भारतीय कृषि अनुसंधान परिषद के
तकनीकी कर्मियों के लिए

संगणक अनुप्रयोग पर प्रशिक्षण कार्यक्रम

TRAINING PROGRAMME ON COMPUTER APPLICATIONS FOR TECHNICAL PERSONNEL OF ICAR

15–21 दिसंबर, 2022

संदर्भ पुस्तिका
REFERENCE MANUAL

पाठ्यक्रम समन्वयक:
डॉ शशि दहिया

पाठ्यक्रम सह समन्वयक:
डॉ चंदन कुमार देब
श्री अक्षय धीरज

Course Coordinator :
Dr. Shashi Dahiya

Course Co-Coordinators :
Dr. Chandan Kumar Deb
Mr. Akshay Dheeraj

संगणक अनुप्रयोग प्रभाग,
भा.कृ.अनु.प. – भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
लाइब्रेरी एवेन्यू , पूसा, नई दिल्ली – 110012

Division of Computer Applications
ICAR - Indian Agricultural Statistics Research Institute (ICAR-IASRI)
Library Avenue, PUSA, New Delhi - 110012

https://iasri.icar.gov.in

तकनीकी कर्मियों
के लिए
कंप्यूटर अनुप्रयोग

Computer Applications
for
Technical Personnel of ICAR

Under the aegis of
HRM Unit, ICAR

15 – 21 December, 2022

Reference Manual

Course Coordinators:
Dr. Shashi Dahiya, Dr. Chandan Kumar Deb, Mr. Akshay Dheeraj

ICAR- Indian Agricultural Statistics Research Institute

Library Avenue, New Delhi – 110012

Editors:

Dr. Shashi Dahiya

Dr. Chandan Kumar Deb

Mr. Akshay Dheeraj

Disclaimer: The information contained in this reference manual has been taken from various web resources. Respective URLs are mentioned in the content.

Citation:

Dahiya S., Deb, C.K., Dheeraj A. (2022). **Computer Applications for Technical Personnel of ICAR**. Reference Manual, ICAR-Indian Agricultural Statistics Research Institute, New Delhi.

प्रस्तावना

भारतीय कृषि सांख्यिकी अनुसंधान संस्थान (भा.कृ.अनु.प.-भा.कृ.सां.अनु.सं.) सांख्यिकीय विज्ञान (सांख्यिकी, संगणक अनुप्रयोग और जैव सूचना विज्ञान) में प्रासंगिकता का एक प्रमुख संस्थान है और कृषि अनुसंधान और सूचित नीति निर्णय लेने की गुणवत्ता को समृद्ध करने के लिए कृषि विज्ञान में उनका विवेकपूर्ण संलयन है। 1930 में अपनी स्थापना के बाद से, तत्कालीन इंपीरियल काउंसिल ऑफ एग्रीकल्चरल रिसर्च के एक छोटे से सांख्यिकीय खंड के रूप में, संस्थान का कद बढ़ा है और इसने राष्ट्रीय और अंतरराष्ट्रीय स्तर पर अपनी उपस्थिति दर्ज कराई है। संस्थान बहुत सक्रिय रूप से सलाहकार सेवा का अनुसरण कर रहा है जिसने संस्थान को राष्ट्रीय कृषि अनुसंधान और शिक्षा प्रणाली (एन.ए.आर.ई.एस.) और राष्ट्रीय कृषि सांख्यिकी प्रणाली (एन.ए.एस.एस.) दोनों में अपनी उपस्थिति दर्ज कराने में सक्षम बनाया है। कृषि अनुसंधान और नीति नियोजन की गुणवत्ता बढ़ाने के लिए उन्नत सांख्यिकीय तकनीकों और संस्थान में विकसित प्रयोगों की कुशल रचना का व्यापक रूप से उपयोग किया गया है। भा.कृ.अनु.प.-भा.कृ.अनु.प. कृषि में आधुनिक, अत्याधुनिक सूचना और संचार प्रौद्योगिकी (आई.सी.टी.) के हस्तक्षेप से संबंधित अनुसंधान, शिक्षण और प्रशिक्षण गतिविधियों में भी सक्रिय रूप से लगा हुआ है।

कंप्यूटर अनुप्रयोगों के बढ़ते उपयोग से कृषि का कायाकल्प हो गया है। वेब और मोबाइल एप्लिकेशन लोकप्रिय सूचना और संचार प्रौद्योगिकियां हैं जो कृषि सूचना प्रणाली को बदल रही हैं। इस तरह के एप्लिकेशन विभिन्न साइबर हमलों के लिए प्रवण होते हैं जिन्हें व्यवस्थित तरीके से प्रबंधित करने की आवश्यकता होती है। कंप्यूटर अनुप्रयोगों की भारी मांग को ध्यान में रखते हुए, इस कार्यक्रम को तकनीकी कार्मिकों के लिए "कंप्यूटर अनुप्रयोग" पर प्रशिक्षण प्रदान करने के लिए डिज़ाइन किया गया है ताकि प्रतिभागियों को वेब और मोबाइल एप्लिकेशन विकास, गूगल फॉर्म विकास, साइबर सुरक्षा प्रबंधन, डेटाबेस प्रबंधन और डेटा विश्लेषण आदि में सक्षम बनाया जा सके। यह कार्यक्रम तकनीकी कर्मियों के बीच कृषि में कंप्यूटर अनुप्रयोगों में जागरूकता लाने और आवश्यक विशेषज्ञता बनाने के लिए आयोजित किया जा रहा है।

प्रशिक्षण कार्यक्रम के लिए शिक्षण संकाय कंप्यूटर अनुप्रयोग के क्षेत्र में अच्छी तरह से स्थापित है। संदर्भ पुस्तिका में व्याख्यान नोट्स विषय की व्याख्या प्रदान करते हैं। मुझे उम्मीद है कि संदर्भ पुस्तिका प्रतिभागियों के लिए काफी उपयोगी होगी। मैं इस अवसर पर पूरे संकाय को एक अद्भुत काम करने के लिए धन्यवाद देता हूं। मैं डॉ. शशि दहिया, डॉ. चंदन कुमार देब और श्री अक्षय धीरज, इस प्रशिक्षण कार्यक्रम के पाठ्यक्रम समन्वयक, को समय पर इस मूल्यवान दस्तावेज को सामने लाने के लिए बधाई देना चाहता हूं। हम इस संदर्भ पुस्तिका को बेहतर बनाने के लिए प्रत्येक भागीदार के सुझावों की प्रतीक्षा कर रहे हैं।

दिनांक: 15 दिसंबर 2022

नई दिल्ली

(राजेन्द्र प्रसाद)

निदेशक, भा.कृ.अनु.प.-भा.कृ.सां.अ.सं.

Foreward

ICAR-Indian Agricultural Statistics Research Institute (ICAR-IASRI) is a premier Institute of relevance in Statistical Sciences (Statistics, Computer Applications and Bioinformatics) and their judicious fusion in agricultural sciences for enriching quality of agricultural research and informed policy decision making. Ever since its inception in 1930, as a small Statistical Section of the then Imperial Council of Agricultural Research, the Institute has grown in stature and made its presence felt both nationally and internationally. The Institute has been very actively pursuing advisory service that has enabled the Institute to make its presence felt both in National Agricultural Research and Education System (NARES) and National Agricultural Statistics System (NASS**).** The advanced statistical techniques and efficient design of experiments developed at the Institute have been widely used for enhancing the quality of agricultural research and policy planning. ICAR-IASRI is also actively engaged in research, teaching and training activities related to modern, cutting-edge Information and Communication Technology (ICT) interventions in agriculture.

Agriculture has been transformed by the increased use of computer applications. Web and mobile applications are the popular information and communication technologies that are transforming agricultural information system. Such applications are prone to various cyber-attacks that need to be managed in a systematic way. In consideration of the huge demand of Computer Applications, this programme has been designed to impart training on **"Computer Applications" for Technical Personnel** of ICAR, to enable participants to work on web and mobile application development, google form development, cyber security management, database management and perform data analysis etc. This programme is being conducted for bringing awareness and building necessary expertise in computer applications in agriculture among the technical personnel.

The Teaching faculty for the training programme is well established in the field of Computer Applications. The lecture notes in the reference manual provide an exposition of the subject. I hope that the reference manual will be quite useful to the participants. I take this opportunity to thank the entire faculty for doing a wonderful job. I wish to complement Dr. Shashi Dahiya, Dr. Chandan Kumar Deb and Mr. Akshay Dheeraj, Course Coordinators of this training programme, for bringing out this valuable document in time. We look forward to suggestions from each participant in improving this reference manual.

Date: 15.12.2022 (Rajender Parsad)

New Delhi Director, ICAR-IASRI

आमुख

भा.कृ.अनु.प.- भारतीय कृषि सांख्यिकी अनुसंधान संस्थान (भा.कृ.अनु.प.-भा.कृ.सां.अ.सं.) सांख्यिकीय विज्ञान (सांख्यिकी, कंप्यूटर अनुप्रयोग और जैव सूचना विज्ञान) में प्रासंगिकता का एक प्रमुख संस्थान है और कृषि अनुसंधान की गुणवत्ता को समृद्ध करने और नीतिगत निर्णय लेने के लिए कृषि विज्ञान में उनका विवेकपूर्ण संलयन है। संस्थान ने कृषि अनुसंधान के लिए उपयोगी विभिन्न सॉफ्टवेयर, वेब एप्लिकेशन, मोबाइल एप्लिकेशन और अब आर्टिफिशियल इंटेलिजेंस (एआई) आधारित उपकरण, तकनीक और कार्यप्रणाली विकसित करने का बीड़ा उठाया है।

कंप्यूटर अनुप्रयोगों के बढ़ते उपयोग से कृषि का कायाकल्प हो गया है। कंप्यूटर के विभिन्न अनुप्रयोग किसानों को निर्णय लेने के लिए कृषि सूचना तक आसान पहुँच प्रदान करते हैं। कंप्यूटर अनुप्रयोगों की भारी मांग को ध्यान में रखते हुए, भारतीय कृषि अनुसंधान परिषद ने 15 दिसंबर से 21 दिसंबर, 2022 तक तकनीकी कर्मियों के लिए "कंप्यूटर अनुप्रयोग" पर एक प्रशिक्षण कार्यक्रम को मंजूरी दी है। इस प्रशिक्षण कार्यक्रम का उद्देश्य कंप्यूटर का अवलोकन प्रदान करना है। कृषि में अनुप्रयोग जिसमें वेब और मोबाइल एप्लिकेशन विकास, गूगल फॉर्म विकास, साइबर सुरक्षा प्रबंधन, डेटाबेस प्रबंधन, डेटा विश्लेषण अनुप्रयोगों आदि के विभिन्न पहलुओं पर सैद्धांतिक और व्यावहारिक सत्र शामिल हैं। इस प्रशिक्षण कार्यक्रम का उद्देश्य तकनीकी कर्मियों के बीच कंप्यूटर अनुप्रयोगों में जागरूकता लाना और आवश्यक निर्माण करना है।

हम इस अवसर पर प्रशिक्षण कार्यक्रम के शिक्षकों को धन्यवाद देना चाहते हैं जिन्होंने इस पाठ्यक्रम को सफल बनाने में अपना बहुमूल्य समय दिया है। उनके सहयोग के बिना इस नियमावली को समय पर पूरा करना संभव नहीं होता। हम इस प्रशिक्षण कार्यक्रम में अपने कर्मचारियों को प्रतिनियुक्त करने के लिए विभिन्न भा. कृ. अनु. प. के संस्थानों के भी आभारी हैं। हम डॉ राजेंद्र प्रसाद, निदेशक, भा. कृ. अनु. प.- भा. कृ. सां. अ. सं., के बहुमूल्य मार्गदर्शन और पाठ्यक्रम के सुचारू संचालन के लिए सभी आवश्यक सुविधाएं उपलब्ध कराने के लिए आभारी हैं। हम उन सभी के आभारी हैं जिन्होंने इस प्रशिक्षण नियमावली को तैयार करने में प्रत्यक्ष व अप्रत्यक्ष रूप से हमारी सहायता की है।

(शशि दहिया)
पाठ्यक्रम संयोजक

(चंदन कुमार देब)
पाठ्यक्रम सह-संयोजक

(अंक्षय धीरज)
पाठ्यक्रम सह-संयोजक

PREFACE

ICAR-Indian Agricultural Statistics Research Institute (ICAR-IASRI) is a premier Institute of relevance in Statistical Sciences (Statistics, Computer Applications and Bioinformatics) and their judicious fusion in agricultural sciences for enriching quality of agricultural research and informed policy decision making. The Institute has taken a lead in developing various Software, Web Applications, Mobile applications and now Artificial Intelligence (AI) based tools, techniques and methodologies useful for Agricultural Research.

Agriculture has been transformed by the increased use of computer applications. Various applications of computer and It enable the farmers an easy access to agricultural information for decision making. Web and mobile applications are the popular information and communication technologies that are transforming agricultural information system. A database is an important component of any web/mobile application as it provides a central location for storing information and business logic. Such applications are prone to various cyber-attacks that need to be managed in a systematic way. Computer networks provide communication possibilities faster than other facilities; hence they are important in the proper functioning of computer applications. In consideration of the huge demand of Computer Applications among Technical Personnel, Indian Council of Agriculture Research has approved a training programme on **"Computer Applications" for Technical Personnel of ICAR** from December 15 to December 21, 2022. This training programme aims to provide an overview of computer applications in agriculture that includes theory and hands on sessions on different aspects of web and mobile application development, google form development, cyber security management, database management, data analysis applications etc. The objective of this training programme is to bring awareness and to build necessary expertise in computer applications in agriculture among the technical personnel.

We would like to take this opportunity to thank the faculty of the institute who have spared their valuable time in making this course successful. Without their cooperation timely completion of this manual would not have been possible. We are also thankful to the various ICAR Institutes and State Agricultural Universities for deputing their employees in this training programme. We are grateful to Dr. Rajender Parsad, Director, ICAR-IASRI for his valuable guidance and making all necessary facilities available for smooth conduct of the course. We are thankful to each and every one who has supported us directly or indirectly for preparing this training manual.

(Shashi Dahiya)
Course Coordinator

(Chandan Kumar Deb)
Course Co-Coordinator

(Akshay Dheeraj)
Course Co-Coordinator

CONTENTS

DIVISION OF COMPUTER APPLICATIONS
ICAR - INDIAN AGRICULTURAL STATISTICS RESEARCH INSTITUTE
LIBRARY AVENUE, PUSA, NEW DELHI - 110012

Training Programme on "Computer Application" (15-21 Decemeber'2022) Course Schedule

Date & Day	**09:30-10:00**	**10:00-11:15**	**11:30-1:00**	**2:15-3:30**	**3:45-5:00**
15/12/2022 (Thursday)	Inaugural Programme	Computer system and information technology (Dr. Shashi Dahiya)	IASRI - IT applications for NARES (Dr. Sudeep)	Microsoft Office Data management using MS Excel (Dr. Sanchita Naha)	
Date& Day	**09:15-10:30**	**10:30-11:45**	**12:00-1:15**	**2:15-3:30**	**3:45-5:00**
16/12/2022 (Friday)	Google Forms Development (Mr. Akshay Dheeraj)	Networking Basics, Unified Communication, Electronic Mail System (Sh. Subhash Chand)	Cyber Security Concepts (Dr. Mukesh Kumar)	Cyber Crisis Management Techniques (Dr. Himanshu) DKMA	Web Application Architecture and Development (Dr. Alka Arora)
17/12/2022(Saturday) HOLIDAY					
18/12/2022(Sunday) HOLIDAY					
Date& Day	**09:15-11:45**		**12:00-1:15**	**2:15-5:00**	
19/12/2022 (Monday)	Database Management (Dr. Soumen Pal)		HTML (Dr. S.N Islam)	Open Source Technology: Content Management, Website creation Using DRUPAL (Kamal Batra)	
Date& Day	**09:15-10:30**	**10:30-11:45**	**12:00-1:15**	**2:15-5:00**	
20/12/2022 (Tuesday)	Basics of Programming using Python (Ms.Madhu)	Application Development using Python (Mr. Samarth Godara)	Mobile Application Development (Dr.Chandan Kumar Deb)	Mobile Application Development (Dr. Md.Ashraful Haque)	
Date& Day	**09:15-10:30**	**10:30-11:45**	**12:00-1:15**	**2:15-3:30**	**3:45-5:00**
21/12/2022 (Thursday)	Statistical computing, ICAR Research data repository for knowledge management (Dr. Rajender Parsad)	Feedback Session	Training Management Information System for ICAR (Shashi Dahiya)	Data Science Dr. Anshu Bharadwaj	Concluding Session

Digital Agriculture

Shashi Dahiya

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
Shashi.Dahiya@icar.gov.in

Digital Agriculture

- Using digitization to Increase Quality and yield of Agricultural Products.
- An approach that helps to guide actions needed to transform and reorient agricultural systems to effectively support development and ensure food security in a changing climate.

Figure 1:Different sectors of digital agriculture

Digital Agriculture - GOAL

To digitize agriculture science and convert it into actionable information for the farmers to provide personalized farm management solutions which would guide farmers across the cropping cycle and support in higher quality yield and getting good price."

Sustainable Development Goal

- Achieving the UN Sustainable Development Goal of a 'world with zero hunger' by 2030 will require more productive, efficient, sustainable, inclusive, transparent and resilient food systems (FAO, 2017b p. 140).
- This will require an urgent transformation of the current agrifood system.

- Digital innovations and technologies will be part of the solution**.**

A Significant Step

- The immense amount of Agricultural data gives insights, understanding and quick learnings.
- Digital Farming is particularly revolutionary for small holders as it has the potential to remove the current information asymmetry.
- The digitalization of agriculture will cause a significant shift in farming and food production over the coming years.

Figure 2: Digitalization of agriculture

Digital Agricultural Revolution

- Historically, agriculture has undergone a series of revolutions that have driven efficiency, yield and profitability.
- Market forecasts for the next decade suggest a 'digital agricultural revolution' will be the newest shift which could help ensure agriculture meets the needs of the global population into the future.

Digital Technologies

- The Internet, Mobile technologies and devices,
- Data analytics,
- Artificial intelligence,
- Internet of Things,
- Digitally-delivered services and apps
- Geographical Information Systems and Remote satellite Data.

Digital technologies are creating new opportunities to integrate smallholders in a digitally driven Agri-Food system (USAID, 2018).

Conditions for a Digital Transformation

- Basic conditions are the minimum conditions required to use technology and include:
  - Availability,
  - Connectivity,
  - Affordability,
  - ICT in education and supportive policies and programmes for digital strategies.

BASIC CONDITION1
IT infrastructure and networks in rural areas.

- Globally, mobile cellular subscriptions have been growing over recent years.
- Much of this recent growth has been driven by countries in Africa and Asia and the Pacific.
- Access to computers and internet has also been increasing in Least Developed Countries and developing economies.
- Challenge: Network coverage in rural areas remains limited. Despite 4G becoming the most common mobile connection globally and 90% of being able to access the internet through 3G or higher quality network, only around a third of rural populations in LDCs receive coverage by 3G networks (GSMA, 2019a).

Access to Mobiles and Internet

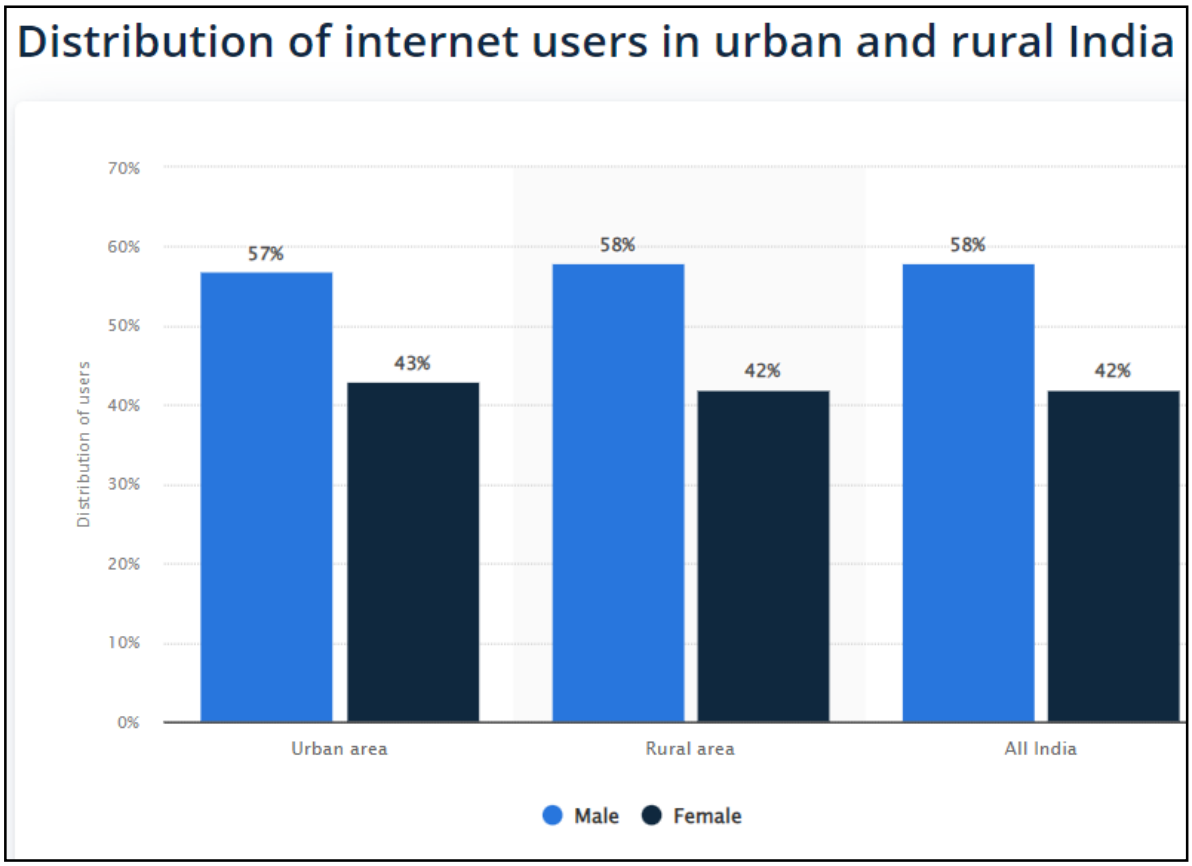


Figure 3: Distribution of Internet users in urban and rural India

Educational attainment, digital literacy and employment in rural areas

- The use of digital technologies requires basic literacy and numeracy as well as special technical knowledge and skills.
- People without such competencies can end up marginalized in increasingly digitally driven societies.
- 'Digital literacy' is critical for using digital technologies.
- Unlike in many developed countries, where students regularly use advanced technologies and digital skills in their education and day-to-day lives, ICT knowledge and skills lag behind in developing countries.
- In many developing countries, basic computer courses are not integrated in primary or secondary education due to a lack of interest from governments and the private sector to invest in building new digital skills.

- A lack of digital tools, such as tablets and laptops, in schools is identified by teachers as a major obstacle to IT education (European Commission, 2019). There is also a lack of relevant skills among teachers.

Policies and Programmes for enabling Digital Agriculture

- Development of government e-services has often been particularly slow in the agricultural sector and very few countries provide e-Agriculture services.
- Developed countries are leading on implementing national level strategies on digital agriculture.
- In developing countries, most of the e-Agriculture services are embedded within e-government or ICT strategies where the main objective is to provide basic e-Agriculture services such as early alert notifications and general information.

ENABLING CONDITIONS FOR DIGITAL AGRICULTURE TRANSFORMATION

- Three key enablers are:

1. The use of internet, mobile and social networks among farmers and agricultural extension officers,
2. Digital skills among the rural population.
3. A culture which encourages digital agripreneurship and innovation.

Establishing a 'digital agriculture ecosystem'

- Digitalization creates demand for digital skills and for people who are competent in using digital devices, understanding outputs and developing programmes and applications.
- This requires not only basic literacy and numeracy but also data handling and communication skills. In populations where such skills are lacking, education must improve quickly; ICT is developing at an incredibly rapid pace and rates of learning must keep up.
- Digital skills among rural populations Youth have a particular role to play in this process.
- They often have the advantage of digital literacy and the capacity for innovative solutions. When digital topics are integrated in educational programs they can also gain an understanding of the uses of digital tools and the skills to create them.

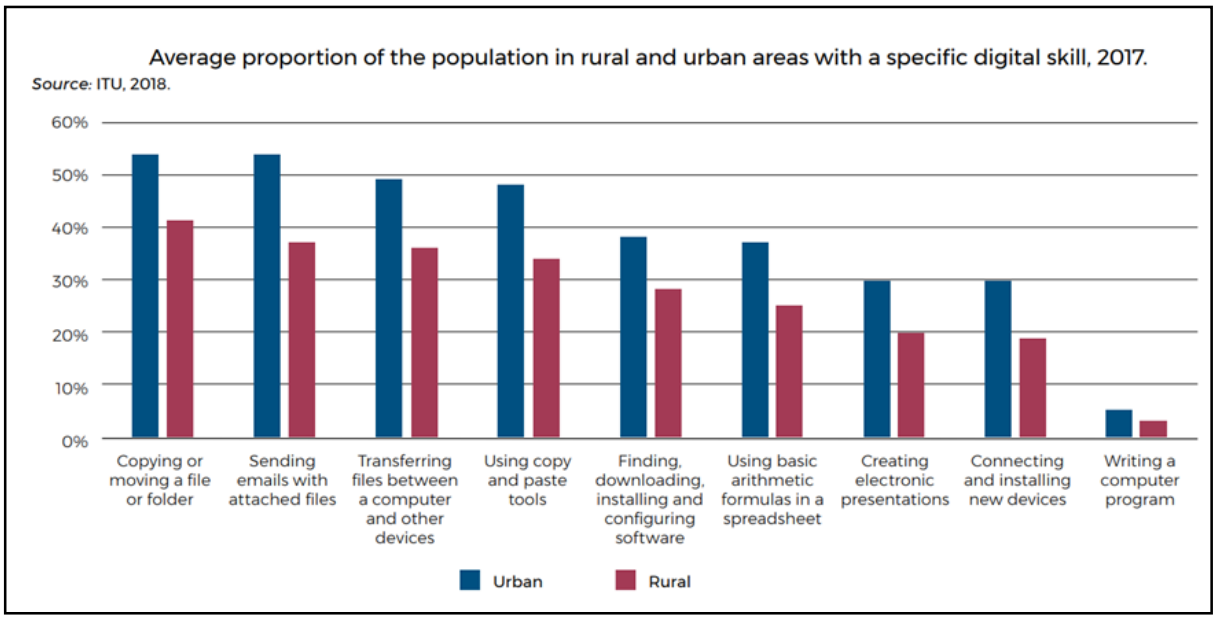


Figure 4: Average proportion of the population in rural and urban areas with a specific difital skill

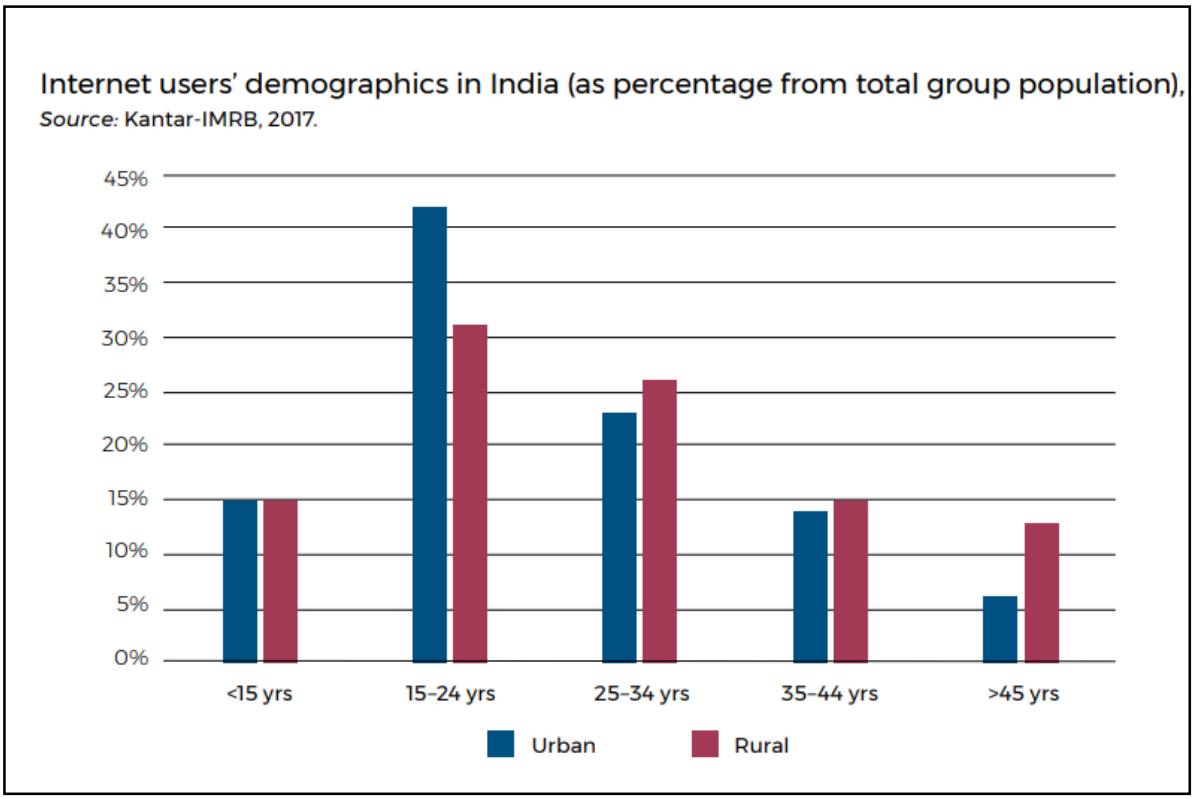


Figure 5: Internet users demographics in India

Farm and Farmer Management Systems

- Empowers farmers through delivering Information, expertise and resources, to increase productivity and profitability, hence improving standard of living.
- The collaborative platform strives to combine cutting edge technology (Big Data, machine learning, smartphones/tablets, etc.), innovative business model (agriculture platform as a

service), and focused human efforts (agriculture insights, products, and services) to serve smallholder farmers.

- Facilitates farmers in taking and executing optimum decisions by providing geomapping, crop planning, individual farm plans and farm automation customized for each farmer based on weather, soil, pest and crop data on an almost real-time basis.
- They are sustainable, data-driven, scalable, intelligent, self-learning, real-time collaborative Agrifood system, which serves as a farm as well as farmer management solution, predictive analytics and monitoring tool, decision support system and agriculture (buy/sales side) e-commerce platform.

Digital technologies to improve the efficiency and functioning of agrifood systems

- The use of mobile applications providing all crop related information to farmers.
- Providing price information to farmers can reduce market distortions.
- They help farmers to plan production processes.
- mKisan SMS Portal for farmers enables all Central and State government organizations in agriculture and allied sectors to give information/services/advisories to farmers by SMS in their language, preference of agricultural practices and location. (https://mkisan.gov.in)

Mobile Telephony

- Under the National e-Governance Plan - Agriculture (NeGP-A), various modes of delivery of services have been envisaged.
- These include internet, touch screen kiosks, agri-clinics, private kiosks, mass media, Common Service Centres, Kisan Call Centres, and integrated platforms in the departmental offices coupled with physical outreach of extension personnel equipped with hand held devices.
- However, mobile telephony (with or without internet) is the most potent and omnipresent tool of agricultural extension.

Figure 6: Future of food and agriculture

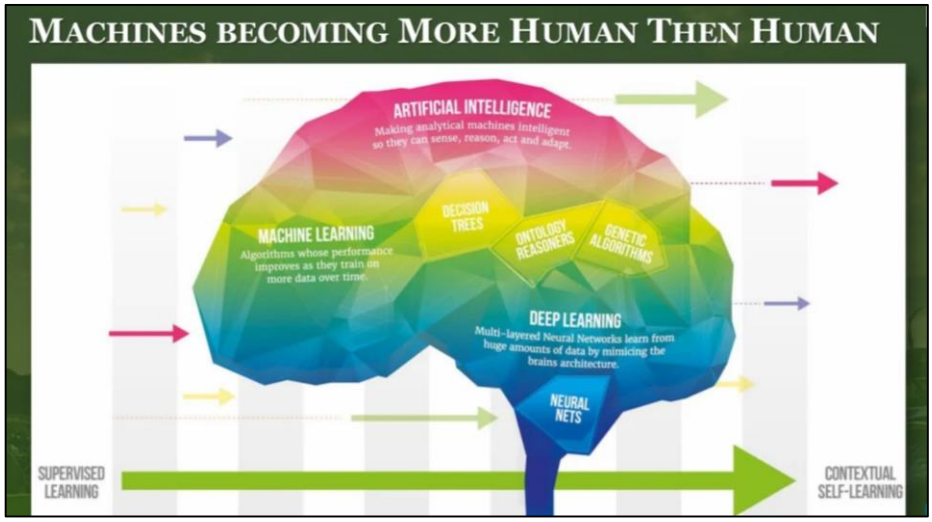


Figure 7: Transmission of machine to human

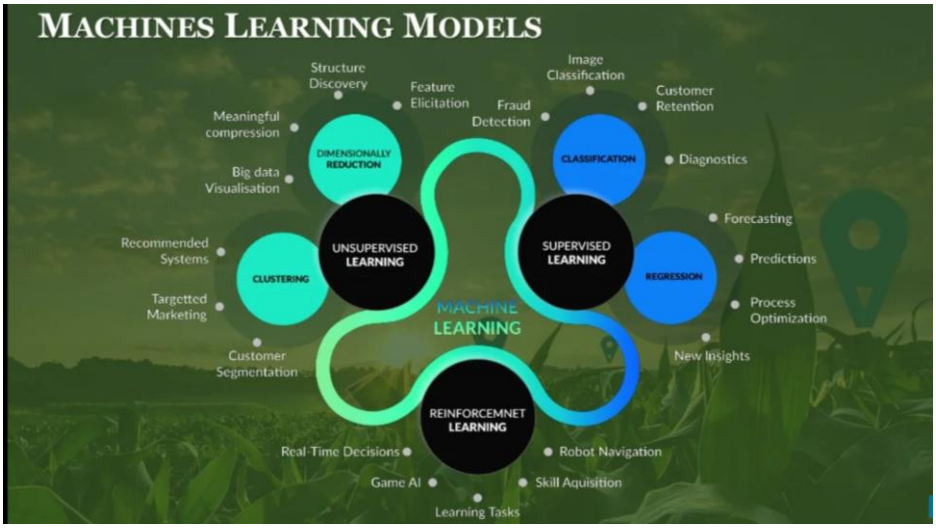


Figure 8: Machine Learning model

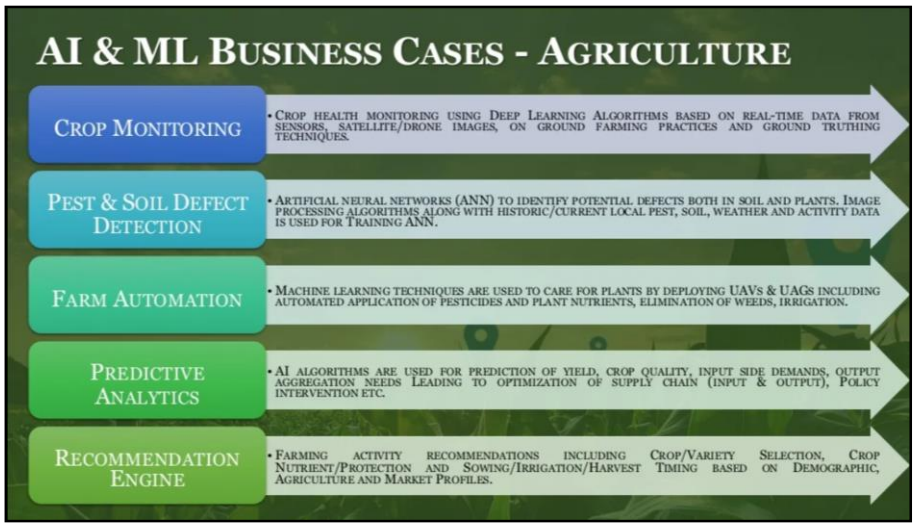


Figure 9: AI&ML Business Cases

Figure 10: 5 V's of Bigdata

Figure 11: Analytical Model

Figure 12: Big Data Business case

Internet of Things

- Internet of Things (IoT)-based application development platforms have the potential to run farm management tools capable of monitoring real-time events when integrated into interactive innovation models.
- Their capabilities must extend to flexible reconfiguration of programmed actions.
- IoT platforms require complex smart decision-making systems based on data-analysis and data mining of big data sets.

Where IOT can be used?

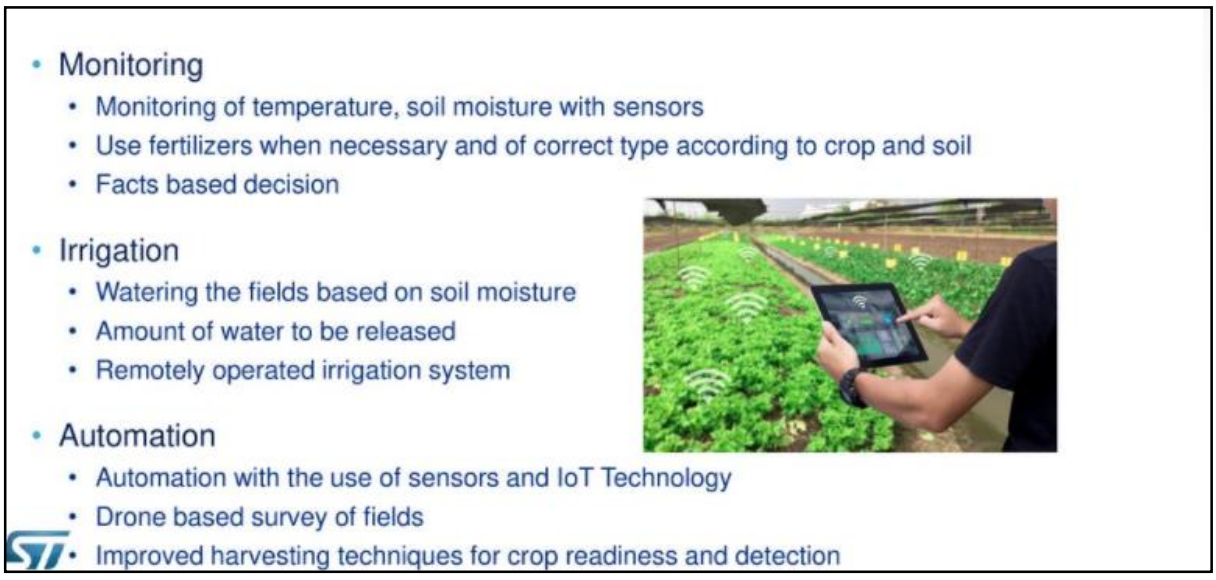


Figure 13: Use of IoT in agriculture

Sensors

- There are different types of sensor
  - Generic sensors such as Temperature, Humidity and Moisture sensors
  - Specific Sensors for macro-nutrients such as NPK (Nitrogen, Phosphorous and Potassium)
- Air temperature, humidity, light, soil temperature and soil moisture are primary concerns for smaller farmers
- Motion sensors, GPS during the harvesting of the crops

Figure 14: Types of sensor used in agriculture

How IOT can be used?

- Edge Nodes : devices, nodes, and sensors that are actually placed in the farm land, fields, orchards that actually collects the data and sends over one of the connectivity technologies
- Gateway : Physical hardware or piece of software on Mobile that allows data to transmit from the sensory device to the platform. The gateway can be assumed to be a data concentrator with improved network performance
- Apps : Mobile Apps that allows the end user to view the data and take fact based decision
- Security : SW/HW that protects your data as they are transmitted between nodes and gateway

Figure 15: Steps for using IoT

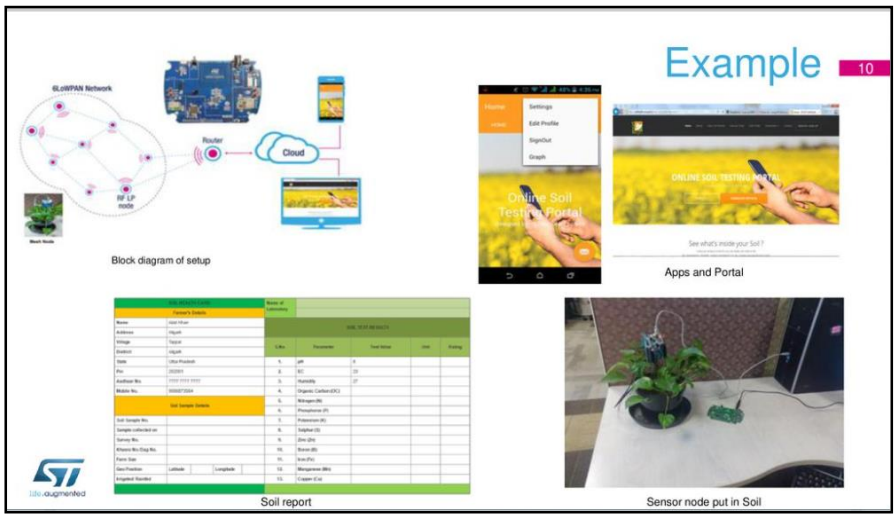


Figure 16: Illustration of IoT application in agriculture

Artificial Intelligence

- " It is the science and engineering of making intelligent machines, especially intelligent computer programs.
- It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable." John McCarthy, 2004.

AI in Agriculture

Artificial Intelligence (AI) in Agriculture

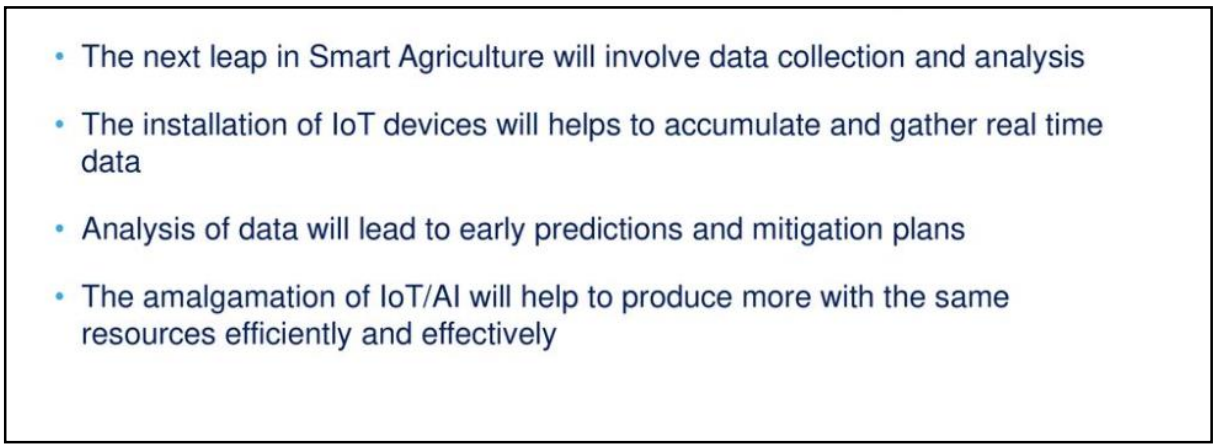


Figure 17: AI in Agriculture

Artificial Intelligence (AI) in Agriculture

- Over the last few years, the AI technology has strengthened agrobased decisions.
- Using AI farmer can monitor every stage of the crop production cycle.
- AI technology is transforming the agricultural sector, as farmers can depend on the data that satellite or UAV record to determine the state of the farm rather than walking all the distance. AI can improve resource use, support early decision making through predictive models and maintain 24/7 monitoring systems;
- AI and Machine Learning based DSS, Prediction Models, Forewarning Models, Forecasting Models and Robots are very helpful in Agriculture.

AI based Systems/ Activities

- Predictions on future climate scenarios.
- Development of forewarning models to alert policymakers and farmers.
- The pattern and potential distribution of target diseases and insect pests under future climate scenarios at zonal, regional and state levels can be forecasted in advance.
- Real-time structured surveillance for insect pests and diseases.
- Hot spots identified in the process and GIS-based risk maps (spatial and temporal) will be developed for targeted diseases and insect/pests.

- This knowledge once available in advance for scientists and Integrated pest management (IPM) practitioners will help in developing country-specific strategies to ultimately support the greater resilience of smallholder farmers.

AI in Agricultural Management

Crop Management

- Disease Prediction
- Pest Infestation Prediction
- Yield Prediction (Automatic count, branch detection, No. of immature fruits, fruit detection)

Livestock Management (SVM for Cattle)

- Classification of Animal Behavior
- Early detection and Warning for problems in Livestock Production

Soil testing

- Two technologies that stand for intelligent data fusion are:
  - Proximity Sensing
  - Remote Sensing.
- One use case of this high-resolution data is Soil Testing.
- While remote sensing requires sensors to be built into airborne or satellite systems, proximity sensing requires sensors in contact with soil or at a very close range.
- This helps in soil characterization based on the soil below the surface in a particular place.

Visualization

- Image-based insight generation
- Drone-based images can help in:
  - In-depth field analysis,
  - Crop monitoring,
  - Scanning of fields and so on. They can be combined with computer vision technology and IOT to ensure rapid actions by farmers. These feeds can generate real time weather alerts for farmers.

Detecting Crop Diseases

- Images of various crops are captured using Computer Vision Technology under white/UV light.
- Farmers can then arrange the produce into separate stacks before sending it to the market.
- Pre-processing of images ensures the leaf images are segmented into areas for further diagnosis.

- Such a technique would identify pests more distinctly.

Recommendations

- Based on multiple parameters like:
  - Soil condition
  - Weather outlook
  - Type of seeds
  - Infestation around a certain area,
- Cognitive computing makes recommendations to farmers on the simplest choice of crops and seeds.
- The advice is further personalized basis on the farm's requirement, local conditions, and past successes.
- External factors like marketplace trends, prices or consumer needs can also be recommended through artificial intelligence.

Monitoring crop health

- Remote sensing techniques alongside hyper spectral imaging and 3D laser scanning are essential to create crop metrics across thousands of acres.
- It could usher in a revolutionary change in terms of how croplands are monitored by farmers in terms of time and energy.
- This technology will monitor crops along their entire life-cycle and generate reports for detecting anomalies, if any.

Agricultural Robots

- Agricultural robots ('agrobots') are seen as a key trend that will deeply influence agriculture in the future.
- Field agrobots are already being deployed to help farmers measure, map and optimize water and irrigation use.
- Fleets of small lightweight robots are now seen as a replacement for traditional high mass tractors, allowing a gradual reduction of compaction, re-aeration of the soil and benefits to soil function.
- The weeding robot allows vegetable farmers to manage crop weeding with a high level of precision, while helping them save time all through the season.

To help farmers tackle the increasing regulations on phytosanitary products, the growing concerns with pesticides, and the lack of workers in the agricultural sector, robots provides a new and effective solution

Weather-based Agro-Advisory Messages

- Technologies can also support farmers to anticipate and respond through timely weather-based agro-advisory messages to reduce:
    - Pest attacks
    - Crop failures and
    - Climatic change impacts

Precision Agriculture (PA)

- Precision Agriculture (is an example of an application of the Internet of Things (Internet of Things) in agriculture.
- The use of Guidance Systems during planting and fertilizer application can lead to cost savings in terms of seed, fertilizer and tractor fuel, and can reduce working hours in the field.
- Variable Rate Technologies (VRT) and drones (UAV) can also reduce water and pesticide use and reduce labour and resource costs.

Blockchain Solution

- Aggregates the farm produce, provides e-marketplace services and implements traceability.
- Blockchain technology is used to capture the information of the produce at various levels in the supply chain.
- Comprehensive information about the commodities (growing information, pre- and post-harvest, transportation, warehousing etc.) is pushed through blockchain to generate a digital identity for a physical commodity.
- It eventually build traceability to prove provenance and movement of commodities from farm to table.
- On the other hand, AI leverage the comprehensive aggregation of data at various points in supply chain to proactively advice farmers on sowing, pest control, harvesting etc.

Figure 18: Blockchain Agriculture Use Cases

Price Forecasting Model

- An AI price forecasting model has been developed by Karnataka Agricultural Price Commission (KAPC) and Microsoft combining artificial intelligence, cloud machine learning, satellite-imaging and other advanced technologies.
- The model considers datasets on historical sowing areas, production yields, weather patterns and other relevant information, and it uses remote sensing data from geo-stationary satellite images to predict crop yields at every stage of the farming process. The resulting output from the model includes predictions about arrival dates and crop volumes, enabling local governments and farmers to predict commodity prices three months in advance for major crop markets. With this information, the Karnataka government can more accurately plan ahead to set the minimum support price.

AI-sowing app

- International Crops Research Institute for the Semi-Arid Tropics (ICRISAT) and Microsoft collaboratively developed an AI-sowing app. The app is powered by Microsoft Cortana Intelligence Suite and Power Business Intelligence.
- The Cortana Intelligence Suite includes technology that helps to increase the value of data by converting it into readily actionable forms.
- Using this technology, the app is able to use weather models and data on local crop yield and rainfall to more accurately predict and advise local farmers on when they should plant their seeds.

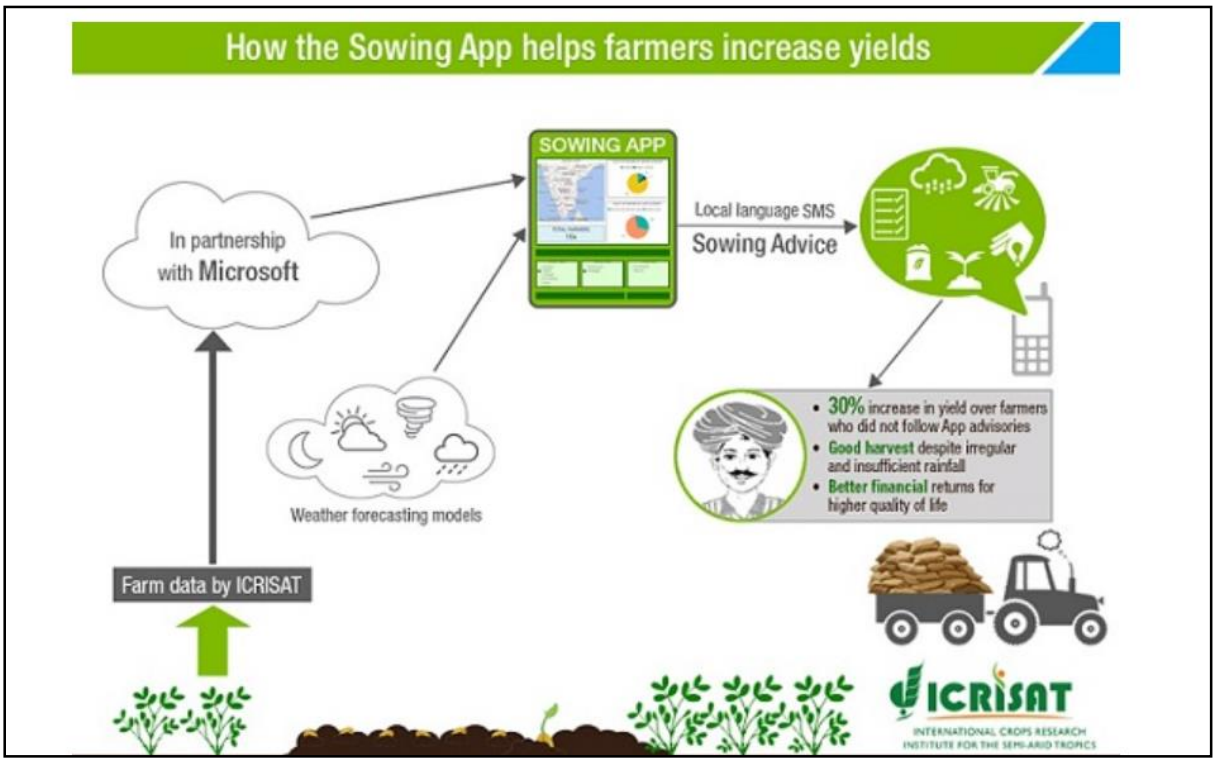


Figure 19: Use of AI-Sowing app

AI Sowing App

- To calculate the crop-sowing period, historic climate data (spanning over 30 years from 1986 to 2015) for the specific area in Andhra Pradesh was analyzed using AI.
- To determine the optimal sowing period, the Moisture Adequacy Index (MAI) was calculated. MAI is the standardized measure used for assessing the degree of adequacy of rainfall and soil moisture to meet the potential water requirement of crops.

A photo of a text message from the Microsoft India AI Sowing App reads "Observe the Baron and Zinc deficiency in field. Spray AGRBOR or Borax". Source: Microsoft

Figure 20: Text message showing advisories issued by AI-Sowing app

Image Recognition Technology

- This AI based technology can recognize objects, faces, flora fauna and tag them in any image.
- It uses deep learning algorithms on which a new generation of intelligent applications are being built for agriculture.
- Small farmers around the world follow traditional farming practices due to lack of access to scientific understanding of crop lifecycle, pests, quality metrics and the latest micro-fertilizers.
- The Image based solutions provide insights on the crops' health during the growing season and its final harvested quality by click of photograph.

Agricultural Product Grading using Image Analysis

- Automated quality analysis of images of food products is an accurate and reliable method for grading fresh products (fruits, grains, vegetables, cotton etc.) characterized by color, size and shape.
- The solution reads the image that a farmer has taken on his phone and determines the product quality in real time, without any manual intervention.

Alerts on Crop Infestation

- Farmers can click an image of their crop and use the solution to understand :
    - The pests, diseases, and foreign plants (weeds) growing in their farms.
    - The solution uses deep learning and image processing models to identify any crop diseases or pest infestation in the crops.
    - Along with the parameters, it gives recommendations on how that disease can be cured and prevented from increasing further.

Pest Risk Prediction App

- This app also leverages AI and machine learning to indicate in advance the risk of pest attack.
- In a few dozen villages in Telangana, Maharashtra, and Madhya Pradesh, farmers receive automated voice calls alerting them whether their crops are at risk of a pest attack based on weather conditions and stage of the crop.

Image Recognition and Soil Science

- Based in the Indian state of Madhya Pradesh, It help farmers with timely information, technology, and right kind of inputs to achieve better yields.
- Uses AI and machine learning to predict pest and disease, forecast commodity prices for better price realizations and recommends products to farmers.
- Uses temperature, humidity and pathology/entomology data to give accurate advisory to the farmers.

Challenges in building the 'digital agriculture ecosystem

- Low overall smartphone ownership in rural areas combined with the cost of internet and limited network coverage present challenges to the use of mobile agricultural applications and limit the scope to use social networks to facilitate agricultural support and information flows between farmers.
- Such availability of information could support farmers to make better farming decisions which could contribute to increasing yields, reduced environmental impacts and improved livelihoods.
- The diversity of available technologies and the lack of standardization and compatibility between them, for example for the exchange of data, also create a barrier to use by farmers. The adaptability of technologies is limited and it is often not possible to integrate machinery from different brands so farmers must decide which brand to invest in.
- There is a lack of independent advisory services to support farmers in making these decisions.

Reaping the benefits of digital technologies in agriculture requires the participation and co-operation of farmers, researchers, private sector, non-profits and government.

Lecture Notes on Microsoft Office and Data Handling with MS Excel

Sanchita Naha

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

Sanchita.naha@icar.gov.in

1. **Introduction to Microsoft Office:** Microsoft Office or popularly MS office is a family of multiple application software provided by Microsoft. It contains the following applications in one software suite:

    a) MS Word (word/ document editor),

    b) MS Excel (spreadsheet program),

    c) MS Power Point (presentation program),

    d) MS Outlook (email client),

    e) MS Access (database),

    f) OneNote (information gathering and multi user access),

    g) OneDrive (cloud storage),

    h) MS Teams (collaborative platform for video conferencing) and many more.

    In this lecture our intent is to familiarize the participants with most frequently used functionalities of various tools from the MS Office suite and data handling with MS Excel. MS Office suite is a proprietary software and distributed under the license terms of Microsoft. Assuming you have MS Office installed in your system let us follow through the following steps to start with.

2. **Microsoft Word:** It is a word/document processing software. It enables you to create, edit and save professional documents like letters, reports and notices.

2.1. Steps to create a word document and insert text in it:

Step1: Go to the start menu and look for Microsoft Word icon or type 'Microsoft word' in search button.

Step2: Click the word icon to open Microsoft Word

Step3: You will see a blinking cursor or insertion point in the text area below the ribbon

Step4: As you start typing, words will appear on the screen in the text area

Step5: Once you type something and save the document in following manner you have successfully created a word file.

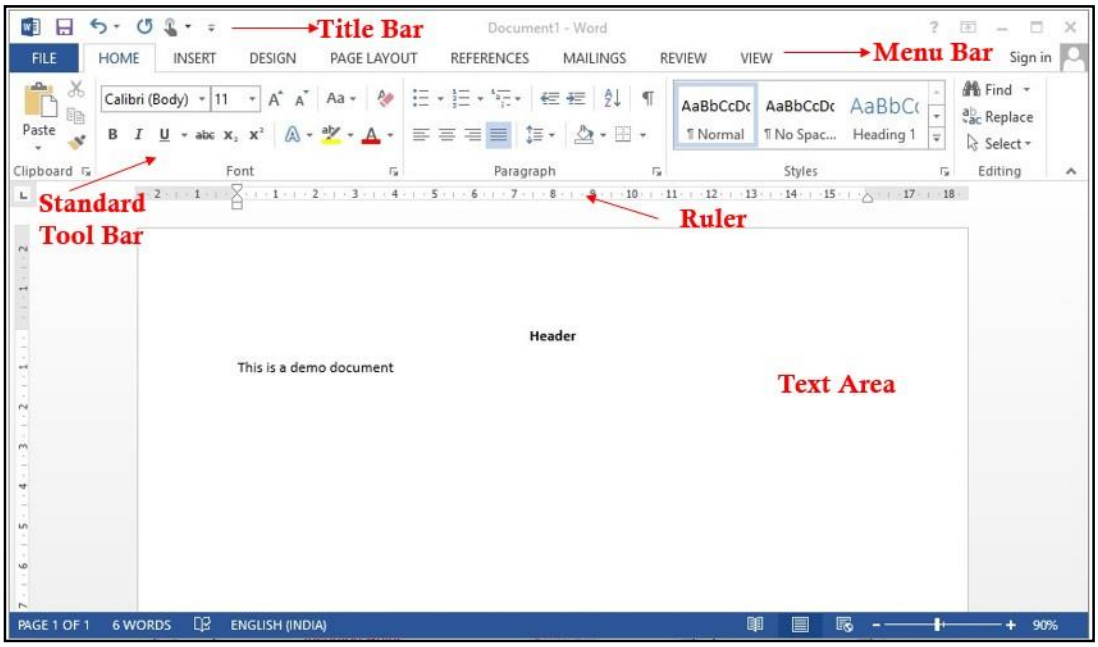


Figure 1: Basics of Microsoft Word wizard

2.2. Steps to save the document in MS Word:

After creating a document it is important to save the document so that it can be viewed or reused later.

Step1: Click the 'File' option on top left of the document

Step2: Click the 'Save As' option

Step3: Select the desired location in your computer and click 'save' option with a desired name for the document in the 'File Name' option of 'save as' dialogue box.

Word files are saved with .doc extension.

2.3. Steps to open a word document:

Step1: Open MS word

Step2: Go To 'File' Option

Step3: Click 'open'

Step4: Select the location of your document

Step5: Select the document you want to open and click 'open'

2.4. Steps to print the document:

Step1: Open the word document you want to print

Step2: Select File > Click 'Print' option.

Step3: To preview each page, select the forward and backward arrows at the bottom of the page. If the text is too small to read, use the zoom slider at the bottom of the page to enlarge it.

Step4: Select File > Click 'Print' option.

2.5. Text Formatting:

2.5.1. Steps to change Font Size in MS Word:

Step1: Select the text that you want to modify

Step2: In Font group click the drop-down arrow next to font size box

Step3: From the Font size menu select the desired font size

Step4: Select the text and click the increase or decrease font size buttons

2.5.2. Steps to Change Font Style in MS Word:

Step1: Select the text you want to modify

Step2: Select the Home tab and locate the Font group

Step3: Click the drop-down arrow next to font style box

Step4: From Font style menu, select the desired font style with a left click

Step5: To change the font to bold or italics or underlined, click the 'B' or 'I' or 'U' icons from the Font bar

2.5.3. Format Font Color in MS Word:

Step1: Select the text that you want to modify

Step2: In Home tab locate the Font group

Step3: Click the drop-down arrow next to Font color button

Step4: From Font color menu, select the desired font color with a left click

2.5.4. Steps to change Text Alignment in MS Word:

Step1: Select the content you want to modify

Step2: In Home tab locate the Paragraph group

Step3: It has four alignment options

Align Text Left: Aligns the text towards left margin

Center Align: Brings the text at the center

Align Text Right: Aligns the text towards right margin

Justify: Aligns the text to both left and right margins

Step4: Select the desired alignment option with a left click

3. **Microsoft PowerPoint:** Microsoft PowerPoint is an application program used to Design Presentations.

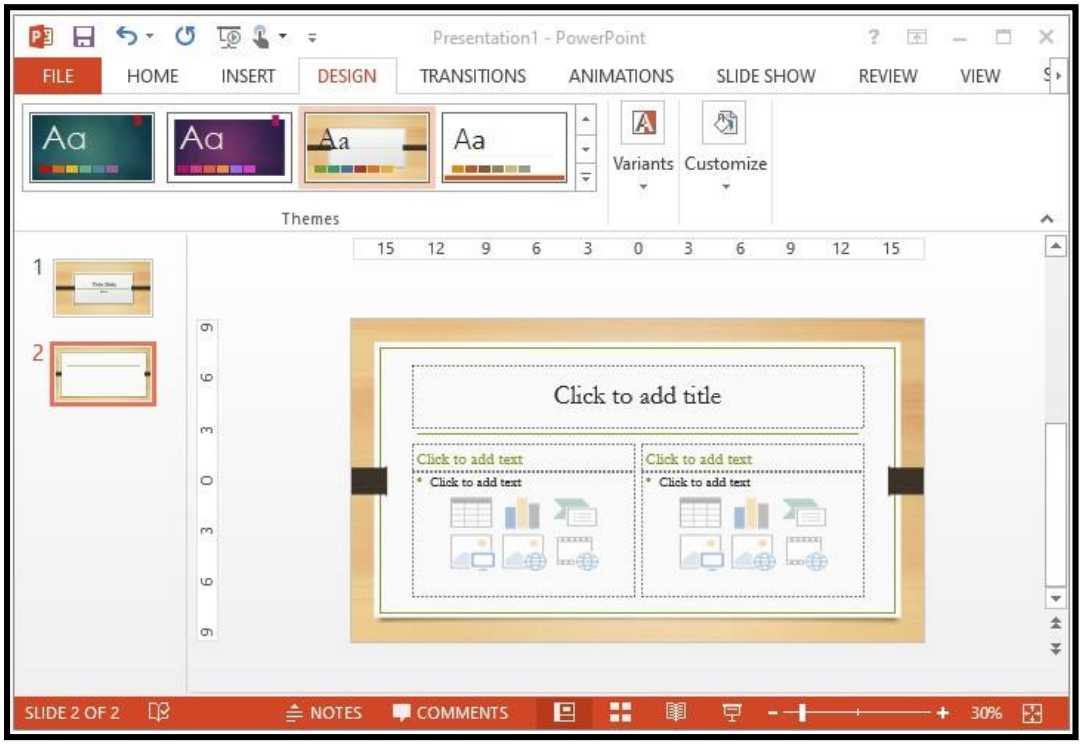


Figure 2: Microsoft PowerPoint

3.1. Steps to create a Presentation:

Step1: Go to the start menu and look for Microsoft PowerPoint icon or type 'Microsoft PowerPoint' in search button.

Step2: Click the PowerPoint icon to open PowerPoint

Step3: Once you open PowerPoint window by default a slide appears. The slide has two placeholders or text boxes. Additional text boxes can be added from the Insert tab.

Step4: To start creating presentation click on the placeholder or text box a blinking cursor will appear. Then type the title and click outside the box. The text box will disappear.

Step5: Click outside the text box.

Step6: Save the .ppt file using 'Save as' option from 'File' menu.

3.2. Steps to Save a Presentation:

Step1: Click the 'File' option in the presentation, select 'Save as' option.

Step2: Select the desired location in the computer, type the name with which you want to save the presentation in 'File Name' option.

Step3: Select .ppt in 'Save as type' option.

3.3. Steps to add slides in the presentation:

Step1: Select the slide next to which you want the new slide to appear
Step2: In Home tab, click the drop-down arrow on the New Slide button
Step3: It will display the office themes
Step4: Select the slide choice that suits your requirement

3.4. Apply different themes to your presentation:

Themes are design templates that make the presentation colorful and more readable.

Step1: Click on Design tab
Step2: Locate the Themes group which will display the office themes

Step4: Select the slide choice that suits your requirement

Step5: Click the desired theme

Step6: Theme will be added to the entire presentation

3.5. Steps to apply animation effect to your presentation:

Step1: Select the text or object you want to animate

Step2: From the Animations tab, expand the Animations group drop-down arrow next to animate option

Step3: Animation options will appear

Step4: Move the cursor over different options to see live preview on the slide

Step5: Select the desired animation

3.6. Steps to Apply Slide Transition Effects:

Transition effects appear when one slide changes into next slide in Slide Show.

Step1: Select the slide to which you want to apply the effect

Step2: Select the Animation tab

Step3: In Transition to This Slide group you will see the transition effects

Step4: Click the drop-down arrow to see menu of transition effects

Step5: Select the desired transition effect

Step6: Click apply to all to apply the effect to all slides

3.7. View slides in presentation mode:

Step1: Select the slide to you want to present

Step2: Click slide show option from menu bar

Step3: You can choose to present from beginning or show current slides

Step4: Right click on the presentation and select end show to come back to previous menu or press 'Esc' to exit from slide show.

4. Microsoft Excel:

Microsoft Excel stores data in tabular form and allow users to perform manipulations/ operations on them. Each sheet in one excel file is called spreadsheet or worksheet. Mostly it is used for accounting or numerical calculations.

4.1. Steps to open MS Excel files:

Step1: Open MS excel

Step2: Go To 'File' Option

Step3: Click 'open'

Step4: Select the location of your document

Step5: Select the document with .xlsx or .xls extension and click 'open'

4.2. Steps to create a excel worksheet and insert values in it:

4.2.1. Worksheet:

An excel worksheet is made of rows and columns that intersect each other to form cells where data values are entered. It is capable of performing multiple tasks like calculations, data analysis, and integrating data. In Excel worksheet, rows are represented by numbers [1, 2, 3 …] and columns by alphabets [A, B, C …]. Therefore the first cell in excel workbook is represented by A1, and if we move on the 2nd column in the same row the selected cell is represented by B1 and so on. You can insert any values by selecting on the particular cell.

4.3. Steps to save Excel files:

Step1: Click File > Save As.

Step2: Under Save As, choose the location where you want to save your workbook. **Step3**: Click Browse to find the location you want in your computer.

Step4: In the File name box, enter a name for a new workbook. Enter a different name if you're creating a copy of an existing workbook.

Step5: To save your workbook in a different file format (like .xls or .txt), in the Save as type list (under the File name box), pick the format you want

Step6: Click Save.

4.4. Steps to Delete Data from Workbook:

Select the data you want to delete, right click on it then select delete option from the menu. You can also delete it by pressing Delete key from the keyboard.

4.4.1. Steps to Delete a Row:

Select the row with a left click on the row number then right click and select the Delete option. If you want to delete more rows drag the mouse downward to select more rows.

4.4.2. How to Delete a Column:

Select the column with a left click on the column header then right click and select Delete option. To delete more columns drag the mouse horizontally left or right to select more columns and then delete.

4.5. Data Manipulation techniques: Data analysis is a challenging task and it requires good data manipulation skills. Here we have described most common data manipulation techniques that are used in Excel for data analysis.

The most common data manipulation techniques are as following:

a) Filtering
b) Sorting
c) Grouping
d) Pivoting
e) Transposing a Matrix
f) Matrix Addition
g) Matrix Multiplication
h) Finding determinant of Matrices
i) Data Filter in MS Excel

4.5.1. Filtering range of data:

Step1: Select any cell within the range.

Step2: Select Data > Filter.

Step3: Select the column header arrow.

Step4: Select Text Filters or Number Filters**,** and then select a comparison, like Between.

Step5: Enter the filter criteria and select OK.

4.5.2. Sorting: Sorting can be done by the filter option under 'Data' menu or directly through 'Sort' option which is available under the 'Data' menu. Once you select 'Data' tab from the menu bar, 'Sort' option is available in the standard tool bar.

Step1: Select the column whose values you want to sort.

Step2: Select sort from data menu

Step3: Specify the column name, select values under 'sort on' option and choose either 'Smallest to Largest' or 'Largest to smallest' under 'Order' option, click ok.

4.5.3. Grouping: With grouping feature you can group values belonging to one category. For example, in a sales table, data can be grouped product wise. We can also get the sub total of each product category by using the following steps.

Step1: Select the similar category of data that you want to group.

Step2: Go to 'Data' tab and find 'Group', select group and press 'OK'. You can see a drill down/ drill up option on the left side scroll bar. On click of the drill up option, similarly grouped data will be hidden from the rest of the table.

Step3: From data tab, find 'Ungroup' and select it to ungroup the data.

Step4: To get subtotal of similar values (whether grouped/ ungrouped), select the 'subtotal' option and select the column for whose unique values you want to get the subtotal values in 'For each change in' option. Then select the function you want to perform on that column and click 'ok'.

4.5.4. Pivoting:

Step1: Select the cells you want to create Pivot table from.

Step2: Click 'Insert' then select 'Pivot table'. This will prompt you to select the range of cells and the worksheet you want to have the pivot table on. This will create a report on the selected datasheet.

Step3: Then drag the column name to filter option based on which you want to filter the row values.

Step4: One by one check or drag the desired columns under the 'Rows' menu.

4.5.5. Matrix in MS Excel: A Matrix is an arrangement of data values in rows and columns. It shows data in a tabular fashion. It is equivalent to storing data elements in a two-dimensional array. You can press 'Tab' to traverse through data horizontally and press 'Enter' for vertical data entry.

Renaming a Matrix: Select all the values in the matrix and right click on it. Find the option 'Define name' and give it a name. Further, you can use the same name for all sorts of matrix multiplications.

Transposing a Matrix: Transposing the matrix is changing the rows into columns or vice-versa. Select the original data array. Click "copy". Select an empty range. Expand the Paste button. Open the "Paste Special" menu. Select 'Transpose' operation. Close the dialog by clicking OK. Make sure row length will become length of the column in the new matrix and vice versa.

Matrix Addition:

Step1: Create two matrices of similar number of rows and columns

Step2: Select another empty cell in the worksheet and type in the formula for adding the first value from the first matrix and the first value from the second matrix.

Step3: Drag the arrow row wise first and then column wise. It will give you added values for all the cells in the matrix.

OR

Step1: If you have named both the matrices you want to add.

Step2: Select same dimension of cells as the size of the matrices to be added, then in the formular bar enter =(matrix_name_1 + matrix_name_2).

Matrix Multiplication: Two matrices can only be applied if the row size of the first matrix is equal to the column size of the matrix 2 and the column size of the first matrix is equal to the row size of the second matrix. Size of the resultant matrix will be row size of the matrix one and column size of the second matrix.

Step1: Select both the matrices to be multiplied. Determine the size of the resultant matrix.

Step2: Select the dimension of the resultant matrix. Type in =MMULT(matrix_1,matrix_2) and press Ctrl + Shift + Enter.

Finding determinant of Matrices:

Step1: Type in =mdeterm(matrix_name) in the formula bar

Step2: Select the cells of the matrix you want to find determinant for

Step3: Press Enter.

4.5.6. Data Filter in MS Excel: The IF function is one of the most popular functions in Excel, and it allows you to make logical comparisons between a value and what you expect. So IF statement can have two results. The first result is if your comparison is True, the second if your comparison is False.

For example, =IF(C2= "Yes", 1, 2) says IF(C2 = Yes, then return a 1, otherwise return a 2).

e.g.:

=IF(B3>C3, "Goal Met", "Goal Not Met").

In this example, the condition is B3>C3, meaning "If the value of B3 is greater than C3."

If the value of B3 is greater than C3, then Goal Met will appear in the cell.

If the value of B3 is less than the value of C3, then Goal Not Met will appear in the cell.

=IF(B3=C3, "Goal Met").

In this example, the condition is B3=C3, meaning, "If the value of B3 is equal to the value of C3."

If the values are equal, "Goal Met" will appear in the cell.

If the values are not equal, a zero will appear in the field because we did not define a third (ELSE) value.

=IF(B3*2>C3, C3*400, "Good").

In this example the condition is B3*2>C3, meaning "If the value of B3 multiplied by 2 is greater than the value of C3."

If the value of B3*2 is greater than the value of C3, the value of C3 multiplied by 400 will appear in this cell.

If the value of B3*2 is not greater than the value of C3, the word "Good" will appear in the cell.

=IF(B3="Sold", "1", "")

In this example, the condition is B3="Sold", meaning "If the value of cell B3 contains the word Sold."

If B3 contains the word "Sold," a "1" will appear in the cell.

If B3 says anything other than the word "Sold," the condition is false. Since our ELSE value is " ", which is two double quote marks with nothing in between, the cell value will be blank.

=IF(D3="Taxed", F3*.07, "0").

In this example, the condition is D3="Taxed", meaning "If the value of D3 is the word Taxed."
If D3 contains the word "Taxed," the result will be the value of F3 multiplied by .07.
If D3 contains anything other than the word "Taxed," the result will be 0.

4.6. Steps to use multiple functions for Data Handling in MS Excel:

4.6.1. Auto Sum Functions:

SUM: If you need to ADD the values in a column or row of numbers then select a cell next to the numbers you want to sum, click AutoSum on the Home tab, and press Enter.
If you click AutoSum, Excel automatically enters a formula (that uses the SUM function) to sum the numbers.

AVERAGE: You can find the Average of a value by selecting the cell below or to the right of values that you want to Average**.**
Then go to the Editing Group, select the Auto-Sum Tool. You will see a drop-down list of the functions, select Average.

MIN: Select the cell below or right, then go to the Auto-Sum Tool in the Editing Group and select the Min Function or select More Function**;** an Insert Function dialog box will pop up and choose Average, Then OK. Enter the range of cells in the Functions Argument box, then OK.

MAX: Choose the cell below or right, then go to the Auto-Sum Tool in the Editing Group and select the Max Function; it will automatically give you the result or you can click more function an Insert Dialog box will appear, Enter the range of cells in the Functions Argument box, then OK.

4.7. Create Diagrams in MS Excel:

4.7.1. Create Line Diagram:

Step1: Select the data you want to plot the graph for
Step2: Click insert tab > go to insert column chart > pick any chart of your preference
Excel will create the graphical representation as following.

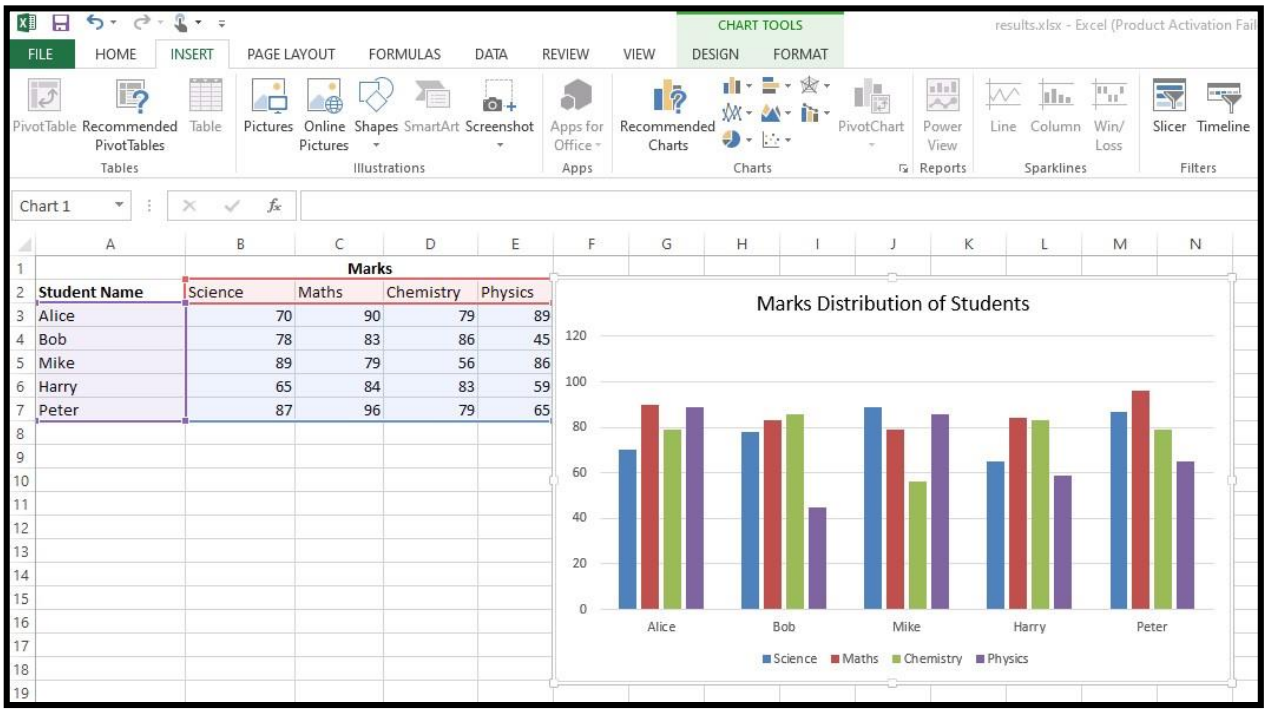


Figure 3: Plotting Line chart in MS Excel

4.7.2. Create pie chart:

Pie chart represents the data in slices of a circle. Each slice represents the percentage contribution of each data section in the sum of individual data values.

Step1: Select the data you want to plot the pie for

Step2: Click insert tab > go to insert pie or doughnut chart > pick any chart of your preference

Excel will create the graphical representation as following.

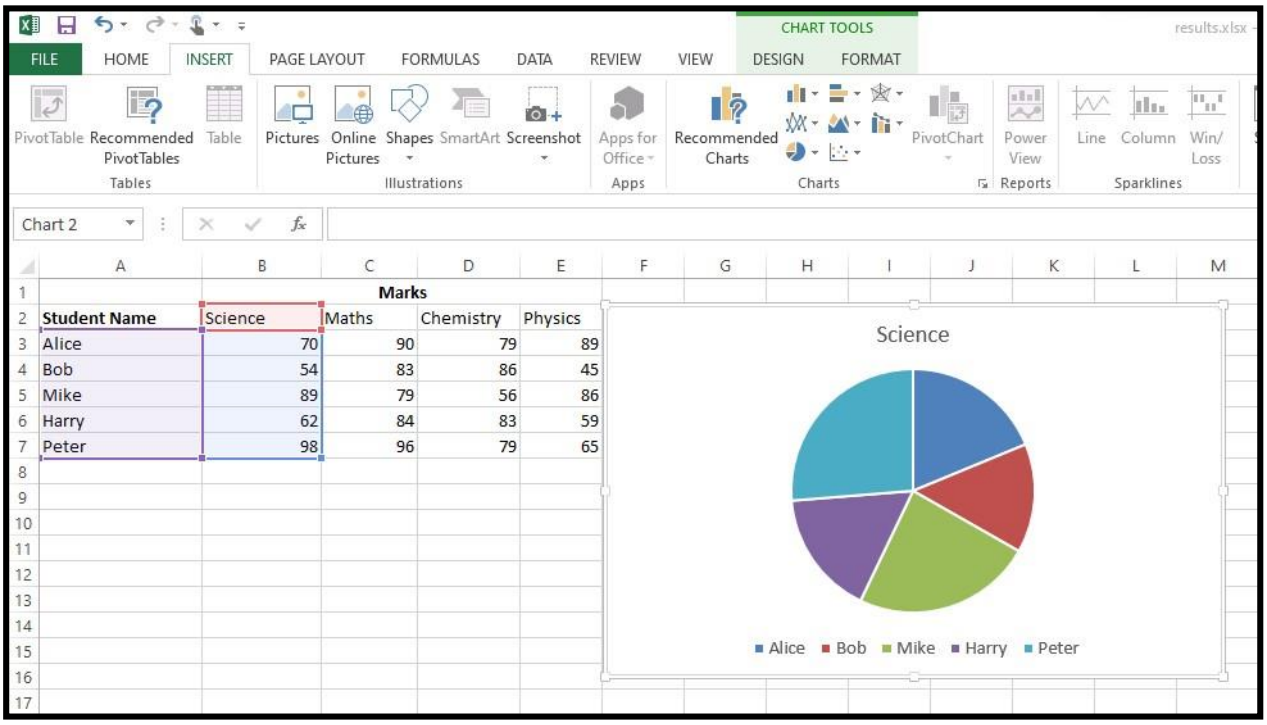


Figure 4: Plotting Pie chart in MS Excel

Google Forms Development

Akshay Dheeraj
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
akshay.dheeraj@icar.gov.in

Introduction

Google Forms is one of the programs available in Google Drive, along with Docs, Sheets, and Slides. You can use Google Forms to create a survey and gather responses. The answers are stored on a Google Sheet spreadsheet, so it's helpful to be somewhat familiar with Sheets. Some uses of Forms beyond the simple survey are self-grading quizzes.

Creating a New Form

Log in to your Google Drive account (drive.google.com). Click on **New**, then mouse down to Google Forms and hover over the arrow next to it. (If you haven't used it before, it might be hiding under **More**.) The arrow gives you two options: **Blank form** and **from a template**. If youjust click on Google Forms, it will open a new blank form.

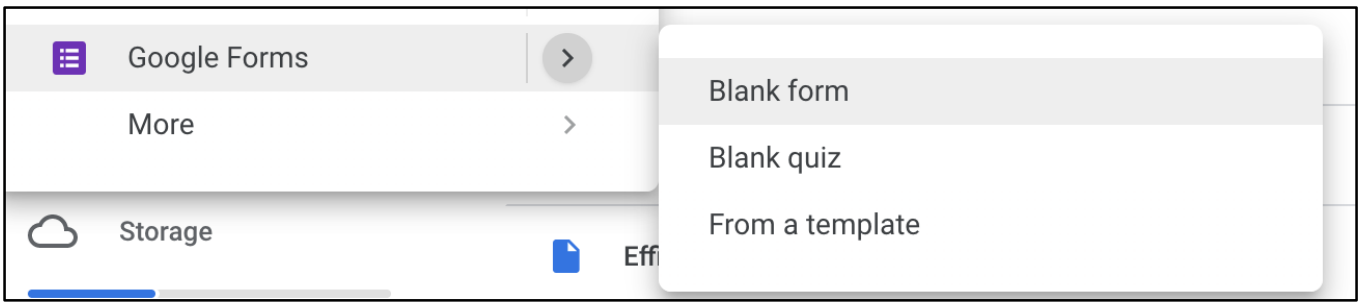


Figure 1: New form creation

Click on **From a template** to see all of the pre-created template options you can use. The templates include a party invitation, a form for deciding on the best time to meet, a feedbackform, an order form, and more.

Screen Layout

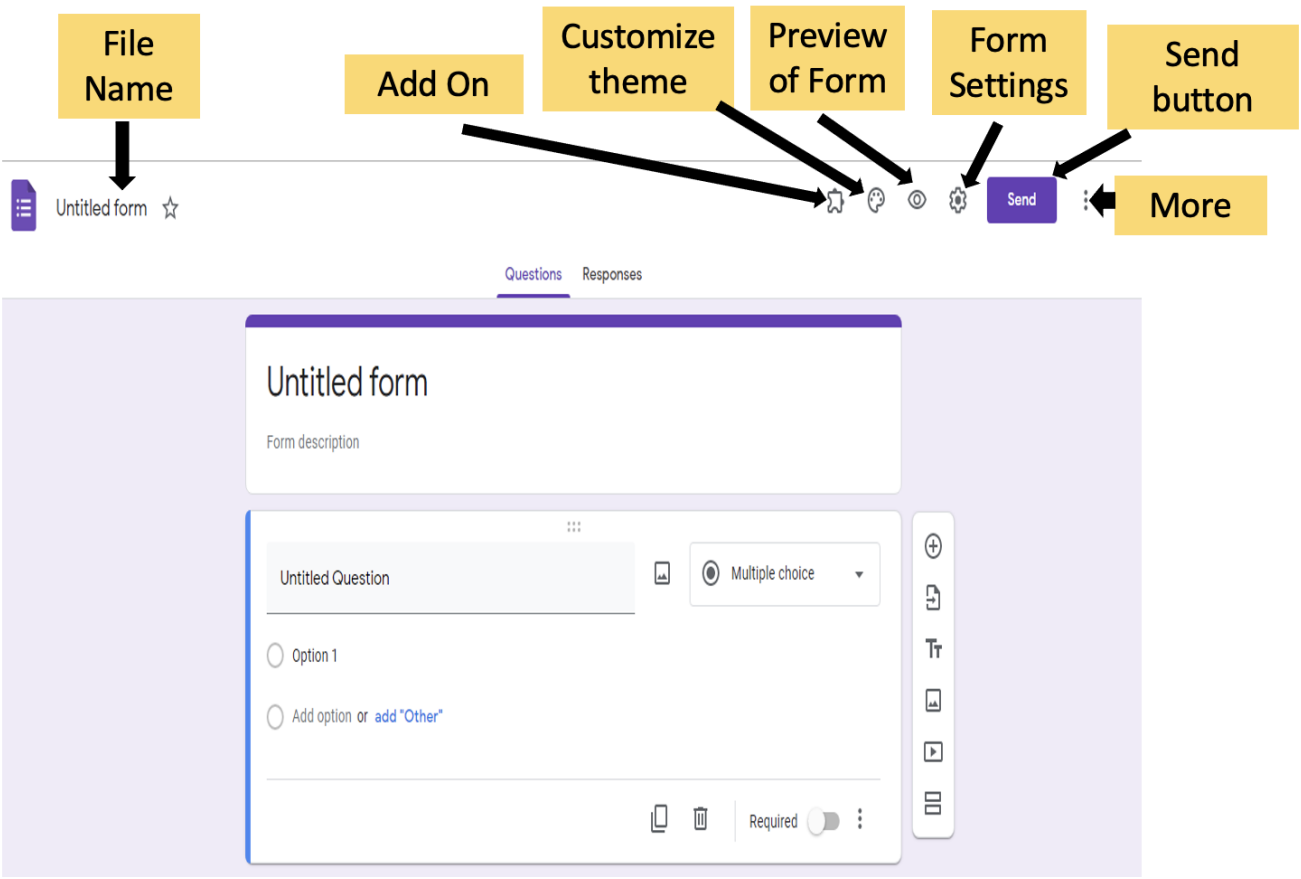


Figure 2: Screen layout

You can rename the file in the upper left by clicking on the name. You can move it by clicking onthe folder icon, or you can start it, to make it easier to find later. In the upper right are more detailed controls. The paint palette icon is **Customize Theme**. You can change the header image, pick new theme colors, or change the font. The eye icon is **Preview of Form**, which lets you see what the live version of the form looks like. It opens in a new browser tab. The gear is **Form Settings**, where you can control how people will interact with the form. Use the **Send** button when you want to email your completed form to people to invite them to fill it out. The three vertical dots are **More**, where you can find **Undo**, **make a copy**, **Add-Ons** and other commands.

Note: Most actions can be undone using **Undo**. You can also use CTRL-Z on your keyboard. CTRL-Y is **Redo**.

Question Types

The essence of Google Forms is creating questions and providing a way for people to answer them. There are ten (10) question types that dictate how a person can answer. Most types haveadditional options under the ⋮ icon in the question box.

Short answer

A very short text answer. Will display a small textbox. You can controlwhat type of answer is valid with **Response validation**.

Paragraph –

A longer text answer. Will display a longer textbox that will line-wrap if necessary.You can control what's valid with **Response validation**.

Multiple choice

A list of options, only one of which can be selected. Displays as radio buttons(circles). 'Other' can be an option, which includes a textbox next to it. You can **Shuffle option order** or **Go to a section based on the answer**.

Checkboxes

A list of options with the potential to select more than one. Displays as checkboxes (squares). 'Other' can be an option, which includes a textbox. **Response validation**here controls how many boxes a respondent can select. You can also **Shuffle option order**.

Dropdown

Displays options as a dropdown menu. Only one option can be selected. Useful ifyou have a lot of items (such as 50 states). While you can add 'Other' as a choice, it will not create a textbox. You can **Shuffle option order** or **Go to a section based on the answer**.

File upload

Allows respondents to upload a file to your Google Drive. You can control whatfile types are allowed, how many files they can upload, and what their maximum size can be.

Linear scale

Displays a row of radio buttons from 0 or 1 to X, with labels on either side. (Forexample, Not Satisfied->Satisfied on a scale of 1-10.)

Multiple choice grid

Displays a grid of radio buttons. Only one option can be selected fromeach row. You can **Shuffle option order** or **Limit to one response per column**.

Checkbox grid

Displays a grid of checkboxes. Any number of boxes can be checked. You can **Shuffle option order** or **Limit to one response per column**.

Date

Displays mm/dd/yyyy which can be filled in by typing or clicking on to display a calendarthat a date can be selected. You can choose to include the time or the year.

Time

By default, displays an empty time to be filled in. ": "and AM/PM dropdown menu. Youcan change this to **Duration**, which asks HRS: MIN: SEC.

Title and Description

On a blank form, it says **Untitled form**. Click here and you can change the title of your form. Click on the **Form description** to describe what the form is to your future respondents. If you choose to write nothing, it will display as blank.

The Question Box

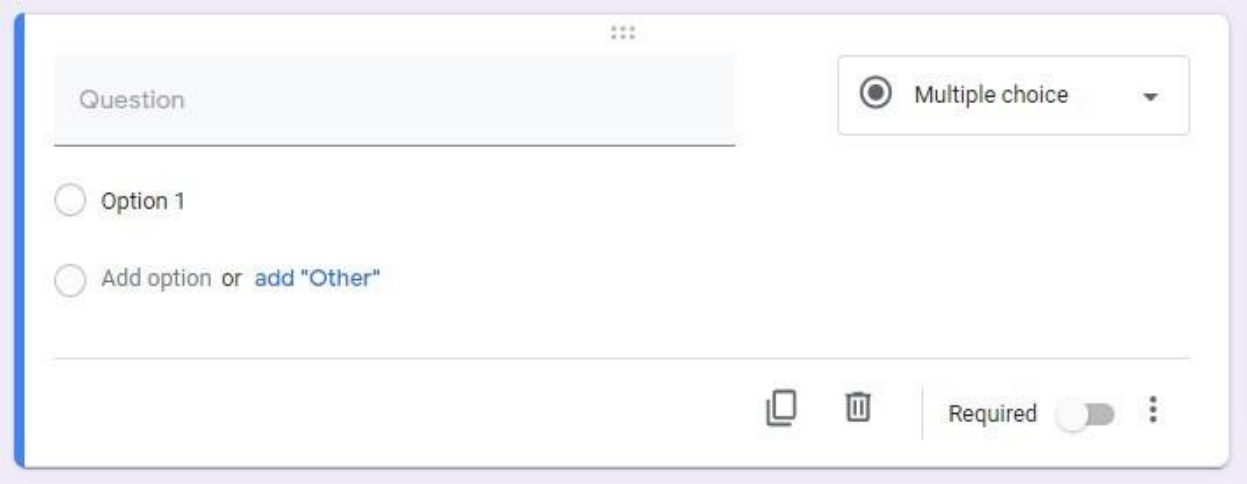


Figure 3: Question box in Google Form

Google Forms starts you off with one question box. No matter what type of question you choose, some of the controls remain the same.

Click on **Untitled Question** or **Question** to put in the text of your question. It can be something as simple as asking for "First Name" or an entire paragraph detailing a math problem.

When you hover over the Question text box, a **photo icon** appears. Click on it to add a photo to that question. You can upload from your computer or Google Drive, or do a Google Imagesearch. Be mindful of copyright if you use a picture from the internet.

The drop-down box in the upper right is where you select your question type.

At the bottom of the question box are icons to **Duplicate** or **Delete** the question.

The **Required** toggle is for making a question optional or required. Click the toggle to require it.Click it again to make it not required.

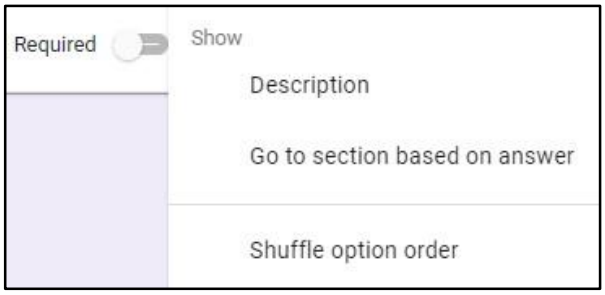


Figure 4: Required toggle

The vertical ellipsis, ⋮, has more options. You can add a **Description** box to your question. Other options that appear on this menu varybased on the type of question you've selected. **Go to a section based on the answer** canbe used in more advanced surveys when you want to direct people to different questions.

The shuffle **option order** will display the options in a different order each time the form is loaded. Useful if you want to avoid biases based on the order. Lessuseful if there's a logical order to your answers (such as age ranges).

The Floating Bar

When you click on a question to edit or add a new one, a bar appears floating to the right of it.

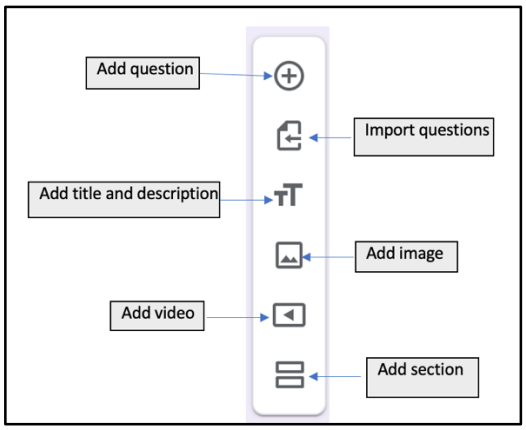


Figure 5: Floating bar

Add question – Add a new question.

Import questions – Import questions from another Google Form.

Add title and description – Create another title and description box like the one at the top.**Add Image** – Add an image between questions (rather than in a question or answer itself). **Add video** – Add a video. Has to be a video on YouTube.
Add section – Add a new section. This adds the **Next** button and takes people to a new page of your form.

If you'd like to rearrange the order of your questions or other elements, hover over one of the boxes until a little grid-shaped icon appears. Click on it and you can drag that element up or down the page and drop it where you want it 

Customizing Theme

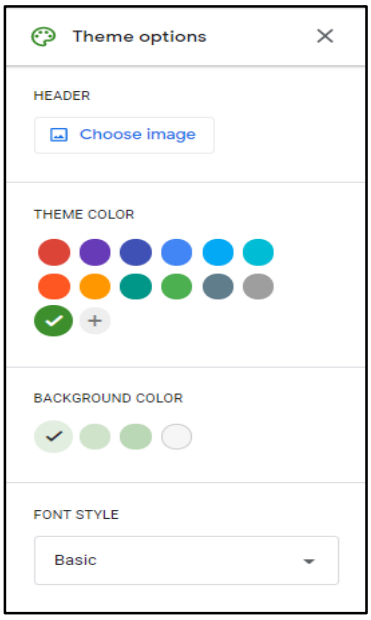


Figure 6: Customizing theme

You can change (or add) an image at the top of your form and customize the colours and fonts by clicking the **Customize Theme** icon in the upper right that looks like a paint palette.

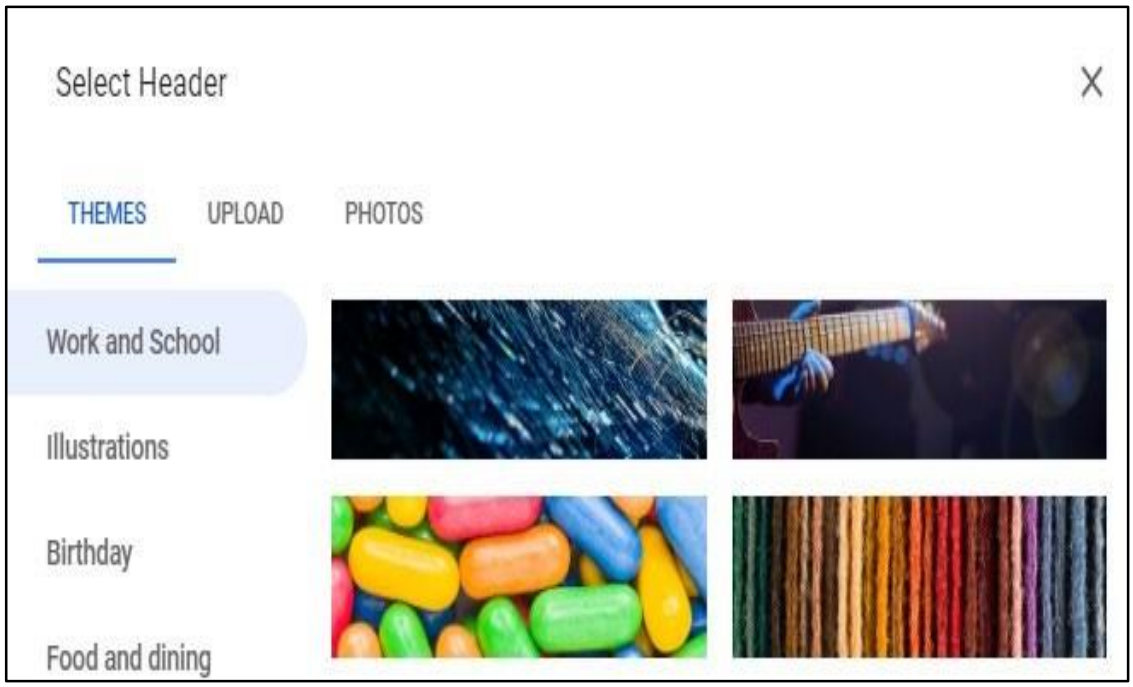


Figure 7: Theme and header

Settings

The gear icon in the upper right is **Settings**. Here you can control whether you're going to collect email addresses and provide receipts, whether you want respondents to be able to fill out the form more than once, whether they can edit their answers after they submit them, and whether they can see how others have responded.

On the **Presentations** tab of the **Settings** box, you can decide if respondents will see a progress bar asthey complete your form, whether you want to shuffle the order of the questions, and if you allow multiple submissions, a link to where they can fill out the form again. You can also customize your **Confirmation message**.

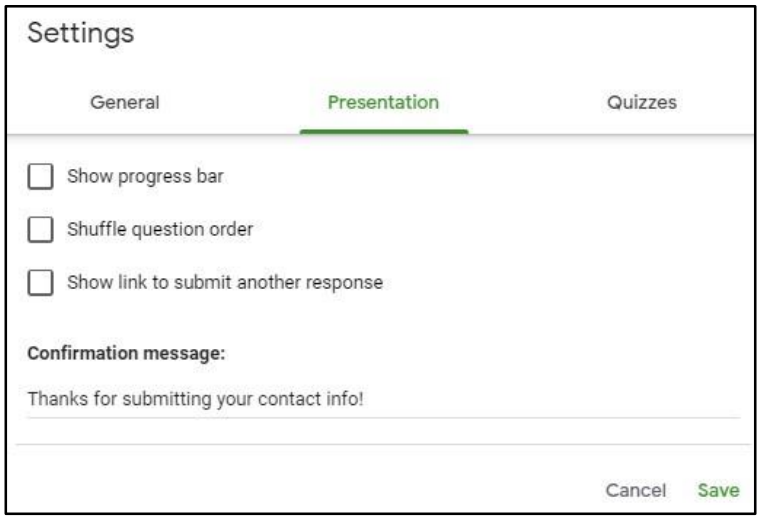


Figure 8: Different settings under presentation

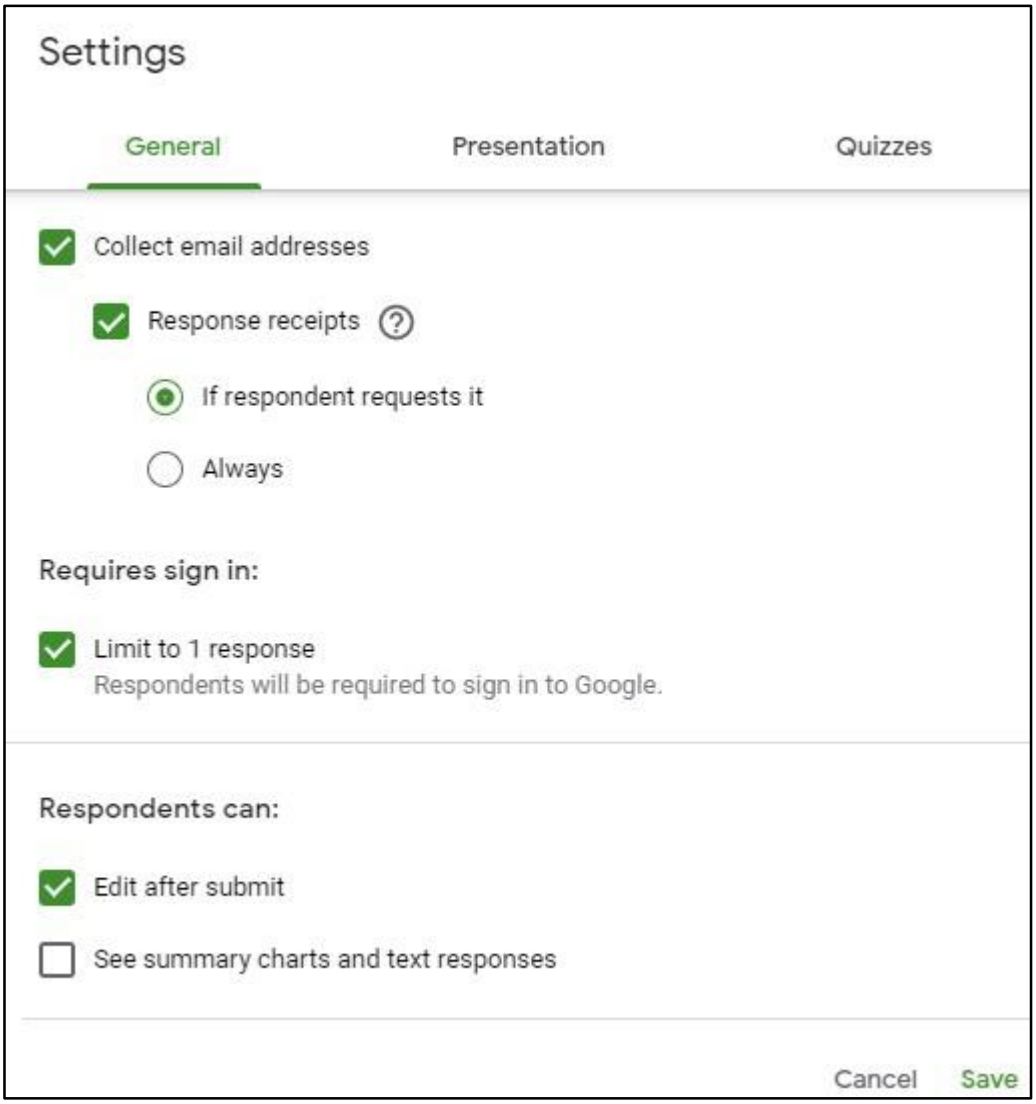


Figure 9: Different settings

Creating a Quiz

To turn a form into a quiz, click the **Settings** gear icon in the upper right. Click on the **Quizzes** tab. Toggle **Make this a quiz**. You're given the option that people answering the quiz can see the results **Immediately after each submission** or, if you'd like to manually grade them yourself, or give a group of people their results all at once, choose **Later, after manual review**.
You can also choose whether people taking your quiz will be able to see **Missed questions**, **Correct answers**, or **Point values**. If it's just a fun quiz, you probably want them all checked. If it's for a class, you might choose to keep some or all of those hidden to reduce cheating amongst classmates. Or if you'd like to give someone a chance to keep trying, maybe show them which questions they missed, but not the correct answers.

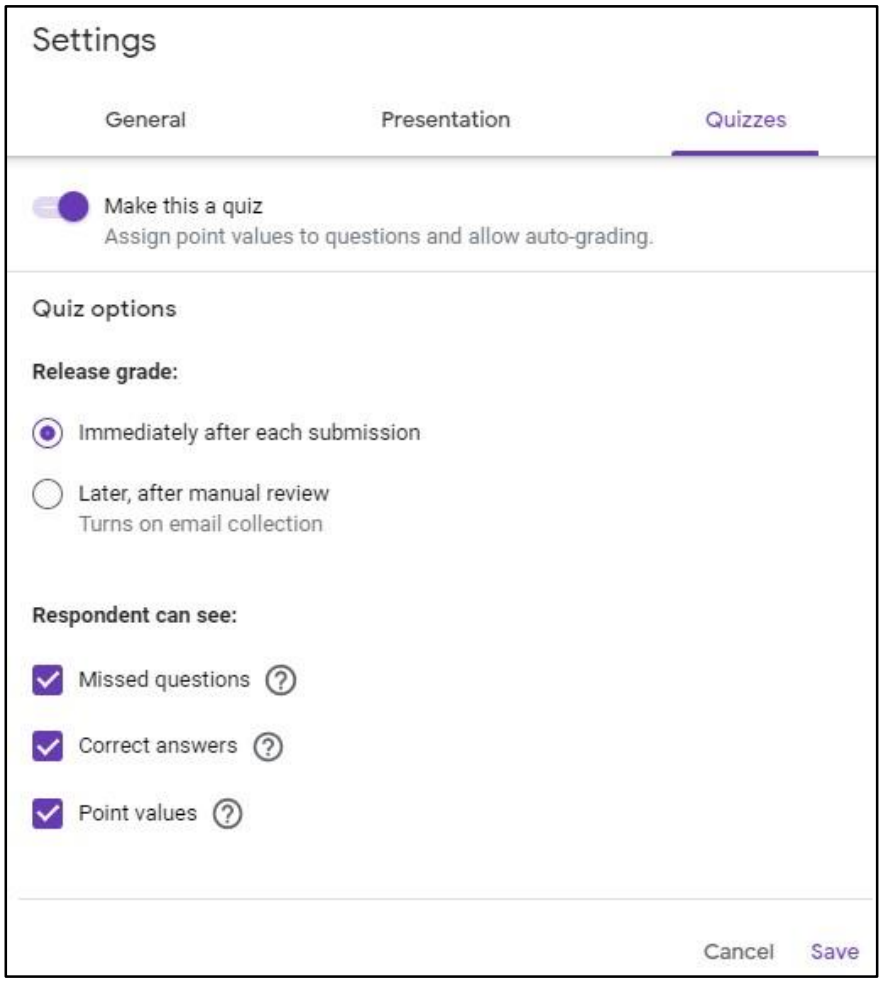


Figure 10: Different settings for quiz

Quiz Questions and Answers

Create sections and questions as you would for a survey, but now your question box will have a place for you to put the correct answer(s).

Figure 11: Quiz question and answer

Clicking on the **Answer key** will change the box to look something like this:

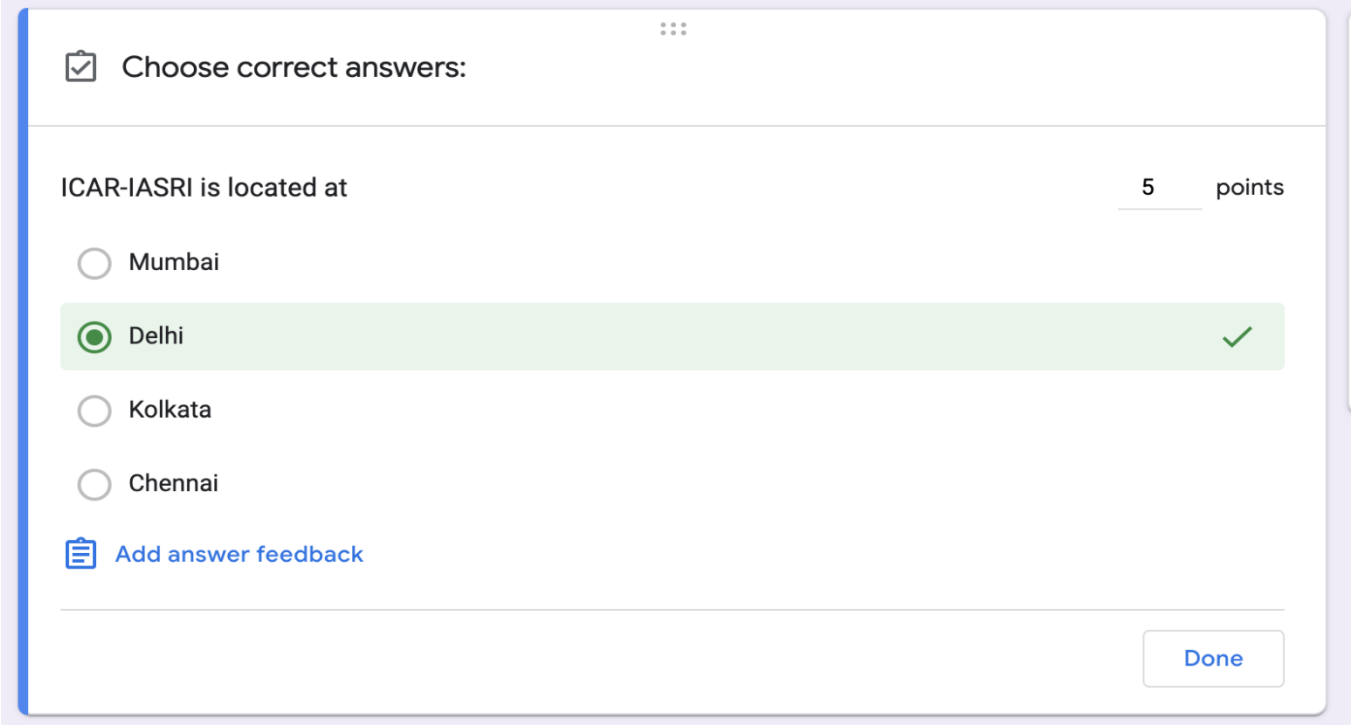


Figure 12: Question template

As this example is Multiple Choice, only one answer can be the correct one. (Use Checkboxes if you want a multiple-choice question with multiple correct answers.) Select the correct answer.

For Short Answer questions, you can add as many correct answers to your **Answer key** as necessary. This is case-sensitive, so you'll want to provide a few options.

Next, in the upper right, you can assign a point value to this question. Click **Add answer feedback** if you'd like to explain why an answer is correct, or why other answers are incorrect. You can add a picture or a link to your feedback.

Preview and Responses

Click the **Preview** eye icon in the upper left of the form to preview the quiz and take it yourself. It's always a good idea to test it out.
Once your quiz is live and some people have taken it (even if it's only you!), in the top middle of the screen next to **Questions** is **Responses**. It will have a number next to it telling you how many responses you've had. Click on **Responses** to get a summary of how your respondents are doing.

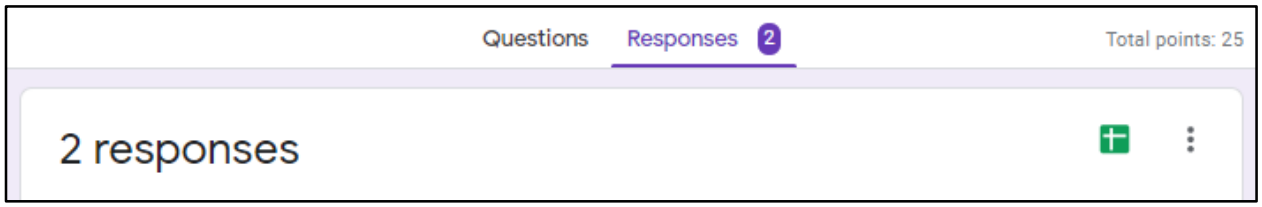


Figure 13: Responses received

Click on the green **Create spreadsheet** icon to do just that. You can create a new Sheets spreadsheet or add to an existing one.

Click on the vertical ellipsis for more options. You can **Get email notifications for new responses**. You can **Download responses** to a CSV file (to import into Excel, for example), you can **Print all responses**, and can **Delete all responses**. You might want to **Delete all responses** after you've done your testing and are ready to share the quiz with others. If you have already created a spreadsheet, the responses will not be deleted from it.

Sending the Form to Potential Respondents

Now that you have a survey, a quiz, or another form, you probably want to share it with other people so they can fill it out. You have many options of ways to share it. First, click on the large **Send** button in the upper right.

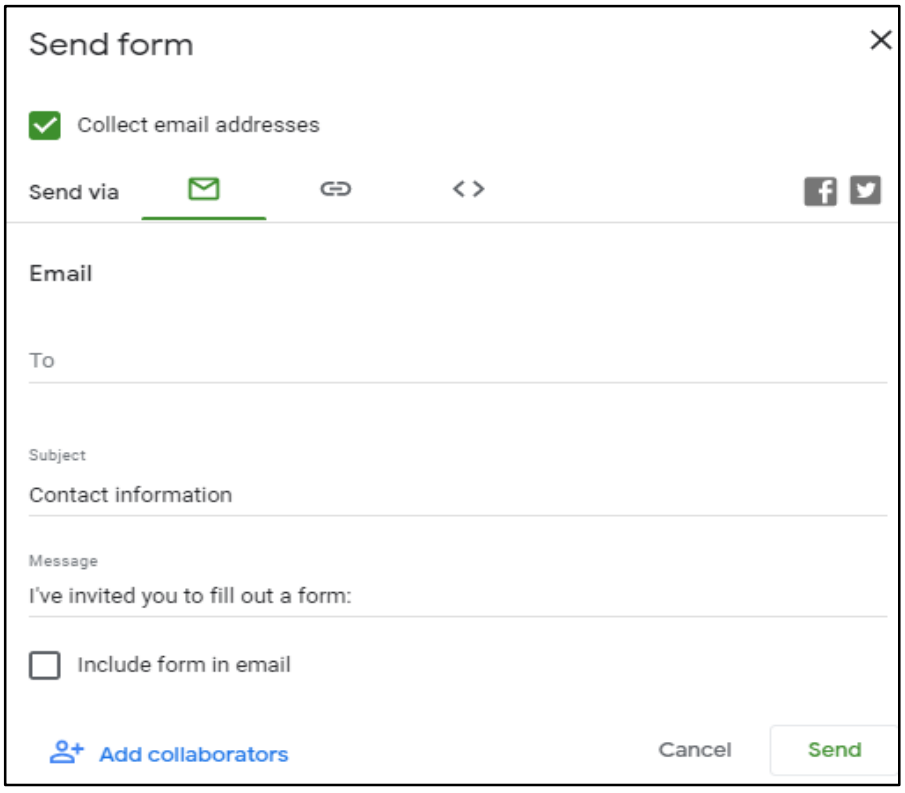


Figure 14: Send form option

Your main options are in the three tabs and the two social media buttons.

Send via Email will email invitations to the people or mailing lists you send it to.

Send via the link will give you a URL you can copy and paste.

Send via embed HTML will give you code to put on your website if you'd like the form toappear there.

Facebook or **Twitter** will start a new post on those platforms under whatever account you're logged in there as.

You can use as many of these options as you want.

Add collaborators is for adding people you want to give permission to view and modify the form.

Cyber Hygiene

Subhash Chand

ICAR-Indian Agricultural Research Institute, New Delhi - 110 012

subhash.chand@icar.gov.in

Agenda

- Overview of Cyber Hygiene
- Common Cyber Hygiene problems.
- Benefit of Cyber Hygiene
- Ways to protect yourself and your organization
- **Cyber Security Pledge to build safe online environment**

Why is Cyber Hygiene Important?

- Cyber hygiene is the cybersecurity equivalent to the concept of personal hygiene in public health literature.

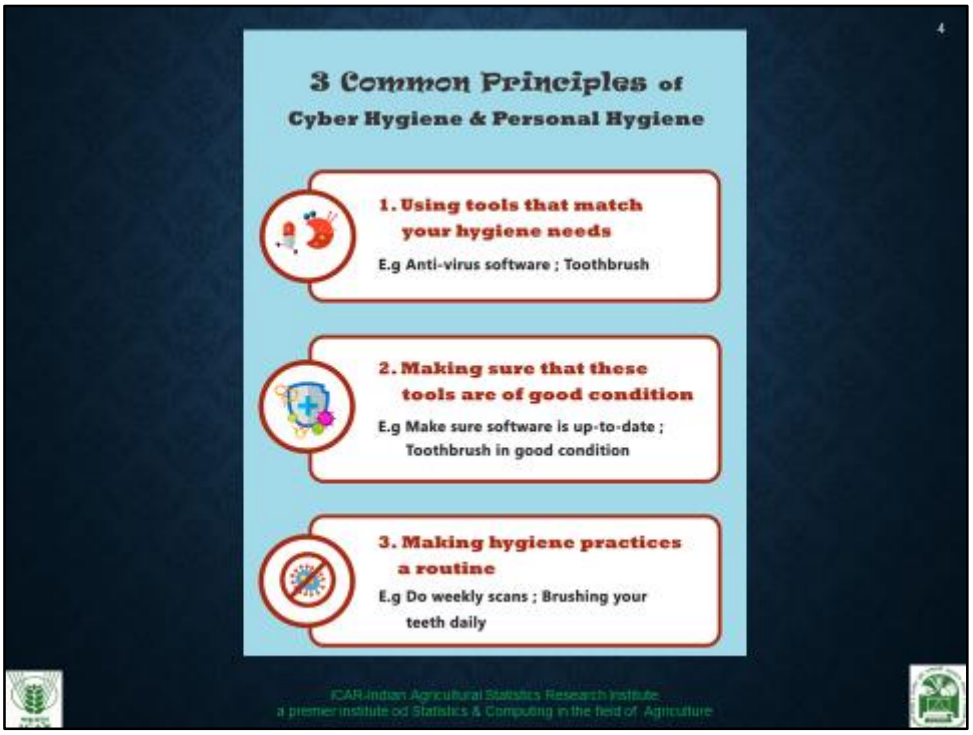


Figure 1: Principles of cyber hygiene

Personal hygiene and Cyber hygiene

- Cyber hygiene is often compared to personal hygiene. Much like an individual engages in certain personal hygiene practices to maintain good health and well-being, cyber hygiene practices can keep data safe and well-protected. In turn, this aids in maintaining properly functioning devices by protecting them from outside attacks, such as malware, which can hinder functionality. Cyber hygiene relates to the practices and precautions users take with the aim of keeping sensitive data organized, safe, and secure from theft and outside attacks.

What is Cyber Hygiene and Why Does it Matter?

- Practices to protect yourself, your organization, and your data
- Financial fraud and identity theft
- Legal compliance
- Business operations
- Reputation and trust

What are Common Cyber Hygiene Problems?

- Loss of data: Hard drives, online cloud storage and SaaS apps that store sensitive data that isn't backed up or maintained can be vulnerable to hacking, corruption, data leaks, and data breaches.
- Misplaced data: Poor cyber hygiene could mean losing data in other ways, while it may not be corrupted or gone for good, it's increasingly common to misplace data due to the myriad of places it can be stored.
- Security breaches: Data breaches are becoming increasingly common, and costly. Spear phishing, whaling attacks, lack of configuration management, and poor network security . This can result in customer identity theft.
- Outdated software: Software applications must have security patches applied regularly to prevent known vulnerabilities. The success of the WannaCry ransomware computer worm is a great example of why patching operating systems is an important part of good cyber hygiene.
- Old security software: Antivirus software and other security software must be kept up to date to keep pace with the ever-changing threat landscape.
- Poor or lack of vendor risk management: Gone are the days where you can think solely about your organization's security posture. Chances are a number of your third-party vendors and service providers have access to your Wi-Fi networks or process sensitive data on your behalf.

Expanding Scope of Cyber

- At work
- At home or while traveling
- Types of data
    - Your own personal information

- Employee personal information
- Organization data
- Third Party data

Data Privacy and Cyber Security

- Data privacy
    - What information is collected
    - How that information is used
    - To whom that information is disclosed
- Cybersecurity
    - Policies and procedures designed to avoid the loss or unauthorized disclosure of collected information

Email Compromise:

Motivation for Threat Actors

- Wire transfers (lead to losses due to fraudulent wire and lawsuits)
- Reconnaissance/targeted spam
- Payroll redirect
- Access sensitive information in the inbox

Ransomware & Cyber Extortion

- Ransomware
    - Business disruption
    - Loss of customers
    - Often deployed after unauthorized access
    - Potential breach notification obligations
- True Extortion
    - Theft or Destruction of Digital Assets
    - Demand for payment

Potential Legal Obligations

- Legal obligations arising from the unauthorized access of personal information
    - Statutory data breach notification obligations to individuals, regulators and business partners
        - State or Central law

- This may include notification to primary owner, key customers, unfriendly parties (e.g., litigation adversaries)

- Contractual obligations to third parties

Data Breach Response

- Have you mitigated the security risk?
- Was "personal information" involved in a "security breach?"
- Do you need to notify impacted persons?
- Do you need to notify state regulators?
- Do you need to notify any federal regulators?

From Common Scenarios

- How should you respond if the incident occurs?
- How could the incident have been prevented?
- How could the impact have been minimized?

Email Compromise Example 1

From: jsmith@ABC.com

To: jdoe@companyx.com

Sent: June 1, 9:30 a.m.

Subject: Secure Message

Dear John Doe,

This is a secured message for you. You can access it by logging onto the following secure website: Link.

Jane Smith
Director of Finance
ABC Company

Jane Smith Director of Finance ABC Company

Email Compromise Example 1 (Contact)

- No compromise at your company, yet.
- Don't click the link, enter any passwords, or open any attachments.
- Was the sender from a known contact? Were you expecting this email?

Email Compromise (2)

From: Jane Smith

To: jdoe@ABC.com

Sent: June 1, 9:30 a.m.

Subject: URGENT

Please complete the attached survey ASAP. You will need to login to the site using your company email address and password. We need this for insurance purposes.

Jane Smith
Director of Finance
ABC Company

Email Compromise (2) (Impersonation)

From: Jane Smith (jsmith99@gmail.com)

To: jdoe@ABC.com

Sent: June 1, 9:30 a.m.

Subject: URGENT

Please complete the attached survey ASAP. You will need to login to the site using your company email address and password. We need this for insurance purposes.

Jane Smith
Director of Finance
ABC Company

Email Compromise (2) (Impersonation)

- Always confirm the identity of the sender and beware of impersonators.
- Add an "External Email" banner
- Train employees regularly

Email Compromise (3)

From: jsmith@ABC.com

To: jdoe@ABC.com

Sent: June 1, 9:30 a.m.

Subject: Survey

Please complete the attached survey ASAP. You will need to login to the site using your company email address and password. We need this for insurance purposes.

Jane Smith
Director of Finance
ABC Company

Email Compromise (3) (Active Compromise)

- Now we know we have a problem.
- Additional facts learned upon investigation:
    - Employee clicked a link in a phishing email and entered their username and password for their work email account.
    - Threat actor accessed the account
        - Searched for emails with wire instructions for outstanding invoices and emailed the customer with changes to the instructions.
        - Established rules that deleted incoming emails from that customer.
        - Established forwarding rules to send emails to an external account.
        - Later sent further phishing emails to contacts inside and outside the organization.
        - Several employees clicked on the links and entered credentials, including employees who routinely send spreadsheets containing employee information.

Email Compromise (3)
(Active Compromise)

- How should you respond if the incident occurs?
    - Follow your incident response plan
    - Remediation
        - Password changes
        - Remove rules
        - Detect any fraudulent wire transfers
    - Investigation
        - Determine if any personal information may have been accessed or acquired
    - Notification
        - Comply with any applicable notification requirements
        - Notify customers or contacts who may be at risk
    - Prevention
        - Mitigate the risk that led to the compromise

Email Compromise (3)
(Active Compromise)

- How could the incident have been prevented?
    - Methods to prevent the first compromise
        - External banner

- Training
- Multifactor authentication
- Blocking certain foreign IP addresses or unapproved IP addresses

Email Compromise (3) (Active Compromise)

- How could the impact have been minimized?
  - Earlier detection
    - Alerts when rules are created
    - Alerts regarding suspicious logins from unknown IP addresses
  - Regular password changes (and rule removals)
  - Reduce or eliminate personal information from email accounts
    - Do not send actual copies of sensitive records.
    - Send through hyperlinks to internal documents or through links to secure file transfer sites
    - Delete older emails

Ransomware Example

- After turning on your workstation in the morning, you see that all of your files are encrypted.
- A single text file is readable and contains a ransom note.
- Other workstations are also encrypted.

Ransomware Example

- How should you respond if the incident occurs?
  - Report to the information security team
    - Individual employees should not pay the ransom
  - Balancing operational needs with preserving forensic evidence
  - Disconnect impacted systems from the network and from the internet
  - Leave the machine alone.
    - Do not turn off the machine and do not run any anti-virus scans.
  - Determining availability of backups
  - Forensic investigation

Ransomware Example

- How could the incident have been prevented?

- Forensic investigation is key to determining entry point
- Securing remote desktop connections
    - Multifactor authentication
    - Blocking foreign or unapproved IP addresses
- Stronger network passwords
- Ensuring patches and updates are installed

Ransomware Example

- How could the impact have been minimized?
    - Backups
        - Not encrypted
        - Up to date
    - Preserving forensic evidence to aid with legal compliance
        - Proving no data was exfiltrated from the company's servers

Avoiding Scams

- Calls/texts/emails from:
    - The IRS
    - The utility company
    - Law enforcement or the courts
    - Tech support
- Red flags
    - Urgent
    - Payment in gift cards
- When in doubt, call the actual entity

Credit Monitoring/Identity Theft Protection

- Watch your credit report
- Review your bank statements
- Fraud alerts
- Credit freeze
- Take advantage of credit monitoring services when offered

What are the benefits of cyber hygiene?

Good cyber hygiene offers several benefits that ultimately put your organization in a better position to defend against cyberattacks.

What are the benefits of cyber hygiene?

- **Locate unmanaged assets:** You can't protect what you can't see. That's why an accurate inventory of all your assets is the foundation for strong cybersecurity.
- **Protect customer data:** Cyber hygiene supports a range of proven security practices, such as patch management, password discipline, appropriate administrator privileges, and other measures that improve data protection.
- **Find outdated administrator privileges:** It's easy to lose track of administrative rights as people move from one role or department to another or leave the organisation.
- **Identify rogue software:** Remote work has led many workers to install unsanctioned software on the devices and endpoints they use to connect to your network.
- **Meet compliance requirements:** By identifying and prioritizing security risks and empowering IT teams to quickly remediate them, cyber hygiene makes it easier to track and report your organization's security status and ensures it's always aligned with regulatory and compliance requirements.

What are cyber hygiene best practices?

- **Maintain an IT asset inventory:** When a criminal attacker gets inside your network, they will look for your organization's most valuable assets the same way a home burglar looks for cash and jewelry.
- **Maintain complex passwords:** Complex passwords that are changed regularly are a strong first line of defense against an array of security threats.
- **Regularly update software:** Poor patch management is essentially an open-door policy for cybercriminals.
- **Control admin privileges:** As high-level administrative privileges pose one of the biggest security risks in any organization, it's important to give admin-level access to programs and systems only to those who need it.
- **Regularly back up data:** Backup procedures should be performed on a regular schedule and verified to confirm their integrity.
- **Manage end-of-life systems:** End-of-life systems are computer hardware and software that are no longer supported with security patches and updates from their manufacturers or developers.
- **Implement an incident response plan:** Businesses of all sizes should have an incident response plan in place to mitigate the damage and minimize the downtime from an attack.

What are some examples of good cyber hygiene?

- One common example of good cyber hygiene would be practicing vigilance when sending or receiving emails. Because of its ubiquitous use in every organization, email has become a popular way for cybercriminals to disseminate malware to unsuspecting users. A typical tactic

is to pose as a person or business the recipient knows and trick them into clicking on a malicious link that steals their credentials or downloads malware onto their computer.

How does business cyber hygiene differ from individual cyber hygiene?

- Individual or personal cyber hygiene is concerned with protecting an individual from security threats while business cyber hygiene mitigates risk for an organization.
- Some practices are common to both, such as using complex passwords, running antivirus software, being vigilant when responding to emails, and backing up data.

What is an ideal cyber hygiene checklist?

An ideal cyber hygiene checklist is informed by the best practices outlined earlier and includes the following tasks:

- Create and maintain an inventory of all hardware and software on the organization's network.
- Identify your "crown jewel" data, where it's located, and who has access to it.
- Set and enforce strong password policies.
- Limit administrative-level privileges to those who need them.
- Regulate how end users install software, either by limiting their access to only trusted programs or requiring IT approval for any installation.
- Keep operating systems and software applications up-to-date and apply patches promptly.
- Implement a process for regularly performing, verifying, and testing data backups. Keep multiple copies and back up both on-premises and in the cloud.
- Track end-of-life systems and remove them from use.
- Create a vendor risk-management plan outlining agreed-upon behaviors, access, and service levels.
- Educate employees on good cyber hygiene practices, including password management, email vigilance, and how to use the organization's network securely.

More Tips for Good Cyber Hygiene

- Multifactor authentication whenever possible
- Lock down access to email accounts or remote desktop applications that can be accessed anywhere in the world
- Have active backups separate from your primary systems
- Encrypt laptops, phones, and storage devices (password protection is not enough)

More Tips for Good Cyber Hygiene

- Do not install unapproved software
- Do not permit automated forwarding of work emails to personal accounts

- Obtain verbal confirmation for changes to wire instructions
- Maintain antivirus and antimalware software
- Policies for ensuring terminated employees cannot access company information
- Enable and preserve audit logging for all available systems

More Tips for Good Cyber Hygiene

- Keep tabs on your vendors
- Conduct periodic risk assessments
- Have an incident response plan
- Delete data after it is no longer needed. More data more risk.
- Cyberinsurance

Information security awareness for Govt. Employee

Government staff must ensure that Confidentiality, Privacy and Commercial Sensitivity Standards, Practices and Requirements are followed to use the organization's equipment, storage, retrieval and access of information to/from systems and networks.

- Please answer the following questions:
- Do you have an official e-mail ID?
- Do you follow internet ethics?
- Do you have an access policy in your organization?
- Do you have any policy guidelines for accessing /using the organization resources?
- Do you maintain any confidential data in your organization?
- Do you make any download on your official computer?
- Do you have security policies in your organization?

If your answer is 'yes' to all these questions then go through the guidelines provided to safeguard yourself and organized resources.

Cyber Security Pledge to build safe online environment

- https://infosecawareness.in/pledge#

Information Security Awareness

CERT-in: Indian - Computer Emergency Response Team (cert-in.org.in)

- CERT-In is operational since 2004. The constituency of CERT-In in the Indian Cyber Community. CERT-In is the national nodal agency for responding to computer security incidents as and when they occur.

- In the Information Technology Amendment Act 2008, Cert-In has been designed to serve as the national agency to perform the following functions in the area of cyber security.;
- Collection, analysis and dissemination of information on cyber incidents.
- Forecast and alerts of cyber security incidents.
- Emergency measures for handling cyber security incidents.
- Coordination of cyber incident response activities.
- Issue guidelines, advisories, vulnerability notes and whitepapers relating to information security practices, procedures, presentation, response and reporting of cyber incidents.
- Such other functions relating to cyber security as may be prescribed.

Cyber Swachhta Kendra

- The " Cyber Swachhta Kendra " (Botnet Cleaning and Malware Analysis Centre) is a part of the Government of India's Digital India initiative under the Ministry of Electronics and Information Technology (MeitY) to create a secure cyber space by detecting botnet infections in India and to notify, enable cleaning and securing systems of end users so as to prevent further infections. The " Cyber Swachhta Kendra " (Botnet Cleaning and Malware Analysis Centre) is set up in accordance with the objectives of the "National Cyber Security Policy", which envisages creating a secure cyber eco system in the country. This centre operates in close coordination and collaboration with Internet Service Providers and Product/Antivirus companies. This website provides information and tools to users to secure their systems/devices. This centre is being operated by the Indian Computer Emergency Response Team (CERT-In) under provisions of Section 70B of the Information Technology Act, 2000.

Cyber Swachhta Kendra: Security Tools

- Free Bot Removal Tool - For Microsoft Windows
- Free Bot Removal Tool - For Android
- Alerts
- **Security Best Practices**
- Digital Payment Suraksha
- Digital Payment Brochure For MerChants
- Digital Payment Brochure For Customers
- Digital Payment Suraksha - Awareness Video in English
- Security for personal computer

Security Tips for common users

Desktop Security

Broadband Security

USB Security

Avoid Phishing Attacks

Mobile Phone Security

Information Security Awareness resources

infosecawareness.in/handbooks

https://infosecawareness.in/brochures

Govt Employee - Information Security Awareness (infosecawareness.in)

https://infosecawareness.in/gallery/handbooks/Cyber-Awareness/Cyber-Safety-Security-English-Hindi.pdf

https://infosecawareness.in/gallery/brochures/Himachal-Pradesh-Police/Broadband-Security.pdf

https://infosecawareness.in/gallery/brochures/ISEA/Computer-Virus.pdf

https://infosecawareness.in/gallery/brochures/Himachal-Pradesh-Police/Desktop-Security.pdf

https://infosecawareness.in/gallery/brochures/ISEA/Email.pdf

https://infosecawareness.in/gallery/brochures/ISEA/File-Sharing-Downloading-Uploading.pdf

https://infosecawareness.in/gallery/brochures/ISEA/Internet-Ethics.pdf

https://infosecawareness.in/gallery/brochures/ISEA/Mitigating-Ransomware.pdf

- https://infosecawareness.in/gallery/brochures/Himachal-Pradesh-Police/Mobile-Security.pdf
- https://infosecawareness.in/gallery/brochures/ISEA/Password-Threats.pdf
- https://infosecawareness.in/gallery/brochures/ISEA/Phishing.pdf
- https://infosecawareness.in/gallery/brochures/ISEA/Ransomware.pdf
- https://infosecawareness.in/gallery/brochures/ISEA/Spam.pdf
- https://infosecawareness.in/gallery/brochures/ISEA/Tips-while-you-download.pdf
- https://infosecawareness.in/gallery/brochures/Himachal-Pradesh-Police/USB-Security.pdf
- https://infosecawareness.in/gallery/brochures/Himachal-Pradesh-Police/WiFi-Security.jpg
- https://infosecawareness.in/cartoon-stories
- https://infosecawareness.in/handbooks
- https://infosecawareness.in/brochures
- https://infosecawareness.in/alerts

Cyber/Information Security Concepts

Dr. Mukesh Kumar
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
mukesh.kumar@icar.gov.in

What is Security?

"The quality or state of being secure—to be free from danger"
A successful organization should have multiple layers of security in place:

- Physical security
- Personal security
- Operations security
- Communications security
- Network security/Information security

Cyber Security:

- **Computer security**, is known as **cyber security** or **IT security**, is the protection of information systems from theft or damage to the hardware, the software, and to the information on them, as well as from disruption or misdirection of the services they provide.
- Cyber security is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access.
- Cyber security strives to ensure the attainment and maintenance of the security properties of the organization and user's assets against relevant security risks in the cyber environment.
- Information security: a "well-informed sense of assurance that the information risks and controls are in balance." —Jim Anderson, Inovant (2002)
- Cyber security is also associated with the technical term, information security, which is explained in federal law as protecting information and information systems from illegal access, use, disclosure, disruption, modification, or damage in order to provide integrity, confidentiality and availability.
- Governments departments, military, corporations, various institutions, hospitals and other originations gather process and store huge confidential information on computers and pass on that data across networks to other computers.
- In view of complexity of cyber-attacks, there is need more attention to protect sensitive information and personal information, as well as protect national security.

The History of Information Security

- Began immediately after the first mainframes were developed
- Groups developing code-breaking computations during World War II created the first modern computers

- Physical controls to limit access to sensitive military locations to authorized personnel
- Fundamental in defending against physical theft, espionage, and disruption

In the 1960

- Advanced Research Procurement Agency (ARPA) began to examine feasibility of redundant networked communications
- Larry Roberts developed ARPANET from its inception

The 1970s and 80s

- ARPANET grew in popularity as did its potential for misuse
- Fundamental problems with ARPANET security were identified
    - No safety procedures for dial-up connections to ARPANET
    - Non-existent user identification and authorization to system
- Late 1970s: microprocessor expanded computing capabilities and security threats

R-609

- Information security began with Rand Report R-609 (paper that started the study of computer security)
- Scope of computer security grew from physical security to include:
    - Safety of data
    - Limiting unauthorized access to data
    - Involvement of personnel from multiple levels of an organization

The 1990s

- Networks of computers became more common; so too did the need to interconnect networks
- Internet became first appearance of a global network of networks
- In early Internet deployments, security was treated as a low priority

The Present

- The Internet brings millions of computer networks into communication with each other—many of them unsecured
- Ability to secure a computer's data influenced by the security of every computer to which it is connected

Information Security Components

- The protection of information and its critical elements, including systems and hardware that use, store, and transmit that information
- Necessary tools: policy, awareness, training, education, technology
- C.I.A. triangle was standard based on confidentiality, integrity, and availability
- C.I.A. triangle now expanded into list of critical characteristics of information

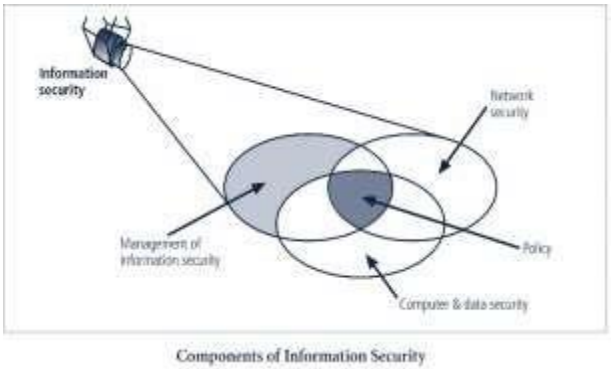


Figure 1: Component of Information Security

Critical Characteristics of Information

The value of information comes from the characteristics it possesses:

- Availability
- Accuracy
- Authenticity
- Confidentiality
- Integrity
- Utility
- Possession

- The National Security Telecommunications and Information Systems Security Committee (NSTISSC) was established by President Bush under National Security Directive 42 (NSD 42) entitled, "National Policy for the Security of National Security Telecommunications and Information Systems," dated July 5, 1990.
- NSTISSC Security Model

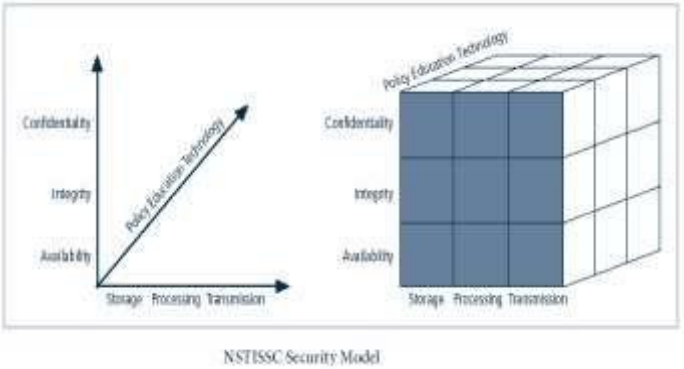


Figure 2: NSTISSC Security Model

Model Key Security Concepts

- Access: Authorized users have legal access to a system, whereas hackers have illegal access to a system.
- Asset: The organizational resource that is being protected. An asset can be logical, such as a Web site, information, or data; or an asset can be physical, such as a person, computer system

- Attack: An intentional or unintentional act that can cause damage to or otherwise compromise information and/or the systems that support it.
- Control, safeguard, or countermeasure: Security mechanisms, policies, or procedures that can successfully counter attacks, reduce risk, resolve vulnerabilities.
- Exploit: Threat agents may attempt to exploit a system or other information asset by using it illegally for their personal gain.
- Exposure: Information security, exposure exists when a vulnerability known to an attacker is present.
- Loss: A single instance of an information asset suffering damage or unintended or unauthorized modification or disclosure.
- Protection profile or security posture: It includes policy, education, training and awareness, and technology, that the organization implements to protect the asset.
- Risk: The probability that something unwanted will happen.
- Subjects and objects: A computer can be either the subject of an attack—an agent entity used to conduct the attack—or the object of an attack—the target entity.
- Threat: A category of objects, persons, or other entities that presents a danger to an asset.
- Vulnerability: A weaknesses or fault in a system or protection mechanism that opens it to attack or damage.

Information Security Breach

Any unauthorized access that leads to disclosure, destruction or any change in information is considered as a security breach.

- An unauthorized individual is accessing confidential data about another person, such as residential address, contact number, credit card and bank account etc. form aninsurance company database.
- A private advertising company is accessing the subscriber details stored in the databases of an email service providing company and sending out spam email to all subscribers. This activity is occurring without the knowledge of the service providing company.
- The computer systems of an organization are being used to launch attacks on the network of another organization. This amounts to a breach in security of both organizations.
- Someone spread a virus on an organization's network, which increase the volume of traffic on network. As result, network will be slow and may be halt.
- Stolen of hard disk, broken of disk, also the security breach
- Someone host the website on another organization's server

Principles of Cyber/Information Security

The Information Technology Industry Council (ITI) provides complete set of cyber security principles for industry and government. ITI comprise the world's leading technology companies, both producers and consumers of cyber security products and services. ITI has developed six principles to improve cyber security.

- Organizations must leverage public-private partnerships and build upon existing initiatives and resource commitments.
- Organizations reflect the borderless, interconnected, and global nature of today's cyber environment. Cyberspace is international and unified system that spans geographic borders and traverses national jurisdictions.
- Firms must be able to adapt rapidly to emerging threats, technologies, and business models and be based on effective risk management. Efforts to improve cyber security must be based on risk management.
- Efforts to improve cyber security must focus on awareness. The principle of cyber security is to focus on raising public awareness. Cyberspace's owners include consumers, businesses, governments, and infrastructure owners and operators.
- Efforts to improve cyber security must more directly focus on bad actors and their threats.
- In cyberspace, as in the physical world, adversaries use instruments to do crime, spying, or warfare. Cyber security policies must allow governments to better use current laws, efforts, and information sharing practices.

Needs to be Secured

Organizations needs to focus on securing the following:

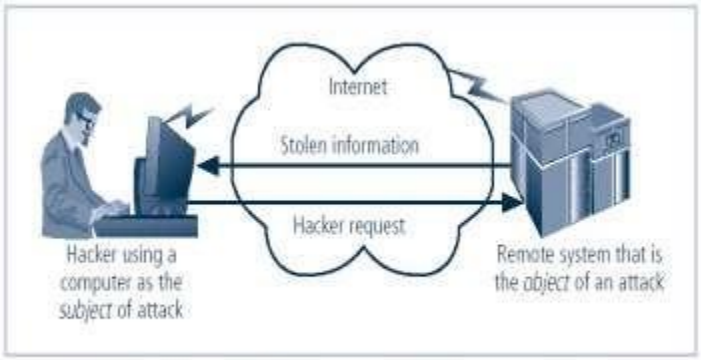
- Software,
- Databases,
- Data files,
- Communication system,
- Computer network
- Web applications against unauthorized access.

Only people who are authorized to access particular information should be allowed to access it.

Components of an Information System

Information System (IS) is entire set of software, hardware, data, people, procedures, and networks necessary to use information as a resource in the organization Securing Components

- Computer can be subject of an attack and/or the object of an attack
  - When the subject of an attack, computer is used as an active tool to conduct attack
  - When the object of an attack, computer is the entity being attacked

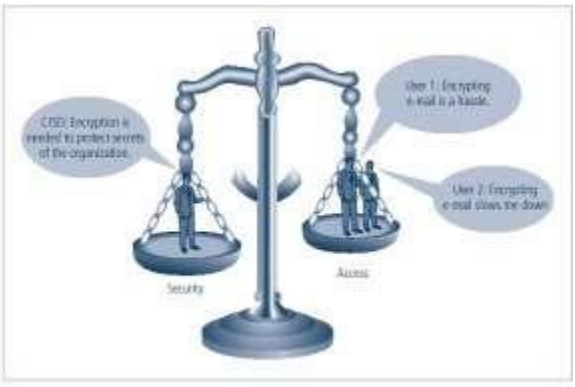


Computer as the Subject and Object of an Attack

Figure 3: Subject of an attack and/or the object of an attack

Balancing Information Security and Access

- Impossible to obtain perfect security—it is a process, not an absolute
- Security should be considered balance between protection and availability
- To achieve balance, level of security must allow reasonable access, yet protect against threats



Balancing Information Security and Access

Figure 4: Balancing Information Security and Access

Approaches to Information Security Implementation: Bottom-Up Approach

- Grassroots effort: systems administrators attempt to improve security of their systems
- Key advantage: technical expertise of individual administrators
- Seldom works, as it lacks a number of critical features:
  - Participant support
  - Organizational staying power

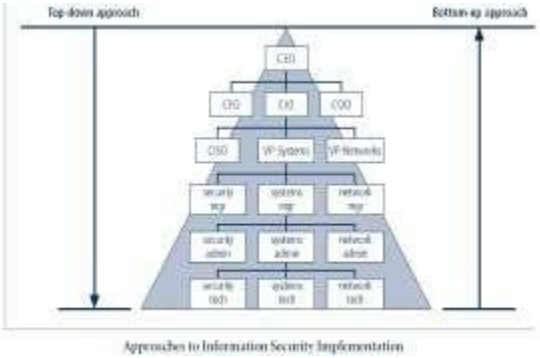


Figure 5: Bottom Up Approach

Approaches to Information Security Implementation: Top-Down Approach

- Initiated by upper management
  - Issue policy, procedures and processes
  - Dictate goals and expected outcomes of project
  - Determine accountability for each required action
- The most successful also involve formal development strategy referred to as systems development life cycle

The Security Systems Development Life Cycle

The same phases used in traditional SDLC may be adapted to support specialized implementation of an IS project

- Identification of specific threats and creating controls to counter them
- SecSDLC is a coherent program rather than a series of random, seemingly unconnected actions

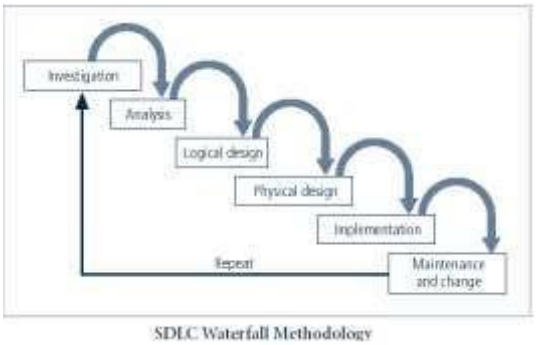


Figure 6: Security Systems Development Life Cycle

Investigation

- Identifies process, outcomes, goals, and constraints of the project
- Begins with enterprise information security policy
- Organizational feasibility analysis is performed

Analysis

- Documents from investigation phase are studied

- Analyzes existing security policies or programs, along with documented current threats and associated controls
- Includes analysis of relevant legal issues that could impact design of the security solution
- The risk management task begins

Logical Design

- Creates and develops blueprints for information security
- Incident response actions planned:
    - Continuity planning
    - Incident response
    - Disaster recovery
- Feasibility analysis to determine whether project should continue or be outsourced

Physical Design

- Needed security technology is evaluated, alternatives generated, and final design selected
- At end of phase, feasibility study determines readiness of organization for project

Implementation

- Security solutions are acquired, tested, implemented, and tested again
- Personnel issues evaluated; specific training and education programs conducted
- Entire tested package is presented to management for final approval

Maintenance and Change

- Perhaps the most important phase, given the ever-changing threat environment
- Often, reparation and restoration of information is a constant duel with an unseen adversary
- Information security profile of an organization requires constant adaptation as new threats emerge and old threats evolve.

Security Professionals and the Organization

- Wide range of professionals required to support a diverse information security program.
- Senior management is key component; also, additional administrative support and technical expertise required to implement details of IS program.

Understanding Threats towards organization

- **Security Systems Development Life Cycle** - involves the unauthorized use, duplication, and distribution of protected IP. For example, piracy being the unauthorized reproduction or distribution of a copyrighted work or other IP.
- **Espionage or Trespass** - occurs when an unauthorized individual attempts to gain illegal access to organizational information.
- **Human errors / failures** - Human error is an unintentional action or decision. Violations are intentional failures – deliberately doing the wrong thing. There are three types of human error: slips and lapses, and mistakes. These types of human error can happen to even the most experienced and well-trained person.
- **Information Extortion** - Cyber extortion occurs when cybercriminals threaten to disable the operations of a target business or compromise its confidential data unless they receive a payment. The two most common types of cyber extortion are ransomware and DDoS (Distributed Denial of Service) attacks.
- **Internal Threats** - An internal threat refers to the risk of somebody from the inside of a company who could exploit a system in a way to cause damage or steal data. These kinds of threats are particularly troubling, as employees are expected to be trusted individuals that are granted extended privileges, which can easily be abused.
- **External Threats etc**.- An external threat refers to the risk of somebody from the outside of a company who attempts to exploit system vulnerabilities through the use of malicious software, hacking, sabotage or social engineering.
- **Botnet:** A combination of the words "robot" and "network", a Botnet is a group of private computers infected with malicious software and controlled as a group without the owners' knowledge to deliver large volumes of spam, carryout DDoS attacks and steal data/credentials. Botnets have the collective computing power to act as a force multiplier for groups looking to disrupt or break into targets' systems.
- **Cryptojacking:** Cryptojacking is the unauthorized use of someone else's computer to mine cryptocurrency. Hackers usually trick the victim into clicking on a malicious email link which loads cryptomining code on the computer, or by infecting a website or online ad with code that auto-executes once loaded in the victim's browser. The cryptomining code then works in the background as unsuspecting victims use their computers normally. The victim may notice the computer's slower performance while they're working, but otherwise it can go undetected.
- **Ransomware:** is malicious malware that threatens to publish or steal victims' data or prevents users from accessing their systems until a ransom is paid. Ransomware has grown to be one of the biggest problems in network security because it can paralyze large organizations and even whole cities, with Atlanta and Baltimore as recent examples.

Understanding Attacks towards organization

- **Malicious Activities** - Malware, or malicious software, is any program or file that is intentionally harmful to a computer, network or server. Types of malware include

computer viruses, worms, Trojan horses, ransomware and spyware.

- **Hoaxes** - are emails typically arriving in chain letter fashion that often describe impossible events, highly damaging malware or urban legends. Their intent is to frighten and mislead recipients and get them to forward to friends.
- **Backdoors** - A backdoor is a malware type that negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to remotely issue system commands and update malware.
- **Password Cracks**- Password cracking is the process of attempting to gain Unauthorized access to restricted systems using common passwords or algorithms that guess passwords. In other words, it's an art of obtaining the correct password that gives access to a system protected by an authentication method
- **Man-in- Middle Attacks** - A man in the middle (MITM) attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway. The goal of an attack is to steal personal information, such as login credentials, account details and credit card numbers. Targets are typically the users of financial applications, SaaS businesses, e-commerce sites and other websites where logging in is required.
- **Spam:** unsolicited usually commercial messages (such as emails, text messages, or Internet postings) sent to a large number of recipients or posted in a large number of places
- **Mail Bombing** - A mail bomb is the sending of a massive amount of e-mail to a specific person or system. A huge amount of mail may simply fill up the recipient's disk space on the server or, in some cases, may be too much for a server to handle and may cause the server to stop functioning.
- **Pharming** - Pharming is like phishing in that it is a threat that tricks users into exposing private information, but instead of relying on email as the attack vector, pharming uses malicious code executed on the victim's device to redirect to an attacker-controlled website.
- **Worms:** A worm is self-replicating malware that duplicates itself to spread to uninfected computers. Its primary function is to infect other computers while remaining active on infected systems.
- **Phishing:** A phishing attack involves using email to trick employees into believing a message is from a legitimate, trustworthy source. Then, when they click a link in the email or open an attachment, their computer becomes infected. The phisher could be someone pretending to be from the employee's company, or perhaps a company he/she does business with.
- **DDoS (Distributed Denial of Service) Attack:** the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the internet. DDoS is typically accomplished by flooding the targeted machine from many different sources with superfluous requests in an attempt to overload systems and prevent some or all

legitimate requests from being fulfilled.

- **APT (Advanced Persistent Threats)**: APT is an undercover, ongoing computer network attack in which a person or group gains unauthorized access to a network with the goal of going undetected for the longest period possible in order to spy, place custom malicious code on multiple computers for specific tasks, gather information, and access sensitive, classified information.

IMPROVING CYBER/INFORMATION SECURITY

Organizations must have Password Management to prevent from cyber-attack. Strong passwords are first defensive technique when it comes to controlling access to protected systems and information. The following should be explored and analysed systematically before establishing policy in a formal way:

- Procedures for protecting password files and administrator accounts
- Random password generation, one-time passwords and two-factor authentication
- Length of a password's life
- Password expiration and renewal
- Procedures for cleansing ex-employee access
- Length and qualities of acceptable passwords
- Antivirus is an effective cyber security management policy that checks the vulnerabilities exist for an organization's resources before formalizing processes and procedures.
- Once weaknesses are acknowledged, the policy will specify both commercial and internally developed solutions to avoid the introduction of malicious code on the company's perimeter defence systems, servers and desktops, how deployment is to unfold, and who is responsible for deployment.
- Backup and Recovery also protect cyber threat. Policy needs to highlight the primary importance of backup and recovery processes for desktops, file servers and mainframes. Responsibilities should be clearly acknowledged.
- Batch processing and storage capacity plans needs to be vital parts of the operational planning process. A plan for disaster recovery from offsite backups should be considered.

Maintaining of Cyber Security at Organization Level

The organizations must need to adopt effective cyber security strategy to shield their own business, their customers, and their data from growing cyber security threats. The following factor may be considred:

- Train employees in security principles: It is necessary to establish basic security practices and policies for workers, such as requiring strong passwords, and establishing proper Internet use guiding principle that detail penalties for violating company cyber security policies.
- Protect information, computers and networks from cyber attacks: It is very important to keep clean machines and have the latest security software, web browser, and operating system and secure with the best defences against viruses, malware, and other

online threats.

- Set antivirus software to run a scan after each update.
- Provide firewall security for Internet connection.
- Make backup copies of important business data and information
- Control physical access to computers and create user accounts for each employee: It is very important to control cybercrime.
- Secure Wi-Fi networks: If there is a Wi-Fi network for workplace, make sure it is secure, encrypted, and hidden. To hide Wi-Fi network, establish wireless access point or router.
- Limit employee access to data and information, limit authority to install software.
- Passwords and authentication: The organizations must facilitate employees to use unique passwords and change passwords every three months.

Key Terms

Access	Asset	Attack
Control, Safeguard or Countermeasure	Exploit	Exposure
Hacking	Object	Risk
Security Blueprint Security Model	Security Posture or Security Profile	
Subject	Threats	Threat Agent
Vulnerability		

Cyber Crisis Management Plan/Techniques (CCMP)

Dr. Himanshu

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
himanshu@icar.gov.in

Purpose of CCMP

Government has mandated that in any organization, there should be a well defined plan and framework in place so that in case of cyber attack/ cyber crisis we are able to address it in a systematic way. So, purpose of the CCMP is :

- To establish a strategic framework and guide actions to prepare for, respond to, and begin to coordinate recovery from a cyber incident.
- To ensure that interruption of manipulations of critical functions/ services in critical sector organisations are brief, infrequent and manageable and cause least possible damage.
- To assist organizations to put in place mechanisms to effectively deal with cyber security crisis and be able to pin point responsibilities and accountability right down to individual level.

Mandate of CCMP

- The Ministries/ Departmenbts of Central Govt., State Govts. And Union Territories to draw-up their own sectoral Cyber Crisis Management Plans in lin with the Cyber Crisis Management Plan for Countering Cyber attacks and acts of Cyber Terrorism.
- Equip suitably for implementation, supervision of implementation and ensure compliance among all the organizational units (both public and private) within their domain.
- CERT-In/ MeitY to conduct mock exercises with Ministries/ Organizations
- MeitY to seek necessary compliance information on implementation of the best IT Security practices from all the organizational units of the Ministries/ Departments of Central Government, State Governments and Union Territories on a regular basis and apprise the NCMC of the progress.

The structure of CCMP for countering Cyber Terrorism has seven sections which deal with the following:

- Concept of CCMP
- Nature of Cyber crisis contingencies
- Cyber Security threat landscape
- Building Cyber Security Capabilities
- Incident prevention measures
- Crisis recognition, mitigation and management
- Post Incident Activity

In addition CCMP document contains guidelines on:

- Implementation of Information Security Management System (ISMS)
- Planning and preparation of incident handling
- Incident Response Activities in first hour and first twenty four hours

- Data Center Guidelines
- Network Infrastructure Security Best Practices

Various Types of Cyber crisis and contingencies:

- Targeted Scanning, Probing and Reconnaissance of Networks and IT Infrastructure
- Large scale defacement of websites
- Malicious Code attacks (virus/ worms/ Botnets/ Trojans/ Ransomware)
- Malware affecting Mobile devides
- Large scale SPAM attacks
- Identity Theft attacks
- Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks
- Application Level attacks
- Cyber Espionage abd Advanced Persistent Threats
- Financial Frauds, threats due to digital payment systems
- Threats due to emerging technologies (IoT, Big Data, AI)
- Hybrid threat and Misinformation campaigns
- Attacks on ICS/ SCADA Systems

Cyber Security threat Landscape

Targets

- Critical Infrastructure
- Business Intelligence
- Personally Identifiable Information

Actors

- Nation States
- Cyber Criminals
- Hacker Groups
- Malicious Insiders

Motives

- Disruption of Services
- Cyber Espionage
- Financial Frauds

Vectors and Mediums

- Botnets
- Vulnerabilities and Exploit Tool Kits
- Social Engineering
- Ignorant Users

Five essential controls that need to be implemented by organizations for CCMP:

1. Inventory of devices and software
2. Controlled use of administrative privileges
3. Secure configuration of Hardware and software
4. Malware defence
5. Vulnerabilities and patch management

Main goals and principles for achieving Cyber Resilience:

- Anticipate
- Withstand
- Recover
- Evolve

Database Management Using MS Access

Soumen Pal

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

soumen.pal@icar.gov.in

A database is an organized collection of interrelated data. A database management system (DBMS) is computer software that facilitates the process of defining, constructing and manipulating databases for various applications. Examples of Information Systems include Bank, Library and Railway Reservation which use DBMS. MS Access is the database software in the Microsoft Office suite that allows to order, manage, search, and report large amounts of information.

Create Access Database

The first step in creating an Access database is to create a blank database file. This is done from the Getting Started Screen when Access is launched. The file is saved into one of the specified folders in computer. The procedure for doing this is outlined below.

1. Launch Access

To begin, launch Access by clicking on the desktop icon, or choose Access from the start menu. This brings up the Getting Started with Microsoft Office Access screen.

Figure 1: Microsoft Office Default Page

2. Select Blank Database Template

Figure 2: Button to create database

Towards the top left of the screen, there is a "Blank Database" icon. Click this icon to bring up the Blank Database side bar on the right hand side of the screen. This is where one has to enter details about the database file to be created.

3. Enter filename for Access database.

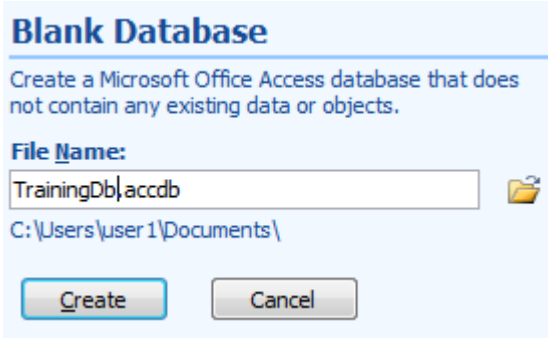


Figure 3: Wizard for naming of database

Type TrainingDb in the **File Name** textbox.

4. Browse and select folder

Next click the folder icon adjacent to **File Name** textbox and browse for a folder to put the database. Click the **OK** button once the file path has been selected.

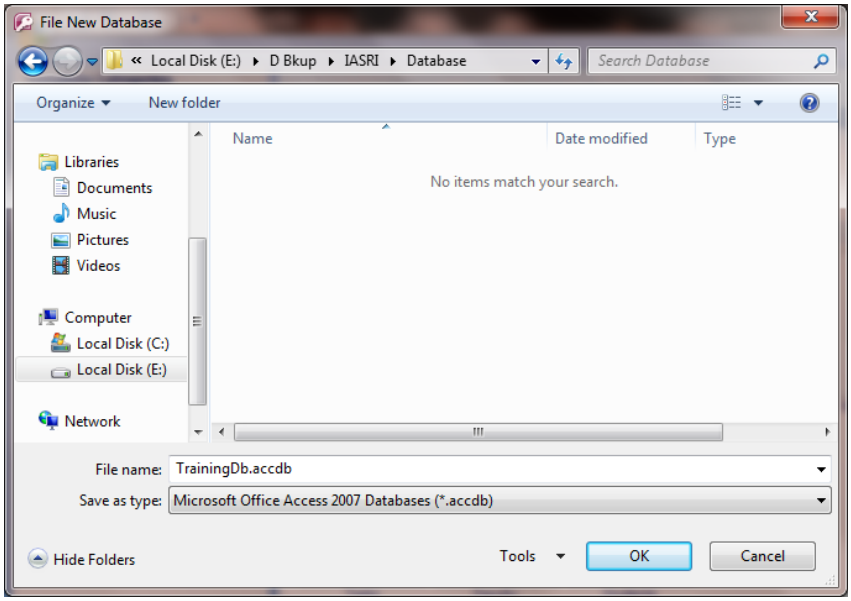


Figure 4: Wizard for saving location of database

5. Click Create

Now the selected file path can be seen below the **File Name** textbox. Once the **Create** button is clicked, the database file is saved to the specified location and opened to work on.

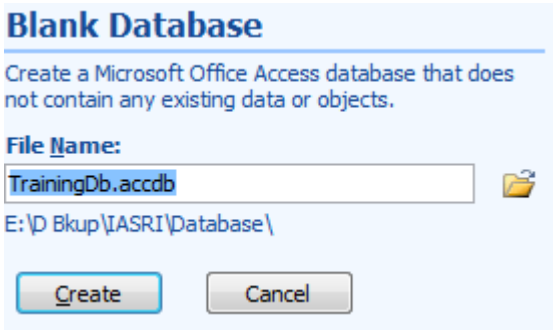


Figure 5: Wizard for naming of database

6. The window for the TrainingDb database will open.

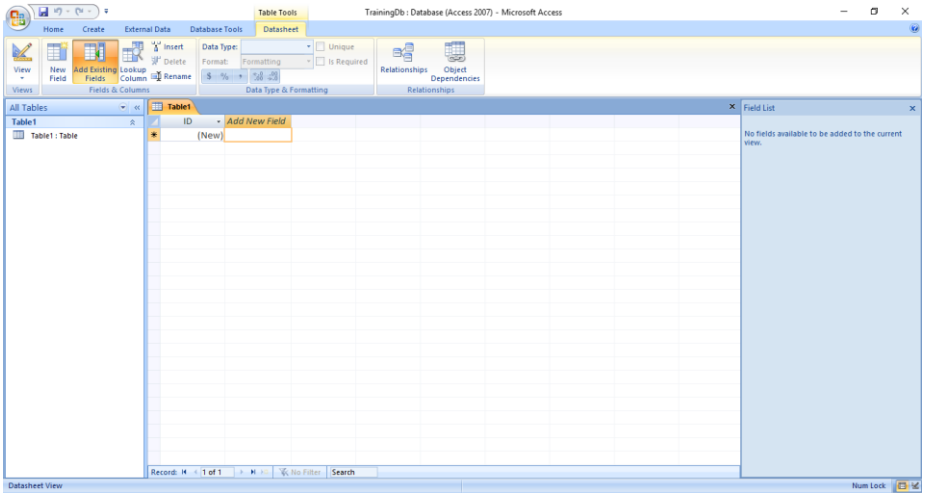


Figure 6: Database table

The newly created database file is ready to be worked upon.

Create Access Table

Tables are the foundation of an Access database. Access stores data in tables. They look like the cells of a spreadsheet with columns and rows. Each horizontal column represents a table record, and each vertical column represents a table field.

1. Click the **Datasheet** tab. In the Ribbon, click **View**.

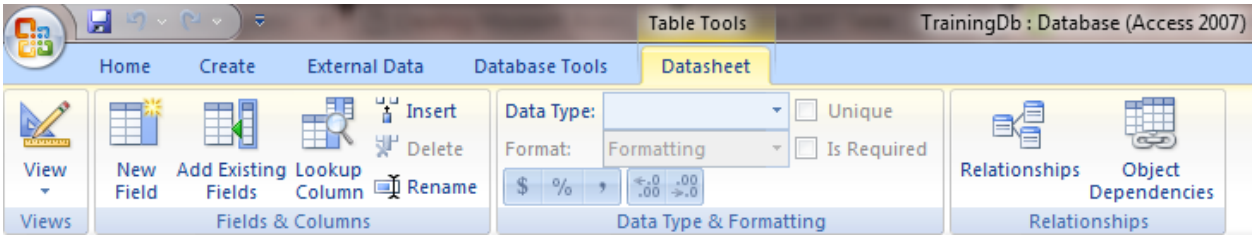
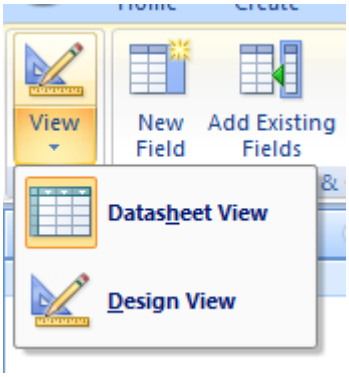
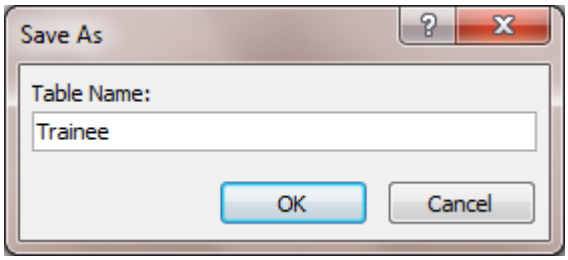


Figure 8: MS Access table creation

2. When the menu appears, click **Design View**.



3. A **Save As** window will appear, type **Trainee** in the **Table Name** box. This is the first database table to be created.



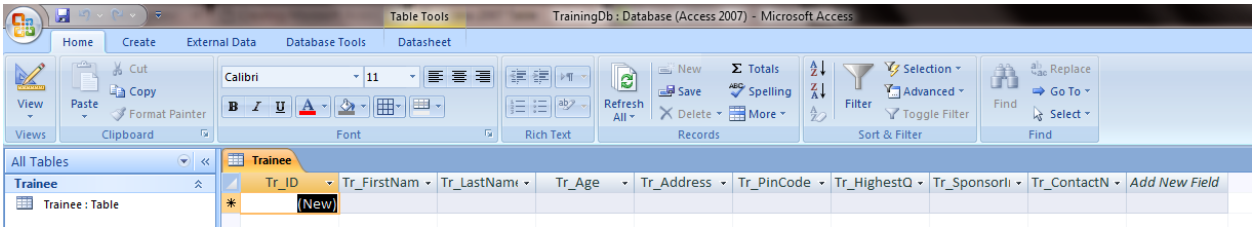
Then click on the OK button.

Create fields in Design View

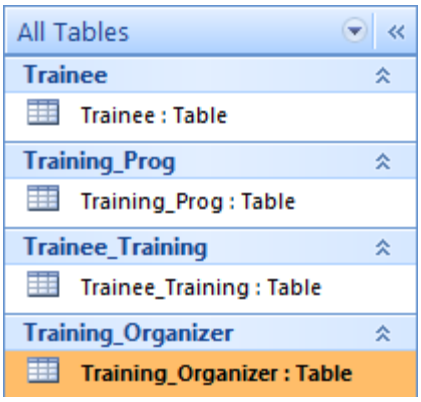
This brings up the Table Design Grid where each field name and its data type can be entered. The first field created is the Tr_ID field which is going to contain a unique reference number for each trainee record. This column is by default the primary key field. A primary key is a field or combination of fields that uniquely identify each record in a table. Enter the name "Tr_ID" into the first column of the first row in the grid. To automatically generate a unique reference number, **AutoNumber** is to be selected from the drop down list in the data type column. One can also enter a description for each field, but this is not essential.

On the next row the field is going to be called Tr_Firstname and the data type is going to be Text. On the third row the field name is Tr_LastName with the data type again being Text. Likewise, one can add as many fields as required. And finally, the last field name is Tr_ContactNo and the data type here is going to be Number.

Now the table can be saved by clicking the save icon on the top left of the screen above the Access Ribbon. To view the table, select **Datasheet View** from the **Views** group. This brings up the datasheet view of the table that is just created. One can see the field headings running across the top of the table.



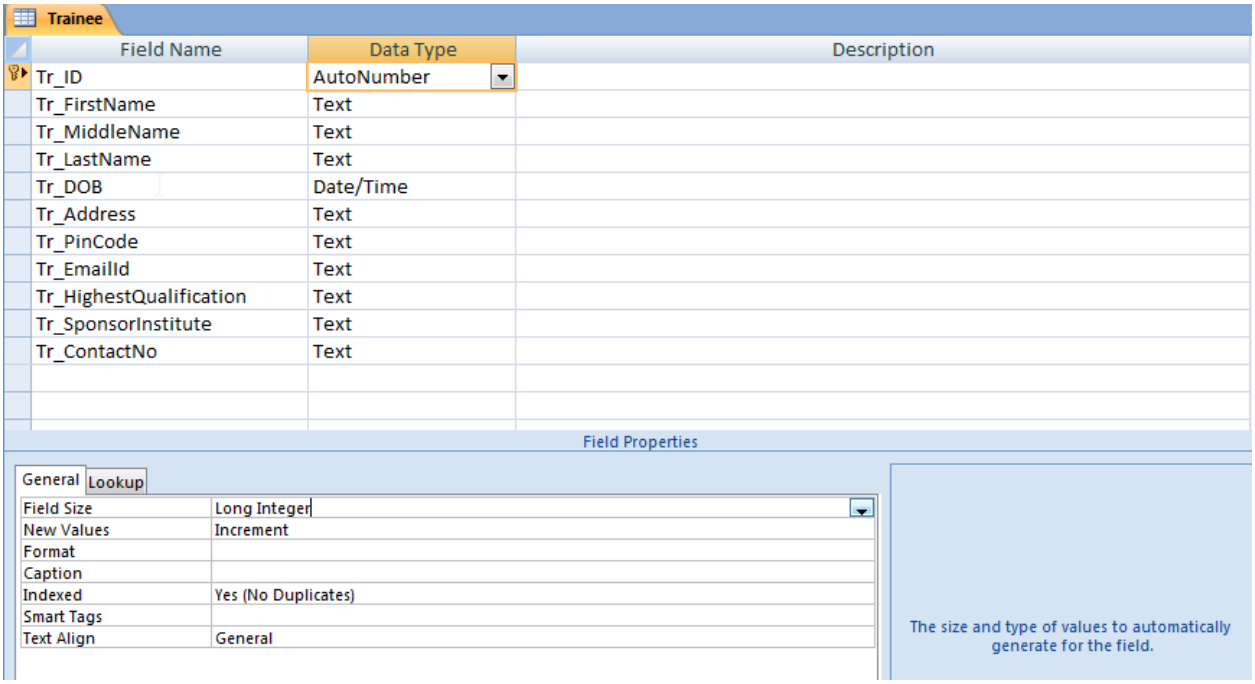
The following tables have been created: Trainee, Training_Prog, Trainee_Training, Training_Organizer.



The Trainee Table contains all the details of the trainees attending different training programmes. This table has the following fields:

Field Name	Data Type
Tr_ID	AutoNumber
Tr_FirstName	Text
Tr_MiddleName	Text
Tr_LastName	Text
Tr_DOB	Date/Time
Tr_Address	Text
Tr_PinCode	Text
Tr_EmailId	Text
Tr_HighestQualification	Text
Tr_SponsorInstitute	Text
Tr_ContactNo	Text

The properties of the data types can be viewed and modified in the **General** Tab under **Field Properties**.



Training_Prog Table contains the details of training programmes and it has the following fields: Training_ID (AutoNumber), Training_Title (Text), Training_Start_Date (Date/Time), Training_End_Date (Date/Time), Training_Host_Institute (Text), Course_Coordinator (Number).

Training_Organizer Table contains the details of the training programme organizers and it has the following fields: Organizer_ID (AutoNumber), Organizer_FirstName (Text), Organizer_MiddleName (Text), Organizer_LastName (Text), Organizer_Address (Text), Organizer_EmailId (Text).

Trainee_Training Table contains the information of the trainees participating in the training programmes and it has the following fields: TraineeTraining_ID (AutoNumber), Trainee_ID (Number), Training_ID (Number).

Building table relationships

In Access, data are stored in multiple tables. Relationships are used to join the tables. After creating relationships, data can be used from all of the related tables in a query, form, or report.

Along with primary key, the foreign key concept is required in building table relationship. A foreign key is a value in one table that must match the primary key in another table. Primary keys and foreign keys are used to join tables together. In other words, primary keys and foreign keys are used to create relationships.

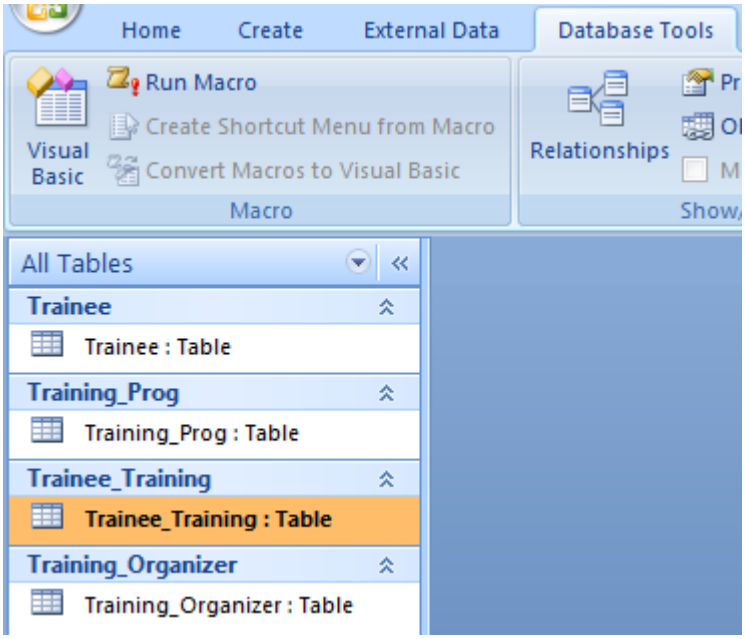
There are three types of relationships: one-to-one, one-to-many (or many-to-one) and many-to-many. In a one-to-one relationship, for every occurrence of a value in table A, there can only be one matching occurrence of that value in table B, and for every occurrence of a value in table B, there can only be one matching occurrence of that value in table A. One-to-one relationships are rare because if there is a one-to-

one relationship, the data is usually stored in a single table. However, a one-to-one relationship can occur when one wants to store the information in a separate table for security reasons, when tables have a large number of fields, or for other reasons. In a one-to-many relationship, for every occurrence of a value in table A, there can be zero or more matching occurrences in table B, and for every one occurrence in table B, there can only be one matching occurrence in table A. In a many-to-many relationship, for every occurrence of a value in table A, there can be zero or more matching occurrences in table B, and for every one occurrence in table B, there can be zero or more matching occurrences in table A.

In the present scenario, one Training Organizer can be Course_Coordinator in one or more training programmes, however, one particular training programme can only have one Course_Coordinator. This is a one-to-many relationship. Now, one trainee can participate in one or more training programmes and one training programme has more than one participants. So, this is an example of many-to-many relationship. In such scenario, another Table viz. Trainee_Training is introduced to break the many-to-many relationship into two one-to-many relationships.

To establish a relationship between tables:

1. Click the **Relationships** button in the **Show/Hide** group on the **Database Tools** tab. It is important that tables must be closed in order to establish relationships.

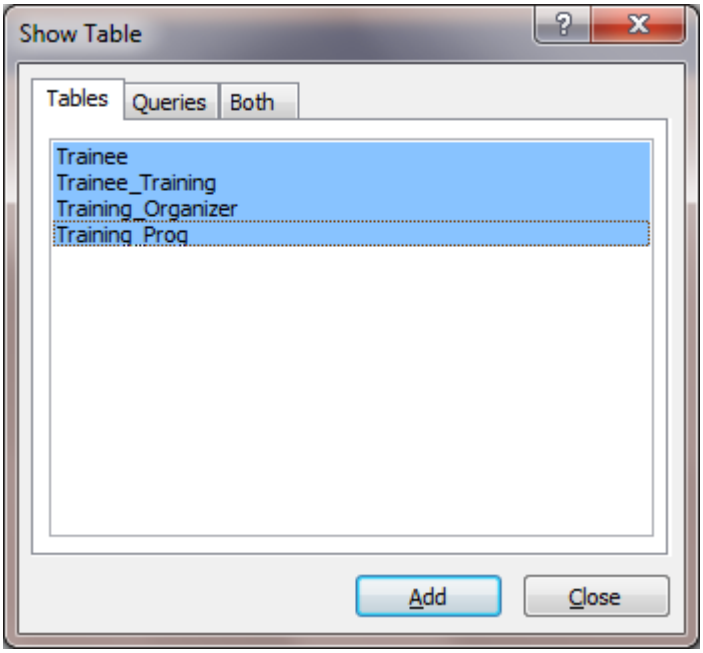


2. Click the **Show Table** button in the Relationships group. The Show Table dialog box appears.

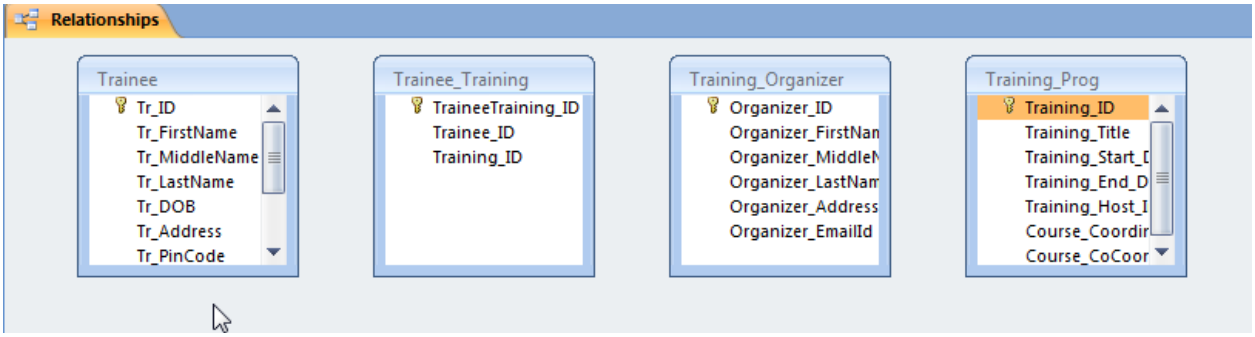
3. Activate the **Tables** tab if the relationships will be based on tables, activate the **Queries** tab if the relationships will be based on queries, or activate the **Both** tab if the relationships will be based on both.

4. Select each table name and then click **Add** for the tables to be related. One can also select multiple tables at a time by pressing the Ctrl Key and then click **Add**.

5. Click the **Close** button to close the **Show Table** dialog box.



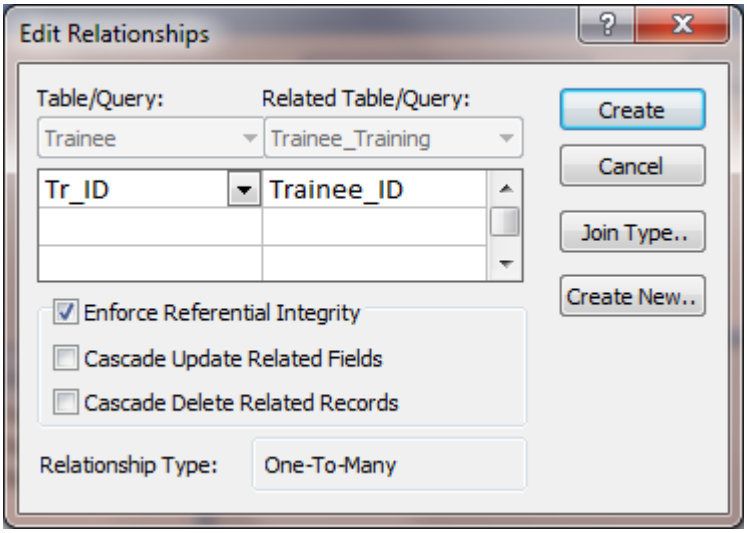
Now, one can see a relationship map that contains all of the tables those were selected.



To move a table that appears in the relationship map:

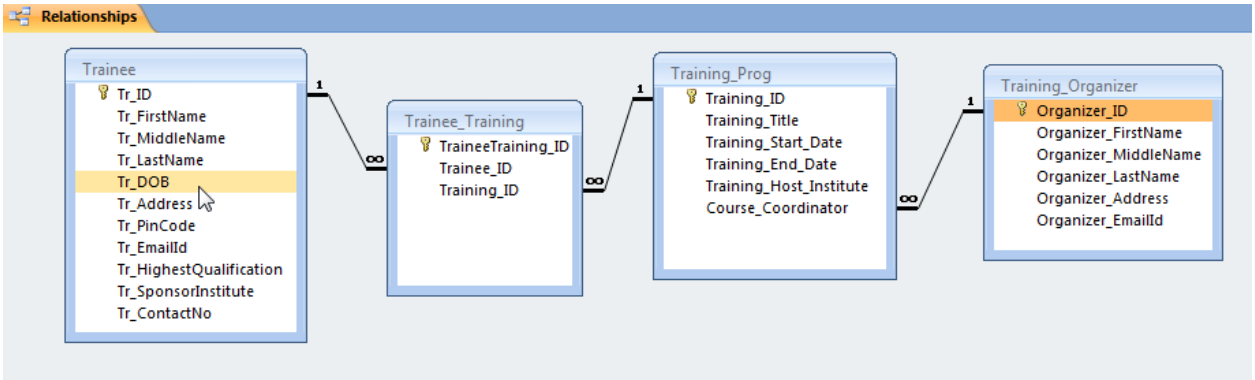
- Place the mouse over the table to be moved.
- Hold down the left mouse button, then drag the table to a new location.
- Release the mouse button to drop the table in its new place.

6. Drag the Primary table's primary key over the related table's foreign key. After dragging the primary key to the related table's box, the cursor changes to an arrow. Make sure that the arrow points to the foreign key. The **Edit Relationships** Dialog box will appear.



7. Click the Enforce Referential Integrity checkbox.

8. Click Create. Access creates a one-to-many relationship between the tables.



Note: After a relationship has been created between two tables, one must delete the relationship before making modifications to the fields on which the relationship is based. To delete a relationship:

1. Click the line that connects the tables.

2. Press the Delete key.

The other facilities available in Access include Queries, Forms and Reports. Query is used to view a subset of the data or to answer questions about the data. Access Forms are used to enter, edit or display data and they are based on tables. Reports organize and summarize data for viewing online or for printing. A detail report displays all of the selected records. One can include summary data such as totals, counts, and percentages in a detail report.

References

1. Date, C. J. (2006). *An Introduction to Database Systems*. Pearson Education.
2. http://www.baycongroup.com/access2007/
3. http://www.dealing-with-data.net/
4. http://www.gcflearnfree.org/access2007/

Web Application Architecture &Development

Alka Arora
ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
alka.arora@icar.gov.in

Web application allows user to access information on a web browser which may be static or retrieved dynamically from a database on the backend. The information is dynamically created by using Hyper Text Markup Language (HTML) or Dynamic HTML (DHTML) which can easily be viewed in any web browser. While building web applications, information can be retrieved from various RDBMS (Relational Database Management) software such as MS-Access, MySQL, Oracle, SQL Server or Microsoft Data Engine (MSDE). This information can be fetched by the script or code behind pages of the web forms which is shown on the client browser.

Web application is more than just creating dynamic web pages or web sites. We can create set of pages into a web site that subsequently begins to function like a web application. Application and Session objects enable these applications for sharing information between an individual's requests to a site, and even between all the users of a site.

As we make this transition for building larger and more scalable web based applications, we can take advantage of the technologies that allow us to do so with greater ease. One of the key elements of any application design is the **system architecture**. The system architecture defines how the pieces of the application interact with each other, and what functionality each piece is responsible for performing.

A **distributed application** utilizes the resources of multiple machines or at least multiple process spaces, by separating the application functionality into more manageable group of tasks that can be deployed in a wide variety of configurations.

Numerous applications run in a client/server environment, this means that client computers (computers forming part of the network) contact a server, generally a very powerful computer in terms of input/output, which provides services to the client computers. These services are programmes which provide data such as the time, files, a connection, etc.

The services are used by programs client programs which run on client computers. This is why the term "client" is applied (FTP client, email client, etc.), where a program is designed to run on a client computer, capable of processing data received from a server (in the case of the FTP client we are dealing with files whereas for the email client we deal with email email).

Ultimately, dividing up an application in this manner results in the creation of a series of application layers or **tiers**, each of which is responsible for an individual, or atomic element of the application's processing.

Advantages of Client/Server Architecture

The client/server model is particularly recommended for networks requiring a high degree of reliability, the main advantages being:

- **Centralized resources**: given that the server is the centre of the network, it can manage resources that are common to all users, for example: a central database would be used to avoid problems caused by redundant and inconsistent data
- **Improved security**: as the number of entry points giving access to data is not so important
- **Server level administration**: as clients do not play a major role in this model, they require less administration
- **Scalable network**: thanks to this architecture it is possible to remove or add clients without affecting the operation of the network and without the need for major modification

Disadvantages of the client/server model

Client/Server architecture also has the following drawbacks:

- **Increased cost**: due to the technical complexity of the server
- **A weak link**: the server is the only weak link in the client/server network, given that the entire network is built around it! Fortunately, the server is highly fault tolerant (primarily thanks to the RAID system)

Client/Server system operation

A client/server system operates as outlined in the following diagram:

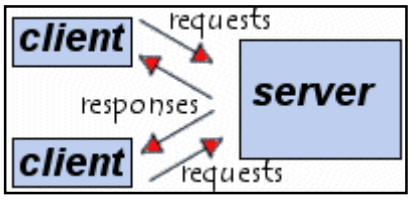


Figure 1: Client/Server system operation

- The client sends a request to the server using its IP address and the port, which is reserved for a particular service running on the server.
- The server receives the request and responds using the client IP address and port

Tiered Applications

Tiered applications can be characterized by the number of layers that information will pass through on its journey from the data tier (where it is stored in a database typically) to the presentation tier (where it is displayed to the client). Each layer generally runs on a different system or in a different process space on the same system, than the other layers.

Introduction to 2-Tier Architecture

2-tier architecture is used to describe client/server systems where the client requests resources and the server responds directly to the request, using its own resources. This means that the server does not call on another application in order to provide part of the service.

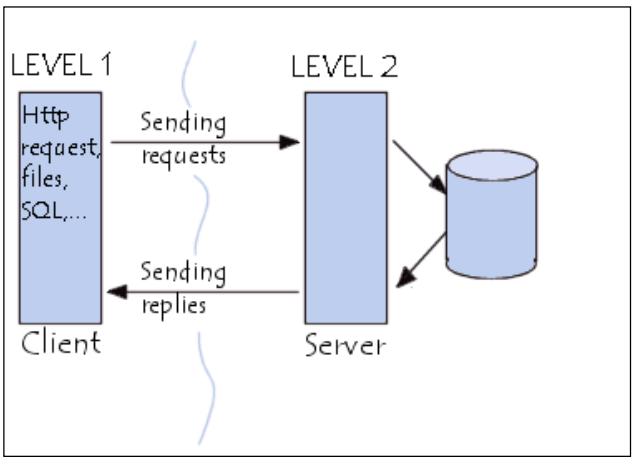


Figure 2: Two-Tier architecture

Introduction to 3-Tier Architecture

In 3-tier architecture, there is an intermediary level, meaning the architecture is generally split up between:

1. A client, i.e. the computer, which requests the resources, equipped with a user interface (usually a web browser) for presentation purposes
2. The application server (also called **middleware**), whose task it is to provide the requested resources, but by calling on another server
3. The data server, which provides the application server with the data it requires

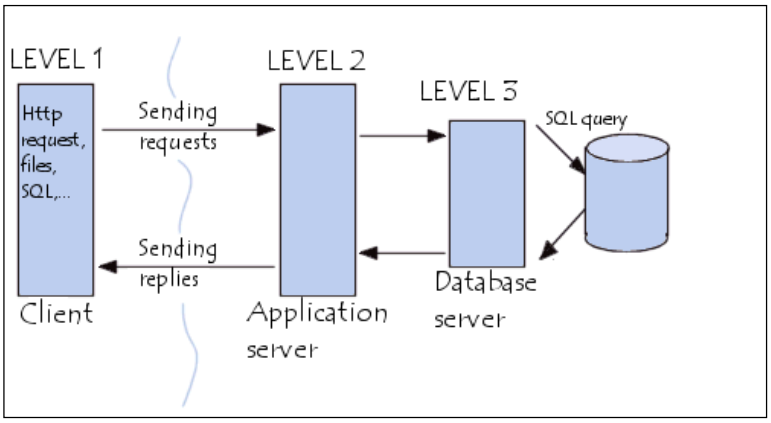


Figure 3: Three-Tier architecture

The widespread use of the term 3-tier architecture also denotes the following architectures:

- Application sharing between a client, middleware and enterprise server
- Application sharing between a client, application server and enterprise database server.

Comparing both types of architecture

2-tier architecture is therefore a client-server architecture where the server is versatile, i.e. it is capable of directly responding to all of the client's resource requests.

In 3-tier architecture however, the server-level applications are remote from one another, i.e. each server is specialized with a certain task (for example: web server/database server). 3-tier architecture provides:

- A greater degree of flexibility
- Increased security, as security can be defined for each service, and at each level
- Increased performance, as tasks are shared between servers

Multi-Tiered Architecture

In 3-tier architecture, each server (tier 2 and 3) performs a specialized task (a service). A server can therefore use services from other servers in order to provide its own service. As a result, 3-tier architecture is potentially an n-tiered architecture

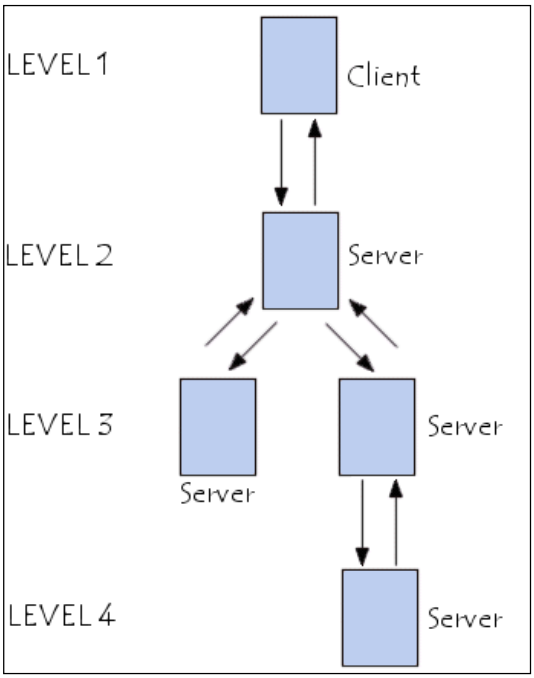


Figure 4: Multi-Tiered Architecture

The three tiers of web applications gain a detailed understanding of the roles of technologies that run in each tier

- User Services (Client-side technologies such as HTML, CSS, JavaScript, and Flash)
- Business Services (Server-side technologies such as CGI/Perl, ASP, ColdFusion, JSP and servlets, ASP.NET, PHP)
- Data Services (Back-end data sources such as databases, mainframe applications, and web services)

The separation of the various aspects of an application into n tiers allows any part of that application to be modified without having to change the other parts, allowing developers to specialize in designing and developing a specific tier or tiers.

The Structure of the Web

The structure of the web is interconnected pieces of HTML-formatted pages, distributed from computers, known as web servers, to client computers, and viewed using tools called web browsers. In general, there is no difference between a server that has thousands of pages and a server that just has one or two pages. It is by linking these pages together some form becomes a web site, and by adding some additional logic, a web site may become a web application.

Web Page

The basic unit of a web interaction is the Web page itself. A Web page is a text file that is marked up using HTML (**H**yper **T**ext **M**ark-up **L**anguage). It is sent to a browser from a Web server based on a request from that browser. The browser parses the information in the HTML file, and the resulting user interface is displayed within the browser itself. The role of the Web server is to listen for a request from the client, process the request to determine the page that the client requested, retrieve that file from the server's storage area, then sent that file to the client. At this point, the server forgets everything about sending that file to the client, except for maybe placing an entry into a log file.

Traditional web page design can be divided into two ways

Static Web Pages: It contains mainly built-in HTML and non-interactive web pages.

Dynamic Web Pages: Developed using any one of these technologies such as CSS, Flash, server side scripting technologies: ASP, JSP, PHP, CGI, .Net etc. that helps in building up interactive web sites, which can receive data from client & respond back accordingly.

Web Site

A Web site consists of a set of related Web pages grouped together by some means. Generally, a Web site is all of the pages that exist on a server, or within a folder on that server. For example, all of the pages that are on the **http://www.iasri.res.in** server are considered part of that Web site. The correlation between the pages on a site is maintained by the links within each page on the site. The links on each page in the site will take users to other pages within the site. In this way, the pages that make up the site internally maintain the hierarchy of the site. These set of related Web pages will make up a Web site in the beginning to look more like an application.

Web Applications

A traditional application requires a special set of files during development, but distributes different outputs. For example, a Visual Basic application has a .vbp project file, multiple .frm, .cls, and .bas files, as well as a set of some components that make up the application project. Prior to the application being distributed, these files are compiled into a set of executable files for distribution and execution. The resulting executable does not require the presence of the source code files that were used to develop it.
Script-based Web applications, on the other hand, are composed of the same set of files used during development and after deployment. There is no compiled executable file produced that becomes the Web application. For example, the .htm, .asp, .php, .jsp etc. in your Web project are the same files you deliver to your production Web server. The source code, or script, in these Web files is executed on the client or server only when a browser requests the Web page. Creating these applications builds upon the architecture of the World Wide Web, but there is some added complexity and functionality in order to have these files function as an application.

Web Application Design

In building a Web application, there are a number of new aspects of application design and development that the developer must take into consideration. In a Web application, we will look at using a browser as the primary user interface. The information in our Web application will flow from server to client using the HTTP protocol. Our Application server will be a web server (Microsoft's Internet Information Server or Apache or Tomcat) functioning as both a Web and Application server. Finally, our business and data access logic will be linked together to make it as an application.

The Browser as the User Interface

For a Web application, the user interface is presented within a Web browser. This means that the client presentation can either be Browser-Enhanced or Browser-Reliant. If you can guarantee that most of your users will be using a particular browser level, then you should consider Browser-Enhanced client presentation type. If there are a wide variety of browsers in use, then you may only be able to support a Browser-Reliant client presentation type. One of the advantages of using server side scripting environment to deliver the application is that it has the capability to determine what browser the current user is accessing the application. By knowing the browser type of the current user, you can dynamically change your client presentation to support the enhanced characteristics of that browser.

HTTP as the Transport

The communication layer between the client and the server is critical to the design and implementation of the application. The HyperText Transport Protocol (HTTP) defines how the request made by the client is received and handled by the server, and then how the information is sent back to the client. Depending on the type of client presentation being supported, there can be different types of information that flow using this protocol. For the basic support of a Browser-Reliant client, the information that flows over HTTP is limited to the HTML that makes up the page, graphical images to enrich the interface presentation, information-bearing cookies, and possibly some client-side scripting to provide interactivity on the client. With an Enhanced client, special features over HTTP may also be communicated over the transport protocol. But support of these is reliant on the capabilities of the browser at the client.

Application Server or Web Server

The combination of web server and corresponding server side scripting environment as the Application server provides the application component of our Web application. For Web pages and Web sites that serve up static Web pages for display, web server can function by itself to provide the information. But when a Web application demands a dynamic display of information, we need to link the Web serving capabilities of web server with the dynamic page generation and object integration capabilities of server side script to deliver a more robust and dynamic Web application. The scripting capabilities allow us to support business logic inside of our scripts, link with business logic components, directly access databases to retrieve information.

Layers in a Application

User Interface Layer

The first layer of our Web application is the user interface layer. It is in this layer that the information generated by the middle tier of the application is presented to the client. This information can be presented in a better presentable form by creating the template in HTML or DHTML to which dynamic information is added.

A server side scripting can be embedded within an HTML page, to acquired the dynamic values into an HTML template. These templates are generally used in making uniform pages in a similar manner that help the workload of a developer.

- **Repetitive Presentation:** There are usually certain parts of a Web application that are repeated in a number of different pages – possibly with different information. A frameless Web site that is trying to replicate a framed look-and-feel generally has to recreate parts of the page over and over again. This can be achieved by using template by using an include file capable of rendering the information to the browser.
- **Performance:** Performance is always a big issue when it comes to building scalable Web sites. The more performance we can ring out of any aspect of the application, the better the application as a whole will perform. Repeated HTML blocks can be stored as strings in script, and then store that script in an application-level variable or can be created in a separate file that can later be included in the page.

Business Layer

The middle of our three tiers is the business logic layer or the application layer. It provide the logical functionality for the application using some condition/looping. This could mean managing a information system, accessing the information stored in the database, incentives to the employees on the HR section of the intranet, or calculating the best route between two locations in a mapping application.

Some of the reasons are similar to the reasons for using data access layer:

- **Business Rule:** Rules can be changed any time. It will not affect the affect the design of the database.
- **Application Reuse:** It will be very easy to handle a particular business rule for all parts of the application. For example, a information system for one crop is developed, this can be replicated for the other crops.
- **Performance:** In developing Web applications, it is very straightforward to use script to create the business rules of the application. This makes it easy to get an application up and running quickly.

Data-Access Layer

The bottom tier of our three-tier architecture is the data access layer. This layer is responsible for integrating with the data sources that our application needs. These data sources could be SQL Server or Access databases or MySQL, Oracle, Exchange message stores, MSMQ message queues, or UNIX legacy applications. They could exist on the server itself, on some other server on the LAN, or somewhere across the Internet. The layer is not only responsible for accessing the data, but also for making the location of that data transparent to the application as well.

There are a number of reasons why data-access layer is necessary in a three-tier architecture. Some of these are as follows:

- **It restricts the developer from the inner structure of the database:** If the internal workings of a database will be exposed to the developer, then it will become very difficult to secure the data.
- **It provides consistent data access to different data sources:** By encapsulating the access to disparate data sources in a common interface, developers can use similar methods to access data regardless of where the data actually resides. This means that access to data stored in a SQL Server database can be accomplished using the same methods as data stored in a flat file on a UNIX system.
- Most of the Net Applications use the Client Server architecture. These terms refer to the two processes or two applications which will be communicating with each other to exchange some information. One of the two processes acts as a client process and another process acts as a server.

Client Process:

- This is the process which typically makes a request for information. After getting the response this process may terminate or may do some other processing.
- **For example:** Internet Browser works as a client application which sends a request to Web Server to get one HTML web page.

Server Process:

- This is the process which takes a request from the clients. After getting a request from the client, this process will do required processing and will gather requested information and will send it to the requestor client. Once done, it becomes ready to serve another client. Server process are always alert and ready to serve incoming requests.
- **For example:** Web Server keeps waiting for requests from Internet Browsers and as soon as it gets any request from a browser, it picks up a requested HTML page and sends it back to that Browser.
- Notice that the client needs to know of the existence and the address of the server, but the server does not need to know the address or even the existence of the client prior to the connection being established. Once a connection is established, both sides can send and receive information.

HyperText Markup Language (HTML)

S N Islam

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
sn.islam@icar.gov.in

Introduction to HTML

- HTML stands for Hyper Text Markup Language
- It is a simple text formatting language used to create hypertext documents.
- It is a platform independent language unlike most other programming languages in terms of hardware independence as well as software.

World Wide Web

- It is an architectural framework for accessing link documents spread out over billions of machines across the globe
- It began in 1989 at CERN (European Centre for Nuclear Research)
- Web is based on client server system
- The base of Client Side Scripting is HTML

Hyper Text Transfer Protocols

- The standard web transfer protocol is HTTP
- HTTP protocols consists of two fairly distinct items
  - Set of Requests from browsers to servers
  - Set of response given back to the browser
- HTTP standards describe message headers and bodies in considerable detail
- It identifies HTML as Standard Markup language

What is a markup language?

- A notation for writing text with markup tags
- The tags indicate the structure of the text
- Tags have names and attributes
- Tags may enclose a part of the text

Standardization of HTML

- When Mosaic was the only browser the language it interpreted was HTML 1.0
- It was the de facto standard
- When new browser came along a formal standardization was needed
- As a result HTML 2.0 was produced

- HTML 3.0 was initially created as a research effort to add many new features like tables, toolbars, mathematical formulas, style sheets etc.

Origin and Background

- HTML came into existence in 1990. The development of HTML was initiated by Tim Berners Lee and was later taken over by Dan Conolly, Dave Ragett and his team
- SGML was developed by ISO (International Standards Organization) and was first used by the US DOD (Department of Defense).
- There are certain visual HTML editors that make web programming very easy.

Evolution of HTML

Level 0: Introduced only the basic structural elements and assumed that all browsers support every feature of level 0.

Level 1: Brought in hypertext and graphics and assumed that the browser will support it depending on its capability.

Level 2: It has introduced the feature of fill-out forms on the internet making web pages interactive. These features are also very much in use.

Level 3: Is currently in use on the WWW and has brought in features like frames, tables, inline video, sound etc. These features of level 3 are gaining popularity.

What is an HTML File?

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- An HTML file is a text file containing small **markup tags**
- The markup tags tell the Web browser **how to display** the page
- An HTML file must have an **htm** or **html** file extension
- An HTML file can be created using a **simple text editor**

Creating HTML document

- An HTML document is created with the help of elements
- An HTML document also has the head, body and the title part which are specified with the help of elements.
- It is similar to writing a letter where you have to specify the address part, the subject, followed by the body of the letter and the end.

COMMON HTML ELEMENTS

```
<HTML>
</HTML>
```

- It indicates to the browser that the following document is an HTML document. Here both the start and the end tags are optional.
- **<HEAD>...</HEAD>**
- The HEAD element contains information about the current document, such as its title, keywords that may be useful to search engines and other data that is not considered document content.

The end tag is optional.

<TITLE>..</TITLE>

This element is used to specify a title for the document. Internet Explorer uses this to display it on the title bar of the window.

Note: - This tag is valid only within the HEAD tag and if we use the TITLE element also, the end tag of the Head element becomes mandatory.

<BODY>..</BODY>

- The body element of a document contains the contents of the document.
- This tag allows you to write text for a document.
- An image, graph or a table can also be inserted in the body.
- Some of the attributes like the background image, the background color, the link colors, and the top and left margins of the page can be set by the help of this tag.

Attributes of Body Elements

- **Bgcolor** specify the background color of the document.
- **Background** The value of this attribute is a URL that designate an image resource. The image generally tiles the background in the browser.
- Text This attribute sets the foreground color for text.

Attributes of Body Elements

- **Link** sets the color of text marking unvisited hypertext links.
- **vlink** sets the color of text marking visited hypertext links.
- **alink** sets the color of text marking hypertext links when selected by the user.

Color Attributes

link=color or vlink=color or alink=color

The color specified is a six digit number and letter combinations represent colors by giving their RGB (red, green, blue) value.

The six digits are three two-digit numbers in sequence, representing the amount of red,green, and blue as a hexadecimal value in the range 00-FF(0-255).

E.g. 000000 represent no color (black)

FF0000 is Red

00FF00 is Green

0000FF is Blue

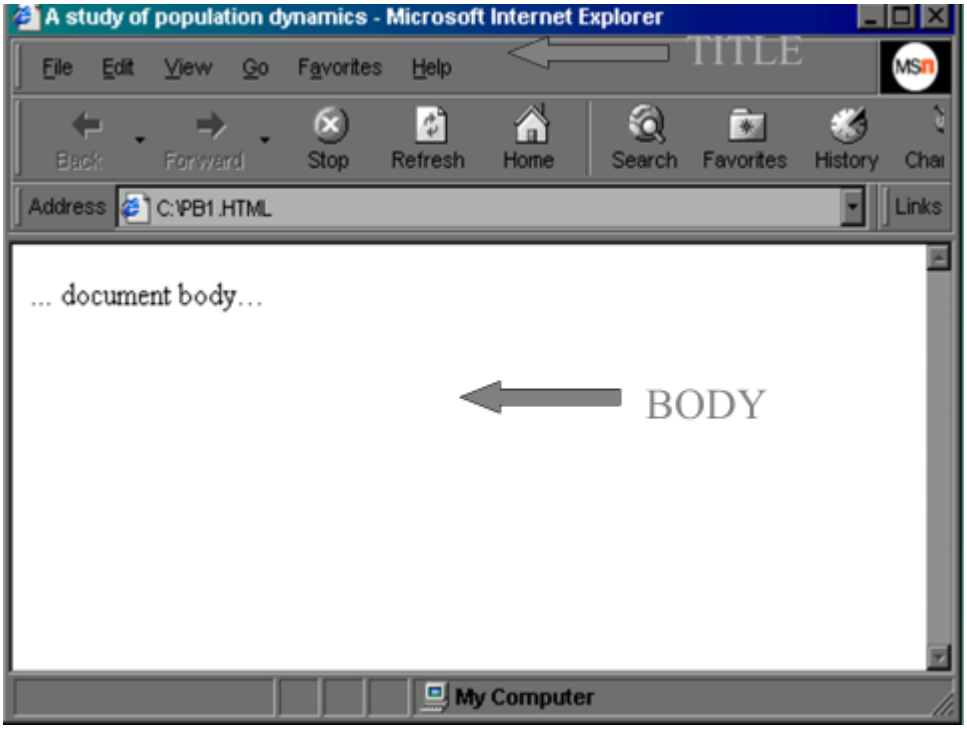
Browser Specific

For some basic colors we can use the name of the color instead of an RGB value.

E.g.

Black, Red, Blue, Cyan

All browsers may not understand all color names but any browser that can display colors will understand RGB values.



Elements

- An element consists of a start-tag, an end tag and the data characters enclosed by the two tags.
- A tag starts with a less than (<) sign and ends with a greater than (>) sign.
- Tag and attribute names are not case-sensitive, but are typically written in uppercase to distinguish them from the data characters.
- An end-tag consists of the tag name immediately preceded by a slash (/).

HTML Tags

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets

- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, means the same as <B>

Example with Body and BR Tags

```
<BR>
```

This element inserts a line break in the paragraph and rest of the characters will appear on the next line.

Example:

```
<HTML>

<HEAD> <TITLE>Introduction to HTML</TITLE></HEAD>

<BODY> <p> HTML I found to be very useful and productive tool.This is the first time I am writing the code of HTML. It is so so simple I hope by the end of the day we can be expert enough to design web pages for our site. </p>

<p>

This is the first time I am writing the code of HTML document. <br> Let's practice in the Lab

</BODY>

</HEAD>

</HTML>
```

Alignment

align = left | center | right | justify

This attribute specifies the horizontal alignment of its element with respect to the surrounding context. Possible values:

left: text lines are rendered flush left.

center: text lines are centered.

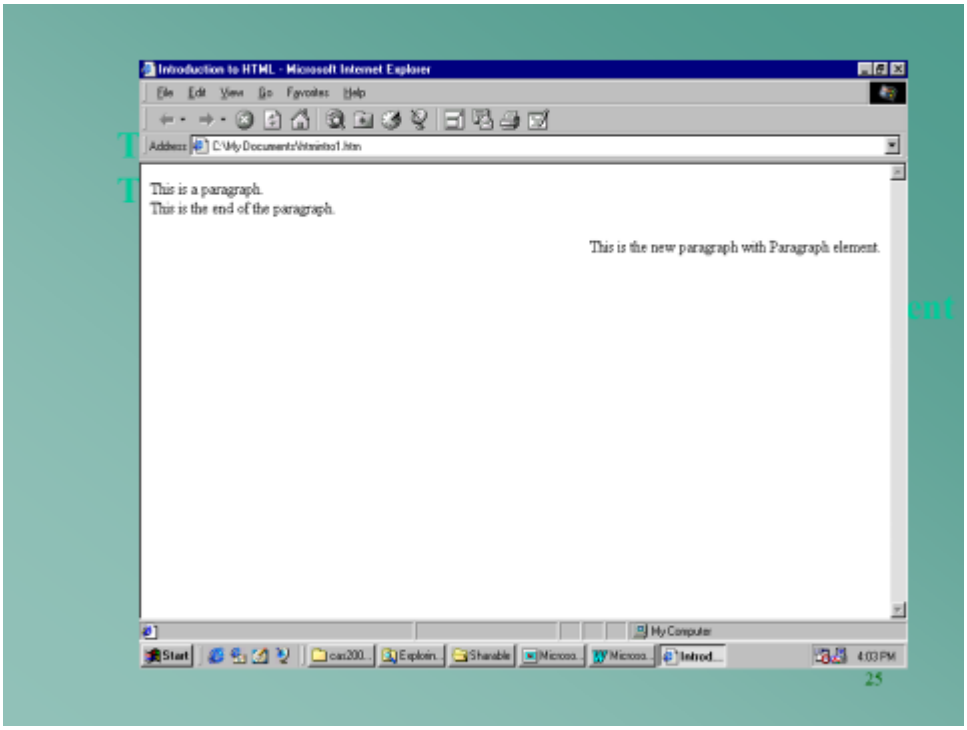
right: text lines are rendered flush right.

justify: text lines are justified to both margins.

HTML code with paragraph

```
<html>

<head>

<title>

Introduction to HTML

</title>
```

```
</head>
<body >
This is a paragraph.<br>
This is the end of the paragraph.
<p align=right>This is the new paragraph with Paragraph element.</p>
</body>
</html>
```



FONTS

- The <FONT> element changes the font size and color for text in its contents.
- Size
  - If you use the FONT tag to change text size, you can specify either a fixed or relative size. A fixed size is a number in the range 1 through 7.
  - Example <FONT SIZE = 3> this text is of size 3 </FONT>
- Color
  - In addition to size color can also be specified
  - Example <FONT SIZE=4 COLOR=Red> Size of the alphabets is large and the color is red.</FONT>
- Face
  - Face of the font can also be defined
  - <FONT SIZE=4 FACE = "Arial"> Size of the alphabets is large and the typeface is Arial.</FONT>

HTML code for fonts

```
<html>
<head>
<title>
Introduction to HTML
</title>
</head>
<body>
<font size=4 color=red>This is a paragraph.</font><br>
This is the end of the paragraph.
<p align=right>This is the new paragraph with Paragraph element.</p>
</body>
</html>
```

Character formatting

<BLINK>.....</BLINK>

The BLINK tag gives a blinking effect on the text marked for blinking, but unfortunately this tag is Netscape Specific and will not render any effect on the Internet Explorer window.

The following example displays" Try Me" as blinking text.

<BLINK> Try Me </BLINK> can be annoying, so use it sparingly.

<Marquee>...</Marquee>

This element has many attributes to support the way your marquee looks on the screen.

This element renders the text to be moving giving it an animated look.

BEHAVIOR=type

Specifies how the text should behave. The type can be one of these values:

- SCROLL=Start completely off one side, scroll all the way across and completely off, and then start again. This is the default.
- SLIDE=Start completely off one side, scroll in, and stop as soon as the text touches the other margin.
- ALTERNATE=Bounce back and forth within the marquee.

```
<html>
<head>
<title>
Dot Net CAS Training
</title>
</head>
<body>
<marquee behavior=alternate>Keep Learning </marquee>
</body>
</html>
```

BGCOLOR=color

Specifies a background color for the marquee.

```
<html>
<head>
<title>
Dot Net Training
</title>
</head>
<body >
<marquee behavior=alternate bgcolor=blue>Keep Smiling while Learning</marquee>
</body>
</html>
```

<Marquee>...</Marquee>

DIRECTION=direction

Specifies which direction the text should scroll. The direction can be LEFT or RIGHT. The default is LEFT, which means scrolling to the left from the right.

```
<marquee direction=right>Keep Smiling</marquee>
```

HEIGHT=n

Specifies the height of the marquee, either in pixels or as a percentage of the screen height. To specify a percentage, the n must be end with a percent (%) sign.

```
<marquee bgcolor=yellow height=20%>Keep Smiling</marquee>
```

HSPACE=n

Specifies left and right margins for the outside of the marquee, in pixels.

```
<marquee bgcolor=yellow height=20% hspace=100>Keep Smiling</marquee>
```

LOOP=n

Specifies how many times a marquee will loop when activated. If n=-1, or if LOOP=INFINITE is specified, it will loop indefinitely.

```
<marquee loop=1>Keep Smiling</marquee>
```

VSPACE=n

Specifies top and bottom margins for the outside of the marquee, in pixels.

```
<marquee bgcolor=yellow height=20% vspace=100>Keep Smiling</marquee>
```

WIDTH=n

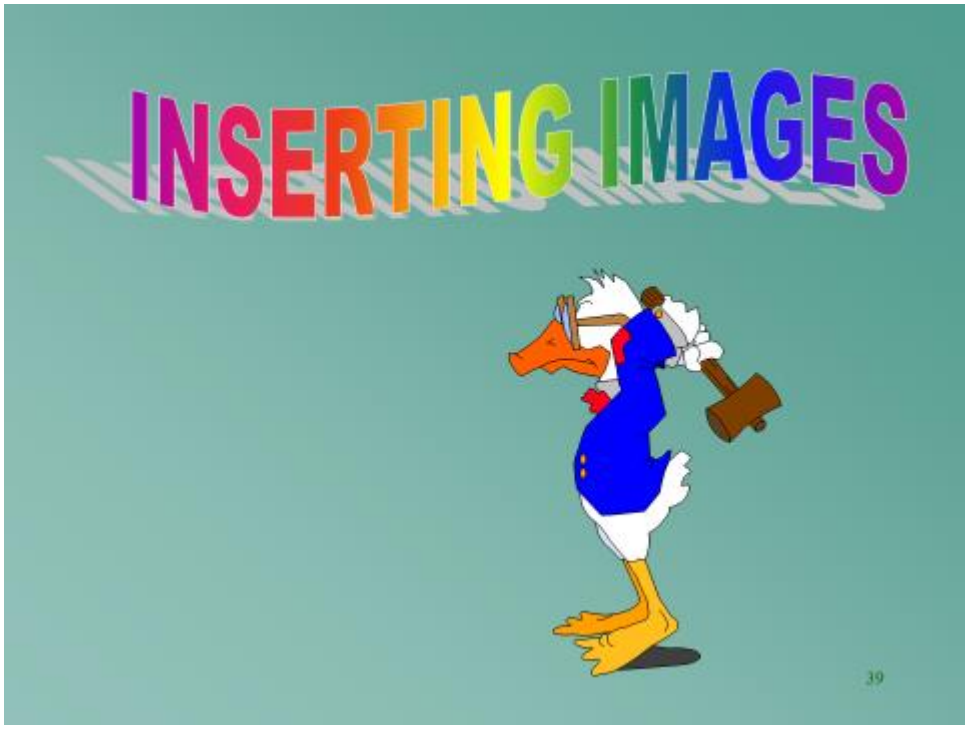
Sets the width of the marquee, either in pixels or as a percentage of the screen width. To specify a percentage, the n must end with a percent (%) sign.

```
<marquee bgcolor=yellow width=50%>Keep Smiling</marquee>
```

SCROLLDELAY=n

Specifies the number of milliseconds between each successive draw of the marquee text.

```
<marquee bgcolor=yellow width=50% scrolldelay=200>Keep Smiling</marquee>
```



<IMG>

- This element is used to insert an image into HTML document to make it more interactive.
- Most Web browsers can display inline images (that is, images next to text) that are in Bitmap (BMP), GIF, or JPEG format.
- Other image formats are also being incorporated into Web browsers [e.g., the Portable Network Graphic (PNG),DAT,MOV format].

```
<html>
<head><title>rfs</title></head>
<body>
<img src="c:\parul\parul.bmp">
</body>
</html>
```

ALT

```
<html>
<head><title>rfs</title></head>
<body>
<IMG alt=”click here" src="c:/parul/parul.bmp">
</body>
</html>
```

SRC

```
<html>
<head><title>rfs</title></head>
<body>
<img src="c:/parul/parul.bmp" >
```

```
</body>
</html>
```

BORDER

```
<html>
<head><title>rfs</title></head>
<body>
<img  border=10 src=" c:/parul/parul.bmp" >
</body>
</html>
```

FOR NO BORDER

```
<html>
<head><title>rfs</title></head>
<body>
<img  border=0 src=" c:/parul/parul.bmp " >
</body>
</html>
```

Height & Width

```
<img border=5 height=40 width=70 src=" c:/windows/web/ent_wl.gif " >
```

HSPACE & VSPACE

```
<img hspace=200 vspace=200 src="c:/windows/web/ent_wl.gif" >
```

DYNSRC

```
<html>
<head><title>rfs</title></head>
<body>
<img dynsrc="c:/windows/tit2/abcdatabc.dat">
</body>
</html>
```

CREATING LIST ELEMENTS

- In the List formatting we can create a list, menu, definition etc. to represent the information in the particular format like the content of items can be displayed in the list format or in the directory format.
- You can create a variety of lists in your document by using the UL, OL, MENU, and DIR tags in conjunction with the LI tag.

<UL>

One can create a bulleted list, consisting of individual items preceded by a bullet character, by using UL and LI, as in the following example:

```
<UL>
<LI>Bulleted Lists
<LI>Ordered Lists
```

The output will be:

Bulleted Lists

Ordered Lists

Directory Lists

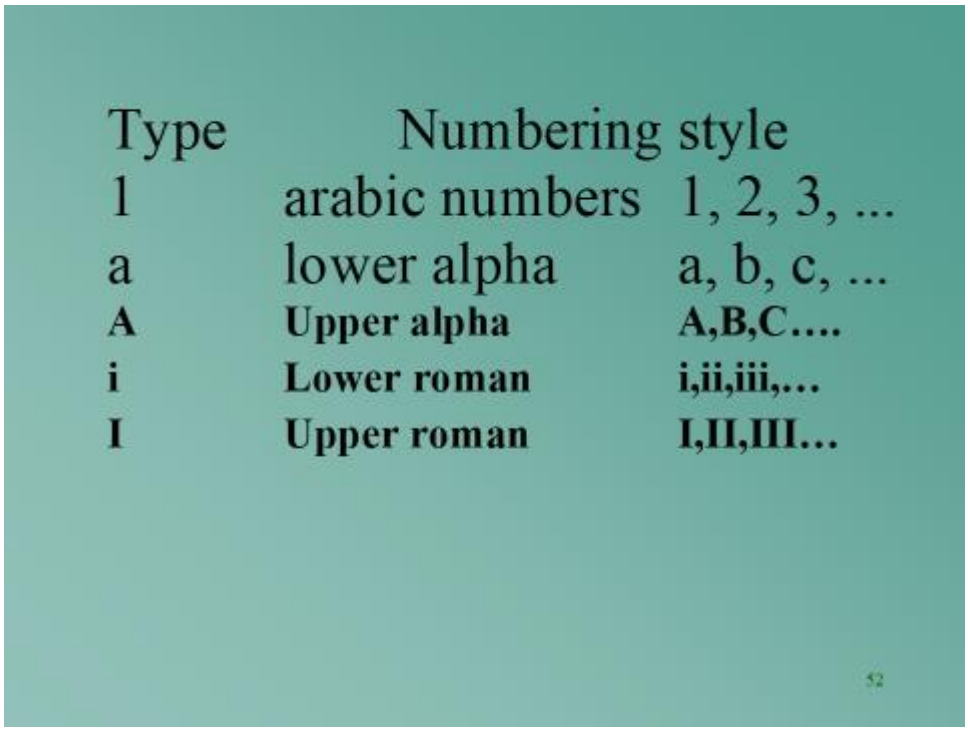
<LI>Directory Lists

</UL>

<OL>

One can use the OL and LI tags to create an ordered list. The list consists of individual items that are sequentially numbered or lettered. To set the style of numbering or lettering, you use the TYPE= attribute in OL.

Similarly, you use the START= attribute to set the initial number or letter. By default, the style is integer numbers starting at 1.




```
<OL START=1>
```

```
<LI>Bulleted Lists
<LI>Ordered Lists
<LI>Directory Lists
</OL>
```

The output will be:

1.Bulleted Lists

2.Ordered Lists

3.Directory Lists

```
<OL type=A start=3 >
<LI>Bulleted Lists
<LI>Ordered Lists
<LI>Directory Lists
</OL>
```

C. Bulleted Lists

D. Ordered Lists

E. Directory Lists

DIR & MENU

The <DIR> element was designed to be used for creating multi-column directory lists. The <MENU> element was designed to be used for single column menu lists. Both elements have the same structure as <UL>, just different in rendering. In practice the user agent will render <DIR> or MENU> list exactly as <UL> list. We strongly recommend using <UL> instead of these elements.

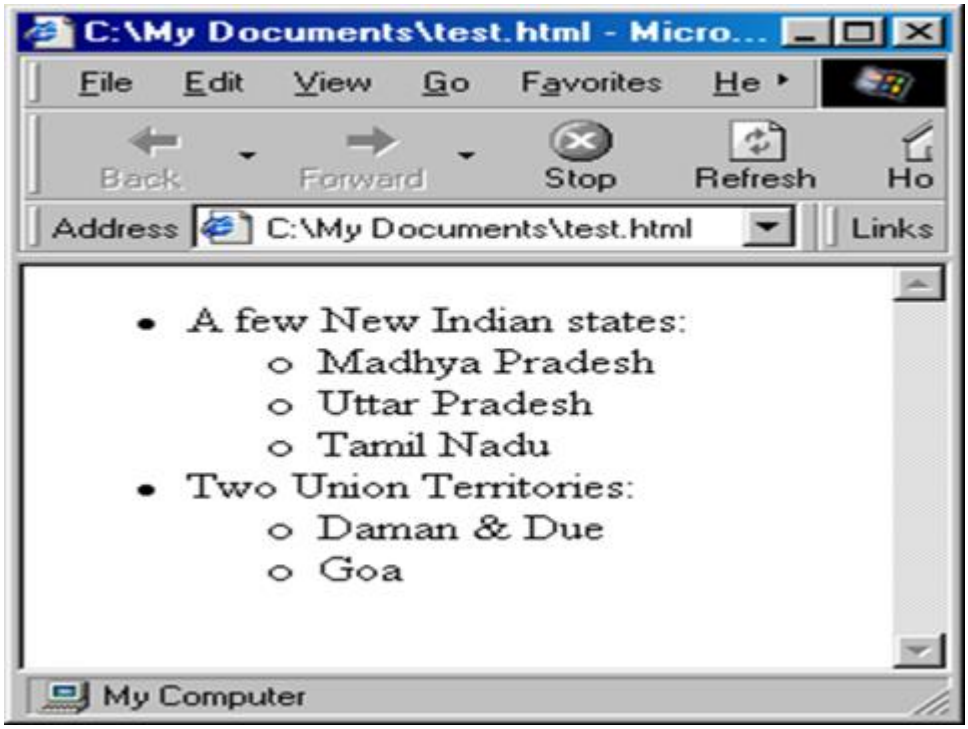
```
<DL>
       <DT>Dweeb
        <DD>young excitable person who may mature into a Nerd Geek
        <DT>Cracker
        <DD>hacker on the Internet
        <DT>Nerd
        <DD>male so into the Net that he forgets his wife's birthday
</DL>
```

Nested Lists

Lists can be nested. You can also have a number of paragraphs, each containing a nested list, in a single list item.

```
<UL>
```

```
<LI> A few New Indian states:
         <UL>
          <LI>  Madhya Pradesh
          <LI>   Uttar Pradesh
          <LI>   Tamil Nadu
          </UL>
<LI>   Two Union Territories:
       <UL>
       <LI>  Daman & Due
       <LI>   Goa
            </UL>
</UL>
```



Logical Formatting

If you use logical styles, stick with them within a document but keep in mind that future releases of HTML might not support certain logical styles, which could mean that browsers will not display your logical-style coding.

<DFN>

For a word being defined. Typically displayed in Italics. (*NCSA Mosaic* is a World Wide Web browser.)

<EM>

For emphasis. Typically displayed in Italics. (*Consultants cannot reset your password unless you call the help line.*)

<CITE>

For titles of books, films etc. Typically displayed in Italics.

<CODE>

For programming code. Displayed in a fixed-width font

<KBD>

For user keyboard entry. Typically displayed in plain fixed-width font.

<SAMP>

For a sequence of literal characters. Displayed in a fixed-width font.

<STRONG>

For emphasis. Typically displayed in bold.

- **<VAR>**

For a variable, where you will replace the variable with specific information. Typically displayed in italics. (**rm** *filename* deletes the file.)

Physical Styles

- Instead of that in the physical formatting if you want something to be displayed in italics (for example) and do not want a browser's setting to display it differently, you should use physical styles
- Physical styles, therefore, offer consistency in that something you tag a certain way will always be displayed that way for readers of your document.

EG:

<B> for bold text

<I> for italic text

<TT> for typewriter text

Escape Sequences

Character entities have two functions:

- escaping special characters
- displaying other characters not available in the plain ASCII character set
- Three ASCII characters--the left angle bracket (<), the right angle bracket (>), and the ampersand (&)--have special meanings in HTML and therefore cannot be used "as is" in text.
- To use one of the three characters in an HTML document, you must enter its escape sequence instead:
- &lt the escape sequence for <
- &gt the escape sequence for >

- &amp the escape sequence for &
- Additional escape sequences support accented characters, such as:
- &ouml a lowercase o with an umlaut: ö
- &ntilde a lowercase n with a tilde: ñ
- &Egrave an uppercase E with a grave accent: È
- You can substitute other letters for the o, n, and E shown above.

IMAGES

- People like to see pretty pictures.
- Images can have many different purposes.
- Images can be static, have links associated with them, or they can be moving.

<IMG> Command

Images can be included in any Web page, using the image element <IMG>

```
<IMG  SRC="Location of Image File"  WIDTH="Width of Image" HEIGHT="Height of Image" BORDER="1 to 6, Size of Border">
```

Inserting Images

- Images included in Web pages are called inline images .
- Images are inserted within a line of body text, and it doesn't start a new paragraph automatically.
- To make an image appear as a separate paragraph, enclose it within the paragraph element like this.

```
<P>
<IMG SRC=”C:/My Documents/samir / images/turtleshirt.jpg ”>
</P>
```

If the image is in the same directory as your HTML file, abbreviate the URL and use a tag like this

```
<IMG SRC=” turtleshirt.jpg ”>
```

IMAGE ELEMENT ATTRIBUTES

The <IMG> tag's attributes are principally intended to tell a browser how the page should be laid out with the image so that text can flow properly around the image.

Alignment Attributes:

To align an image in a line, choose one of the following attributes for the image element:

ALIGN=”TOP”

ALIGN=”MIDDLE”

ALIGN=”BOTTOM”

CHOOSE:

ALIGN="LEFT"

ALIGN="RIGHT"

To make an image "float" to the left or right side and cause paragraphs to wrap around the image.

The text after the <IMG> tag will flow around the image.

Other Attributes

WIDTH and HEIGHT:

Indicate the exact size of your image, in pixels. For example:

```
<IMG SRC="sbamm.jpg" WIDTH="50" "HEIGHT="50" >
```

HSPACE and VSPACE:

Specifies the amount of horizontal space with the HSPACE attribute and the amount of vertical space with the VSPACE attribute. For example

```
<IMG SRC="sbamm.jpg" HSPACE="50" >
```

Describing Images with Alternate Text

The Alt attribute is used to describe the image in some way. For any browser that isn't displaying images, the alternate text contained inside the ALT attribute is displayed instead.

Example:

```
<IMG SRC=" images/turtleshirt.jpg" ALT =" Mickey Mouse ">
```

External Images

Image open as a separate document when a user activates a link on either a word or a smaller inline version of the image included in the document is called an external image.

To include a reference to an external image.

```
 <A HREF="MyImage.gif">link anchor</A>
```

We can also use a smaller image as a link to a larger image.

```
<AHREF="LargerImage.gif"><IMG SRC="SmallImage.gif"></A>
```

HTML Links

The chief power of HTML comes from its ability to link text and/or an image to another document or section of a document. A browser highlights the identified text or image with color and/or underlines to indicate that it is a hypertext link (often shortened to hyperlink or just link).

Anchor Element

Link can be inserted in HTML document by an anchor and specifying the path of the document or its part using this element

Start the anchor with <A, specify the document you're linking to by entering the parameter HREF.

HREF="*filename*" followed by a closing right angle bracket (>)

```
<A HREF="MaineStats.html">Maine</A>
```

Linking a particular section in the document

Name attribute of the anchor element is used to insert the location link (Bookmark) and to refer this link.

Example:

Anchors can also be used to move a reader to a <A HREF name="position"> particular section </A> in a document (either the same or a different document) rather than to the top, which is the default.

```
< A HREF="#position" > Bookmark </A>
```

Linking a particular section in another document

You can specify an anchor on the specified section of another document.

```
<A HREF ="path of file #specific location"> link </a>
```

<Mailto>

You can make it easy for a reader to send electronic mail to a specific person or mail alias by including the mail to attribute in a hyperlink.

```
<A HREF="mailto:emailinfo@host">Name</a>
```

Enhancements in the <BODY> Element

<BACKGROUND>

Include Background Images using the <BACKGROUND> element, make sure your text can be read easily when displayed on top of the image.

The background images should preferably be in Graphic Interchange format (.GIF) or in Joint Photographic Express Group (. JPG)

Including the background

The tag to include a background image is included in the <BODY> statement as an attribute:

```
<BODY BACKGROUND="filename.gif">
```

Using a feature called tiling, a browser takes the image and repeats it across and down to fill your browser window.

Changing the Link Colors

You change the color of

- links <LINK>
- visited links <VLINK>
- active links <ALINK>

using further attributes of the <BODY>

An Example

```
BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC">
```

This creates a window with a black background (BGCOLOR), white text (TEXT), and silvery hyperlinks (LINK).

Headings

A heading element briefly describes the topic of the section it introduces. There are six levels of headings in HTML with <H1> as the most important and <H6> as the least. Visual browsers usually render more important headings in larger fonts than less important ones.

An Example

```
<H1>ABC</H1>        ABC
<H3>ABC</H3>        ABC
<H6>ABC</H6>        ABC
```

Identifying the Document Properties

The <META> element can be used to identify properties of a document (e.g., author, expiration date, a list of key words, etc.) and assign values to those properties.

<META> tag attributes

- name = name

This attribute identifies a property name. This specification does not list legal values for this attribute.

For example:

```
<Meta name="keywords" …..>
<Meta name="Author" …..>
<Meta name="date" …..>
```

<META> tag attributes

- content = cdata

This attribute specifies a property's value. This specification does not list legal values for this attribute.

For example:

```
<META name="author" content="John Doe">
<META name="copyright" content="&copy; 1997 Acme Corp.">
```

<META> tag attributes

- http-equiv = name

This attribute may be used in place of the name attribute. HTTP servers use this attribute to gather information for HTTP response message headers.

<META> tag attributes

A common use for META is to specify keywords that a search engine may use to improve the quality of search results.

For example,

<-- For speakers of British English -->

```
<META name="keywords" lang="en"
    content="holiday, Greece, sunshine">
```

<-- For speakers of French -->

```
<META name="keywords" lang="fr"
    content="vacances, Gr&egrave;ce, soleil">
```

Introducing Frames

Frames allow authors to present documents in multiple views, which may be independent windows or sub-windows. Multiple views offer designers a way to keep certain information visible, while other views are scrolled or replaced.

For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled though or replaced by navigating in the second frame.

Structure of framed page

- An HTML document that describes frame layout (called a frame-set *document*) has a HEAD, and a FRAMESET in place of the BODY.
- Inside a frame-set, there can be one or more frame tags, each of which describes separate frame.

Where you can use frames

- A frame with "table of contents" links pages that display in an adjacent frame when clicked.
- A frame containing a query and an adjacent frame that displays the results of the processed query.

- A static frame that displays a page that is always visible. The frame could contain title graphics and copyright notices. As the user navigates in other frames, the contents of the static frame do not redraw.

Sample Frame Page

```
 <html>
<head><title>First FrameSet Page</title></head>
<frameset cols=50%,50%>
<frame src=notes.htm>
<frame src=changes.htm>
</frameset>
</html>
```

<FRAMESET>..</FRAMESET>

- The FRAMESET section of a document specifies the layout of views in the main user agent window.
- A standard document has one HEAD section and one BODY. A frame-set document has a HEAD, and a FRAMESET in place of the BODY.
- Elements that might normally be placed in the BODY element must not appear before the first FRAMESET element or the FRAMESET will be ignored.

ATTRIBUTES....

rows = multi-length-list

This attribute specifies the layout of horizontal frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%, meaning one frame.

cols = multi-length-list

This attribute specifies the layout of vertical frames. It is a comma-separated list of pixels, percentages, and relative lengths. The default value is 100%.

Setting Frame Attributes

Setting the rows attribute defines the number of horizontal subspaces in a frameset. Setting the cols attribute defines the number of vertical subspaces.

If the rows attribute is not set, each column extends the entire length of the page.

If the cols attribute is not set, each row extends the entire width of the page.

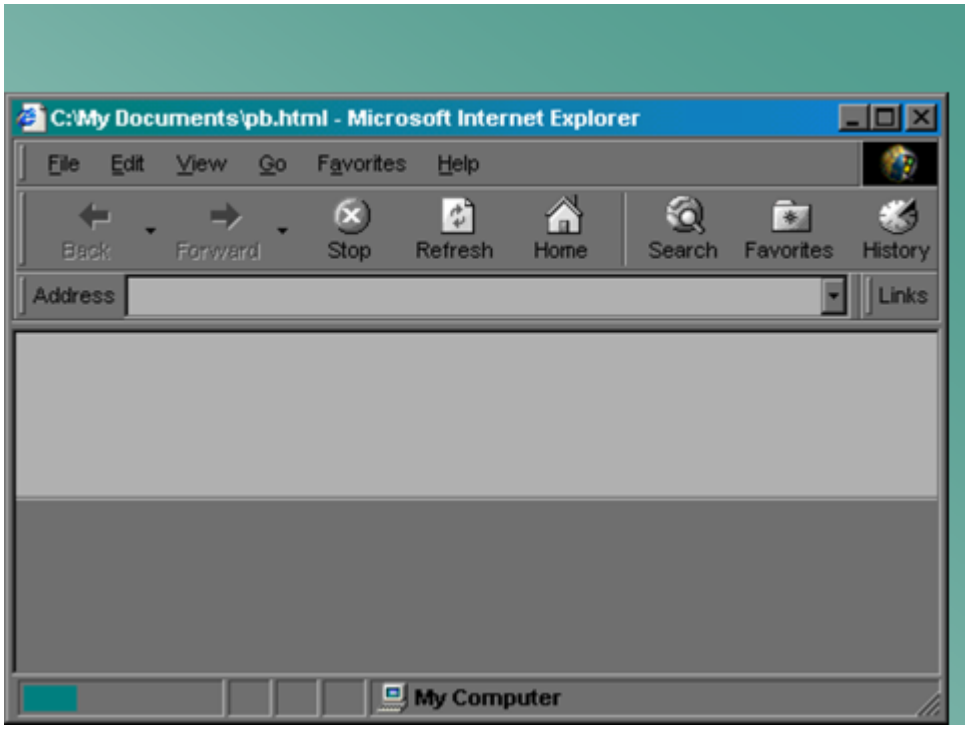
Example:

This example divides the screen vertically in two (i.e., creates a top half and a bottom half).

```
<FRAMESET rows="50%, 50%">
```

...the rest of the definition...

```
</FRAMESET>
```



<FRAME>

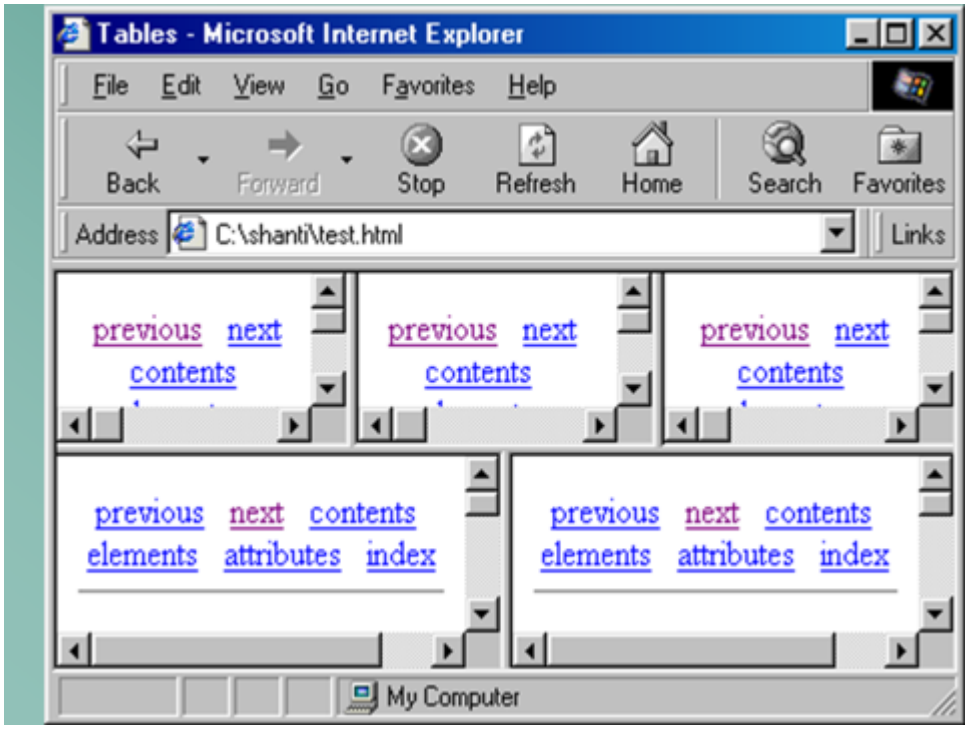
The FRAME element defines the contents and appearance of a single frame.

```
<FRAME
ALIGN=align-type     FRAMEBORDER=yes-no
MARGINHEIGHT=height   MARGINWIDTH=width
NAME=name    SCROLLING=yes-no
SRC=address >
```

SRC

The src attribute specifies the initial document the frame will contain.

```
<HTML>
<HEAD>
<TITLE>Tables</TITLE>
 <FRAMESET rows="20%,50%">
             <FRAMESET cols="33%,34%,33%">
             <FRAME src="notes.html">
             <FRAME src="notes.html">
             <FRAME src="notes.html">
</FRAMESET>
<FRAMESET cols="50%,50%">
            <FRAME src="changes.html">
          <FRAME src="changes.html">
</FRAMESET>
</FRAMESET>
 </html>
```



Noresize : When present, this boolean attribute tells the user agent that the frame window must not be resizeable.

scrolling = auto|yes|no

This attribute specifies scroll information for the frame window. Possible values

- auto: This value tells the user agent to provide scrolling devices for the frame window when necessary. This is the default value.
- yes: This value tells the user agent to always provide scrolling devices for the frame window.
- no: This value tells the user agent not to provide scrolling devices for the frame window.

ALIGN=*align-type*

- Sets the alignment of the frame or of the surrounding text. The *align-type* can be one of these values:
- TOP Surrounding text is aligned with the top of the frame.
- MIDDLE Surrounding text is aligned with the middle of the frame.
- BOTTOM surrounding text is aligned with the bottom of the frame.
- LEFT The frame is drawn as a left-flush "floating frame," and text flows around it.
- RIGHT The frame is drawn as a right-flush "floating frame," and text flows around it.

Frameborder = 1|0 :

This attribute provides the user agent with information about the frame border. Possible values are 1 & 0. Default is 1.

Marginwidth = *pixels*

This attribute specifies the amount of space to be left between the frame's contents in its left and right margins. The value must be greater than one pixel. The default value depends on the user agent.

Marginheight = *pixels* :

This attribute specifies the amount of space to be left between the frame's contents in its top and bottom margins. The value must be greater than one pixel. The default value depends on the user agent.

target = -target frame

This attribute specifies the name of a frame where a document is to be opened.

By assigning a name to a frame via the name attribute, authors can refer to it as the "target" of links defined by other elements. The target attribute may be set for elements that create links (A, LINK), image maps (AREA), and forms (FORM).

```
<HTML>
<HEAD><TITLE>A frameset document</TITLE></HEAD>
<FRAMESET rows="50%,50%">
   <FRAME name="fixed" src="init_fixed.html">
   <FRAME name="dynamic" src="init_dynamic.html">
</FRAMESET>
</HTML>
Then, in init_dynamic.html, we link to the frame named "dynamic".
<HTML>
<HEAD>
<TITLE>Anchors with specific targets</TITLE>
</HEAD>
<BODY>
...beginning of the document...
<P>Now you may advance to
  <A href="slide2.html" target="dynamic">slide 2.</A>
...more document...
<P>You're doing great. Now on to
   <A href="slide3.html" target="dynamic">slide 3.</A>
</BODY>
</HTML>
```

Forms

Forms are the best way to provide user- interaction on Internet

In a form, users can select from several choices, using a variety of selection methods, such as buttons, fill in the blanks, and selection lists.

The way the users select any information depends on the form elements you use when you create the form.

Designing Form

<FORM> .... </FORM>

The <FORM> tag creates an HTML form, which lets users input text and make choices from elements such as checkboxes, radio buttons, and selection lists.

The way your browser or the server handles your form depends on the way you specify attributes to your form.

ATTRIBUTES

- action = *uri*

This attribute specifies a form-processing agent. For example, the value might be an HTTP URI (to submit the form to a program) or a mailto URI (to email the form).

- **enctype = *content-type***

This attribute specifies the content type used to submit the form to the server (when the value of method is "post"). The default value for this attribute is "application/x-www-form-urlencoded". The value "multipart / form-data" should be used in combination with the INPUT element, type="file".

ATTRIBUTES

- method = get | post

This attribute specifies which HTTP method will be used to submit the form data set. Possible (case-insensitive) values are "get" (the default) and "post".

- get: With the HTTP "get" method, the form data set is appended to the URI specified by the action attribute and this new URI is sent to the processing agent.
- post: With the HTTP "post" method, the form data set is included in the body of the form and sent to the processing agent.

ATTRIBUTES

- accept-charset = *charset list*

This attribute specifies the list of character encoding for input data that must be accepted by the server processing this form. The value is a space- and/or comma-delimited list of charset values. The server must interpret this list as an exclusive-or list, i.e., the server must be able to accept any single character encoding per entity received.

ATTRIBUTES

- accept = *content-type-list*

This attribute specifies a comma-separated list of content types that a server processing this form will handle correctly. User agents may use this information to filter out non-conforming files when prompting a user to select files to be sent to the server (cf. the INPUT element when type="file").

FORM ELEMENTS

<INPUT >

A text field lets the user enter a word, phrase, or series of numbers. Use the <INPUT> tag to place text input fields on an HTML form.

Type = text| password| checkbox| radio| submit| reset| file| hidden| image| button

```
<FORM>      <B>Last name:</B> <INPUT TYPE="text" NAME="last_name" SIZE=25>
</FORM>
```

The output looks like: Last name:

CHECHBOX

A checkbox is a toggle switch that lets the user set a value on or off. Use the INPUT tag to implement checkboxes on an HTML form. When a form is submitted to the server, the name & value pair for the checkbox is sent only if the checkbox is checked. Multiple checkbox elements can have the same values for the NAME attribute if they have different values for the VALUE attribute.

OPTION BUTTON

A set of radio buttons lets the user choose one item from the set. Use the INPUT tag to implement radio buttons on an HTML form.

All radio buttons in a group have the same value for the NAME attribute. When a form is submitted to the server, the name & value pair for the radio button is sent only if the radio button is selected. For a group of radio buttons, only one name & value pair is sent to the server, because only one button in a group can be selected.

PASSWORD ELEMENT

Password elements are text-input fields on an HTML form that conceal their value by displaying asterisks (*). When the user enters text into the field, asterisks (*) hide anything entered from view.

HIDDEN ELEMENT

Hidden elements are text elements that don't display on the form. Use the INPUT tag to implement hidden elements. A hidden element is used for passing information to the server when a form is submitted. A hidden element cannot be seen or modified by a user (other than by viewing the source of the HTML), but by using JavaScript, you can programmatically change its value. You can use hidden elements for client/server communication or to pass state information from one script or form to another.

When a form is submitted to the server, a hidden element's name & value pair is always sent.

ATTRIBUTES OF INPUT ELEMENT

- name = *cdata*

This attribute assigns the control name.

- **value = *cdata***

This attribute specifies the initial value of the control. It is optional except when the type attribute has the value "radio".

- **size = *cdata***

This attribute tells the user agent the initial width of the control. The width is given in pixels except when type attribute has the value "text" or "password". In that case, its value refers to the (integer) number of characters.

- **maxlength = *number***

When the type attribute has the value "text" or "password", this attribute specifies the maximum number of characters the user may enter. This number may exceed the specified size, in which case the user agent should offer a scrolling mechanism. The default value for this attribute is an unlimited number.

<SELECT>

The SELECT element creates a menu. Each choice offered by the menu is represented by an OPTION element. A SELECT element must contain at least one OPTION element.

ATTRIBUTES OF SELECT

- name = *cdata*

This attribute assigns the control name.

- **size = *number***

If a SELECT element is presented as a scrolled list box, this attribute specifies the number of rows in the list that should be visible at the same time. Visual user agents are not required to present a SELECT element as a list box; they may use any other mechanism, such as a drop-down menu.

- **multiple**

If set, this Boolean attribute allows multiple selections. If not set, the SELECT element only permits single selections.

<OPTION>

When rendering a menu choice, user agents should use the value of the label attribute of the OPTION element as the choice. If this attribute is not specified, user agents should use the contents of the OPTION element.

ATTRIBUTES:-

- selected

When set, this boolean attribute specifies that this option is pre-selected.

- **value = *cdata***
- This attribute specifies the initial value of the control. If this attribute is not set, the initial value is set to the contents of the OPTION element.
- **label = *text***
- This attribute allows authors to specify a shorter label for an option than the content of the OPTION element. When specified, user agents should use the value of this attribute rather than the content of the OPTION element as the option label.

<TEXTAREA>

The TEXTAREA element creates a multi-line text input control. User agents should use the contents of this element as the initial value of the control and should render this text initially.

Attributes:-

- **name = *cdata***

This attribute assigns the control name.

- **rows = *number***

This attribute specifies the number of visible text lines. Users should be able to enter more lines than this, so user agents should provide some means to scroll through the contents of the control when the contents extend beyond the visible area.

- **cols = *number***
- This attribute specifies the visible width in average character widths. Users should be able to enter longer lines than this, so user agents should provide some means to scroll through the contents of the control when the contents extend beyond the visible area. User agents may wrap visible text lines to keep long lines visible without the need for scrolling.

EXAMPLE

```
<HTML>
<FORM NAME="form1">
<BR>
<B>City: </B>
<INPUT TYPE="text" NAME="city" VALUE="Santa Cruz" SIZE="20">
<B>State: </B>
<INPUT TYPE="text" NAME="state" VALUE="CA" SIZE="2">
<P><SELECT NAME="colorChoice">
<OPTION SELECTED> Blue
<OPTION> Yellow
<OPTION> Green
<OPTION> Red
</SELECT>
<P><INPUT TYPE="radio" NAME="musicChoice" VALUE="soul-and-r&b" CHECKED> Soul and
R&B
<BR><INPUT TYPE="radio" NAME="musicChoice" VALUE="jazz"> Jazz
<BR><INPUT TYPE="radio" NAME="musicChoice" VALUE="classical"> Classical
```

```
<CENTER><INPUT TYPE="image" SRC="alr_conn.gif"></CENTER>
<INPUT TYPE="checkbox" NAME="musicpref_rnb" CHECKED> R&B
<BR>
<INPUT TYPE="checkbox"NAME="musicpref_jazz" CHECKED> Jazz
<BR>
```

File upload elements

A file upload element is an element on an HTML form that lets the user supply a text file as input. When the form is submitted, the content of the specified file is sent to the server along with the other form data.

```
<FORM ENCTYPE="multipart/form-data" ACTION="_URL_" METHOD=POST>
          <B>First name:</B><INPUT TYPE="text">
          <BR>
          <B>Last name:</B><INPUT TYPE="text">
          <P>
          <B>Brief description of the problem:</B>
          <BR> <INPUT TYPE="text" SIZE="45">
          <P>
          <B>Please attach a file with your event log.</B>
          <BR>
          <B>File name:</B><INPUT TYPE="file">
          <P>
          <INPUT TYPE="submit" VALUE="Submit Report">
          <INPUT TYPE="button" VALUE="Cancel" onClick="window.close()">
</FORM>
```

Sending a form data to a server

HTML forms collect data, but they do not usually process it. To process a form, you can submit the form to a program stored on a web server. The form and the server-side program should be designed together so that the program can process the form data being sent. When a form is submitted, each form element defined with INPUT, SELECT, or TEXTAREA tag is sent to the server in the format name & value; this is called a name & value pair. The name comes from the tag's NAME attribute, and the value is the value of the form element when submitted.

EXAMPLE

```
<FORM>
<B>Last name:</B>
          <INPUT TYPE="text" NAME="LastName" VALUE="">
          <INPUT TYPE="submit" NAME="SubmitButton" VALUE="Done">
</FORM>
```

Suppose the user enters the value "Mitchell" for LastName and presses the Done button; the form is submitted and the name & value pair LastName=Mitchell is sent to the server.

```
<html><head><title></title> </head>
<FORM action="http://somesite.com/prog/adduser" method="post">
    <P>    <LABEL for="firstname">First name: </LABEL>
```

```
            <INPUT type="text" id="firstname"><BR>
    <LABEL for="lastname">Last name: </LABEL>
            <INPUT type="text" id="lastname"><BR>
    <LABEL for="email">email: </LABEL>
            <INPUT type="text" id="email"><BR>
    <INPUT type="radio" name="sex" value="Male"> Male<BR>
    <INPUT type="radio" name="sex" value="Female"> Female<BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
    </P> </FORM>
</html>
```

A Content Management System

Kamal Batra

ICAR-Indian Agricultural Research Institute, New Delhi - 110 012

Traditional Websites

- Slow
- Inflexible
- Custom-built (no documentation)
- Not well-optimized
- Rigid navigation
- Single, limited template

Problems with old sites

- Required Dreamweaver/Contribute licenses (not free, users tied to computers)
- Out of date + abandoned content + course content
- Permissions buggy and not granular
- Content not classified
- No workflows + no editing process
- Security concerns

What is a Content Management System (CMS)?

A complete web based system that handles

- Content management
- User management
- File (attachment, media - images, audio, video) management and many more

Is called CMS

Open Source vs Commercial

Commercial CMS

- Fees, licensing, upgrading
- Fewer modules selection
- Upgrades and enhancements

Open Source vs Commercial

Open Source CMS

- Generally no cost or licensing
- Free to modify
- Large number of modules

Different CMS

- WebGUI
- MySource Matrix
- Typo3
- Wordpress
- Joomla
- Drupal

Types of Content Management Systems

Complete content management systems

- Wordpress
- Drupal
- Joomla
- BLOGS
- WIKIS
- MediaWiki
- Confluence
- pmwiki
- Movable Type
- Blogger

Under the hood

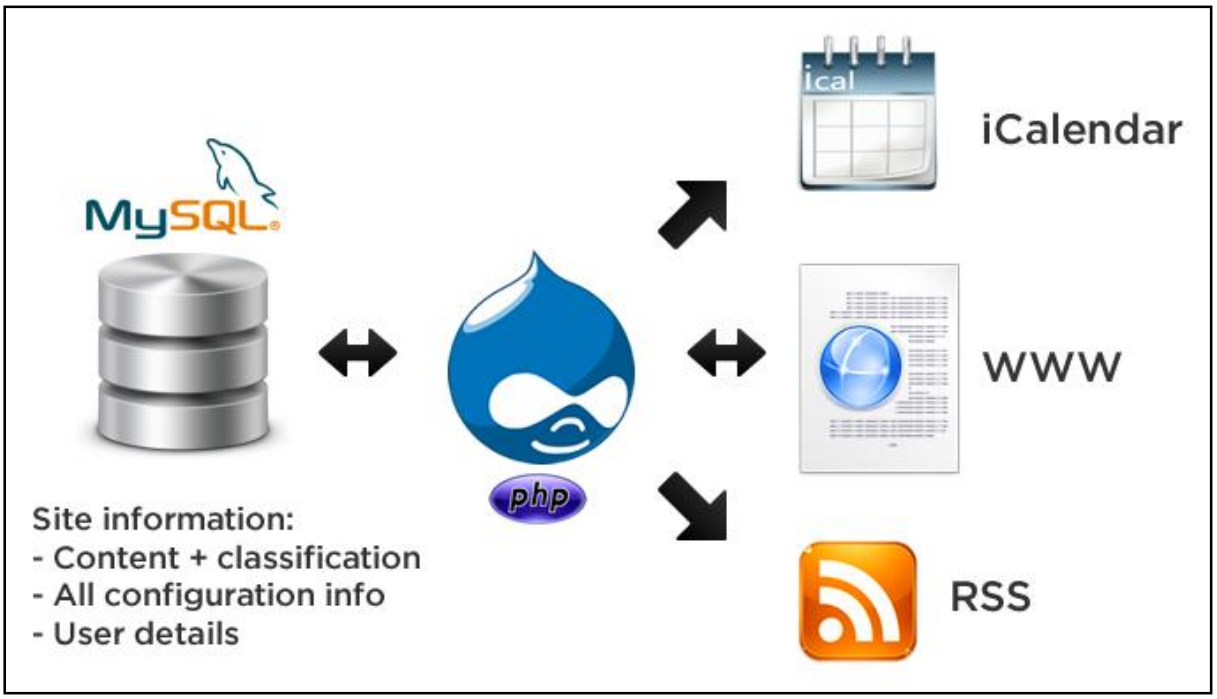


Figure 1: Architecture of web development

Why Wordpress?

- Open Source
- Content Management System
- More secure
- Community Collaboration Tools
- Community Supported
- International Translations

Features of wordpress

- Well-constructed and documented code
- Extensible
- Content Scheduling
- Taxonomy (categorization)
- Inline Page Management
- Image Resizing

More Features

- Poll
- Blog
- Image gallery

- Customized Menus
- Easy file uploading
- Calendar

Some Terminologies

- Pages
- Menus
- Plugins
- Theme
- Modules

Basics of Programming using Python

Madhu

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
madhu.dahiya@icar.gov.in

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Characteristics of Python

Following are important characteristics of **Python Programming** −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python Syntax

```
print('India is my country.')
```

Variables

```
x=2
y="India"
print(x)
print(y)
```

Type Casting

```
x=str(5)
y=int(5.0)
z=float(5)
print(x)
print(y)
print(z)
print(type(x))
print(type(y))
```

Single & Double Quotes

```
x="india"
y='india'
print(x)
print(y)
```

Multiline Strings

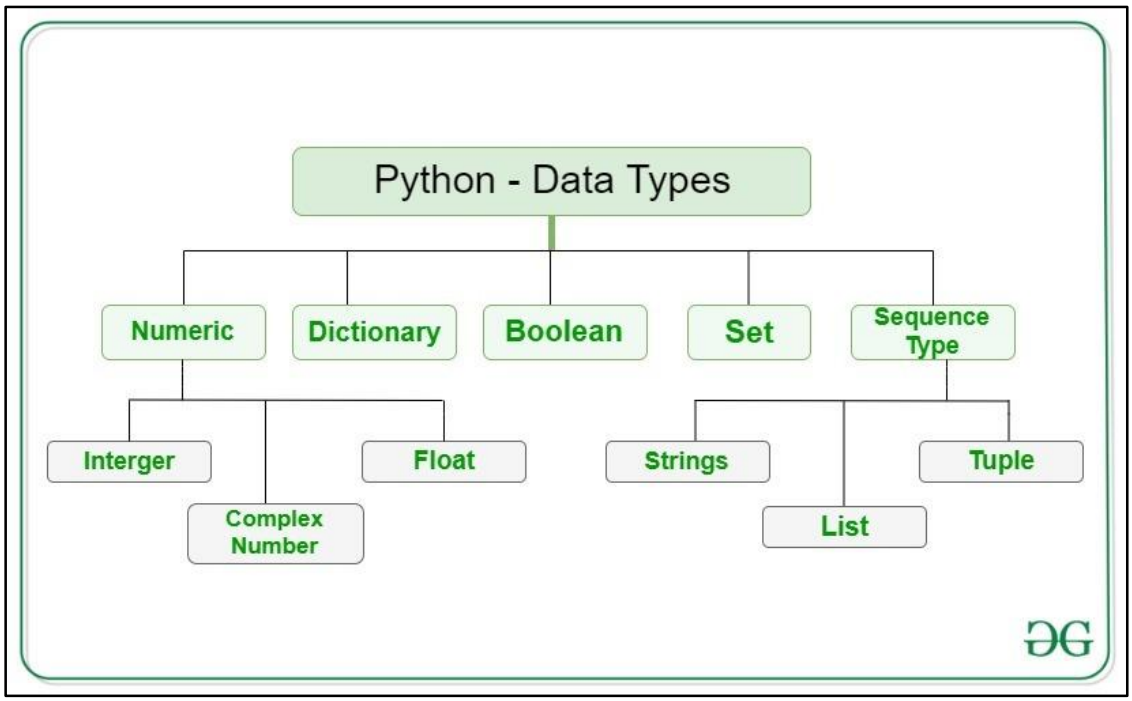
```
a = """hello,
good morning,
h r u,
all."""
print(a)

a = '''hello,
good morning,
h r u
all.'''
print(a)
```

Python Data Types

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Following are the standard or built-in data type of Python:

- Numeric
- Sequence Type
- Boolean
- Set
- Dictionary



Numeric

In Python, numeric data type represent the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

- **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
- **Float** – This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- **Complex Numbers** – Complex number is represented by complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j

Note – type() function is used to determine the type of data type.

Float

```
x = 1.10
y = 1.0
z = -35.59

print(type(x))
print(type(y))
print(type(z))
```

int

```
x = 1
y = 3565
```

```
z = -3255522

print(type(x))
print(type(y))
print(type(z))
```

complex

```
x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))
```

Sequence Type

In Python, sequence is the ordered collection of similar or different data types. Sequences allows to store multiple values in an organized and efficient fashion. There are several sequence types in Python –

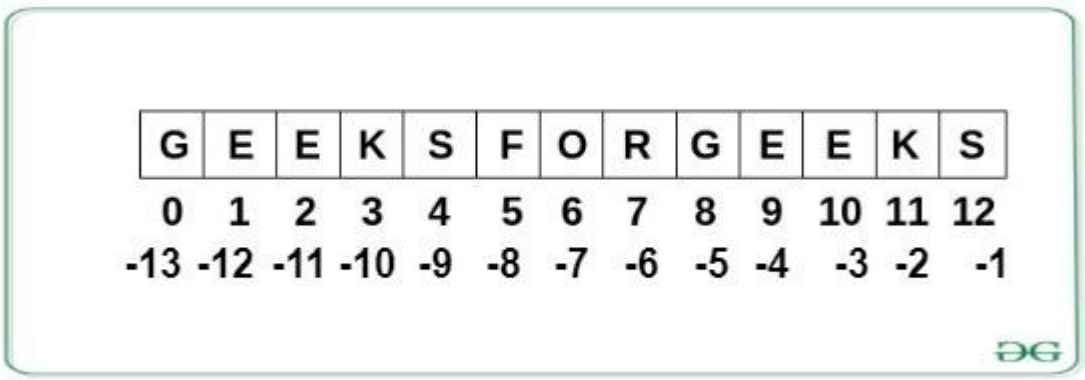
- String
- List
- Tuple

String

In Python, Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.

Creating String
Strings in Python can be created using single quotes or double quotes or even triple quotes.

Accessing elements of String
In Python, individual characters of a String can be accessed by using the method of Indexing. Indexing allows negative address references to access characters from the back of the String, e.g. -1 refers to the last character, -2 refers to the second last character and so on.




```
print("Hello")
print('Hello')

a = "Hello"
print(a)
```

Strings are Arrays

```
a = "Hello, World!"  # 0....n
print(a[1])
print(a[2])
print(a[7])
```

Looping Through a String

```
for x in "banana":
  print(x)
```

String Slicing

```
b = "Hello, World!"
print(b[2:5])

b = "Hello, World!"
print(b[:5])

b = "Hello, World!"
print(b[2:])

b = "Hello, World!"
print(b[-5:-2])
print(b[-1])
```

Strings Functions

```
a = "hello, World!"
```

```
print(a.upper())                  #Converts a string into upper case
print(a.capitalize())             #Converts the first character to upper case
print(a.casefold())             #Converts string into lower case
print(a.split())              #Splits the string at the specified separator, and returns a
list
print(a.lower())              #Converts a string into lower case
print(a.strip())                  # Returns a trimmed version of the string
                                    #returns "Hello, World!"
print(a.replace("h", "J"))            #Returns a string where a specified value is
replaced with a specified value
print(a.isdigit())                  #Returns True if all characters in the string are
digits
print(a.isupper())            #Returns True if all characters in the string are upper case
print(len(a))             #len() function returns the length of a string
```

String Concatenation

```
a = "Hello"
b = "World"
c = a + b
print(c)

a = "Hello"
b = "World"
c = a + "     "     + b
print(c)

"""we cannot combine strings and numbers"""

age = 36
txt = "My name is John, I am " + age
print(txt)
```

we can combine strings and numbers by using the **format() method!**

The format() method takes the passed arguments, formats them, and places them in the string where the placeholders {}

```
age = 36
txt = "My name is John, and I am {}"
print(txt.format(age))
```

The format() method takes unlimited number of arguments, and are placed into the respective placeholders:

You can use index numbers {0} to be sure the arguments are placed in the correct placeholders.

```
quantity = 3
itemno = 567
```

```
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity, itemno, price))

quantity = 3
itemno = 567
price = 49.95
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))
```

LIST [] O,C,DUPLICATE

TUPLES () O, NOT CHANGE,D

DICTIONARY KEY:VALUE

SET {} O,C,NOT DUPLICATE

List

Lists are just like the arrays, declared in other languages which is a ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

Creating List
Lists in Python can be created by just placing the sequence inside the square brackets[].

Accessing elements of List
In order to access the list items refer to the index number. Use the index operator [ ] to access an item in a list. In Python, negative sequence indexes represent positions from the end of the array. Instead of having to compute the offset as in List[len(List)-3], it is enough to just write List[-3]. Negative indexing means beginning from the end, -1 refers to the last item, -2 refers to the second-last item, etc

Lists are used to store multiple items in a single variable.
List items are ordered, changeable, and allow duplicate values. Lists are created using **square brackets.**

Ordered

When we say that lists are ordered, it means that the items have a defined order, and that order will not change.
If you add new items to a list, the new items will be placed at the end of the list.

Changeable

The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

Allow Duplicates

Since lists are indexed, lists can have items with the same value

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
print(type(thislist))

thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)                                          # Allow Duplicates
```

Length of a List

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))                        #len function for finding the number of values
in a tuple

list1 = ["apple", "banana", "cherry"]              #List items can be of any data type
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
print(type(list2))
print(type(list1))

list1 = ["abc", 34, True, 40, "male"]              #A list can contain different data
types
print(type(list1))
```

Indexes of List items

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])

thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
print(thislist[-2])
print(thislist[-3])
print(thislist[0])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
```

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:       # true
  print("Yes, 'apple' is in the fruits list")

thislist = ["apple", "banana", "cherry"]
if "orange" in thislist:        #false
  print("no, 'orange' is not in the fruits list")    #not executed
```

Change Item Value

```
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
thislist[1:3] = ["blackcurrant", "watermelon"]                          #Change the values
"banana" and "cherry" with the values "blackcurrant" and "watermelon
print(thislist)

thislist = ["apple", "banana", "cherry"]
thislist[1:3] = ["watermelon"]
print(thislist)
```

Insert Items

To insert a new list item, without replacing any of the existing values, we can use the **insert() method**.

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")            # insert() method inserts an item at the
specified index

thislist = ["apple", "banana", "cherry"]
thislist.append("orange")                                               #append() method add
an item to the end of the list
print(thislist)

thislist = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]

print(thislist+tropical)

thislist = ["apple", "banana", "cherry"]
thistuple = ("kiwi", "orange")
thislist.extend(thistuple)       #extend() method does not have to append lists only, you
can add any iterable object (tuples, sets, dictionaries etc.).
print(thislist)
print(type(thistuple))

thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")               #remove() method removes the specified item
print(thislist)

thislist = ["apple", "banana", "cherry"]
```

```
thislist.pop(-1)                          #pop() method removes the specified index
print(thislist)

thislist = ["apple", "banana", "cherry"]
thislist.pop()          #do not specify the index, the pop() method removes the last item
print(thislist)

thislist = ["apple", "banana", "cherry"]
del thislist[0]                           #del keyword also removes the specified index
print(thislist)

thislist = ["apple", "banana", "cherry"]
del thislist
print(thislist)                                     #Delete the entire list

thislist = ["apple", "banana", "cherry"]
thislist.clear()                              #list still remains, but it has no content
print(thislist)

thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)

thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
  print(thislist[i])

thislist = ["apple", "banana", "cherry"]
i = 0
while i < len(thislist):          #until true
  print(thislist[i])
  i = i + 1
```

Sorting List

```
thislist = ["orange", "mango", "apricot", "apple", "banana"]
thislist.sort()               #sort() method that will sort the list alphanumerically,
ascending, by default
print(thislist)

thislist = [100, 50, 50, 82, 23]
thislist.sort()
print(thislist)

thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse = False)         #To sort descending, use the keyword argument
reverse = True
print(thislist)
```

Copy a List

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()      #to make a copy, one way is to use the built-in List method
copy()
print(mylist)

thislist = ["apple", "banana", "cherry"]
mylist = list(thislist)     #make a copy is to use the built-in method list()
print(mylist)
```

Join Two Lists

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2       #by using the + operator
print(list3)

list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

for x in list2:
  list1.append(x)           #Another way to join two lists is by appending all the items
from list2 into list1, one by one

print(list1)
```

Tuple

Just like list, tuple is also an ordered collection of Python objects. The only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by tuple class.

Creating Tuple

In Python, tuples are created by placing a sequence of values separated by 'comma' with or without the use of parentheses for grouping of the data sequence. Tuples can contain any number of elements and of any datatype (like strings, integers, list, etc.).

Tuples are used to store multiple items in a single variable.
A tuple is a collection which is ordered and unchangeable.
Tuples are written with **round brackets. ()**

Ordered

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.

Unchangeable

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Allow Duplicates

Since tuples are indexed, they can have items with the same value:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)

thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)                                          #allow duplicates

thistuple = ("apple", "banana", "cherry")
print(len(thistuple))           #len function for finding the number of values in a tuple

thistuple = ("apple",)          #use comma (,) if tuple having single value otherwise it
is considered as string
print(type(thistuple))
print(len(thistuple))

thistuple = ("apple")             #NOT a tuple
print(type(thistuple))

tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
print(type(tuple1))
print(type(tuple2))

tuple1 = ("abc", 34, True, 40, "male")
print(type(tuple1))

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[2:5])             #access the elements of a tuple
```

Change Tuple Values

Once a tuple is created, you cannot change its values. Tuples are unchangeable, or immutable.

You can convert the tuple into a list, change the list, and convert the list back into a tuple

```
x = ("apple", "banana", "cherry")   #can not change the tuple
y = list(x)                 #convert tuple into list
y[1] = "kiwi"               # change list
x = tuple(y)                #convert list into tuple
```

```
print(x)

thistuple = ("apple", "banana", "cherry")                #Since tuples are immutable,
they do not have a build-in append() method
y = list(thistuple)
y.append("orange")                                         # list have append method
thistuple = tuple(y)
print(thistuple)

thistuple = ("apple", "banana", "cherry")
y = ("orange","mango")
thistuple += y            # allowed to add tuples to tuples

print(thistuple)

thistuple = ("apple", "banana", "cherry")         #  they do not have a build-in remove()
method
y = list(thistuple)
y.remove("apple")                                 # list have remove method
thistuple = tuple(y)
print(thistuple)

thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple) #this will raise an error because the tuple no longer exists
```

Packing & Unpacking Tuples

When we create a tuple, we normally assign values to it. This is called **"packing"** a tuple.

In Python, we are also allowed to extract the values back into variables. This is called **"unpacking".**

```
a = ("Delhi", 5000, "Agriculture")   #PACKS values into variable a

(City, student, type_ofcollege) = a     #UNPACKS values of variable a

print(City,student,type_ofcollege)

If the number of variables is less than the number of values, you can add an * to the
variable name and the values will be assigned to the variable as a list.

fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")

(green, yellow, *red) = fruits

print(green)
print(yellow)
print(red)

If the asterisk is added to another variable name than the last, Python will assign values
```

```
to the variable until the number of values left matches the number of variables left.

fruits = ("apple", "mango", "papaya", "pineapple", "cherry")

(green, *tropic, red) = fruits

print(green)
print(tropic)
print(red)
```

Join Tuples

```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)

fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
```

Boolean

Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.
Note – True and False with capital 'T' and 'F' are valid booleans otherwise python will throw an error.
Booleans represent one of two values: True or False.

```
print(10 > 9)
print(10 == 9)
print(10 < 9)

print(bool("Hello"))          #Almost any value is evaluated to True if it has some sort
of content.
print(bool(15))             #Any string is True, except empty strings.
print(bool(0))             # Any number is True, except 0.
print(bool(""))
print(bool(()))
print(bool([]))
print(bool({}))               # Any list, tuple, set, and dictionary are True, except
empty ones.
print(bool(False))
print(bool(None))
```

Set

In Python, Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

Creating Sets

Sets can be created by using the built-in set() function with an iterable object or a sequence by placing the sequence inside curly braces, separated by 'comma'. Type of elements in a set need not be the same, various mixed-up data type values can also be passed to the set.

Accessing elements of Sets

Set items cannot be accessed by referring to an index, since sets are unordered the items has no index. But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.
Sets are used to store multiple items in a single variable.

Set Items

Set items are unordered, unchangeable, and do not allow duplicate values.

Unordered

Unordered means that the items in a set do not have a defined order.
Set items can appear in a different order every time you use them, and cannot be referred to by index or key.

Unchangeable

Set items are unchangeable, meaning that we cannot change the items after the set has been created.

Duplicates Not Allowed

Sets cannot have two items with the same value.

Sets are written with **curly brackets.**

```
thisset = {"apple", "banana", "cherry"}
print(thisset)

thisset = {"apple", "banana", "cherry", "apple"}
print(thisset)

thisset = {"apple", "banana", "cherry"}
print(len(thisset))                    # len() function used to  find length

set1 = {"abc", 34, True, 40, "male"}
print(set1)                  # A set with strings, integers and boolean values
print(type(set1))              # type function return data type

thisset = {"apple", "banana", "cherry"}
```

```
for x in thisset:
  print(x)

thisset = {"apple", "banana", "cherry"}
print("banana" in thisset)
print("kiwi" in thisset)
```

Add an Item

```
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")              #  to add one item to a set use the add() method.
print(thisset)

thisset = {"apple", "banana", "cherry"}
tropical = {"pineapple", "mango", "papaya"}
thisset.update(tropical)                          #To add items from another set into the
current set, use the update() method
print(thisset)

thisset = {"apple", "banana", "cherry"}
mylist = ["kiwi", "orange"]
thisset.update(mylist)                #update() method does not have to be a set only, it
can be any iterable object (tuples, lists, dictionaries etc.)
print(thisset)
```

Remove Item

```
thisset = {"apple", "banana", "cherry"}
thisset.remove("apple")
thisset.remove("kiwi")              #To remove an item in a set, use the remove(), or the
discard() method.
print(thisset)                         #If the item to remove does not exist, remove()
will raise an error

thisset = {"apple", "banana", "cherry"}
thisset.discard("kiwi")                #If the item to remove does not exist, discard()
will NOT raise an error.
print(thisset)

thisset = {"apple", "banana", "cherry"}
x = thisset.pop()                   #   pop() method to remove an item,but this method
will remove the last item
print(x)                            #Remember that sets are unordered, so you will not
know what item that gets removed.The return value of the pop() method is the removed item.
print(thisset)

thisset = {"apple", "banana", "cherry"}
thisset.clear()                      #   clear() method empties the set
print(thisset)
```

```
thisset = {"apple", "banana", "cherry"}
del thisset            #  del keyword will delete the set completely
print(thisset)
```

Join Two Sets

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set3 = set1.union(set2)        # union() method returns a new set with all items from both
sets
print(set3)

set1 = {"a", "b" ,"b" ,"c"}
set2 = {1, 2, 3}

set1.update(set2)           #update() method inserts the items of set2 into set1.
                              #Both union() and update() will exclude any duplicate items
print(set1)
```

Both union() and update() will exclude any duplicate items.

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.intersection_update(y)          #intersection_update() method will keep only the items
that are present in both sets.

print(x)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.intersection(y)     #intersection() method will return a new set, that only contains
                              # the items that are present in both sets.

print(z)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.symmetric_difference_update(y)      #symmetric_difference_update() method will keep only
the elements that are
                                        #NOT present in both sets.

print(x)

x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
```

```
z = x.symmetric_difference(y)           # symmetric_difference() method will return a new
set,
                                        # that contains only the elements that are NOT
present in both sets.

print(z)
```

Dictionary

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a 'comma'.

Creating Dictionary

In Python, a Dictionary can be created by placing a sequence of elements within curly {} braces, separated by 'comma'. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can't be repeated and must be immutable. Dictionary can also be created by the built-in function dict(). An empty dictionary can be created by just placing it to curly braces{}.

Note – Dictionary keys are case sensitive, same name but different cases of Key will be treated distinctly.

Accessing elements of Dictionary

In order to access the items of a dictionary refer to its key name. Key can be used inside square brackets. There is also a method called get() that will also help in accessing the element from a dictionary.

Dictionaries are used to store data values in **key:value pairs**.

A dictionary is a collection which is ordered, changeable and do not allow duplicates.

Ordered

When we say that dictionaries are ordered, it means that the items have a defined order, and that order will not change.

Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Duplicates Not Allowed

Dictionaries cannot have two items with the same key.

```
thisdict = {
  "brand": "Ford",

  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"])
print(len(thisdict))              # len () function returns length
print(type(thisdict))             # type() function show datatype

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020             #Dictionaries cannot have two items with the same key.
                             # Duplicate values will overwrite existing values
}
print(thisdict)
```

Accessing Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict["model"]
y = thisdict.get("model")       #Get the value of the "model" key
z = thisdict.keys()         #keys() method will return a list of all the keys in the
dictionary
b = thisdict.values()        #values() method will return a list of all the values in the
dictionary
print(x)
print(y)
print(z)
print(b)
```

Change Dictionary Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["year"] = 2018
print(thisdict)

thisdict = {
  "brand": "Ford",
```

```
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"year": 2020})   #update() method will update the dictionary with the
items from the given argument.
print(thisdict)
```

Adding Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["color"] = "red"
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"color": "red"})         #update() method will update the dictionary with
the items from a given argument
print(thisdict)
```

Removing Items

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.pop("model")     #pop() method removes the item with the specified key name
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.popitem()          #popitem() method removes the last inserted item
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict["model"]       #del keyword removes the item with the specified key name
```

```
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict              #del keyword can also delete the dictionary completely
print(thisdict)

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.clear()          #clear() method empties the dictionary
print(thisdict)
```

Loop Through a Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
for x in thisdict:        #Print all key names in the dictionary, one by one
  print(x)
for y in thisdict:
  print(thisdict[y])      #Print all values in the dictionary, one by one
for x in thisdict.values(): #values() method to return values of a dictionary
  print(x)
for x in thisdict.keys():
  print(x)                    #keys() method to return the keys of a dictionary
for x, y in thisdict.items(): #items methods to return values for both key and values
  print(x, y)
```

Copy a Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
mydict = thisdict.copy()          # Make a copy of a dictionary with the copy() method
print(mydict)
print(thisdict)
mydict = dict(thisdict)     # to make a copy is to use the built-in function dict()
print(mydict)
```

Nested Dictionaries

A dictionary can contain dictionaries, this is called nested dictionaries.

```
myfamily = {
  "child1" : {
    "name" : "Emil",
    "year" : 2004
  },
  "child2" : {
    "name" : "Tobias",
    "year" : 2007
  },
  "child3" : {
    "name" : "Linus",
    "year" : 2011
  }
}
print(myfamily)
```

```
child1 = {
  "name" : "Emil",
  "year" : 2004
}
child2 = {
  "name" : "Tobias",
  "year" : 2007
}
child3 = {
  "name" : "Linus",
  "year" : 2011
}

myfamily = {
  "child1" : child1,
  "child2" : child2,
  "child3" : child3
}
print(myfamily)
```

Python Arrays

Array in Python can be created by importing array module. array(data_type, value_list) is used to create an array with data type and value list specified in its arguments.

import array as arr

```
# creating an array with integer type
a = arr.array('i', [1, 2, 3])
```

```
# printing original array
print ("The new created array is : ", end =" ")
for i in range (0, 3):
    print (a[i], end =" ")
print()
```

Accessing Python Array Elements

```
import array as arr
a = arr.array('i', [2, 4, 6, 8])

print("First element:", a[0])
print("Second element:", a[1])
print("Last element:", a[-1])
```

Slicing Python Arrays

```
import array as arr

numbers_list = [2, 5, 62, 5, 42, 52, 48, 5]
numbers_array = arr.array('i', numbers_list)

print(numbers_array[2:5]) # 3rd to 5th
print(numbers_array[:-5]) # beginning to 4th
print(numbers_array[5:])  # 6th to end
print(numbers_array[:])   # beginning to end
```

Changing and Adding Elements

```
import array as arr

numbers = arr.array('i', [1, 2, 3, 5, 7, 10])

# changing first element
numbers[0] = 0
print(numbers)     # Output: array('i', [0, 2, 3, 5, 7, 10])

# changing 3rd to 5th element
numbers[2:5] = arr.array('i', [4, 6, 8])
print(numbers)     # Output: array('i', [0, 2, 4, 6, 8, 10])

import array as arr

numbers = arr.array('i', [1, 2, 3])

numbers.append(4)
print(numbers)     # add one item to the array using the append() method

numbers.extend([5, 6, 7])
```

```
print(numbers)     # add several items using the extend() method

import array as arr

odd = arr.array('i', [1, 3, 5])
even = arr.array('i', [2, 4, 6])

numbers = arr.array('i')   # concatenate two arrays using + operator
numbers = odd + even

print(numbers)

import array as arr

number = arr.array('i', [1, 2, 3, 3, 4])

del number[2]  # removing third element
print(number)  # Output: array('i', [1, 2, 3, 4])

del number  # deleting entire array
print(number)  # Error: array is not defined

import array as arr

numbers = arr.array('i', [10, 11, 12, 12, 13])

numbers.remove(12)    # remove() method to remove the given item
print(numbers)
print(numbers.pop(2))   # pop() method to remove an item at the given index
print(numbers)
```

Python Operators

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.

Arithmetic operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, etc.

+ Add two operands or unary plus x + y+ 2:

- Subtract right operand from the left or unary minus x - y- 2

* Multiply two operands x * y

/ Divide left operand by the right one (always results into float) x / y

% Modulus - remainder of the division of left operand by the right x % y (remainder of x/y)

// Floor division - division that results into whole number adjusted to the left in the number line x // y

** Exponent - left operand raised to the power of right x**y (x to the power y)
"""

```
x = 3
y = 2
print('x + y =',x+y)
print('x - y =',x-y)
print('x * y =',x*y)
print('x / y =',x/y)
print('x // y =',x//y)
print('x ** y =',x**y)
```

Comparison operators

Comparison operators are used to compare values. It returns either True or False according to the condition.

> Greater than - True if left operand is greater than the right x > y

< Less than - True if left operand is less than the right x < y

== Equal to - True if both operands are equal x == y

!= Not equal to - True if operands are not equal x != y

>= Greater than or equal to - True if left operand is greater than or equal to the right x >= y

<= Less than or equal to - True if left operand is less than or equal to the right x <= y
"""

```
x = 5
y = 10

print('x > y is',x>y)
print('x < y is',x<y)
print('x == y is',x==y)
print('x != y is',x!=y)
print('x >= y is',x>=y)
print('x <= y is',x<=y)
```

Logical operators

Logical operators are the and, or, not operators.

and True if both the operands are true x and y

or True if either of the operands is true x or y

not True if operand is false (complements the operand) not x
"""

```
x = True
y = False

print('x and y is',x and y)    #true true

print('x or y is',x or y)       #either true

print('not x is',not x)
```

Assignment operators

Assignment operators are used in Python to assign values to variables.

a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

= x = 5 x = 5

+= x += 5 x = x + 5

-= x -= 5 x = x - 5

*= x *= 5 x = x * 5

/= x /= 5 x = x / 5

```
a = 21
b = 10
c = 0

c = a + b
print ("Value of c is ", c)

c += a
print ("Value of c is ", c)
```

```
c *= a
print ("Value of c is ", c)

c /= a
print ("Value of c is ", c)

c  = 2
c %= a
print ("Value of c is ", c)

c **= a
print ("Value of c is ", c)

c //= a
print ("Value of c is ", c)
```

Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

& AND Sets each bit to 1 if both bits are 1

| OR Sets each bit to 1 if one of two bits is 1

^ XOR Sets each bit to 1 if only one of two bits is 1

~ NOT Inverts all the bits

```
a = 10          #1010    0101
b = 4           #0100

# Print bitwise AND operation
print("a & b =", a & b)         #0000

# Print bitwise OR operation
print("a | b =", a | b)       #1110

# Print bitwise NOT operation
print("~a =", ~a)                #  ~a = ~1010
                                 #    = -(1010 + 1)
                                 #    = -(1011)
                                 #     = -11 (Decimal)

# print bitwise XOR operation
print("a ^ b =", a ^ b)         # Returns 1 if one of the bits is 1 and the other is 0
else returns false.
```

Shift Operators

These operators are used to shift the bits of a number left or right thereby multiplying or dividing the number by two respectively.

Bitwise right shift: Shifts the bits of the number to the right and fills 0 on voids left(fills 1 in the case of a negative number) as a result.

Bitwise left shift: Shifts the bits of the number to the left and fills 0 on voids right as a result.

```
a = 10
b = -10

# print bitwise right shift operator
print("a >> 1 =", a >> 1)
print("b >> 1 =", b >> 1)

a = 5
b = -10

# print bitwise left shift operator
print("a << 1 =", a << 1)
print("b << 1 =", b << 1)
```

7) Identity operators

is and is not are the identity operators in Python. They are used to check if two values (or variables) are located on the same part of the memory.

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]
print(x1 is not y1)     # Output: False

print(x2 is y2)         # Output: True

print(x3 is y3)       # Output: False
```

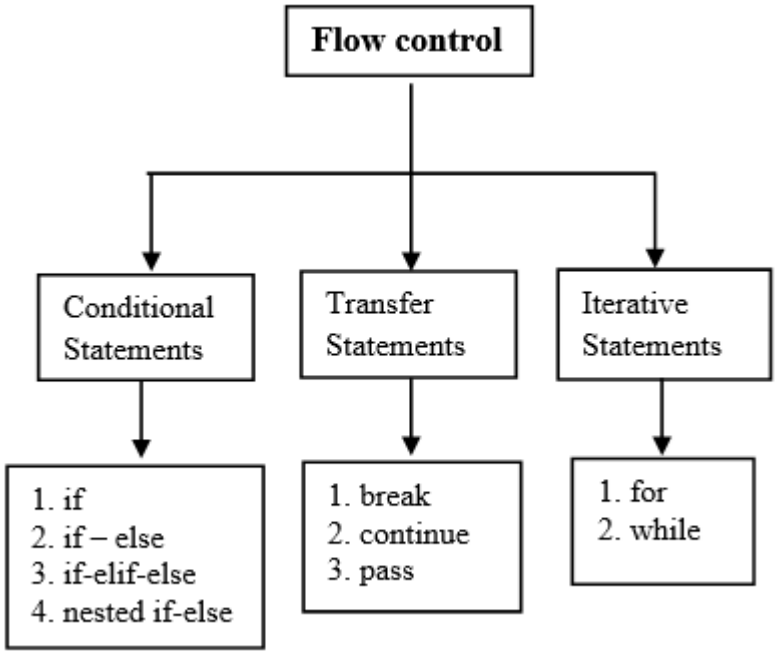
Membership operators

in and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

```
x = 'Hello world'
y = {1:'a',2:'b'}

print('H' in x)         # Output: True

print('hello' not in x)   # Output: True

print(1 in y)             # Output: True

print('a' in y)           # Output: False
```

Control Flow Statements

The flow control statements are divided into three categories
1. Conditional statements
2. Iterative statements.
3. Transfer statements



Python If ... Else

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

Elif

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

Else

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Short Hand If

```
if a > b: print("a is greater than b")
```

Short Hand If ... Else

```
a = 2
b = 330
print("A") if a > b else print("B")
```

Nested If

```
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

The pass Statement

```
a = 33
```

```
b = 200

if b > a:
 pass
```

while Loop

```
i = 1
while i < 6:
  print(i)
  i += 1
```

break Statement

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

continue Statement

```
i = 0     #
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

For Loop

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Looping Through a String

```
for x in "banana":
  print(x)
```

range() Function

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)

for x in range(2, 6):
  print(x)                # not including 6

for x in range(2, 30, 3):             #third parameter is the increment value
  print(x)

for x in range(6):
  print(x)
else:                    #else keyword in a for loop specifies a block of code to be
executed when the loop is finished
  print("Finally finished!")

for x in range(6):
  if x == 3: break                  #else block will NOT be executed if the loop is
stopped by a break statement
  print(x)
else:
  print("Finally finished!")
```

Nested Loops

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

Functions

You use functions in programming to bundle a set of instructions that you want to use repeatedly. That means that a function is a piece of code written to carry out a specified task.

There are three types of functions in Python:

- **User-Defined Functions (UDFs),** which are functions that users create to help them out.
- **Anonymous functions,** which are also called **lambda functions** because they are not declared with the standard def keyword.
- **Built-in functions,** such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal.

Creating a Function

```
def my_function():
  print("Hello from a function")

"""**Calling** **a** **Function** """

def my_function():
  print("Hello from a function")

my_function()

def my_function(fname):                        #A parameter is the variable listed inside
the parentheses in the function definition.
  print(fname + " Refsnes")

my_function("Emil")                          #An argument is the value that is sent to the
function when it is called.
my_function("Tobias")
my_function("Linus")
```

Number of Arguments

```
def my_function(fname, lname):
  print(fname + " " + lname)

my_function("Emil", "Refsnes")

def my_function(*kids):                          #do not know how many arguments that will
be passed into your function, add a * before the parameter name in the function definition
  print("The youngest child is " + kids[2])

my_function("Emil", "Tobias", "Linus")

def my_function(child3, child2, child1):
  print("The youngest child is " + child3)

my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")           #send
arguments with the key = value syntax

def my_function(**kid):                         #number of keyword arguments is unknown, add
a double ** before the parameter name
  print("His last name is " + kid["lname"])

my_function(fname = "Tobias", lname = "Refsnes")
```

Default Parameter Value

```
def my_function(country = "Norway"):
```

```
  print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()                    #If we call the function without argument, it uses the
default value
my_function("Brazil")
```

Passing a List as an Argument

```
def my_function(food):
  for x in food:
    print(x)

fruits = ["apple", "banana", "cherry"]

my_function(fruits)
```

Return Values

```
def my_function(x):
  return 5 * x

print(my_function(3))
print(my_function(5))
print(my_function(9))
```

Python Lambda

A lambda function is a small anonymous function. A lambda function can take any number of arguments, but can only have one expression.

```
x = lambda a : a + 10
print(x(5))

Max = lambda a, b : a if(a > b) else b            # Example of lambda function using if-
else

print(Max(1, 2))
```

Difference Between Lambda functions and def defined function

```
def cube(y):
    return y*y*y

lambda_cube = lambda y: y*y*y
```

```
print(cube(5))
print(lambda_cube(5))
```

Python Built-In Functions

all()

The python all() function accepts an iterable object (such as list, dictionary, etc.). It returns true if all items in passed iterable are true. Otherwise, it returns False. If the iterable object is empty, the all() function returns True.

```
k = [1, 3, 4, 6]              # all values true
   print(all(k))

   k = [0, False]                # all values false
   print(all(k))

   k = [1, 3, 7, 0]            # one false value
   print(all(k))

   k = [0, False, 5]         ## one true value
   print(all(k))

   k = []                      # empty iterable
print(all(k))

test1 = []
   print(test1,'is',bool(test1))
   test1 = [0]
   print(test1,'is',bool(test1))
   test1 = 0.0
   print(test1,'is',bool(test1))
   test1 = None
   print(test1,'is',bool(test1))
   test1 = True
   print(test1,'is',bool(test1))
   test1 = 'Easy string'
print(test1,'is',bool(test1))

x = 10
print('Absolute value of -40 is:', abs(x))    #abs() function is used to return the
absolute value of a number
floating = -20.83
print('Absolute value of -20.83 is:', abs(floating))
y =  bin(x)  #bin() function is used to return the binary representation of a specified
integer.
print (y)
s = sum([1, 2,4 ])               #sum() function is used to get the sum of numbers of an
iterable, i.e., list.
print(s)
```

```
print(float(9))                 # float() function change into float number
print(complex(9))                # complex() function change into complex number
```

Development of applications using Python

Samarth Godara

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
samarth.godara@icar.gov.in

Application software is a type of computer program that performs a specific personal, educational, and business function. Each application is designed to assist end-users in accomplishing a variety of tasks, which may be related to productivity, creativity, or communication. The most common software application platforms are used by millions of people every day. They're designed to help with specific tasks, simplify workflows, and improve communication across teams, example:

- The Microsoft Suite: Office, Excel, Word, PowerPoint, Outlook, etc.
- Internet Browsers: Firefox, Chrome, Safari, Internet Explorer
- Music Software: Pandora, Apple Music, Spotify
- Communication Software: Slack, Skype, Zoom, Teams

In computer programming, computer code refers to the set of instructions, or a system of rules, written in a particular programming language (i.e., the source code). It is also the term used for the source code after it has been processed by a compiler and made ready to run on the computer (i.e., the object code). In addition to building computer programs and mobile applications, code is used heavily for innovative concepts such as artificial intelligence and machine learning. Of course, there are several other uses and applications for the word code, explained in the next section. A code is a set of instructions for a computer to perform. It's a bit like a cooking recipe, for example:

1. Crack an egg into a bowl
2. Whisk the egg
3. Put a pan on medium heat
4. Grease the pan
5. Pour the eggs into the pan
6. And so on

Instead of manipulating ingredients, computers manipulate data. The instructions inside software look more like this:

1. Load some data
2. Load some other data
3. Transform those two bits of data into result data
4. Send the result data to the monitor for display
5. Send the result data over the internet
6. And so on

Mobile Application Development Using Android

Chandan Kumar Deb and Md.Ashraful Haque

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012

chandan.deb@icar.gov.in

ashraful.haque@icar.gov.in

Introduction

Android is a mobile operating system. It is an open-source framework and is based on Linux. The Android framework helps us to develop advanced and user-friendly applications. The applications can be built using Java and Kotlin. The Android operating system has then gone through numerous releases by fixing bugs as well as adding additional features which make our life more comfortable and easy.

History of Android Technology

It is an operating system developed by Android Inc. and then overtaken by Google. Android Inc. was developed in Palo Alto California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White.

, in 2007. The first version was released by Google and the commercial version was released in 2008 known as Android 1.0.

Besides, numeric names, Google has assigned code names to all Android versions. The following picture depicts all the versions and their code names.

Version

- The first version of Android 1.0, released on 23 September 2003.
- The second version, released on 9 Feb 2009.
- Then version 1.5 known as Cupcake, released on 27 April 2009.
- Version 1.6 known as Donut, released on 15 Sept 2009.
- Then on 26 Oct 2009 version 2.0-2.1 was released.
- Froyo version 2.2-2.2.3, released on 20 May 2010.
- On 6 Dec 2010 gingerbread version 2.3-2.3.7 was released.
- Honeycomb version 3.0-3.2.6, released on 22 Feb 2011.
- Ice cream version 4.0-4.0.4, released on 18 Oct 2011.
- Jelly bean version 4.1-4.3.1, released on 9 July 2012.
- Kit-kat version 4.4-4.4.4, released on 31 Oct 2013.
- Lollipop version 5.0-5.1..1, released on 12 Nov 2014.
- Marshmallow 5 version 6.0-6.0.1, released on Oct 2015.
- Nougat version 7.0-7.1.2, released on 22 Aug 2016.
- Oreo version 8.0-8.1, released on 21 Aug 2017.
- Version 9 also known as Pie, released on 6 Aug 2018.
- Version 10 also known as Android 10, released on 3 September 2019.

Why Learn Android Technology?

Android technology is not constrained to only cell phones, nowadays there are many devices in the market that use it as their operating system. Devices like television sets, tablets, Android auto cars, ebook-reader, and wristwatches use Android as the operating system. It is leading the global market. Most of the population uses Android devices. The applications we use every day has brought plenty of jobs available for Android developers in the market. As Android is open-source anyone can learn and build Android applications. Also, Android applications are compatible with a variety of devices. No doubt Android development is one of the enthusiastic and interesting jobs in this period of technology.

Android Features

As we know the Android has changed our lives, let's discuss some of the Android features.

1. **Voice-search**

This feature lets the user search by recording the voice message instead of typing it. Example- If we want to call XYZ person, we just have to speak and the call will be directed to the XYZ person, performing multi-tasking. With this feature, we can watch a video and also play games simultaneously.

2. **Screen-capture**

We can capture the screen using this feature.

3. **Multiple Language Support**

English is the default language but now we can use any local language. Also, Android supports multiple languages.

4. **Gestures**

With the help of gestures, we can use the phone without even touching it.

5. **Tethering**

With this feature, we can share internet connections through the wired./wireless hotspot.

6. Media Support

Android supports the following media H.263, H.264, MPEG-4, AMR, AMR-WB, AAC, HE-AAC, MP3, JPG, PNG, etc.

7. Storage

SQLite is an open-source relational database that is inbuilt in Android.

8. Auto-correct

This feature suggests words and corrects grammatical mistakes.

9. Sensors

Almost, every mobile phone has inbuilt sensors that sense the motion of the phone. Some of the inbuilt sensors are an accelerometer, heart rate, magnetic field sensor, gyroscope.

Android Architecture

When we talk about Android we know that it is an operating system and we use applications that are built on it. Now we need to understand exactly what is Android and what enables it to work the way it does. For that, we need to have an exact vision of its components and functionalities.

Here comes the need and use of Android Architecture. Android Architecture is a software stack of components that are required to build an Android Application. Android Architecture contains 5 major layers that an Architecture works in. Android Architecture explains the complete working of Android through these five layers only.

These five layers are:

System Applications

Java API Framework

Native Libraries

Android Runtime

Linux Kernel

Frameworks for Android Development

Let's discuss the tools required for application development. There are many tools available for Android development. Some of the best Android development tools are listed below:

Android Applications

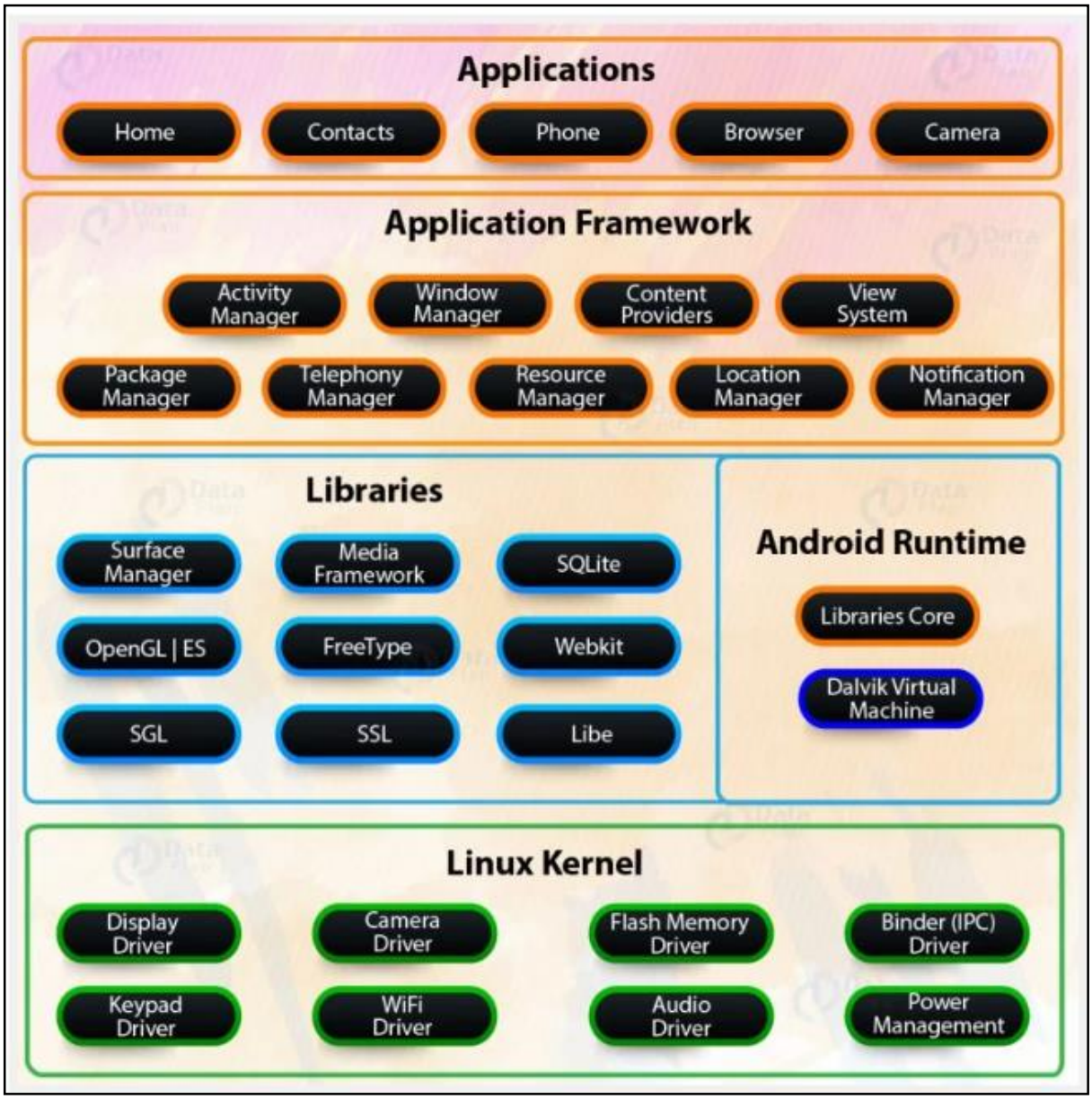
In this Android tutorial, we have discussed various types of applications that can be created using Android development which helps to save time and energy. Various types of applications are Music, Lifestyle, Social Media, Navigation, Finance, Weather, Travel etc.

Top Android Features

Following is the list of unique and best Android features:

Split-Screen, Dark Mode, Auto-correct, Gesture Control, Direct USB Access, Bluetooth, Multiple Language Support, Text To Speech Feature, Storage And Battery Swap, Customize Home Screen, Notification Dots, Snooze Notification, Fingerprint scanner, Alternate Keyboard, Infrared Transmission, Improved Files App, Google Assistant, Overheating USB Warnings.

Android Architecture



When we talk about Android applications, there is a wide range of languages available for Android app Development. Before getting to which one is the best, let us see some languages that you can consider for development. Following are few of the common languages for Android Application development:

Top 7 Programming Languages for Android App Development

- Java
- Kotlin
- C#
- Python
- C/C++
- Corona
- BASIC

Android Application Components

1. Activities

2. Services

3. Content Providers

4. Broadcast Receiver

Additional Components of Android Application

1. Intents

2. Widgets

3. Views

4. Notifications

5. Fragments

6. Layout XML Files

7. App APK files

8. Resources

What is Android Studio?

Android Studio is the Official IDE for Google's Operating System that is Android. It is built on JetBrains' IntelliJ IDEA software, especially for Android Application Development. This can be used on Windows, macOS, and also Linux based operating systems. It has risen as a replacement for Eclipse Android development tools for the native Android app development. Google announced Android Studio as the primary IDE for native android apps on 16th May 2013 at the Google I/O conference.

Java was the official language of Android Studio for Android app development, which was recently replaced by Kotlin on 7th May 2019.

Why is Android Studio used?

Android Studio is a development framework that has every tool required for Android Application Development. The purpose of developing it was to accelerate the development and help developers. It helps the developers to build the apps of the highest quality for every Android Device. It offers many tools to developers for coding, editing, designing, debugging, testing, and whatnot. Android Studio also has some features such as auto code completion and refactoring that makes the project development hasslefree. Android Studio is useful for Android application development if you're good with either Java or Kotlin.

What are Features of Android Studio?

We already know that Android Studio is based on IntelliJ that does all the things that Eclipse did with the ADT plug-in. Other than that, it has many things and many more new and exciting features that we will see. Without any delay let us check them out-

- It has Gradle-built support
- It lets debug the code easily and has quick fixes.
- There are lint tools that catch performance issues, usability, or version compatibility problems.
- It has a Proguard tool that is one of the best things in Android Studio.
- It has a template-based wizard that helps in creating common Android designs and layouts.
- This has a rich editor that allows us to drag and drop the UI components and check it very easily.
- It has a rich color preview editor that lets us add the color and the color name in the resource directory.
- It has a deep code analysis, which means it shows exactly where the bug is and also gives suggestions to correct it without any complexity.

Android Studio Installation

So guys, we know you are interested in Android Application development. To begin with application development in Android Studio, you must begin with its installation. Learn Android installation in easy steps.

First Android App in Android Studio

Once you have Android Studio ready with you, now you can proceed to create your first ever Android App. Before beginning with it, it would be good if you brush up the following skills-

- Object-Oriented Programming
- Java Programming
- Basics of extensible markup language

Android Studio for Android Applications

Android Studio has many exciting things that help and encourages developers to create many new applications. Let us see these things one by one in the following:

a. **Code and debug Faster**

As Android Studio is based on IntelliJ, it provides the fastest turnaround for its coding and workflow. It also has auto code completion that is of great use for the developers.

b**. Easy and fast Run**

It has an instant run feature that immediately runs the app using the codes and the resources provided. It is intelligent enough to understand the changes and show them immediately in the app.

c. **Smart Code editor**

Its code editor helps the developer in writing better code faster and more productive. For this, it offers advanced code completion, refactoring, and code analysis features. As you type, it provides suggestions from which you can easily choose the best fit.

d. **Faster Emulator**

The Android Emulator installs the apps faster than a real device and lets the developers test their app on various Android Virtual Devices. These devices can be smartphones, tablets, Android Wear, and even Android TV devices.

e. **Great Build System**

Android Studio offers rich dependency management and customizable build configurations for apps. The developers can configure the projects and include local as well as hosted libraries. The build variants that include different code and resources can also be defined.

f. **Supports all Android devices**

Android Studio IDE provides a unified environment to build apps for all Android devices. These devices are phones, tablets, Android Wear, and also TVs. It also has the facility of working in modules that allow developers to divide their projects into functional units that can be independently built, tested, and debug.

g. **Templates and sample apps**

Android Studio provides templates and sample applications that make it easy to add well-established patterns such as menu or navigation, etc. The developer can start with a code template or even an API in the editor. It also has this functionality where the code can be directly imported in the Android Studio and run.

h. **Testing tools**

Android Studio provides many tools that help in testing Android apps and their functional UI parts. UI test codes can be easily generated using Espresso Test Recorder, recording the interactions with the app on an emulator or a real device.

i. **Keyboard shortcuts**

Here are 10 shortcuts to help you through your Android Application Development:

- Ctrl + F – It will help you find words in your file.
- Ctrl + R – It will help you replace a word/ name.
- F5 – It will help you to copy.
- Ctrl+ (+/-) – It will let you zoom in or out.
- F11 – It will get the Android Studio in full-screen mode.
- Ctrl + B – It will open the XML file.
- Ctrl + N – It will help you find a class.
- Shift + F6 – It will help you to rename a class name or variable name.
- Ctrl + Shift+ N – It will help you find a file.
- Ctrl + I – It will help to implement methods.

Some other Tips

We would like to conclude this tutorial by providing you some tips and tricks. These will help you out throughout the journey of Android App development using Android Studio-

- You can change the theme as per your convenience as there is a light and a dark theme to use Android Studio.
- You can make a very fast search if you need to find the files or classes or symbols. It is very simple. You can simply press the Shift key two times, and there you go; you can search it.
- While writing the code, you would be given suggestions, pressing the tab key will type that automatically.
- You can also check the CPU, memory, and network usage of the applications in Android Studio. This can be done through the profile check feature in it.
- Also, if there is some issue with the name of the class or variable or method, you can easily change it too. You just need to right click and choose the refactor and then rename. And you are done.

What is Activity?

We know by now, that an activity is a single screen of an application that lets us see and interact to perform an activity. Usually, an application contains many screens and each screen extends **Activity()** class.

When we work on an application what we see is a UI on the screen which is an activity. Most of the applications that we use have many activities. Among all those activities, one is the **MainActivity()** & the rest are its **ChildActivities()**. Generally, the first page that appears on the screen when an application opens is referred to as MainActivity(). This main activity interacts with the child activities and lets the user access them.

Android Activity Lifecycle

An activity can have four states, which are :

1. Running

2. Paused
3. Resumed
4. Stopped

The above are the four states that Android activity can achieve during its whole lifecycle.

1. Running State

An activity is in the **running** state if it's shown in the foreground of the users' screen. Activity is in the running state when the user is interacting with it.

2. Paused State

When an activity is not in the focus but is still alive for the user, it's in a **paused** state. The activity comes in this state when some other activity comes in with a higher position in the window.

3. Resumed State

It is when an activity goes from the paused state to the foreground that is an **active** state.

4. Stopped State

When an activity is no longer in the activity stack and **not visible** to the users.

1. **onCreate()**

The Android oncreate() method is called at the very start when an activity is created. An activity is created as soon as an application is opened. This method is used in order to create an Activity.

2. **onStart()**

The Android onstart() method is invoked as soon as the activity becomes visible to the users. This method is to start an activity. The activity comes on the forescreen of the users when this method is invoked.

3. **onPause()**

The Android onPause() method is invoked when the activity doesn't receive any user input and goes on hold. In the pause state, the activity is partially visible to the user. This is done when the user presses the back or home buttons. Once an activity is in the pause state, it can be followed by either **onResume()** or **onStopped()** callback method.

4. **onRestart()**

The Android onRestart() method is invoked when activity is about to start from the stop state. This method is to restart an activity that had been active some time back. When an activity restarts, it starts working from where it was paused.

5. **onResume()**

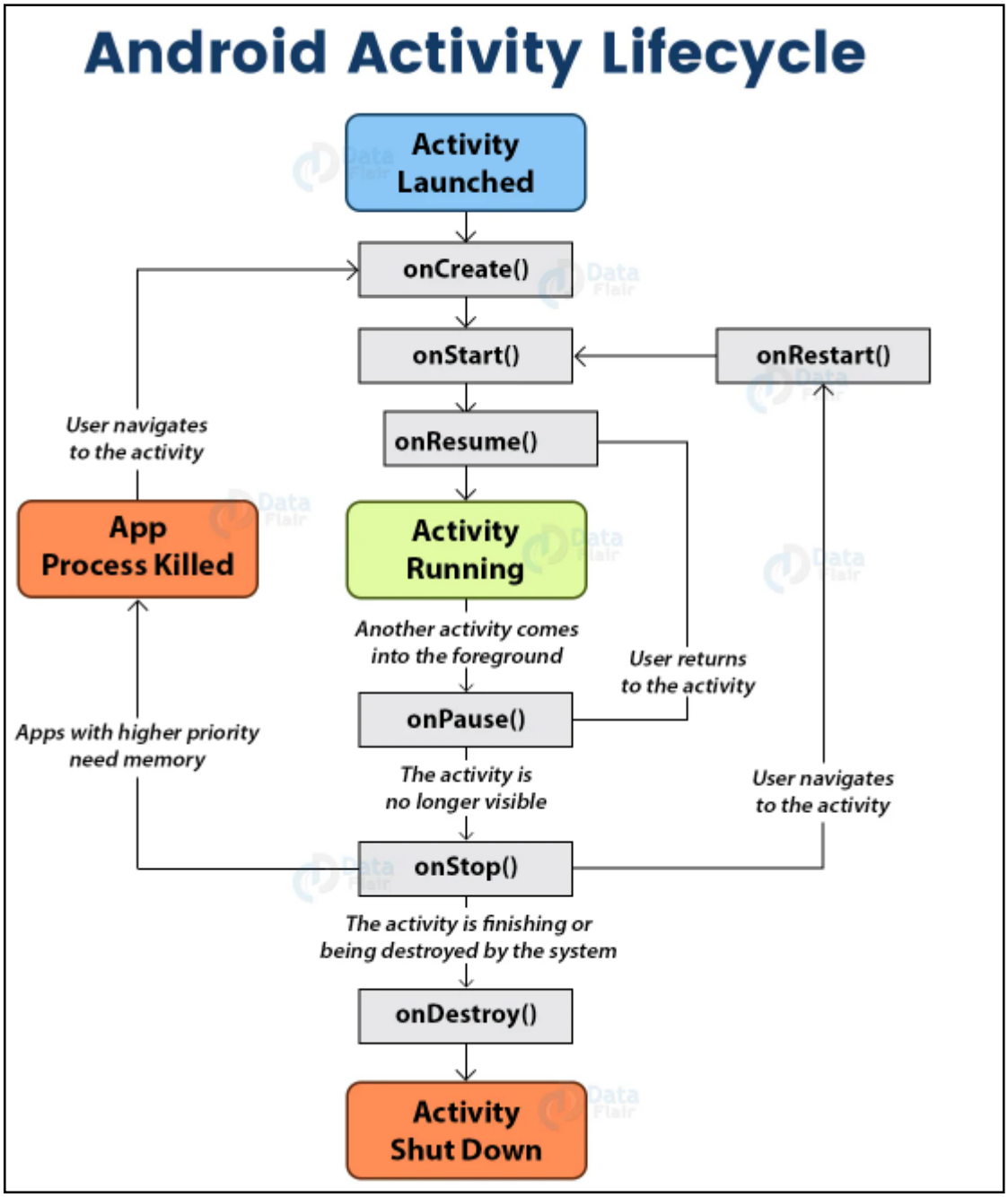
The Android onResume() method is invoked when the user starts interacting with the user. This callback method is followed by **onPause()**. Most of the functionalities of an application are implemented using **onResume()**.

6. **onStop()**

The Android onStop() method is invoked when the activity is no longer visible to the user. The reason for this state is either activity is getting destroyed or another existing activity comes back to **resume** state.

7. **onDestroy()**

The Android onDestroy() is the method that is called when an activity finishes and the user stops using it. It is the final callback method received by activity, as after this it is destroyed.



What are Android Services?

Android Services are the application components that run in the background. We can understand it as a process that doesn't need any direct user interaction. As they perform long-running processes without user intervention, they have no User Interface. They can be connected to other components and do **inter-process communication (IPC)**.

1. **Foreground Services**

Foreground services are those services that are visible to the users. The users can interact with them at ease and track what's happening. These services continue to run even when users are using other applications.

The perfect example of this is Music Player and Downloading.

2. **Background Services**

These services run in the background, such that the user can't see or access them. These are the tasks that don't need the user to know them.

Syncing and Storing data can be the best example.

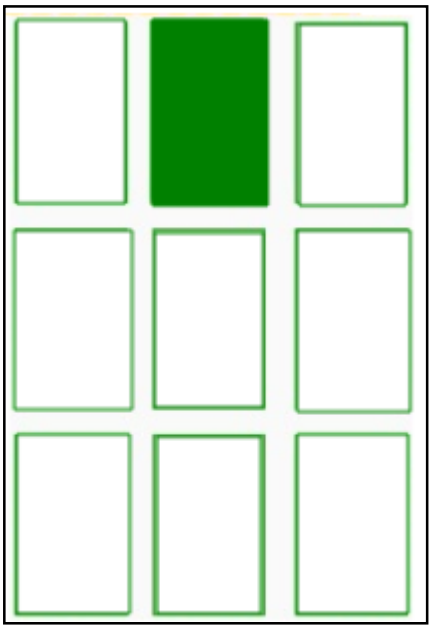
3**. Bound Services**

Bound service runs as long as some other ***application component*** is bound to it. Many components can bind to one service at a time, but once they all unbind, the service will destroy.
To bind an application component to the service, **bindService()** is used.
What is Android Broadcast Receiver?

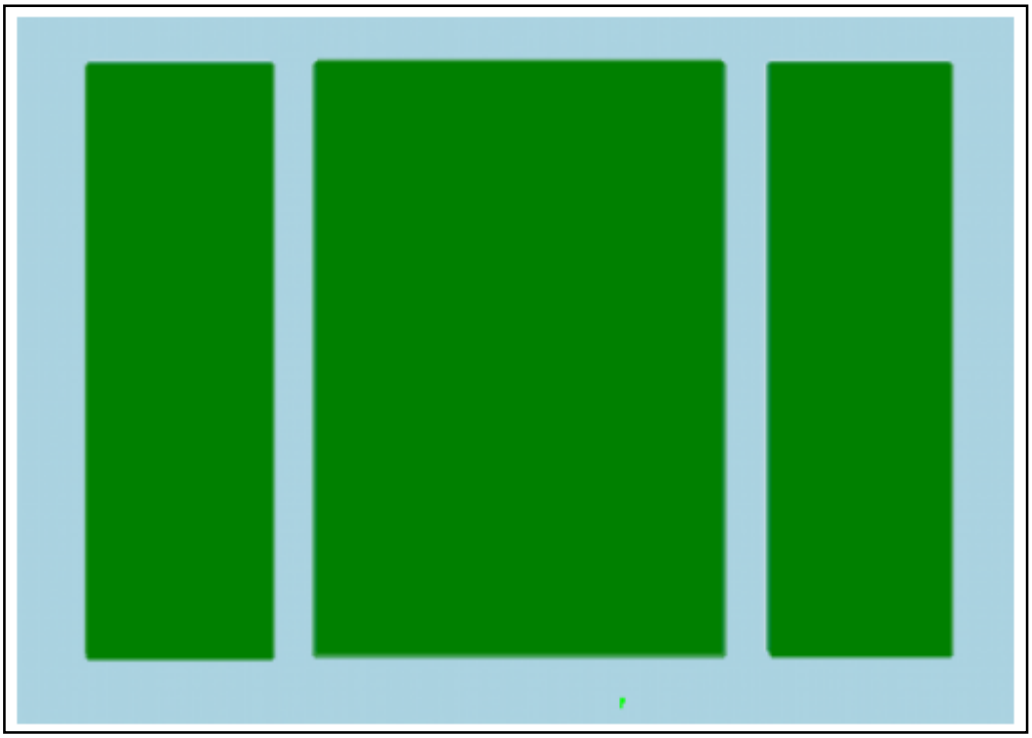
Android Broadcast Receiver is an Android component that is used to broadcast the messages to the system or other applications. The broadcast message is referred to as an Event or Intent. Broadcast receivers, unlike Activities, have no user interface. It's working is similar to that of a publish-subscribe design pattern. It's used for Asynchronous Inter-Process communication.

Layouts in Android UI Design

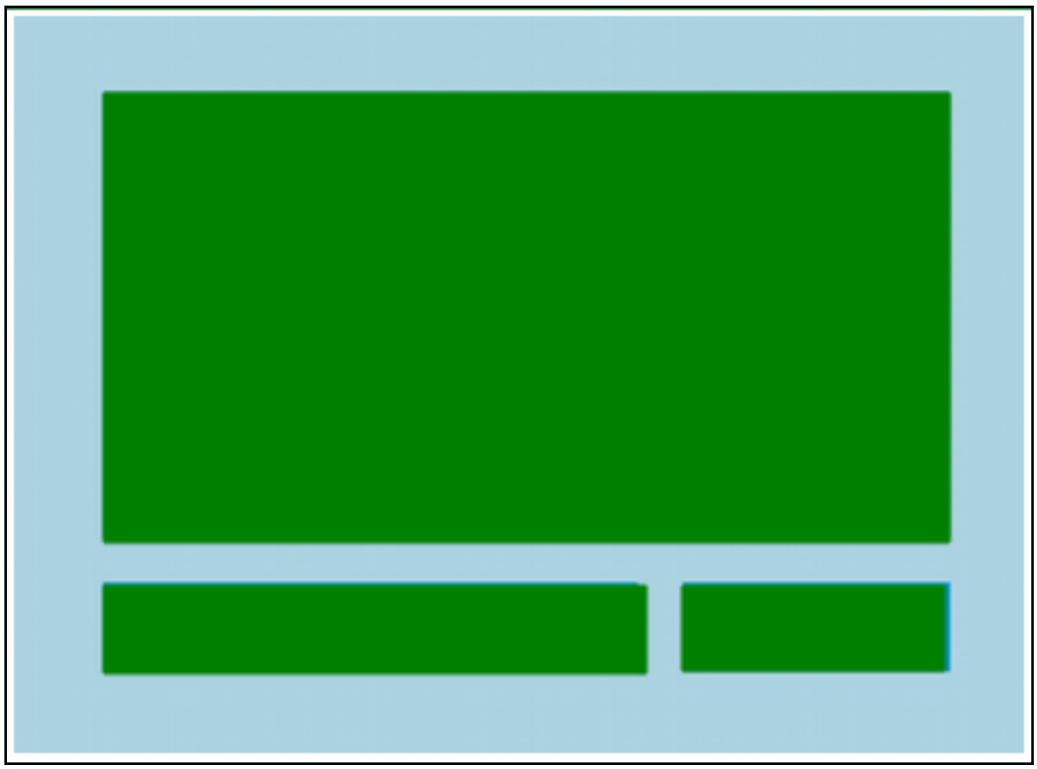
FrameLayout- It is the simplest of the Layout Managers that pins each child view within its frame. By default the position is the top-left corner, though the gravity attribute can be used to alter its locations. You can add multiple children stacks each new child on top of the one before, with each new View potentially obscuring the previous ones.



LinearLayout- A LinearLayout aligns each of the child View in either a vertical or a horizontal line. A vertical layout has a column of Views, whereas in a horizontal layout there is a row of Views. It supports a weight attribute for each child View that can control the relative size of each child View within the available space.



RelativeLayout- It is flexible than other native layouts as it lets us to define the position of each child View relative to the other views and the dimensions of the screen.



- **GridLayout-** It was introduced in Android 4.0 (API level 14), the Grid Layout used a rectangular grid of infinitely thin lines to lay out Views in a series of rows and columns. The Grid Layout is incredibly flexible and can be used to greatly simplify layouts and reduce or eliminate the complex nesting often required to construct UIs using the layouts described before.

Android UI Controls

1. TextView

TextView is a UI Component that displays the text to the user on their Display Screen.

2. EditText

EditText is a user interface control that allows the users to enter some text.

3. Button

This is a UI that is used to perform some action as soon as the user clicks on it.

4. ImageButton

It is the same as a Button but it's used to display an image on the button to perform an Action. In this, we need to give the source of the image so that the system can load it.

5. ToggleButton

The toggle button displays the ON/OFF states of a button with a light indicator.

6. RadioButton

Radio button in Android is the one that has only two possible states, that are either checked or unchecked. Initially, it is in the unchecked state, once it's checked it can't be unchecked.

7. RadioGroup

It's a group of Radio buttons that are alike. In this, only one of all the buttons can be chosen.

8. CheckBox

A CheckBox is the UI control that has two states that are either checked or unchecked. If we have a group of CheckBox, we can select as many as we want, unlike RadioGroup.

9. ProgressBar

In Android, we have a progress bar that shows the progress of some action that is happening like pasting a file to some location. A progress bar can be in two modes:

1. Spinner

The Spinner in Android is a User Interface that is used to select a particular option from a list of given options. Spinner in Android is the same as dropdown in HTML. It provides us with a faster way to select an option of our choice. When we click on the down arrow, it shows a list of values from which we can select one. By default, the first value would be shown as the currently selected Value.

2. TimePicker

Time picker is a UI component that works as an intermediate to select a time of the day. The time chosen by it is shown either in 24 hrs format or in 12hrs format using AM and PM.

It gives a virtual Clock/watch to select it. This virtual clock makes it easy to choose the time.

3. DatePicker

Like we have time picker, we have a date picker as UI control too. In this, the System shows a virtual calendar to the users to choose the day.

This enables the user to choose a particular date using either a calendar or a dropdown. These both are made to make it easier for the user to pick up a date and a time.

4. SeekBar

In Android, Seekbar is an extended Progress bar. A seekbar comes with a pointer that is draggable throughout the bar either in left or right. This pointer helps to set the progress as well. This helps the user to choose a particular range of values.

5. RatingBar

A rating bar in Android is an extended version of a seekbar. It is used to give the rating by touching it. In the rating bar, a user can rate at a scale of 5 with a difference of 0.5.

Its rating is in Stars. The user needs to tap/click the stars.

6. AlertDialog

Alert Dialog Box is a UI that gives the users an Alert or Warning of something. It appears on the screen in a small window. Once it comes, the user needs to decide or choose an option that it shows.

For example, when you enter the wrong password for email id.

7. Switch

In Android, a switch is a two-state UI element that holds either ON or OFF state. ON generally means Yes and OFF means No. By default, a switch is in the OFF state. A user can change its state many times.

8. AutoCompleteTextView

AutoCompleteTextView is an extension of EditText. In this UI element, the user is provided with a few suggestions of some values/texts. The value can be chosen by the user while filling AutoCompleteTextView.

Data Science

Anshu Bharadwaj

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
anshu.bharadwaj@icar.gov.in

Introduction

Data is everywhere and is one of the most important assets of every organization that helps them to flourish by making decisions based on facts, statistical numbers, and trends. Data is on overdrive. It's being generated at break-neck pace, flooding out of the dozens of connected devices we use every day, and it shows no signs of slowing down. In fact, the number of internet users has grown over a billion in the last five years, more than half of the world's web traffic now comes from mobile phones. Nowadays, data is considered the world's most valuable resource. Data is valuable, and so is the science in decoding it. Zillions of bytes of data are being generated, and now its value has surpassed oil as well. Data needs to be read and analyzed. This calls out for the requirement of having a quality of data and understanding how to read it and make data-driven discoveries. Due to this growing scope of data, data science came into picture which is a multidisciplinary field that helping us to discover useful information from the data, answer questions, and even predict the future or the unknown. It uses scientific approaches, procedures, algorithms, the framework to extract the knowledge and insight from a huge amount of data.

What is Data Science?

The accelerating volume of data sources, and subsequently data, has made data science is one of the fastest growing field across every industry. Organizations are increasingly reliant on them to interpret data and provide actionable recommendations to improve business outcomes.

Data Science is an interdisciplinary field that focuses on extracting knowledge from data sets which are typically huge in amount. The field encompasses analysis, preparing data for analysis, and presenting findings to inform high-level decisions in an organization. As such, it incorporates skills from computer science, mathematics, statistics, information visualization, graphic, and business.
Data science combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization's data. These insights can be used to guide decision making and strategic planning.

Why is data science important?

The reason why we need data science is the ability to process and interpret data. This enables companies to make informed decisions around growth, optimization, and performance. Data Science is important because it combines tools, methods, and technology to generate meaning from data. Modern organizations are inundated with data; there is a proliferation of devices that can automatically collect and store information. Online systems and payment portals capture more data in the fields of e-

commerce, medicine, finance, and every other aspect of human life. We have text, audio, video, and image data available in vast quantities.

History of data science

While the term data science is not new, the meanings and connotations have changed over time. The word first appeared in the '60s as an alternative name for statistics. In the late '90s, computer science professionals formalized the term. A proposed definition for data science saw it as a separate field with three aspects: data design, collection, and analysis. It still took another decade for the term to be used outside of academia.

The Data Science Life Cycle

The data science lifecycle involves various roles, tools, and processes, which enables analysts to glean actionable insights. The image below represents the five stages of the data science life cycle:

Capture (data acquisition, data entry, signal reception, data extraction). The lifecycle begins with the data collection--both raw structured and unstructured data from all relevant sources using a variety of methods. These methods can include manual entry, web scraping, and real-time streaming data from systems and devices. Data sources can include structured data, such as customer data, along with unstructured data like log files, video, audio, pictures, the Internet of Things (IoT), social media, and more.

Maintain (data warehousing, data cleansing, data staging, data processing, data architecture): Since data can have different formats and structures, companies need to consider different storage systems based on the type of data that needs to be captured. Data management teams help to set standards around data storage and structure, which facilitate workflows around analytics, machine learning and deep learning models. This stage includes cleaning data, deduplicating, transforming and combining the data using ETL (extract, transform, load) jobs or other data integration technologies. This data preparation is essential for promoting data quality before loading into a data warehouse, data lake, or other repository.

Process (data mining, clustering/classification, data modeling, data summarisation)

Analysis (exploratory/confirmatory, predictive analysis, regression, text mining, qualitative analysis) : Here, data scientists conduct an exploratory data analysis to examine biases, patterns, ranges, and distributions of values within the data. This data analytics exploration drives hypothesis generation for a/b testing. It also allows analysts to determine the data's relevance for use within modeling efforts for predictive analytics, machine learning, and/or deep learning. Depending on a model's accuracy, organizations can become reliant on these insights for business decision making, allowing them to drive more scalability.

Communicate (data reporting, data visualisation, business intelligence, decision making): Finally, insights are presented as reports and other data visualizations that make the

insights—and their impact on business—easier for business analysts and other decision-makers to understand. A data science programming language such as R or Python includes components for generating visualizations; alternately, data scientists can use dedicated visualization tools.

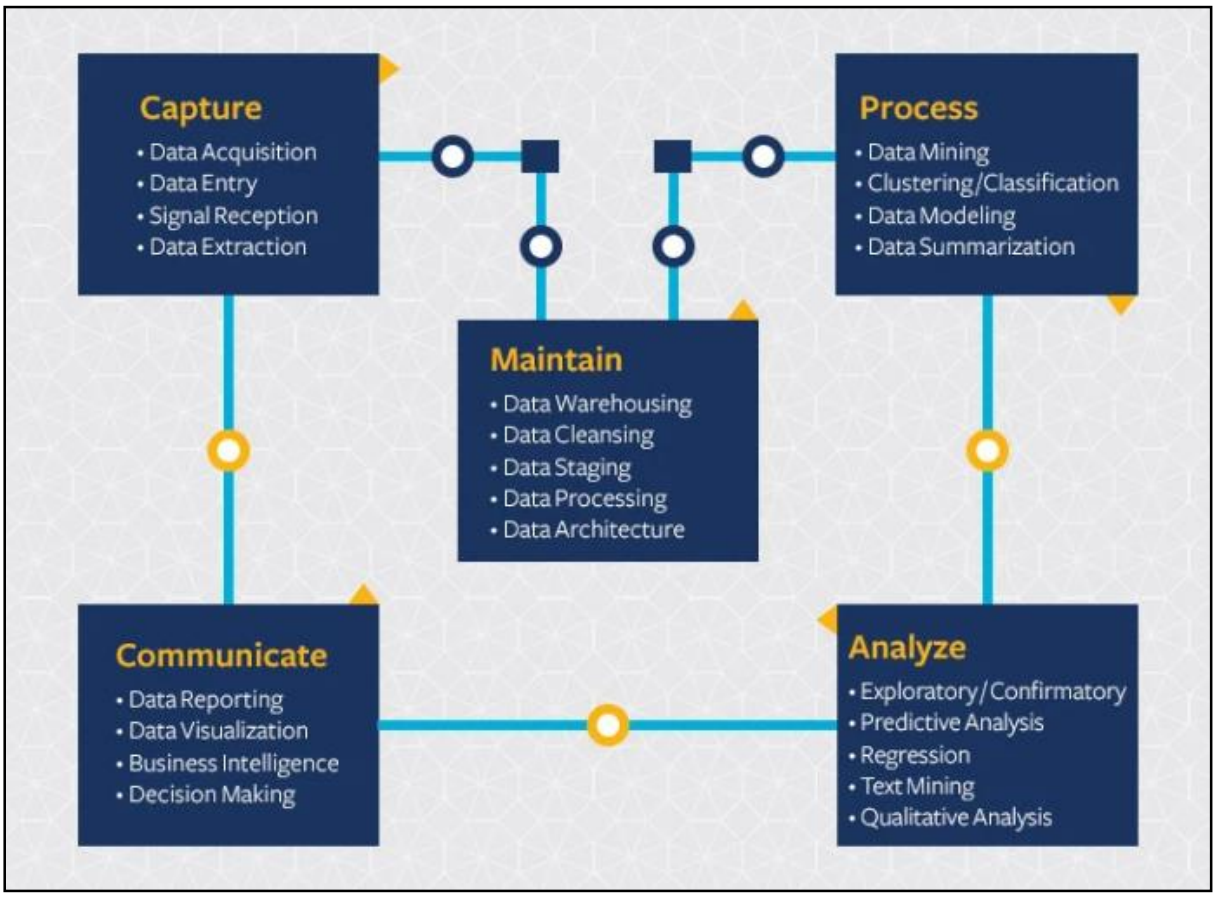


Figure 1: Data science life cycle

What is data science used for?

Data science is used to study data in four main ways:

Descriptive analysis

Descriptive analysis examines data to gain insights into what happened or what is happening in the data environment. It is characterized by data visualizations such as pie charts, bar charts, line graphs, tables, or generated narratives. For example, a flight booking service may record data like the number of tickets booked each day. Descriptive analysis will reveal booking spikes, booking slumps, and high-performing months for this service.

Diagnostic analysis

Diagnostic analysis is a deep-dive or detailed data examination to understand why something happened. It is characterized by techniques such as drill-down, data discovery, data mining, and correlations. Multiple data operations and transformations may be performed on a given data set to discover unique patterns in each of these techniques.For example, the flight service might drill down on a particularly

high-performing month to better understand the booking spike. This may lead to the discovery that many customers visit a particular city to attend a monthly sporting event.

Predictive analysis

Predictive analysis uses historical data to make accurate forecasts about data patterns that may occur in the future. It is characterized by techniques such as machine learning, forecasting, pattern matching, and predictive modeling. In each of these techniques, computers are trained to reverse engineer causality connections in the data.For example, the flight service team might use data science to predict flight booking patterns for the coming year at the start of each year. The computer program or algorithm may look at past data and predict booking spikes for certain destinations in May. Having anticipated their customer's future travel requirements, the company could start targeted advertising for those cities from February.

Prescriptive analysis

Prescriptive analytics takes predictive data to the next level. It not only predicts what is likely to happen but also suggests an optimum response to that outcome. It can analyze the potential implications of different choices and recommend the best course of action. It uses graph analysis, simulation, complex event processing, neural networks, and recommendation engines from machine learning.

Back to the flight booking example, prescriptive analysis could look at historical marketing campaigns to maximize the advantage of the upcoming booking spike. A data scientist could project booking outcomes for different levels of marketing spend on various marketing channels. These data forecasts would give the flight booking company greater confidence in their marketing decisions.

What are the data science techniques?

The top techniques used by data scientist are:

Classification: Classification is the sorting of data into specific groups or categories. Computers are trained to identify and sort data. Known data sets are used to build decision algorithms in a computer that quickly processes and categorises the data. For example:

- Sort products as popular or not popular·
- Sort insurance applications as high risk or low risk·
- Sort social media comments into positive, negative, or neutral.

Regression: Regression is the method of finding a relationship between two seemingly unrelated data points. The connection is usually modeled around a mathematical formula and represented as a graph or curves. When the value of one data point is known, regression is used to predict the other data point. For example:

- The rate of spread of air-borne diseases.
- The relationship between customer satisfaction and the number of employees.·

- The relationship between the number of fire stations and the number of injuries due to fire in a particular location.

Clustering: Clustering is the method of grouping closely related data together to look for patterns and anomalies. Clustering is different from sorting because the data cannot be accurately classified into fixed categories. Hence the data is grouped into most likely relationships. New patterns and relationships can be discovered by clustering. For example:

- Group customers with similar purchase behavior for improved customer service.·
- Group network traffic to identify daily usage patterns and identify a network attack faster.
- Cluster articles into multiple different news categories and use this information to find fake news content.

What are different data science technologies?

Data scientists use the complex technologies such as:

1. Artificial intelligence: Machine learning models and related software are used for predictive and prescriptive analysis.
2. Cloud computing: Cloud technologies have given data scientists the flexibility and processing power required for advanced data analytics.
3. Internet of things: IoT refers to various devices that can automatically connect to the internet. These devices collect data for data science initiatives. They generate massive data which can be used for data mining and data extraction.
4. Quantum computing: Quantum computers can perform complex calculations at high speed. Skilled data scientists use them for building complex quantitative algorithms.

Data science tools

Data scientists rely on popular programming languages to conduct exploratory data analysis and statistical regression. These open source tools support pre-built statistical modeling, machine learning, and graphics capabilities. These languages include the following:

- **R Studio:** An open source programming language and environment for developing statistical computing and graphics.
- **Python:** It is a dynamic and flexible programming language. The Python includes numerous libraries, such as NumPy, Pandas, Matplotlib, for analyzing data quickly.

To facilitate sharing code and other information, data scientists may use GitHub and Jupyter notebooks.

Some data scientists may prefer a user interface, and two common enterprise tools for statistical analysis include:

- **SAS:** A comprehensive tool suite, including visualizations and interactive dashboards, for analyzing, reporting, data mining, and predictive modeling.
- **IBM SPSS:** Offers advanced statistical analysis, a large library of machine learning algorithms, text analysis, open source extensibility, integration with big data, and seamless deployment into applications.

Conclusion

Data science is a process that empowers better business decision-making through interpreting, modelling, and deployment. This helps in visualizing data that is understandable for business stakeholders to build future roadmaps and trajectories. Implementing Data Science for businesses is now a mandate for any business looking to grow. Data science continues to evolve as one of the most promising and in-demand career paths for skilled professionals. Today, successful data professionals understand that they must advance past the traditional skills of analysing large amounts of data, data mining, and programming skills. In order to uncover useful intelligence for their organizations, data scientists must master the full spectrum of the data science life cycle and possess a level of flexibility and understanding to maximize returns at each phase of the process.

Training Management Information System for ICAR (TMIS)

Shashi Dahiya

ICAR-Indian Agricultural Statistics Research Institute, New Delhi - 110 012
shashi.dahiya@icar.gov.in

1. INTRODUCTION

Competencies encompass knowledge, skills and behaviour, which are required in an individual for effectively performing the functions of a post (ICAR, 2018). Systematic approach to develop and continuously improve the individual competencies and capabilities is necessary for achieving the organizational objectives and goals effectively. Department of personnel and training (DoPT), Govt. of India [2] has made it mandatory for all Govt. departments to develop their Training Management Information System and to ensure that each employee should be provided with minimum number of trainings in the service period.
The Indian Council of Agricultural Research (ICAR), is an autonomous organization under the Department of Agricultural Research and Education (DARE), Ministry of Agriculture and Farmers Welfare, Government of India. ICAR is attaching tremendous importance to the management and development of their employees. Presently the information on competency enhancement of employees of ICAR during their service through their experiences, in-house trainings, trainings at different national and international institutes is not available at one place. Such information, if available centrally, will be useful to understand and analyze the strength and weakness of different categories of employees and plan suitable strategies for enhancing the competency of employees of ICAR. A well-planned Management Information System (MIS) can fill this gap and will meet the mandatory condition of DoPT, Govt. of India.

The Training Management Information System for ICAR (TMIS), has been designed and developed for the management of information related to trainings sponsored by the Human Resource Management Unit of ICAR.

2. Training Hierarchy Identification

The following user hierarchy (Fig.1) has been identified at the applicant and various management levels for smooth approval and flow of information. The Training Manager reviews the Annual Training Plan (ATP) for all Institutes and the ICAR Headquarters (HQ). He is also involved in the overall monitoring and evaluation of training related functions of all Institutes/HQs. The Middle Level manager is responsible for discussion with the employee and forwarding/approving the employee applications to the senior manager or the operational level manager. The operational level manager being the HRD, Nodal officer of the institute, is responsible for all training related operations and communications with the employee, middle

level manager and the senior level manager of the Institute/Unit. The senior level manager finally approves or rejects the applications and forwards them to the top-level manager.

Figure 1: TMIS User Hierarchy and Interaction

2.1 Training Process Implementation

The whole training process is divided into four sequential processes:

1) **Training Needs Assessment (Annual Training Plan)** Under the Training Needs Assessment process, the broad/ deficient training areas and the organizations conducting trainings under these areas are identified for each cadre based on the ICAR-HRM policy for training and capacity building. The cadres are divided into 3 main categories of Scientific, Technical, Administrative and Supporting staff. The employee submits its training needs under the broad areas and the HRD Nodal Officer of the institute identifies the trainings in consultation with the employee and his middle level manager. Based on the identified training needs for all the employee, the Annual Training Plan (ATP) for the institute is prepared by the HRD nodal officer which contains the list of employee, the training names, organizing institutes, duration and the financial implications during the year.

2) **Training Application Process (Application Submission and Approval)** The Training Application process involves the online training application submission process. Each employee of ICAR has a provision to apply for a training programme defined under the Annual Training Plan of the institute. The training applications approval mechanism will be done online by the middle level manager

(Reporting Officer) and the senior level manager (Reviewing Officer) involving an interaction among themselves and with the operational level manager (HRD Nodal Officer). Operational level manager also has the authority to approve or reject the application.

3) **Training Feedback Process (Feedback)** The Feedback process starts after the employee attends the approved training programme and submits his feedback about the training. The middle and senior level managers gives their comments and ratings on the feedback before it is finally submitted to the Training Manager.

4) **Training Evaluation Process (Performance Indicators)** Under the training evaluation process, the employee evaluates the training attended by him on certain performance indicators related to its future use and technical skills involved in the training. The employee has to provide a rating of 1 to 5 to each indicator. The managers will give their remarks on the Training Evaluation Report.

2.2. Software Design and Implementation

The Software Requirements Specification (SRS) for the proposed system have been developed from the information gathered from the HRM unit, ICAR. Based on the SRS, The software architecture used for the development of TMIS as a web application, is a modified three-tier client server architecture (Fig.2.). The goal of using 3-tier architecture is to separate the user application from the physical database (Pressman, 2009). It has a user interface client, which runs on user's machine that communicates with business logic software on a server machine, which in turn communicates with the software responsible for the long-term storage of data on database server.

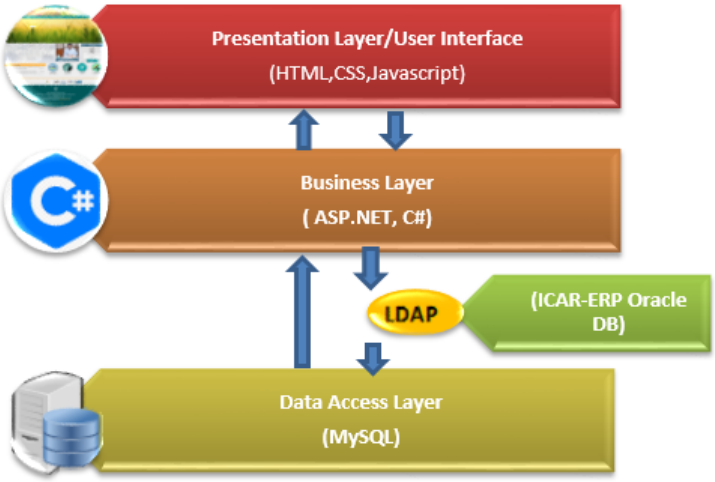


Figure 2: Modified Three Tier Client-Server Architecture

A standard three-tier client-server architecture involves three layers – The Presentation Layer (User Interface), The Business Layer (Application Logic) and The Data Layer (Data Access). An intermediate Lightweight Directory Access Protocol (LDAP) link with the ICAR-ERP [5], database has been established

between the Business Layer and the Data Access layer. LDAP is a software protocol for enabling to locate the data about organizations, individuals and other resources such as files and devices in a network. A common use of LDAP in TMIS is to provide a central place to store usernames, passwords, name, designation, date of birth, gender, institute name and other training details. This allows the application to connect to the LDAP server to validate the users. This link allows the access of employee specific information from the ICAR-ERP into the TMIS.

The Presentation layer or the front-end layer provides the graphical user interface (GUI) to the web application. This layer is used for designing the user interface and to get or set the data back and forth using Hyper Text Markup Language (HTML) and JavaScript. The layer has been designed using the HTML5, JavaScript, Cascading Style Sheet (CSS) and communicates with other layers through the API calls. In this layer, the validation controls of Microsoft .NET framework have been used to validate user input entered into the control fields and display associated error messages for each control.

The Business layer contains the functional application logic, which drives an application's core capabilities. The server side development of TMIS has been done using the C# programming language in Visual Studio 2013, which provides a framework to create the dynamic content on the server. It facilitates the data retrieval, data entry/updating by communicating with the database layer and presenting the results by communicating with the presentation layer.

The data access layer comprises of the database/data storage system. A relational database has been designed which facilitates the fast and convenient storage of data entered through online data management forms and generation of online Performance indicators, Impact Assessment and report generation at various user levels. MySQL has been used for its implementation.

3. RESULTS AND DISCUSSION

The TMIS supports the management in short term planning, target setting and controlling the training related activities in the institution. It is supported by the use of the management tools in the form of a variety of online monitoring reports for managers. These monitoring reports supports in decision making at various levels of management, planning and control. The TMIS, therefore, plays a vital role in the operation, processing and management of trainings in ICAR.

3.1. Training Data Management

The Annual Training Plan of the whole ICAR is made automatically by consolidating the ATP's of all institutes. There is no manual intervention required for file submission, movement and approval. The whole process from application submission to the approval has become online. Automatic Email alerts are sent to the registered email id's of applicants to get them updated about the status of their application.

3.2. HRM Reports

The management reports are required to effectively monitor the implementation of the training policy. The reports have been developed at various levels (Institute Wise, Subject Matter Division Wise and for the whole ICAR), frequency (Monthly, Annually, Dynamically for any time period) and authentication levels (Middle level manager, senior level manager, operational manager and the Top-level manager). The Home Page of TMIS and the management report giving the Application Approval Status of employees are depicted in Fig. 3, Fig. 4 respectively.

Figure 3: TMIS Home Page

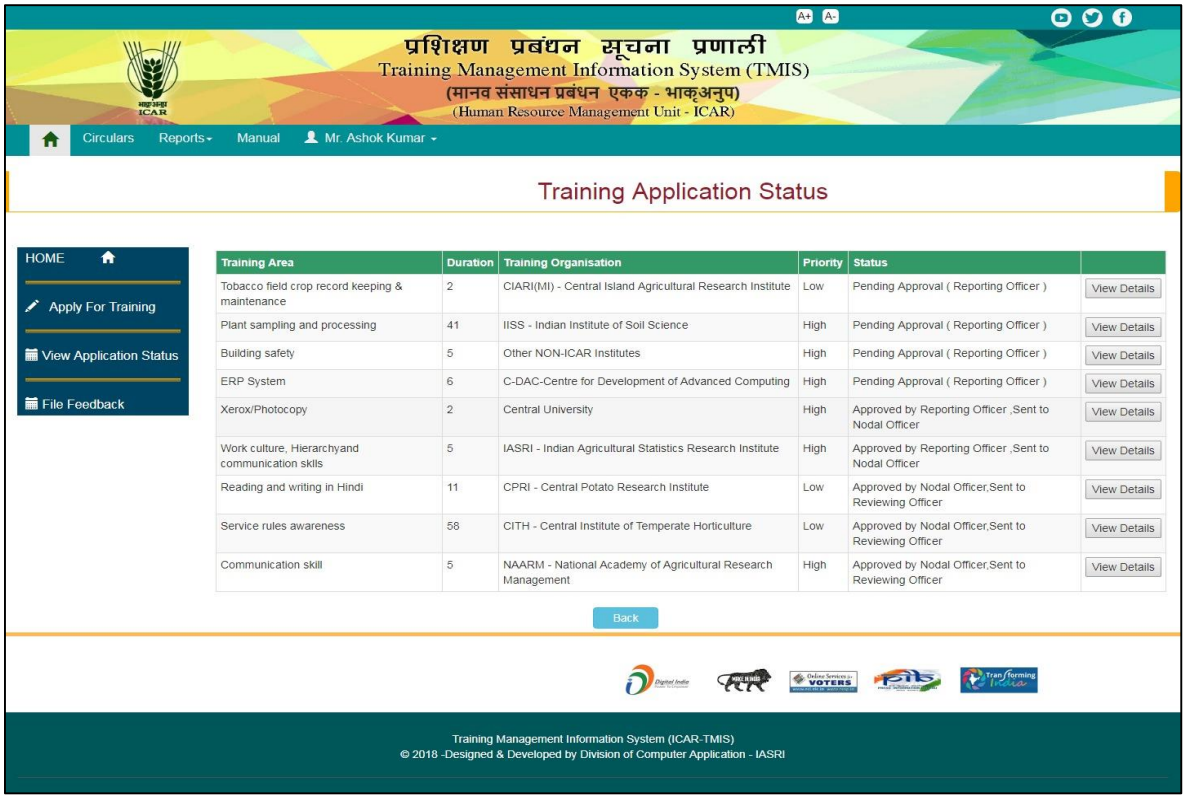


Figure 4: Training Application Approval Status Report in TMIS

3.3. Conclusion

The TMIS for ICAR has been successfully implemented as a web application in all ICAR institutes at the web address: **https://hrm.icar.gov.in.** TMIS is operational across all ICAR institutes. Its centralized database has complete data of trainees, training needs, feedbacks along with the performance indicators for each individual training. The tasks performed by different levels of managers for accomplishing training related approvals have now become online. Timely and proper feedbacks and assessment of trainings undertaken by trainee and there supervisors using online forms helps in creating a high impact and better results of trainings done. Central login through ICAR email-id enhances users experience as this enables them to login using their existing ICAR emails and not creating or signing up with any other email ids on the application. The complete database of training along with the feedback and performance indicators for each individual training supports the HR managers in taking decisions for individual employee future trainings.