

CHAPTER

3

Session

1

Introduction to Data Visualization



Goal

In this Session, you will learn

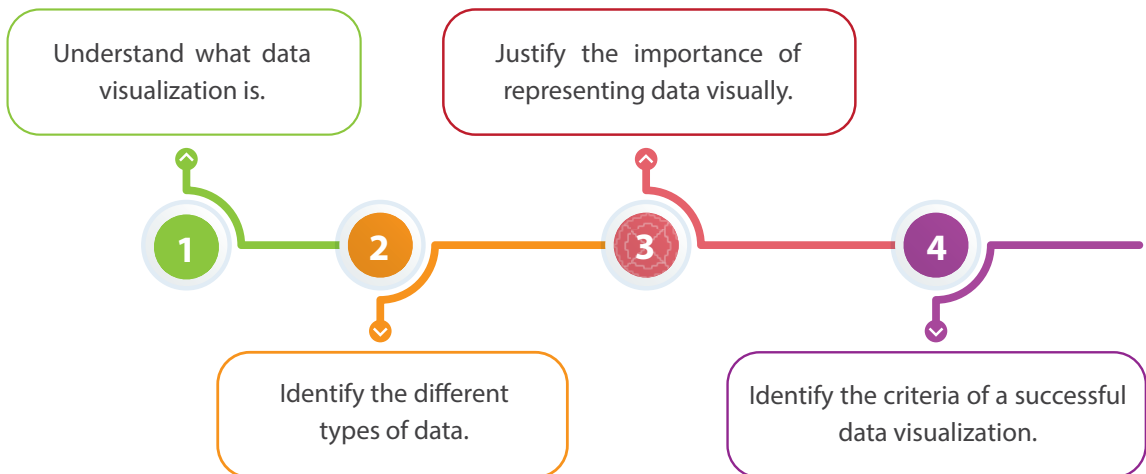
why representing data visually makes a difference and what makes visual data successful.



Learning objectives

Dear Student

By the end of this Session, you should be able to:



Be prepared

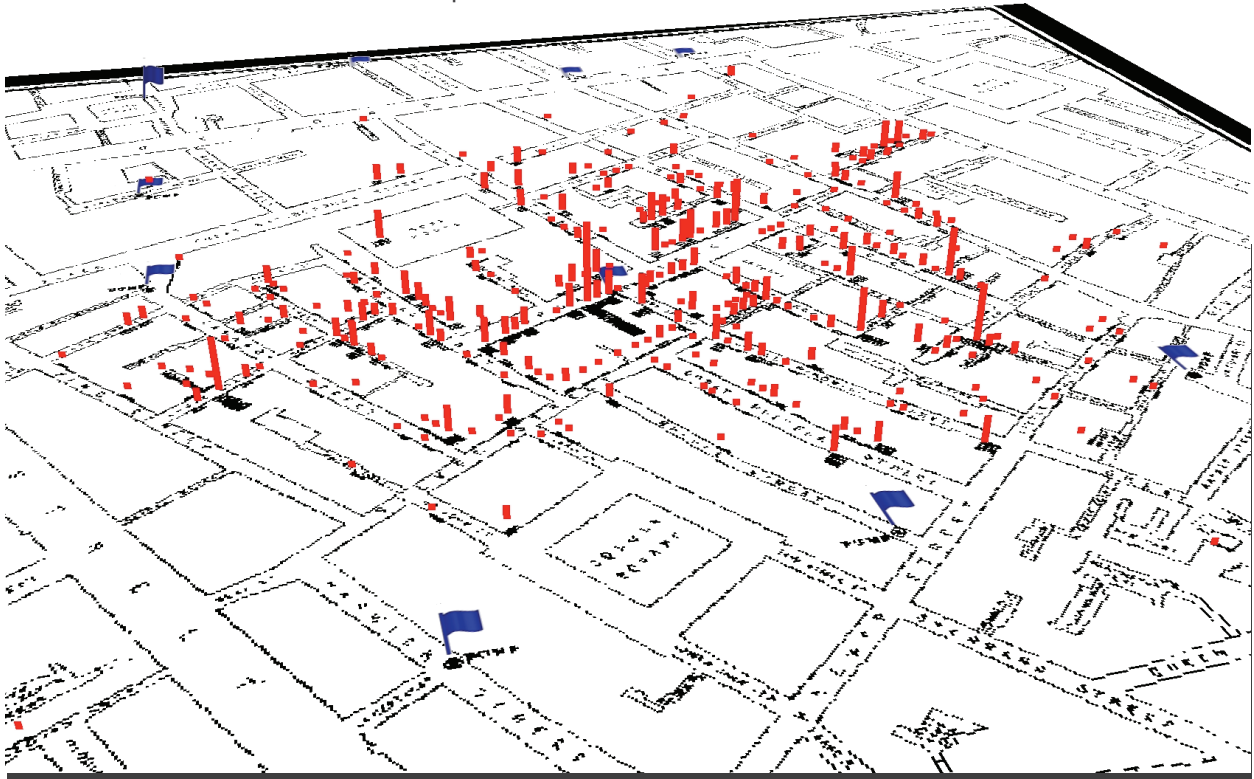
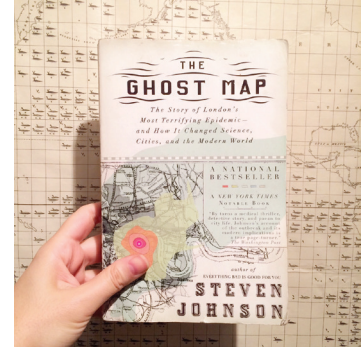
In this session, you will learn what data visualization is, why data visualization is important, and how data visualization can change people's way of thinking.

List of Terms

Data visualization	A method used to display data in graphic format
Numerical data	Data in the form of numbers
Non-numerical data	Data we cannot measure in numbers
Exploratory data visualization	Using the data visualization to find the story behind it
Explanatory data visualization	Using the data visualization to communicate the story

The Ghost Map

In 1854, London was a city of death. A cholera outbreak turned into an epidemic that would eventually claim the lives of 10,000 people. The common belief was that cholera was spread through the air, giving the disease an ethereal quality that earned it the nickname “The Ghost.” However, a physician named John Snow had a different idea. Snow’s hypothesis was that cholera wasn’t spread through air at all; he believed it was spread through water. He published a paper explaining his theory, but the medical community didn’t pay any attention to it — no experiment had been conducted to justify his hypothesis. So, Snow started collecting more information about the spread of cholera in London. He made a breakthrough when he found out where every victim of the epidemic had lived, then marked each of those locations on a map. The result was what came to be known as the “The Ghost Map.”



The map showed the distribution of the affected buildings – buildings where one or more people had died of cholera. When the data were laid out on the map, it was relatively easy to trace the outbreak to a specific point: a water pump on Broad Street.

Snow publicized his Ghost Map, and this time, people listened. The Broad Street pump was shut down, and the number of cases of cholera decreased until the outbreak eventually ended.

John Snow’s decision to visualize the data he’d collected on cholera deaths became a significant step toward solving the mystery of London’s terrifying Ghost.

I hope by now you are getting a sense of why visualizing data is such an important part of solving a problem or communicating a message.



Learn

Introduction to Data

Data is defined as a collection of raw facts, figures, or statistics, which are collected and processed to deduce information, which in turn is used to make decisions.



On a larger scale, humans are currently generating an estimated

2.5 quintillion bytes
of data every single day.

Types of Data for Data Visualization

There are two main types

Numerical Data

Data in the form of numbers is called numerical data. This includes data in quantitative form or in measurable form like: time, weight, height, amount, etc.

Non-numerical Data

Data we can observe is called non-numerical data. This includes data in a qualitative form or the data that we cannot measure in numbers like: a person's gender, address, ethnicity, etc.

Data Visualization



Data visualization is a method used to display data in a graphical format. It can show patterns, trends, or presentations of data in a pictorial format. This is an effective way to help your audience understand concepts or figures in an easier and more attractive way. It enables a presenter to convey a concept or an idea in an understandable way without making much effort.

You see data visualization around you almost every day. Common examples of visuals are a Google map, Riyadh's train map, the scale-style speedometer, and many of the signs and visual instructions you pass on the street.



The human brain processes **images**

60,000

times faster than **text**.

History of Data Visualization

The concept of using maps, figures, and pictures to understand or to convey a message has been around for centuries. Examples include mapmaking, drawings and pictures. It has deep roots, having been used by cartographers and surveyors in the second century. Nowadays we have all types of data and methods of visualizing data.

Why did the concept of data visualization gain so much popularity? The answer is that our brain loves visuals. The visuals communicate more information than tables and text. Our brain can more easily understand the data presented in graphics than in text.



Significance of Data Visualization



“A picture is worth a thousand words.”

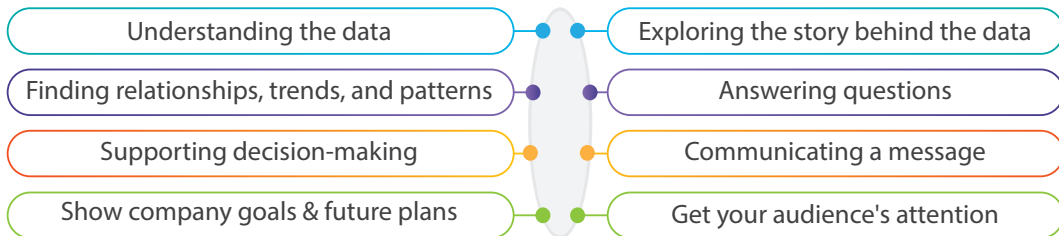
Data visualization helps you to explain the relationships among the data in an easier way. We live in a fast-paced world in which our leaders, our managers, and we, ourselves, are often faced with the need to make quick and accurate decisions. With the flood of information available to us, the process of reviewing all the relevant data seems impossible. However, reviewing charts, figures, and graphs that show the patterns, trends, and relationships inherent in that data decreases the time and effort it takes to make sound decisions.



The more visual the input becomes, the more likely it is to be recognized and recalled.

Neurologist John Medina

By this point, it should be clear to you that the significance of data visualization lies in:



It is worth mentioning that the focus of this chapter will be on explanatory visualization. Hence, in the following sessions, you will learn how to visualize your data using some tools and software to communicate a story and answer questions for your audience effectively and in a visually appealing manner.

Types of Data Visualizations

There are two main types of data visualizations

Exploratory data visualization

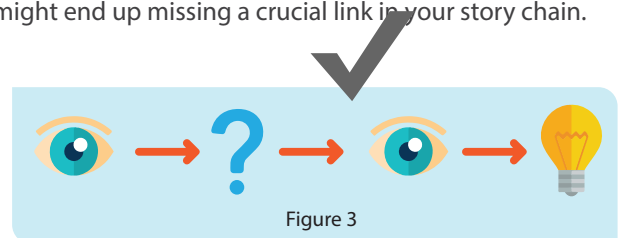
Explanatory data visualization

Exploratory data visualization

Exploratory data visualization is used when you want to look deeply into your collection of data to gain insight about its meaning and to understand how the different variables connect. This type of data visualization is mainly used as a part of a larger data analysis to help the researcher figure out the story behind the data. Exploratory data visualizations tend to be complex and contain lots of information; this is to be expected, since the main goal of this form of representation is to find the story that pulls together as much of the data as possible. In other words, if you start removing everything that looks like data noise, you might end up missing a crucial link in your story chain.

The Process of Exploratory Visualization.

Figure 3 shows the process of finding the story in an exploratory data visualization. The process starts by carefully examining the visual and formulating questions about what you see. Then you examine the visual again to explore answers to those questions or to find relationships or trends.

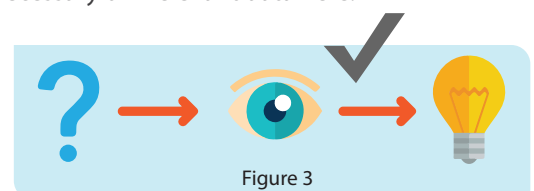


Explanatory data visualization

Unlike exploratory data visualization, explanatory data visualization is used when you already know the story behind your data and you want to communicate that story to an audience. You choose which type of visual representation to use based on the story that you want to share, and you include in that visual representation only the data you want to focus on; you don't want to include any unnecessary or irrelevant data here.

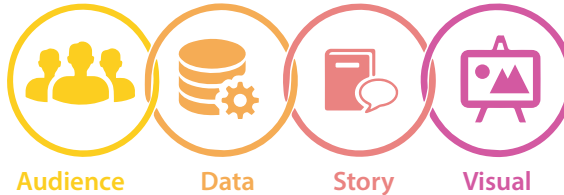
The Process of Explanatory Visualization

Figure 4 explains the process of explanatory data visualization. The questions in the first step are usually asked by the audience, then the visual designer tries to communicate answers to those questions by designing an attractive visual with minimal data noise.



Successful data visualization

The success of data visualization is based on a combination of four elements: audience, data, story, and visual. The next sections explain how these elements should interact with one another to provide a successful data visualization.



Clarity and simplicity

“Maximize impact and minimize noise.” —Chris Dutton

The main goal of explanatory visualization is to communicate a message to your audience. Therefore, it is important to make the message clear and simple to ensure quick and easy understanding. Visualizing data with clarity and simplicity means doing so in a way that gets directly to the point; tells the story the data show, not what you want to show; and avoids any irrelevant data.

Create a narrative

“Don’t show your data, tell a story.” —Chris Dutton

To tell a story effectively through visual aids, you should know your audience as well as the message you want them to get from your data visualization. Maybe your audience is a group of professionals who need to get a sense of how their business or organization is performing so they can make important decisions. In this situation, the data visualization might focus on presenting the changes to their market value over time or comparing earnings over two periods or the performance of two products. Or maybe you want to communicate a general message about your latest assignment to non-specialists – in that case, you might choose a visual that is very direct and contains fewer details. Thus, the storytelling changes according to who the audience is, what they already know about the topic, and the level of information you want your data visualization to convey.

Balance between design and function

“Beautiful is good, functional is better, both is ideal.” —Chris Dutton

You want your data visualization to attract the attention of your audience, so good design is a requirement. However, attractive design is not enough. You don’t just want people to look – remember, your goal is to convey a message. To do this in an effective way, you have to choose the best visual aid to represent your data. There are many types of data visualizations to choose from, and the more familiar you are with your data, your audience, and the message you want to communicate, the easier it will be for you to choose the best one.

The next sessions talk about how you can effectively visualize your data using the concepts we learned in this session. First, you will learn how you can present your numerical data using different types of charts. Then, you will learn about presenting non-numerical data. Finally, you will learn how to pull everything together to create your own dashboard that tells your story in an effective way.



The power of Data Visualization
can change
one’s perspective.

Businesses or Fields that need data visualization

What kinds of businesses or fields need data visualization?

Data visualization is becoming a vital part of every business and organization. Industries and businesses record their data progress and visualize it, to track the company's growth, development, profit and loss, etc. Using data visualization, companies and organizations analyze their progress, what their target is, and where they are. Based on the figures, companies make plans and strategies.

Because of computers, today it is easy to analyze big and complex data using data visualization techniques. It is used in the corporate world, the banking sector, marketing companies, governmental organizations, hospitals, and other industries.



In 1644, **Michael Florent van Langren**, a Flemish astronomer, is believed to have provided the first visual representation of statistical data.



Apply

In pairs, watch the following video, then answer the questions.

Activity
1

The Best Stats You've Ever Seen | Hans Rosling | TED Talks

- What makes the presentation so attractive in your opinion?
- What is the tool used in the presentation?
- What is the type of data in the presentation?



Fill in the blanks with the appropriate words provided below.



(audience, explanatory, numerical data, exploratory, non-numerical)

Activity
2

- _____ data visualizations tend to be complex and contain lots of information.
- _____ data visualization is used when you already know the story behind your data.
- The success of data visualization is based on a combination of four elements: _____, data, story, and visual.
- Data in quantitative form or in measurable form is called _____.
- _____ data is data in qualitative form, data we cannot measure in numbers.

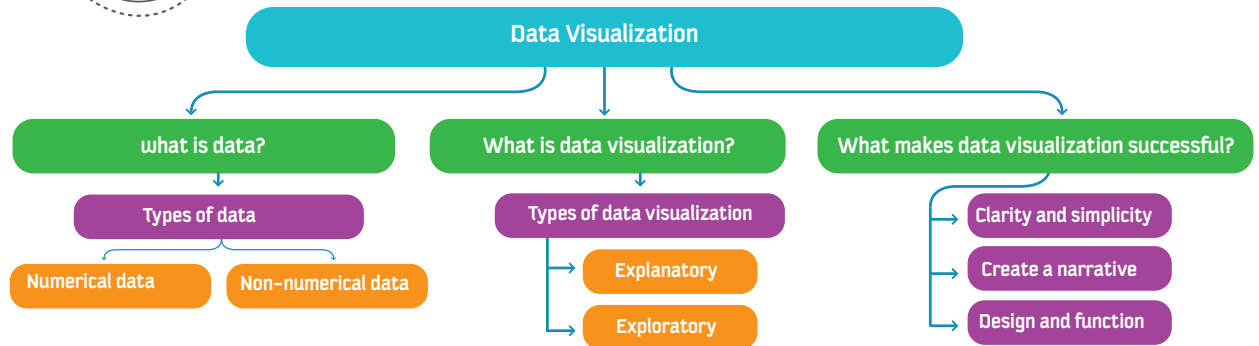
Choose TRUE or FALSE for the following statements.

Activity
3

- To tell a story effectively through visual aids, you should know your audience well. True
 False
- Data we can observe or data in qualitative form is numerical data. True
 False
- Time, weight, height, and amount are examples of numerical data. True
 False
- Data visualization is a method used to display data in graphical format. True
 False



Closing



CHAPTER

3

Session

2

Visualization of Numerical Data



Goal

In this Session, you will learn

how to present numerical data using different types of charts.



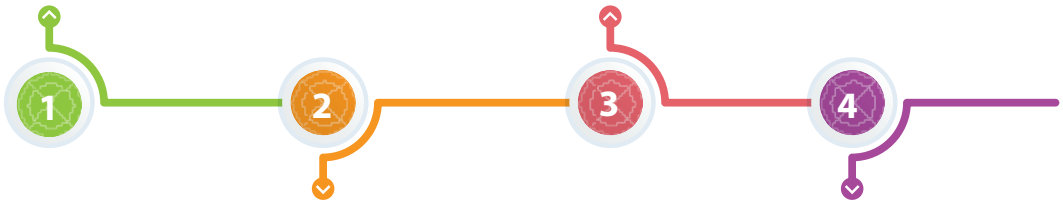
Learning objectives

Dear Student

By the end of this Session, you should be able to:

1 Assign numeric data to the appropriate chart.

3 Prepare numeric data to be presented in the appropriate chart or graph.



2 Understand the purpose of different types of charts.

4 Create band and waterfall charts using Microsoft Excel software.



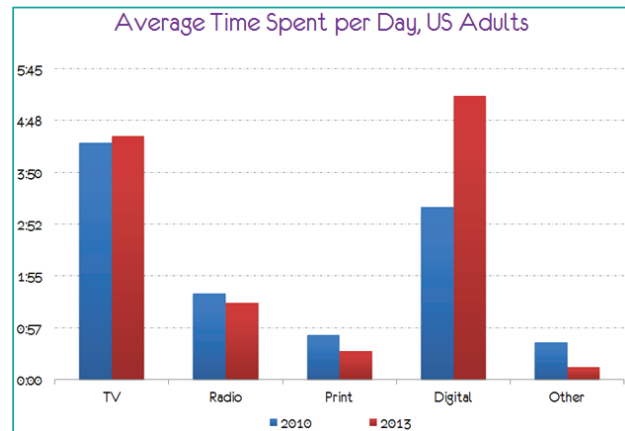
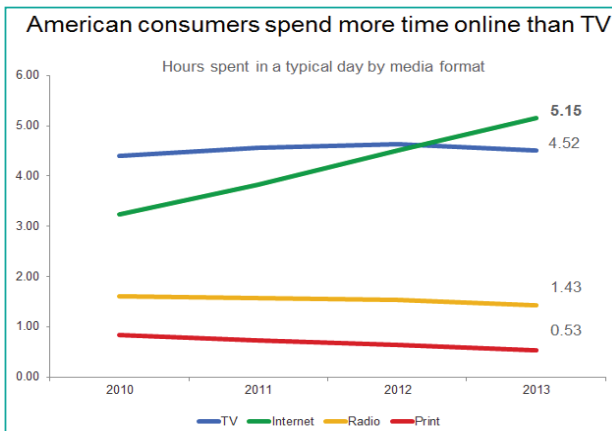
Be prepared

As mentioned in the first session of this chapter, we will learn in sequence how to visually represent numerical and non-numerical data. This session focuses on numerical data, and we will learn how to use different types of charts to represent a given set of data.

With your group, considering what you have learned about a successful data visualization, examine the two charts below for the same data and answer the following questions:



- 1 What is the story behind the two charts?
- 2 Which chart is harder to understand?
- 3 What changes could improve the harder one?



Source of the charts:

<https://www.kaushik.net/avinash/analytics-tips-complex-data-simple-logical-stories/>

List of Terms

Chart

A graph or diagram that represents quantitative or qualitative data such as bar chart and pie chart

Data preparation

The process of adding, deleting, calculating, or reorganizing the data before using it in a chart

Band chart

A line chart combined with background shades

Waterfall chart

A chart that presents the cumulative effects of either positive or negative contribution to reach the net value



Learn

Assigning data to the appropriate chart



Assigning data to the wrong chart type affects the story and the message that you want to deliver. However, how you can choose a suitable chart type to present your data effectively? The answer is mostly based on the purpose of presenting the data and the message that you want your audience to get from your data visualization. The next table shows a classification of some types of charts, based on the purpose which they serve.



90 percent of information transmitted to the brain is visual.

Types of Charts

Data Comparison Charts	Data Composition Charts	Data Distribution Charts	Data Relationship Charts	Data Evolution Charts
• Bar Chart	• Pie Chart	• Histogram Chart	• Scatter Plot	• Line Chart
• Bullet Chart	• Pareto Chart	• Box Plot Chart	• Bubble Chart	• Step Chart
• Heat Map	• Waterfall Chart			• Band Chart

Of course, this table does not cover all purposes or types of charts. However, it gives you an idea about how some charts can serve specific purposes more than others, although some charts can serve several purposes perfectly.

In general, classifications of the types of charts based on their purpose are not sharply divided, because you can find a chart that can serve more than one purpose. Thus, if you have a good sense of your data and you know exactly what message you want your audience to get, you can pick a suitable chart easily.

Data Comparison Charts

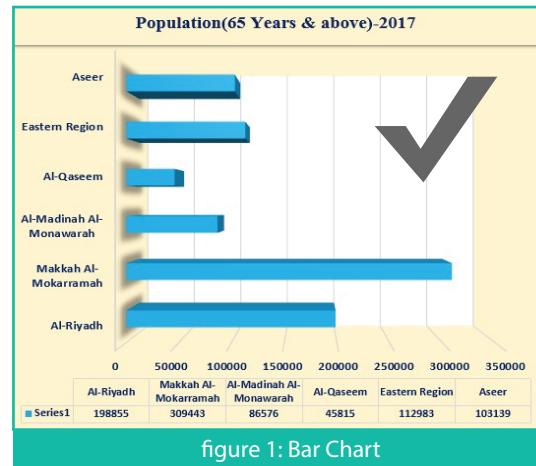
Comparison charts are used to compare the magnitude of values to each other and can be used to easily find the lowest and highest values in the data. These charts are visual comparison tools to compare between many sets of data.

For Example, "what products sells best" or "how are our sales compared to last year".

Following charts are the examples of Comparison Charts.

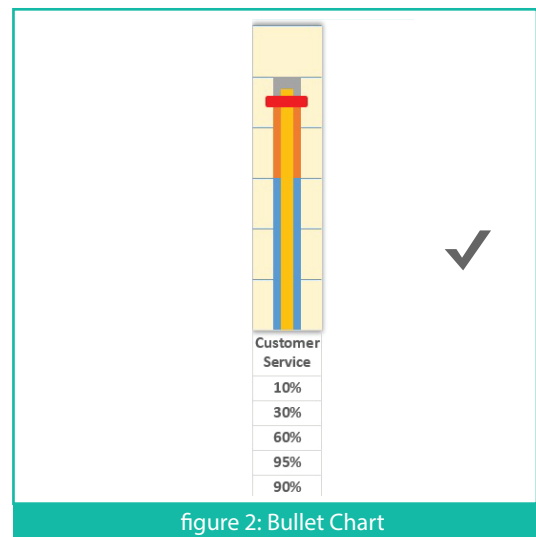
Bar chart

A bar chart is a chart which represents categorical data with bars and their height or length is proportional to the values. It shows comparisons among discrete categories. Refer to figure 1.



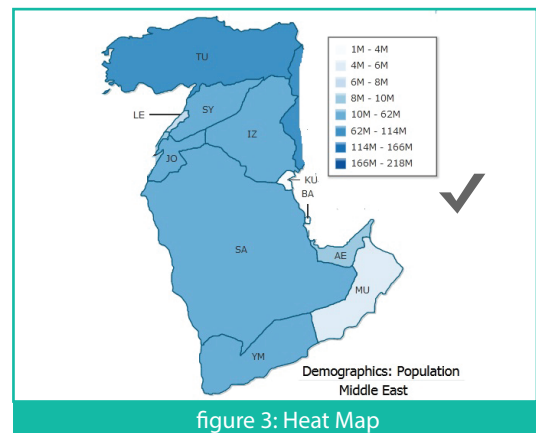
Bullet chart

A bullet chart or bullet graph features a single measure, which compares to one or more measures, and displays it in the context of qualitative ranges of performance, e.g. excellent, good, or poor. A bullet chart or graph is like a traditional thermometer. Refer to figure 2.



Heat map

A heat map is a graphical representation of data that displays values in a matrix and represents them as colors. The high values are represented with dark colors and low values with light colors. Refer to figure 3.



Data Composition Charts

Composition charts are used to see how a part compares to the whole and how a total value can be divided into shares.

For Example, "how big part of the market do we have in a region" or "what areas is our budget divided into".

Following are the examples of Composition Charts.

Pie chart

A pie chart is a circular statistical graph that is divided into slices to illustrate numerical proportion. Pie charts are best used for making part-to-whole comparisons. Refer to figure 4.

Pareto chart

A Pareto chart is a type of chart that contains both bars and a line graph. The Pareto chart is useful for analyzing what problems need attention first. It is based on the Pareto principle, which states that in a given situation, 80 percent of the effects come from 20 percent of the causes (known as 80/20 rule). Refer to figure 5.

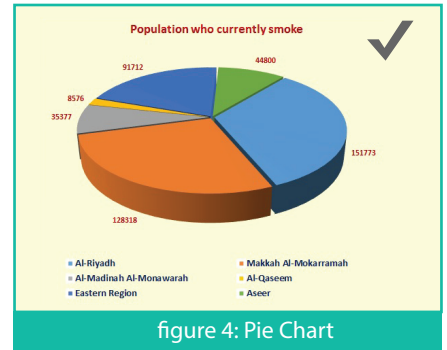


figure 4: Pie Chart

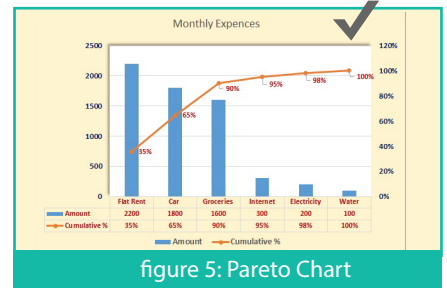


figure 5: Pareto Chart

Data Distribution Charts

Distribution charts are used to see how quantitative values are distributed along an axis from lowest to highest. It can be used to answer questions such as "number of customers per age group" or "how many days late are our payments".

Following are the examples of Distribution Charts.

Histogram

A histogram shows you an accurate distribution of continuous data. The data values are distributed into bins, which are represented as columns. Refer to figure 6

Boxplot

A boxplot is a graphical display of five statistical values (the minimum, lower quartile, median, upper quartile, and maximum) that summarize the distribution of a set of data. Refer to figure 7.

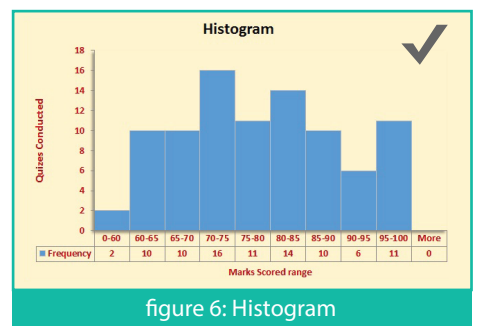


figure 6: Histogram

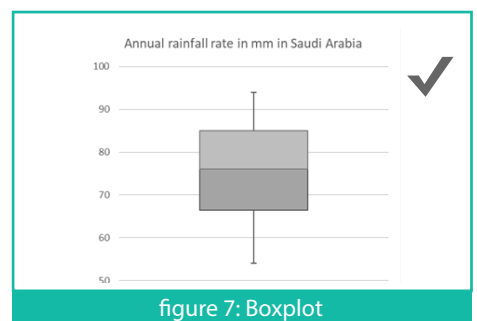


figure 7: Boxplot

Data Evolution Charts

These charts are used to analyze trends in your dataset and how a particular dataset performed during a specific time period,

Line chart

A line chart displays information as a series of data points connected by straight line segments. They are most often used to visualize data that changes over time. Refer to figure 8.

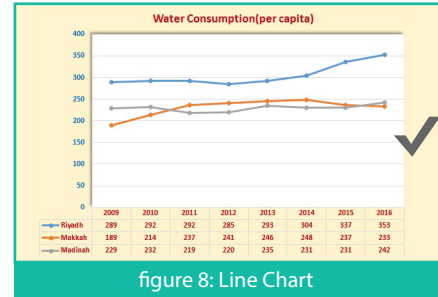


figure 8: Line Chart

Step chart

A step chart displays changes that occur at irregular intervals. It uses vertical and horizontal lines to connect data points which make a step-like progress. Refer to figure 9.

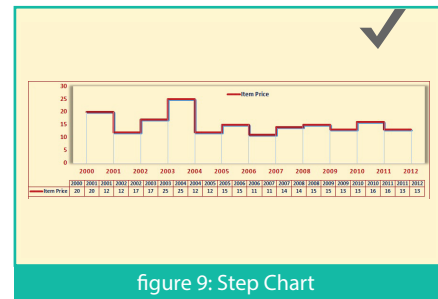


figure 9: Step Chart

Data Relationship Charts

Relationship charts are used to see the relationship between the data and can be used to find correlations, *For example "is there a correlation between Height and weight"*

Scatter plot

A Scatter Plot displays relationships between two variables and determines how closely the two variables are related. One variable is plotted on the horizontal axis and the other is plotted on the vertical axis. The pattern of their intersecting points can graphically show relationship patterns. Refer to figure 10.

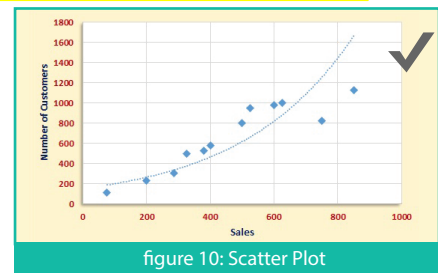


figure 10: Scatter Plot

Bubble chart

A bubble chart shows relationship between data in 3 numeric dimensions: x-axis data, y-axis data, and the size of bubble. It is used to compare and display relationships between categorized circles by the use of positioning and proportions. Refer to figure 11.

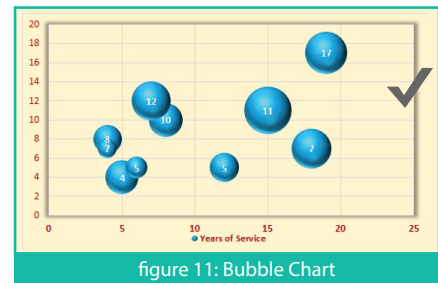


figure 11: Bubble Chart

There are many tools and software programs to create charts, for example Tableau Power BI Microsoft Excel. In the following sections, you will learn how to create band and waterfall charts using Excel. Excel has been chosen since it is the most powerful and flexible software to create charts, especially for average users.

Band chart

A band chart is basically a line chart combining with background shades that represent the upper and lower boundaries of the data. The purpose of using the band chart is representing data during a period against specific intervals. For example, you want to show, track, or monitor the performance of two companies during a period against percentage intervals.

Prepare the data for the band chart

The data should be ready to be used to create the band chart. The steps below will show you how to prepare your data.

Let's start with basic data you may have. Suppose that you have a table of data for the performance of two companies through the financial year

Now let's suppose that you want to compare the performance of the two companies against the following three intervals:

Month	Com.1	Com.2
Apr-15	86.4%	63.0%
May-15	45.8%	58.9%
Jun-15	44.1%	81.6%
Jul-15	77.6%	86.1%
Aug-15	80.7%	95.0%
Sep-15	83.7%	78.2%
Oct-15	78.8%	98.9%
Nov-15	76.0%	88.3%
Dec-15	79.0%	75.5%
Jan-16	77.0%	72.1%
Feb-16	67.1%	93.1%
Mar-16	45.8%	95.7%

Low (<50%)

Medium (50% - 80%)

High (>80%)

Thus, you have to add three columns to show the three intervals:

Month	Com.1	Com.2	Low (<50%)	Medium (50% - 80%)	High (>80%)
Apr-15	86.4%	63.0%	50%	30%	20%
May-15	45.8%	58.9%	50%	30%	20%
Jun-15	44.1%	81.6%	50%	30%	20%
Jul-15	77.6%	86.1%	50%	30%	20%
Aug-15	80.7%	95.0%	50%	30%	20%
Sep-15	83.7%	78.2%	50%	30%	20%
Oct-15	78.8%	98.9%	50%	30%	20%
Nov-15	76.0%	88.3%	50%	30%	20%
Dec-15	79.0%	75.5%	50%	30%	20%
Jan-16	77.0%	72.1%	50%	30%	20%
Feb-16	67.1%	93.1%	50%	30%	20%
Mar-16	45.8%	95.7%	50%	30%	20%

The value in the Low column represents the band from 0% – 50%

The value in the Medium column represents the band from 50% – 80%

The value in the high column represents the band from 80% – 100%

You will get the value of the bands by subtracting the lower percentage from the higher one. By now, your data is ready to create the band chart.

Create the band chart

To create a Band Chart, Follow the steps given below:

- 1 Select your data, which you already created above. Insert a line chart from the Charts group of Insert Tab. The Line Chart is inserted.

- 2 Select the Chart and click on Change Chart Type from Design Contextual Tab.

- 3 Change the chart type to Combo from displayed dialog box.

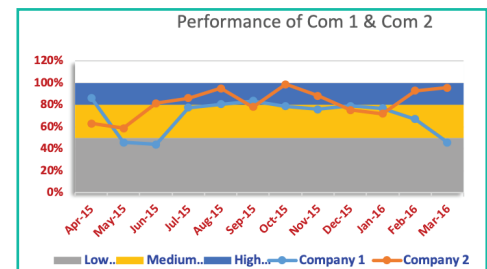
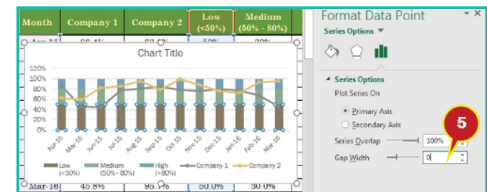
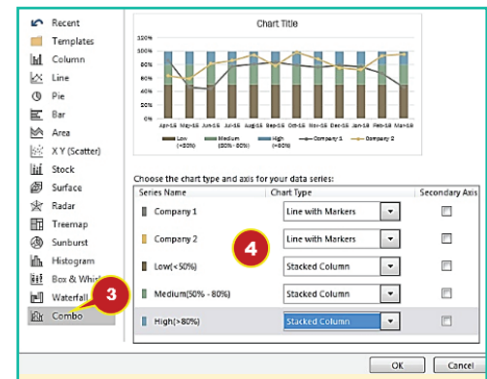
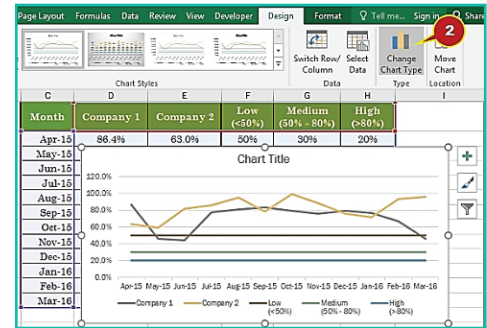
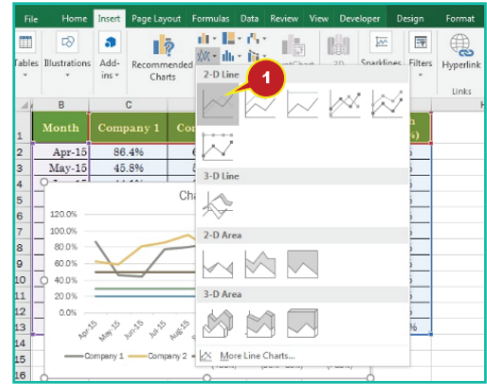
- 4 Change the chart type for the data series Com. 1, Com. 2, Low, Medium, and High as follows:
 - Line with Markers for the series Company 1 & Company 2.
 - Stacked Column for the series Low, Medium, and High.

Click on OK button to apply changes

- 5 Double click on any Data Point, then, in the Format Data Series pane, change the gap width to 0%. As a result, you will get bands rather than columns.

Finally, enter chart title, apply a suitable design, and other formatting from the Design & Format contextual Tabs. Here is your final band chart.

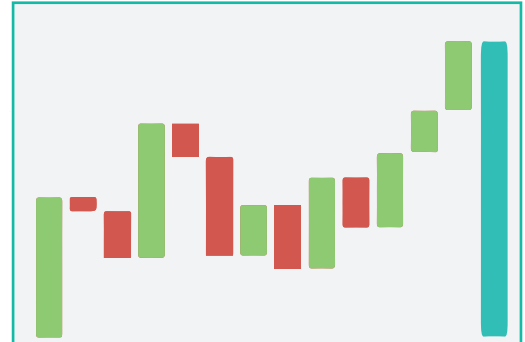
By looking at the band chart now, it is easy and quick to track the performance of the two companies — when the companies are performing well and when they need to improve.



Waterfall chart

A waterfall chart is used to show how an initial value is increased and decreased by a series of intermediate values, leading to a final value. It is also called a flying brick or Mario chart.

The waterfall chart is commonly used in financial departments of small to large businesses. It gains its popularity from its feature of presenting the cumulative effects of either positive or negative contributions to reach the net value.



You can use the waterfall chart for quantitative analysis, such as inventory or performance analysis, and also for contract tracking, which helps the stakeholders to make the right decisions about finishing some contracts or investing more in others.

Thus, if you have start value and end value and you want to show how the changes occur to move from the start to the end value, the waterfall will be the perfect choice.

Creating Waterfall Chart

The given table shows income and expenses for a particular month e.g. March. The Waterfall Chart for the given data will show:

- Increase i.e. income
- Decrease i.e. expenses
- Total i.e. Starting Cash Balance & End Cash Balance

Category	Amount
Starting Cash Balance	90000.00
Income1(Salary)	10000.00
Housing	-30000.00
Grocery	-10000.00
Car Payment	-5000.00
Income2(Shop Rented)	10000.00
End Cash Balance	

To create a Waterfall Chart, Follow the steps given below:

- 1 Calculate the End Cash Balance
Select the cell and click on **AutoSum** command from the Home Tab. Press Enter.
The End Cash Balance is Calculated i.e.

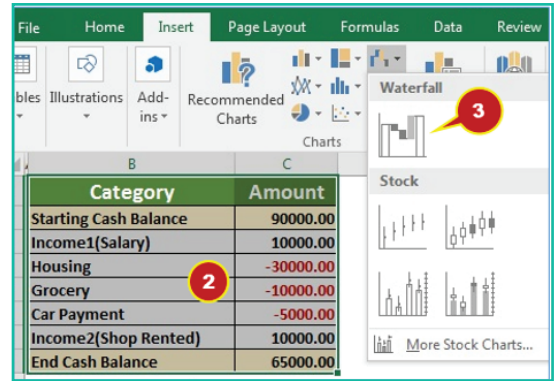
$$\begin{array}{r}
 \text{End Cash Balance} \\
 = \\
 \text{Starting Cash Balance} \\
 + \\
 \text{All Income} \\
 - \\
 \text{All Expenses}
 \end{array}$$

Category	Amount
Starting Cash Balance	90000.00
Income1(Salary)	10000.00
Housing	-30000.00
Grocery	-10000.00
Car Payment	-5000.00
Income2(Shop Rented)	10000.00
End Cash Balance	65000.00

2 Select all the data from the given table

3 Click on **Insert Waterfall or Stock Chart** Command from Charts group of the Insert Tab. Select Waterfall chart

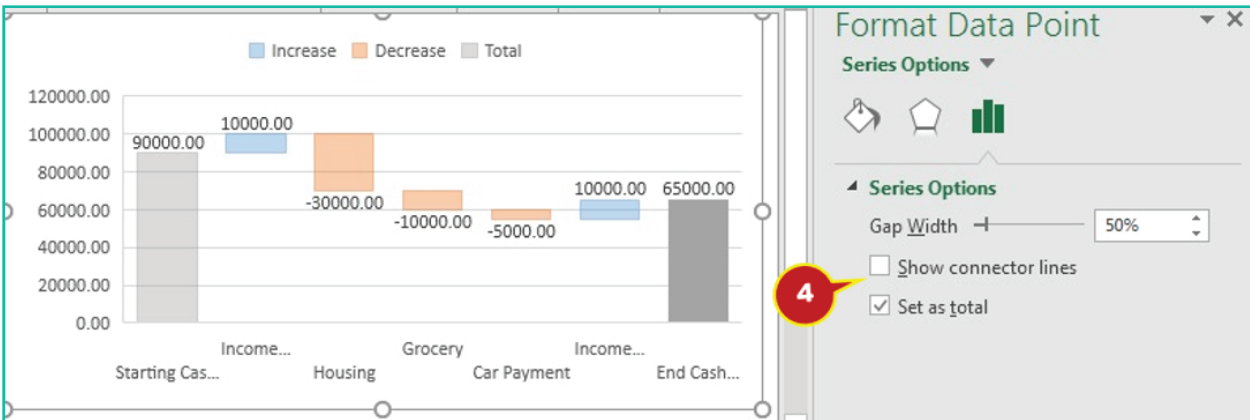
The selected Waterfall Chart is inserted.



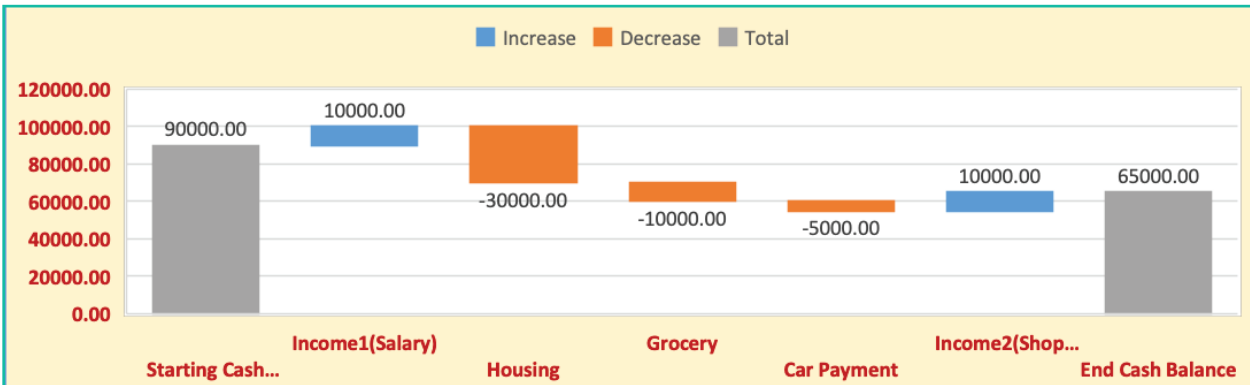
4 Double Click on the **End Cash Balance** Data Point, the Format Data Point Pane is displayed:

- Check **Set as Total** box
- Uncheck **Show Connector Lines** box

Similarly, double Click on the **Starting Cash Balance** Data Point and follow the same process



Select the Chart, apply a suitable design and other formatting from the Design & Format contextual Tabs.





Apply

Activity 1

Visit the website:

<https://www.data-to-viz.com>

Then, in pairs, answer the following questions:

- What is the criterion to choose the most appropriate chart to represent numeric data?
- How many categories of charts are used to represent numeric data, and what are they?

Activity 2

Fill in the blanks with the appropriate words provided below.

(waterfall chart, band chart, histogram, heat map, pie chart)

- A _____ chart is basically a line chart combined with background shades that represent the upper and lower boundaries of the data.
- The _____ chart is commonly used in financial departments of small to large businesses.
- A _____ is a circular statistical graph, which is divided into slices to illustrate numerical proportion.
- A _____ is a graphical representation of data that displays values in a matrix and represents them by colors.

Activity 3

Choose TRUE or FALSE for the following statements.

- Assigning the data to the wrong chart does not affect the story and the message that you want to deliver. () True
() False
- The purpose of using the band chart is representing data during a period against specific intervals. () True
() False
- A bullet chart or graph is like a traditional thermometer. () True
() False
- A bubble chart shows relationship between data in 3 numeric dimensions () True
() False



Closing

Focal points of this session are:

- The purpose of the data visualization is the key to choosing the right chart.
- Preparing your data is a crucial step before creating the chart.
- A band chart is used to compare two sets of data during a period against specific intervals.
- A waterfall chart is used to represent the positive and negative changes that occur between start and end points.

CHAPTER

3

Session

3

Visualizing Non-Numerical Data



Goal

In this Session, you will learn

how to visualize non-numeric data using different visualization techniques.



Learning objectives

Dear Student

By the end of this Session, you should be able to:

Understand what a diagram is.

Differentiate between different types of visualization techniques for non-numeric data.

1

2

3

4

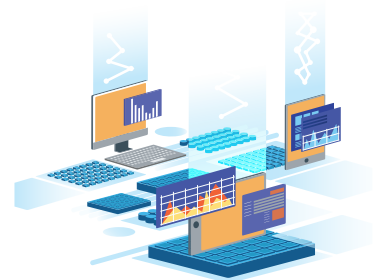
Assign data to an appropriate diagram.

Use Dia software to create different diagrams.



Be prepared

As we mentioned before, the main reason to visualize data is because we have an overload of data around us from different sources, including the internet. This data overload makes it hard to make decisions. The rule of data visualization is to transfer complicated data into a simple form that is easier to understand. In session 1, we learned about the different types of data, and in session 2, we learned how to visualize numerical data. In this session, we will focus on visualizing non-numerical data using several techniques.



In groups, try to convert the following table of an organizational hierarchy into another visual form to make it easier to understand and present.

Employee name	Reports to	Title	Department
Ahmad Nasser	CEO	Executive	
Khalid Saleh	Ahmad Nasser	Development manager	Product development
Ali Rashed	Khalid Saleh	Software developer	Product development
Maher Fares	Ahmad Nasser	Marketing manager	Marketing
Abdullah Saud	Maher Fares	Market researcher	Marketing

Table 1: Organizational hierarchy.

List of Terms

Term	Explanation
Non-numeric data	Data in a qualitative form or data we cannot measure in numbers
Diagram	A visual illustration of a group of concepts and the relationships between them
Visualization technique	A method to represent data considering the purpose of the visualization
Word cloud	A group of words of different sizes placed together into a cloud shape
Tree diagram	A visualization technique that has a start, root, or parent node, with branches emanating from it
Flowchart	A visualization technique to represent algorithms and processes
Venn diagram	Sets of entities; each set is drawn within a circle



Learn

Introduction

Non-numerical data can be referred to as categorical data or qualitative data. Non-numerical data can be observed but cannot be measured. These data mostly deal with the description of something such as feelings or characteristics.

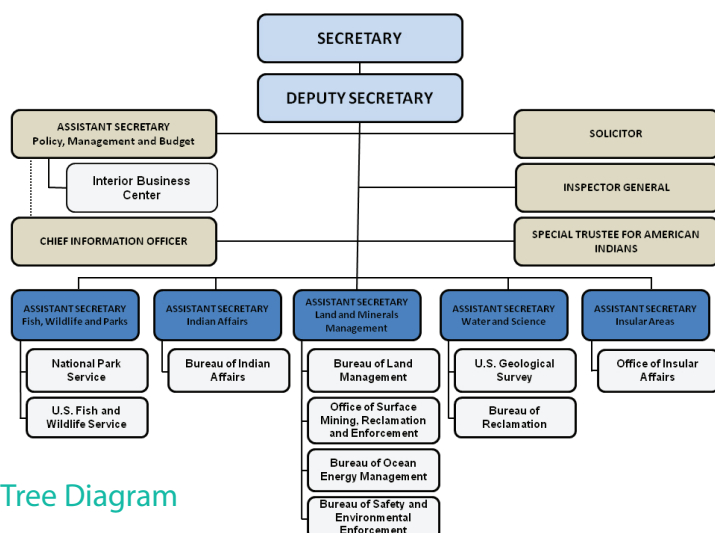
When we talk about non-numerical or categorical data, this does not mean they cannot be numbers. However, the numbers in non-numerical data do not have any mathematical value, and mathematical operations cannot be performed on these numbers. For example, if you collect data for a survey that needs the gender of a client, you can make the task easier by assigning the number "1" as a code for men and the number "2" as a code for women. The numbers "1" and "2" here do not have any mathematical meaning, and you cannot add them.



Most non-numerical data can be represented in a diagram. In the next sections, we define a diagram and describe the most common visualization techniques to draw diagrams.

What is a diagram?

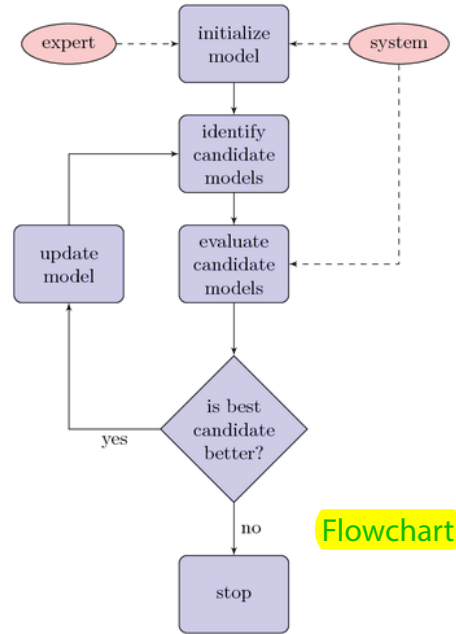
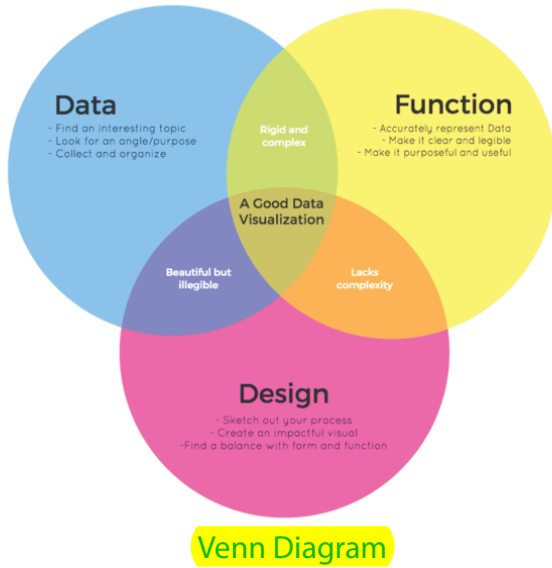
A diagram is a visual illustration of a group of concepts and the relationships between them. Diagrams are mostly used to represent non-numerical or qualitative data. Diagrams are used to convert qualitative data into visual representations to make information clear, simple, easy to understand and remember. Diagrams can use different visualization techniques according to their purpose and the types of concepts and relationships that you want to show. You can see diagrams around you everywhere, in educational settings such as in books, posters, or presentations, and in business settings such as in organizational hierarchies or process maps for task completion.



Tree Diagram



Word Cloud

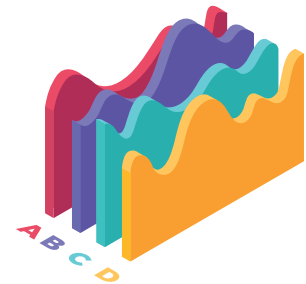


DIFFERENT TYPES OF VISUALIZATION TECHNIQUES FOR NON-NUMERIC DATA

The next section shows how to choose the right diagram to represent your data.

Assign your data to a suitable diagram

As we learned, information communication is the main reason to visualize qualitative data using diagrams, and it provides a crucial advantage. But how do you choose the most suitable diagram to communicate your message to an audience? The main point to consider is the purpose of the diagram. For example, do you want to show the process of how to accomplish a task, to show cause and effect of an issue, or to illustrate relationships between different concepts? The following table shows some of the most common diagrams, classified according to their purpose.



Diagramming purpose	Example of diagramming technique
Demonstrate a process: show the steps to accomplish a task	Flowchart
Brainstorm: create, include, and comprehend ideas about a topic	Mind map
Identify defects: analyze the causes of a problem to correct the defect	Fishbone diagram
Demonstrate relationships: show the connections between concepts or people	Tree diagram
Compare: show similarities and differences between two or more concepts	Venn diagram

The next section covers some of the most common visualization techniques for diagrams.

Common types of diagrams for non numeric data visualization

Mostly, only words are counted in non-numerical data; thus, most visualization techniques for this type of data are about words or concepts and the relationships between them. The most common visualization techniques for non-numerical data are the following.

Word clouds

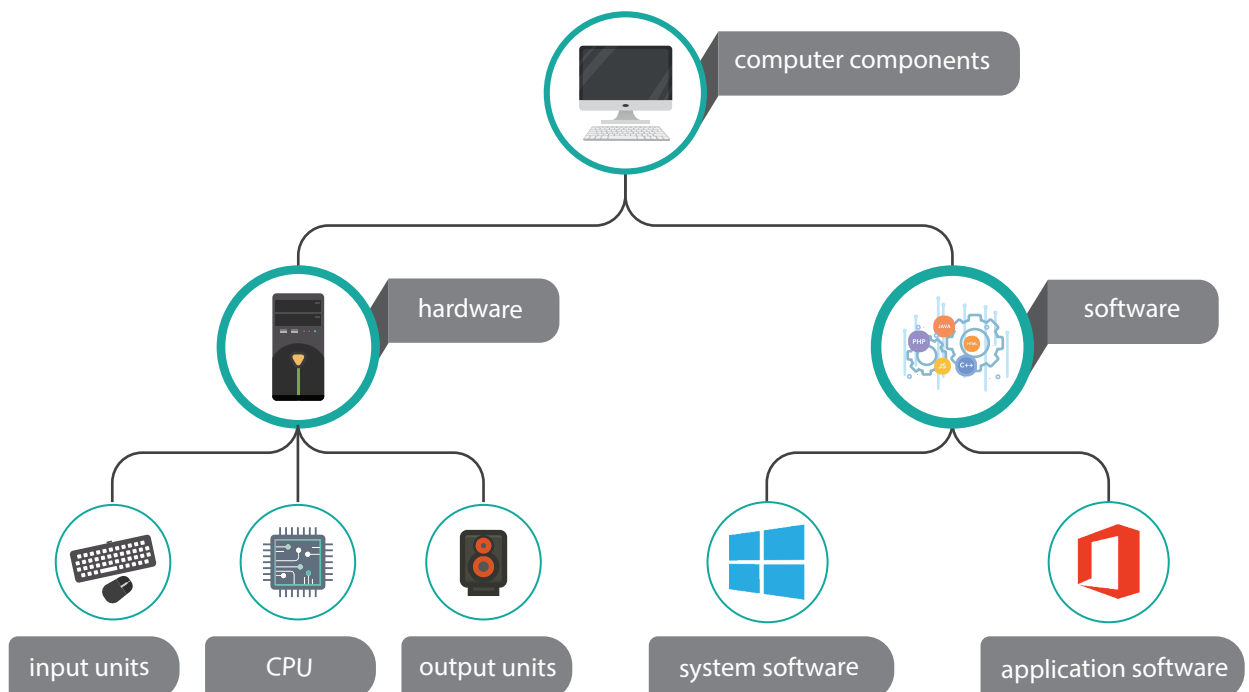
Word clouds or tag clouds are one of the most well-known visualization techniques for non-numerical data. Word clouds are often seen in websites and blogs to show keyword or tag usage. A word cloud appears as a group of words of different sizes placed together into a cloud shape. The size of each word represents the frequency of that word in the text. You can use word clouds to highlight keywords in a text or to compare two different texts.



WORD CLOUD OF THE HOMEPAGE OF THE KING SAUD UNIVERSITY WEBSITE BY [HTTPS://TAGCROWD.COM/](https://tagcrowd.com/)

Tree diagrams

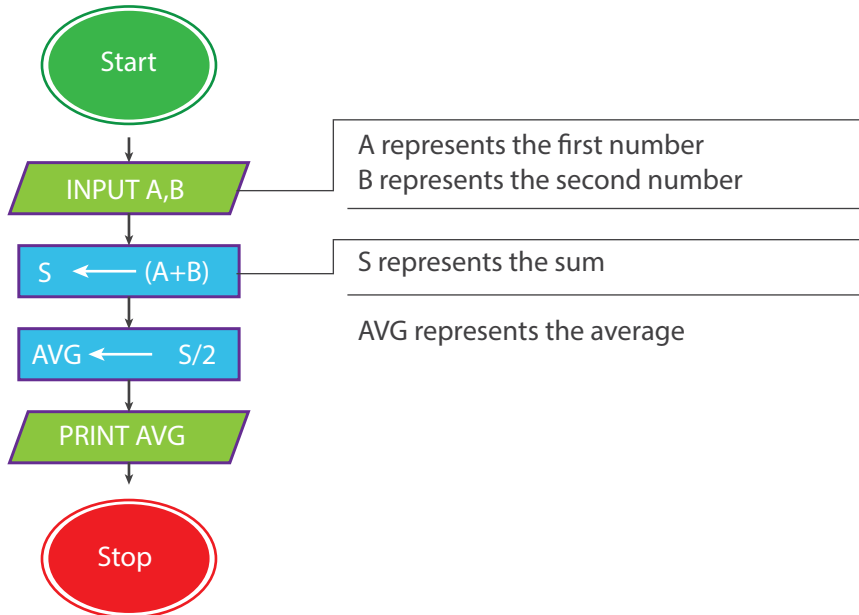
A tree diagram is a visualization technique that has a start, root, or parent node, with branches emanating from it. Each branch leads to a new node, and so on. Tree diagrams are used to make strategic decisions or valuation calculations, or simply to show a classification, as each branch represents a mutually exclusive decision. Each new node is created at a new level, and you continue moving to lower levels until you reach the end, leaf, or child node with a concluding decision.



TREE DIAGRAM OF COMPUTER COMPONENTS

Flowcharts

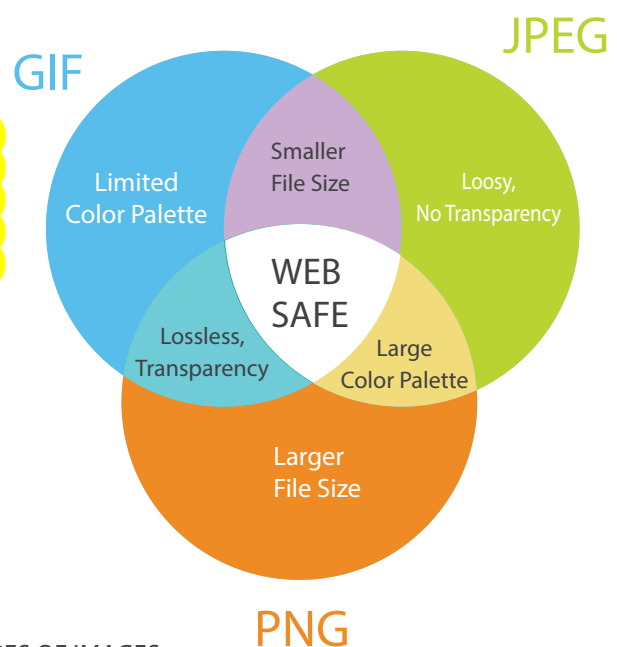
Flowcharts are a visualization technique to represent algorithms and processes. As algorithms contain series of computational operations to solve a problem, they can be hard to understand and track within a text description. A flowchart's role is to convert the algorithm into an easy-to-understand-and-track diagram. Flowcharts use different shapes and connections. Each shape represents a different type of operation, and each arrowhead connector represents the direction of the steps.



A FLOWCHART TO SHOW THE AVERAGE OF TWO NUMBERS

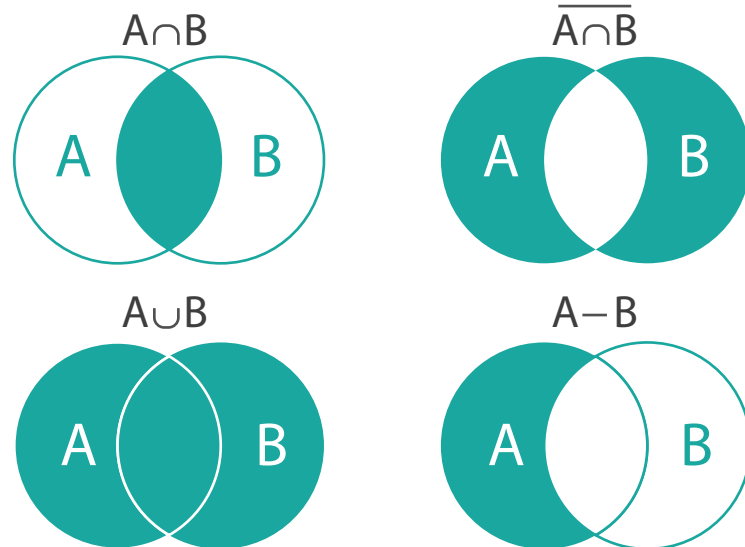
Venn diagrams

Venn diagrams are also known as set diagrams because each collection of entities represents a set. In Venn diagrams, each set is drawn within a circle, and if two sets intersect, it means that one or more entities occur in both sets. The entities outside the intersection area represent the differences between sets.



THE DIFFERENCES AND SIMILARITIES AMONG THREE TYPES OF IMAGES

Venn diagrams are also used in mathematics to represent possible logical relationships between sets.



LOGICAL RELATIONSHIPS BETWEEN TWO SETS

Diagramming Applications

Visio software is a Microsoft Office application and one of the most commonly used tools to create different types of diagrams. However, other, more affordable applications exist with great features, such as SmartDraw, Lucidchart, and Dia.

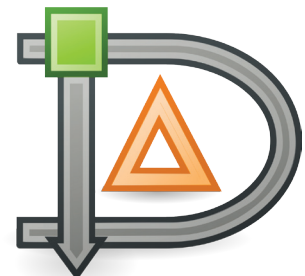


The next section will introduce Dia software in detail.

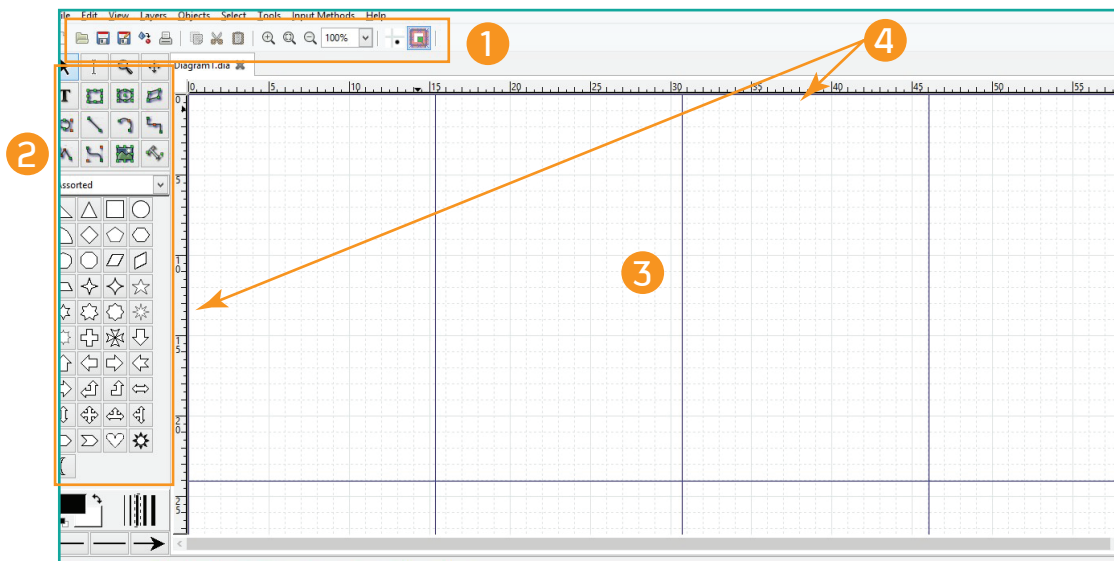
Overview of the Dia application

Dia is a diagramming application that can be used to make a large variety of diagrams. Dia is easy to learn and flexible enough to allow power users to create highly customized diagrams. It has many excellent features, such as the following:

- 1 Dia is free
- 2 It allows multiple page printing
- 3 It can be exported to many formats (.eps, .svg, .cgm, and .png)
- 4 It allows the user to create custom shapes

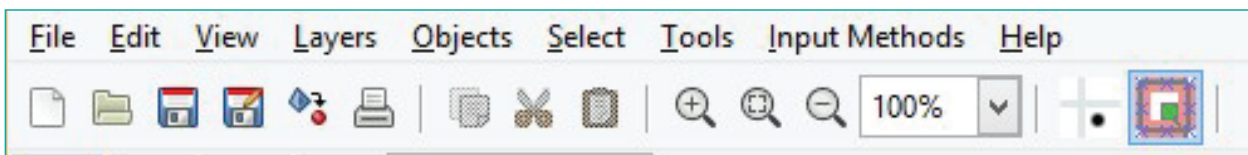


Interface of Dia



As you can see, the main interface of the Dia application is simple and easy to explore.

1 Ribbon



The ribbon has all the commands you need to deal with the file, canvas, and objects. The ribbon has several tabs, and each tab includes a group of commands.

File tab

Includes commands that deal with files, such as new, open, save and export

Edit tab

Includes commands such as do, undo, copy, cut, and paste

View tab

Includes commands concerned with the appearance of the canvas

Layers tab

Allows you to distribute objects into multiple layers on your canvas

Objects tab

Allows you to control objects in ways such as send them back, bring them forward, or group them together

Select tab

Includes options for object selection

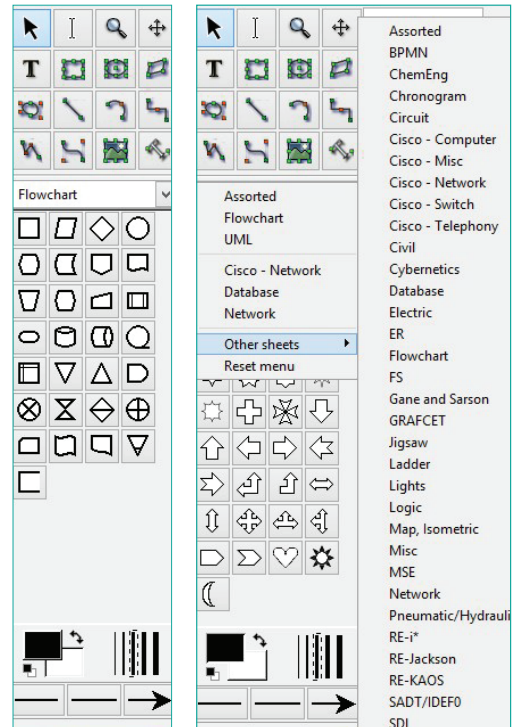
Tools tab

Includes all control tools in the toolbox, such as modify, text edit, and magnify

2 Toolbox

The toolbox contains three parts:

Basic objects	Includes all tools under the Tools tab, such as modify and edit text
Listbox	Allows you to select a sheet of special objects based on the desired diagram
Controls	Allow you to control the foreground, background, line width, and line style of the diagram



3 Canvas The canvas is where all the objects and diagrams are placed. You control the properties of the canvas in the View tab.

4 Rulers The rulers allow you to align objects on the canvas and know how big the objects and canvas are. There are both a horizontal and a vertical ruler.

What can I do with Dia?

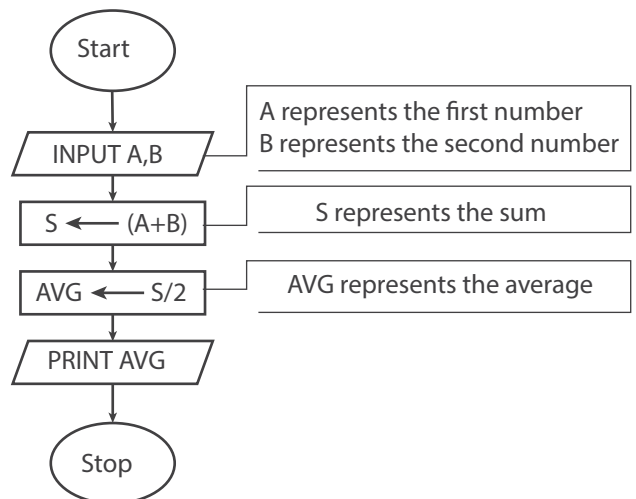
Dia software allows you to draw many types of diagrams, such as flowcharts, networks, and UML (Unified modeling language) diagrams.

The next sections provide example of drawing a flowchart using Dia.

How to create a flowchart?

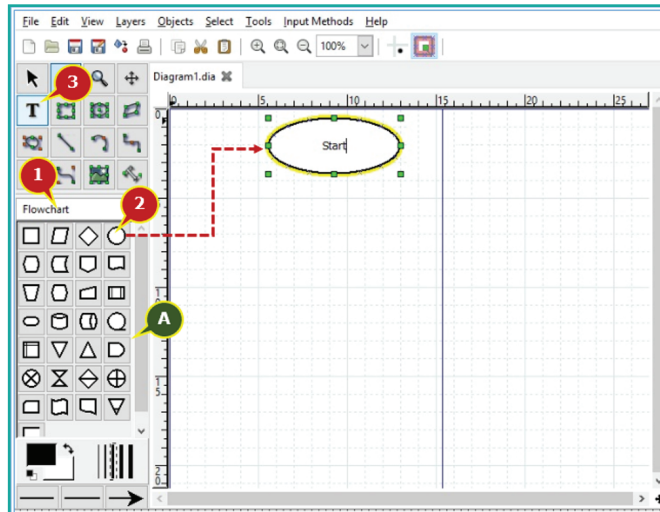
you can draw a flowchart using several shapes. Each shape represents a specific action. For example, the oval shape is used to represent the start and the end of an algorithm.

Now, let's draw the previous example of the flowchart to calculate the average of two numbers, using Dia.



To create a Flow Chart, follow the steps given below:

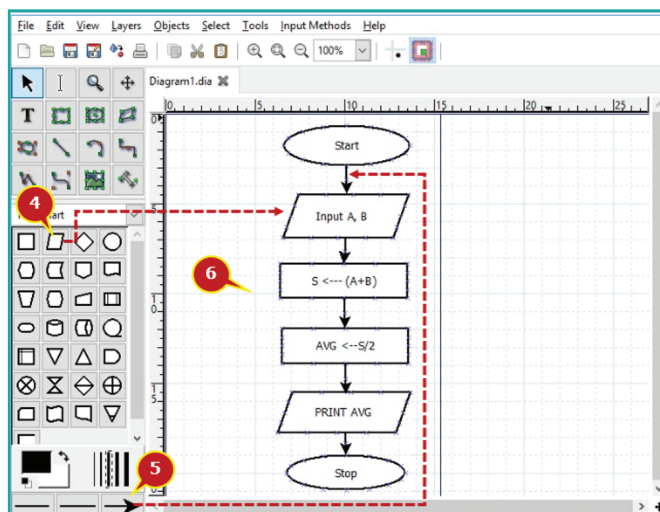
- 1 From the sheet's drop-down menu, select "Flowchart"
 - A Once you select the flowchart sheet, you will see all the objects that relate to flowchart diagrams.
- 2 Select the circle shape to represent the first step in the algorithm, "Start," and draw an oval shape on the canvas.
- 3 Click on the text edit tool to write "Start" inside the shape.



- 4 Select the input/output shape from the objects and draw the shape for the input of the two numbers. Then click on the text edit tool to write "INPUT A, B" inside the shape.
- 5 To connect the two shapes, select the line from the toolbox and draw an arrow between the two shapes. If the line connects to the edge of the shape correctly, the shape will be highlighted in red.

You can then control the style of the beginning and end of the arrow from the tools at the bottom of the toolbox.

- 6 Repeat these steps to complete the flowchart.





Apply

Activity 1

Now try to re-represent Table 1, using a suitable diagram in Dia.

Activity 2

Re-draw the network types (PAN, LAN, MAN, and WAN) from chapter 2 by using the network sheet in Dia.

Activity 3

With your friend, use any search engine to find more diagrams to visualize non-numeric data, other than the ones you have learned about in this session.

Activity 4

Choose TRUE or FALSE for the following statements.

a) Non numerical data mostly deal with the description of something such as feelings or characteristics. True False

b) A Flowchart shows similarities and differences between two or more concepts. True False

c) In Word Cloud, the size of each word represents the frequency of that word in the text. True False

d) Venn diagrams are also known as set diagrams because each collection of entities represents a set. True False



Closing

In this session, you learned:

The definition of a diagram.

The criteria to consider in choosing the appropriate diagram for particular data.

Different types of visualization techniques.

The basic skills to use Dia software.

CHAPTER

3

Session

4

Data Dashboards



Goal

In this Session, you will learn

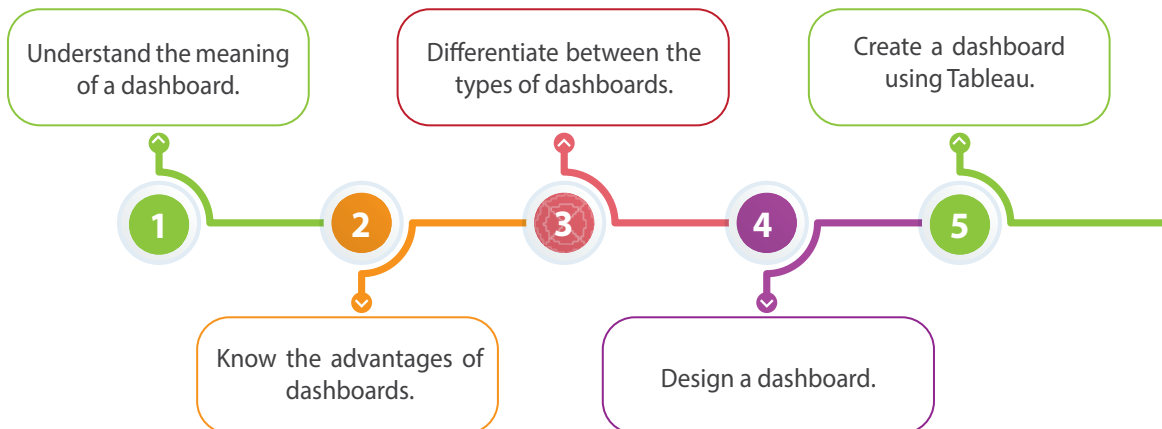
how to design and create a dashboard using Tableau.



Learning objectives

Dear Student

By the end of this Session, you should be able to:





Be prepared

For a long time, people relied on spreadsheet software not just to process numbers and do calculations, but also to represent numbers in a visual format. Nowadays, with the rapid growth of data sources, spreadsheet software is no longer adequate. The dashboard is a powerful tool in the business intelligence field that fills this gap. The idea of a dashboard itself is not new; however, dashboard features and their ability to be more interactive have evolved and are now key to their success.



List of Terms

Term	Explanation
KPI (Key Performance Indicator)	A measurable value that indicates a company's progress towards its goals
Metric	A quantifiable and measurable unit within a KPI that tracks processes
Dashboard	A single visual display combining data from several sources to track the business KPIs, metrics, and key data points
Daily web overview dashboard	An operational dashboard that monitors the daily performance of a website
SaaS (Software as a service) dashboard	A strategic dashboard mostly viewed once a day by executives to monitor their business performance
MRR(Monthly Renewal Rate) dashboard	An analytical dashboard mostly used in the accounting department to predict the new trends



Learn

Introduction

In this session, you will learn about the different types of dashboards, how to design dashboards to answer specific questions or measure specific key performance indicators (KPIs), and how to create a dashboard using Tableau software.

What are KPIs and metrics?

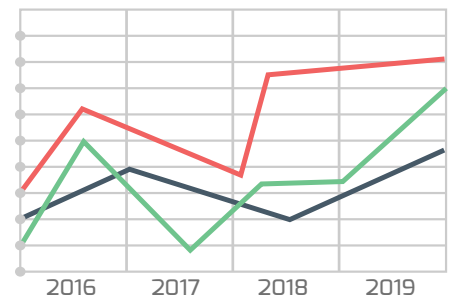
Before learning about dashboards, we will first need to understand the most important factors used in dashboards, i.e. KPIs and metrics.

A KPI is a measurable value that indicates a company's progress towards its goals and how well that company is achieving its key business objectives. Selecting the right KPIs for measuring a company's success is an important need for any organization.

For example, sales by region KPI will help in analyzing which regions are meeting sales objectives and you can provide better feedback for underperforming regions.

Metric is a quantifiable and measurable unit that tracks processes. With the help of metrics, you can track the activities and performance of a set target. All KPI's are metrics but all metrics are not KPI's. Everything we can measure is a metric. KPI's are the metrics but the most important ones. For example:

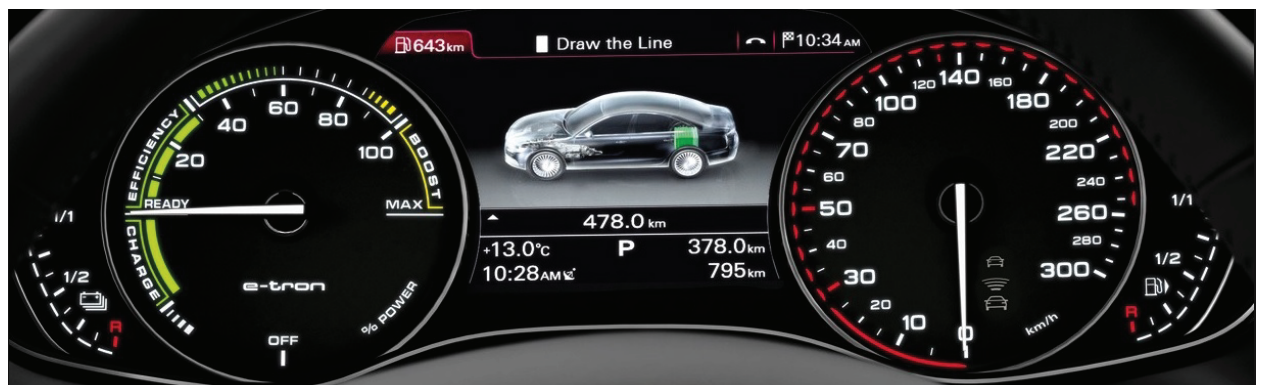
- Net Profit Margin
- Sales by Region
- Monthly Sales Growth
- Sales per Representative
- Net Sales
- Operational Cash Flow

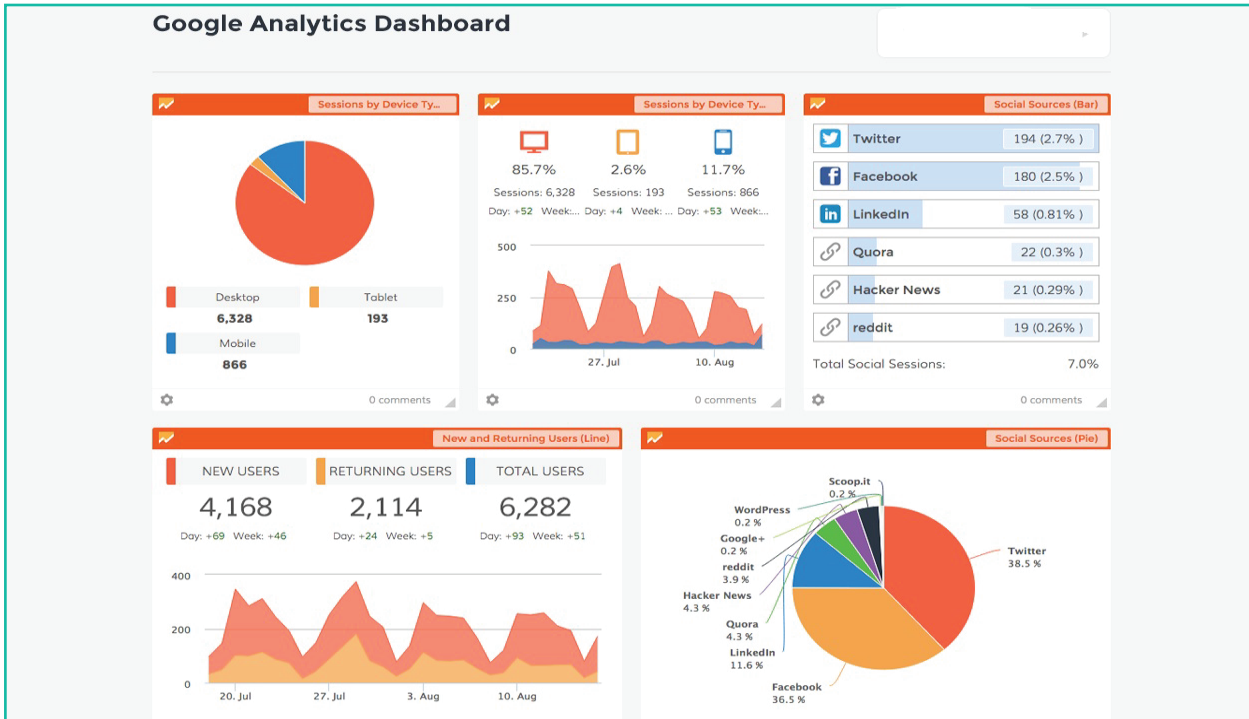


Data Dashboard



The dashboard is a single visual display combining data from several sources to control the health of a business, a department, or a process, by monitoring the business KPIs, metrics, and key data points. The word "dashboard" originates from the car dashboard, where the driver of a car can see all the major functions of the car at a glance. A business dashboard is an information management tool that provides answers to the questions of the decision makers by tracking and analyzing business performance and helping them make the right decisions.





What are the advantages of dashboards?

A dashboard is a powerful business intelligence tool that provides many advantages which a single chart or graph cannot provide. Some advantages of dashboards are that they can:

- 1 provide all the information about your business at a glance
- 2 track the organization's goals and progress
- 3 easily identify the parts of the business that require immediate attention
- 4 provide an overview of business status as well as detailed reports

Characteristics of dashboard

Characteristics	Description
1 Customizable	The dashboard tool gives you the ability to create different dashboards for each departments, using the same data.
2 All in One	By using the dashboard tool, you can represent all data analysis in one place
3 Level of Detail	You can drill into details and show as much information as you need in a dashboard by including the required objects and variables.
4 Easy to Use and Understand	No special training is required.
	Import data from different data sources, such as spreadsheets or databases.
	The software uses drag-and-drop features to create charts or graphs. The graphic design of the software makes it easy to use.
5 Accessibility	They are accessible from anywhere and at any time, even on mobile devices.

Types of dashboards

There are several types of dashboards, including operational, strategic, and analytical dashboards. The differences among these three types depend on the type of information and level of detail in the dashboard report. As a result, each dashboard type targets different teams and different business departments. In this section, we will learn about dashboard types, what type of information each contains, and which teams will benefit from each type of dashboard.

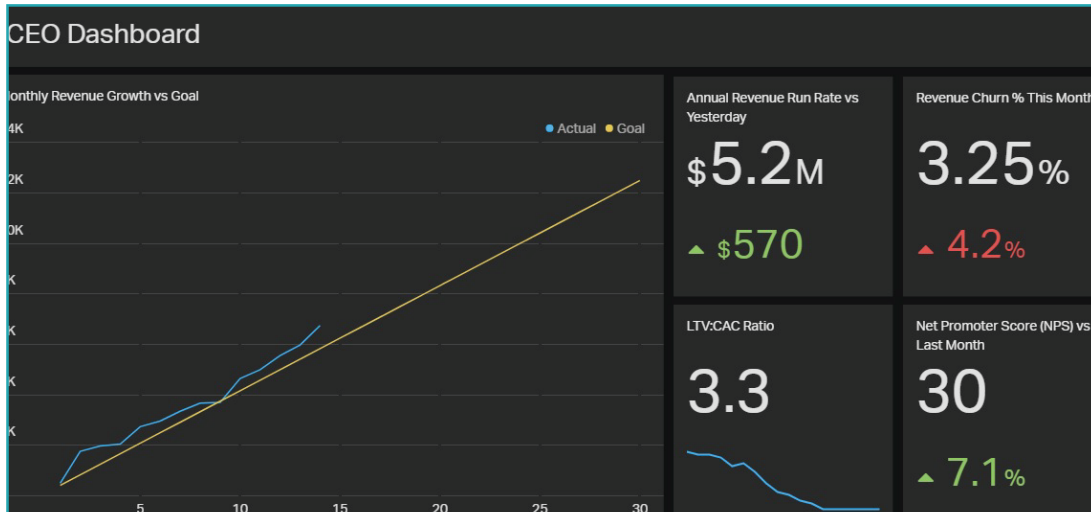
Operational dashboards (where are we now?)

An operational dashboard is a reporting tool that tracks business metrics and KPIs. The content of operational dashboards changes quickly and frequently, and the data used to create it is updated from one minute to another. Thus, it is shown several times a day to measure improvements toward predefined goals. A dashboard that monitors the daily performance of a website (daily web overview dashboard) is an example of an operational dashboard. Usually, this type of dashboard is used by a digital marketing department.



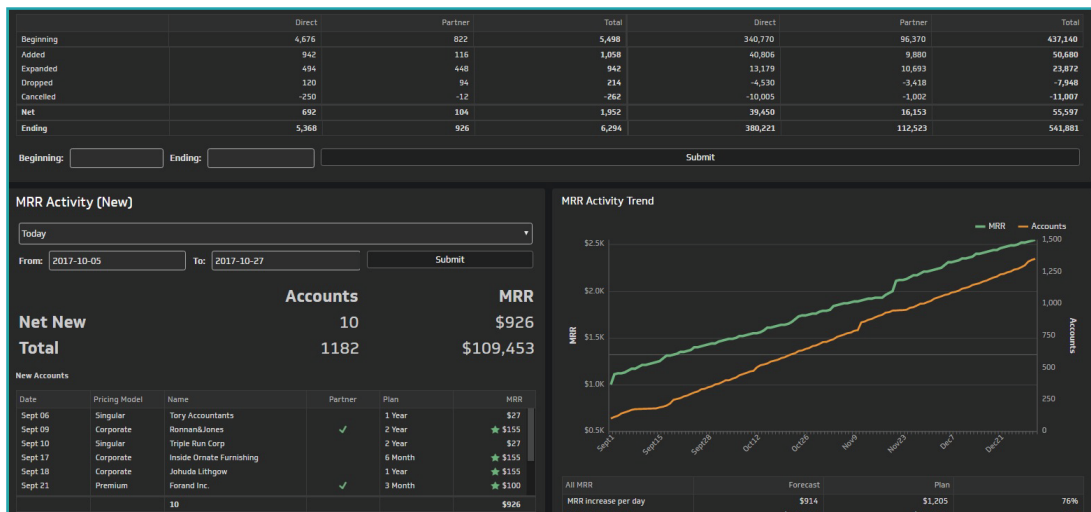
Strategic dashboards (how are we doing?)

The strategic dashboard is a business intelligence tool that tracks the performance of the business by measuring different KPIs. The data in a strategic dashboard changes, but these changes are less frequent than in operational dashboards. Viewing it once a day is generally enough. It is typically used by executives to keep on top of business KPIs and monitor the performance of their businesses. The SaaS (Software as a Service) dashboard, which is mostly used by CEOs, is an example of a strategic dashboard.



Analytical dashboards (where we are going?)

The analytical dashboard could be either an operational dashboard or a strategic dashboard. However, it has drill-down functionality, which allows the analysis of large amounts of data to discover new insights, view trends, or predict conclusions. The analytical dashboard is the most common type of dashboard because the designer and developer of the analytical dashboard is a data analyst. Analytical dashboards need accurate and up-to-date data to provide accurate predictions. However, the data is not updated very frequently. The MRR (Monthly Renewal Rate) dashboard, which is mostly used in the Accounting department, is an example of an analytical dashboard.



Designing the dashboard

Designing a dashboard is an art. While designing dashboards, keep in mind that you are communicating important and large amounts of data which need to be readable at a glance.

Try to make complex data simpler.

Design your dashboard to tell a clear story.

Know your data well to transfer it into a meaningful and informative dashboard.

Choose the right KPIs (this is an especially important skill).

Identify your audience

First, think about your audience. Successful dashboards target a specific team in the business. Once you know who your target team is, it will be easy to pick the data that is most useful to them and that they really need to know. For example, if the executive managers of the business are your audience, then your data should provide a good overview of business performance by choosing KPIs and metrics such as periodic revenue or cost comparisons.



Choose the right dashboard type



Next, you will need to decide which dashboard type is most appropriate to represent the selected data to your audience. As we discussed in the previous section, there are three main types of dashboards: operational dashboards, strategic dashboards, and analytical dashboards. Each type is designed to represent specific data to specific teams in the business. For example, if you want to represent data from everyday operations such as website monitoring, or sales data such as the number of calls or appointments, then the operational dashboard would be your best choice.

Order your data and use space effectively

When you have different groups of data in your dashboard, pay attention to their logical order. For example, part of your data may be related to marketing, while another part is related to sales. Make sure that the marketing data is grouped and ordered logically and that the same approach is taken with the sales data. Additionally, you need to use the space on the dashboard effectively. For example, do not save the top left corner of your dashboard for the business logo. The top left corner is the first space in your dashboard that attracts the attention of western language speakers, as their languages are read from top to bottom and from left to right. Thus, keep this space for an important piece of data.





Connect the data with the audience

This rule depends on the size of the business. For a small business, you may choose to create a single strategic dashboard for the executive manager and the heads of departments. In this case, the dashboard contains several KPIs for different departments. Each person in the audience will find the required data for his department. For a large business, the situation is different. Each department will have a separate dashboard with their required data. In either situation, if you have a mixed audience, make sure that you identify the data requirements for each group while keeping in mind the scope of the whole business, each department, and each individual.

Just relevant and important data

The rule here is simple and easy to follow. Include in your dashboard just the relevant and important data. Do not include items just because you can, or to fill a space. Irrelevant and unnecessary data would distract the attention of your audience from the important message that you want to convey. There is no golden rule to the amount of data you should include on a dashboard; however, you can figure out what important data to include by identifying your audience and their data requirements.



Identify the refresh rate



Deciding how often you will need to refresh the data on the dashboard before actual development will save you time and effort while designing the dashboard and sourcing the required data. The type of dashboard could help in this matter. For example, if you intend to design an operational dashboard, you need real-time data, so the data refresh rate will be quite high. However, a strategic dashboard needs less frequent data refresh.

Examples of dashboard tools

The main rule of business intelligence software is to simplify complicated data and transform it into a form that can be used across all departments in the organization. The transformation process is highly impacted by the business intelligence tool being used as well as by user capabilities. There are many available software tools to help you develop professional dashboards, such as Tableau, Zoho Reports, Segment, and Sisense. Tableau is one of the most powerful professional business intelligence tools for dashboarding. The next section discusses the reasons for choosing Tableau.

Why Tableau?

Tableau has been trusted by more than 50,000 users around the world for many reasons. We will work with Tableau Public as our product of choice in the next practical section. To choose the most appropriate product for your needs, you should consider the following features:

Connectivity the ability of the product to connect to different data sources

Distribution the limitations of the product on sharing dashboards with others

Automation the ability of the product to update the dashboard based on refresh rates of the data sources

Security the level of security (on- or off-premise) that the product provides

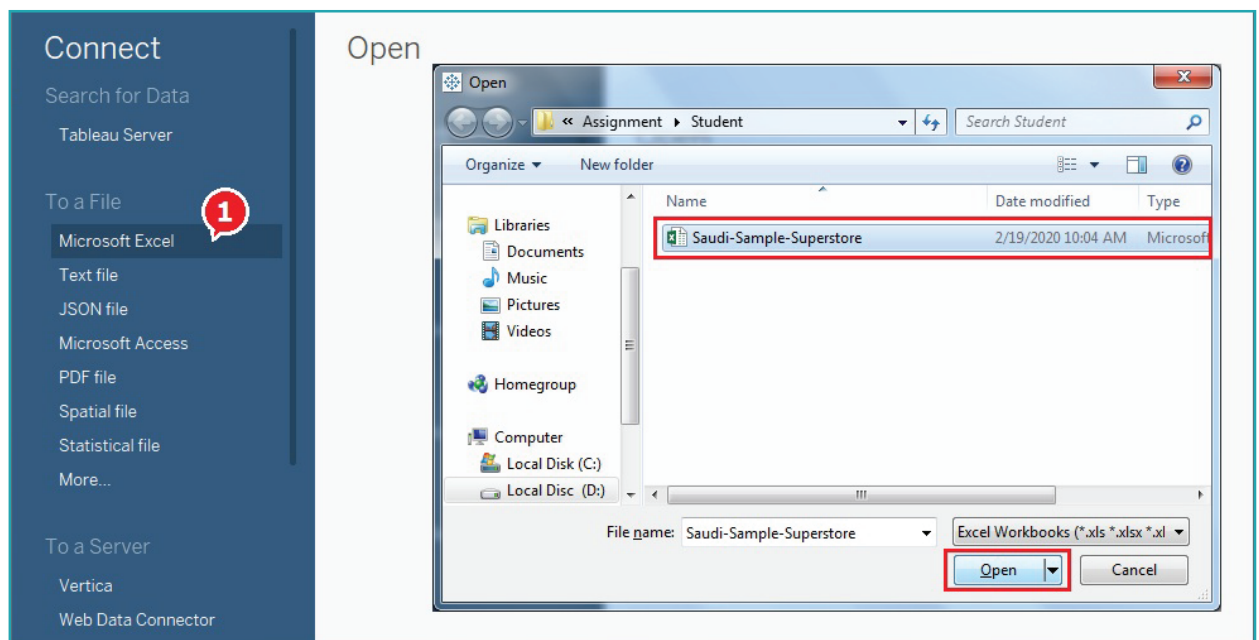
How to create a dashboard using Tableau?

To start the Tableau Desktop App, click on Tableau Icon from desktop or search and locate the app in the app list from start menu.

Connecting to Source Data in Tableau

On opening Tableau, you will get the start page showing various data sources. On the left, is the data pane showing the various data connection options. These options allow you to import data into Tableau.

1- Click on Microsoft Excel, choose Excel file “Saudi-Superstore.xls”.



2- The excel file has three sheets named **Orders**, **People** and **Returns**. Drag the sheet “Orders” to the upper empty space of the screen.

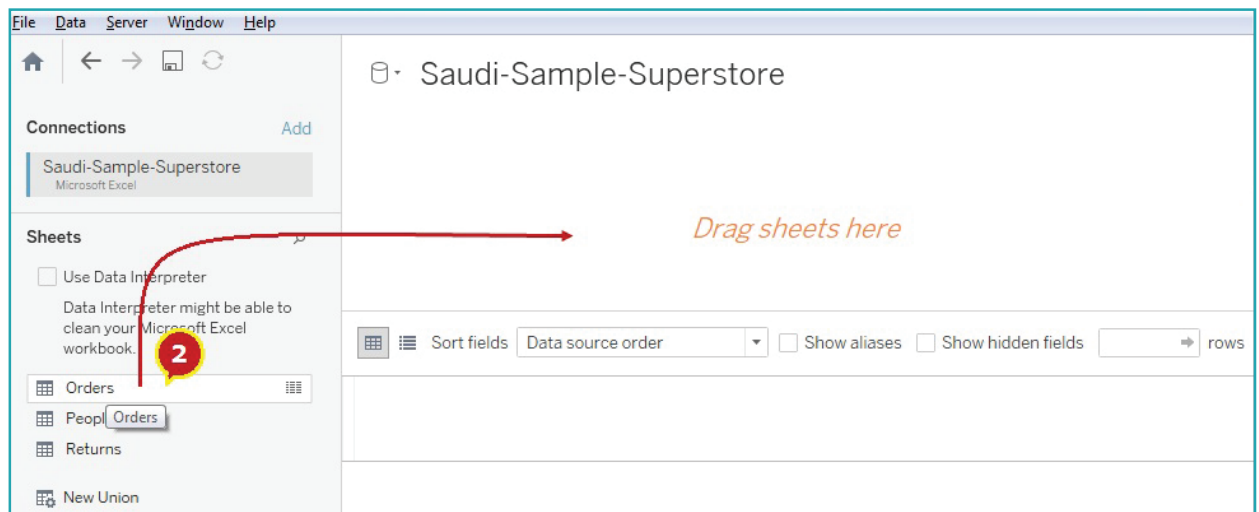
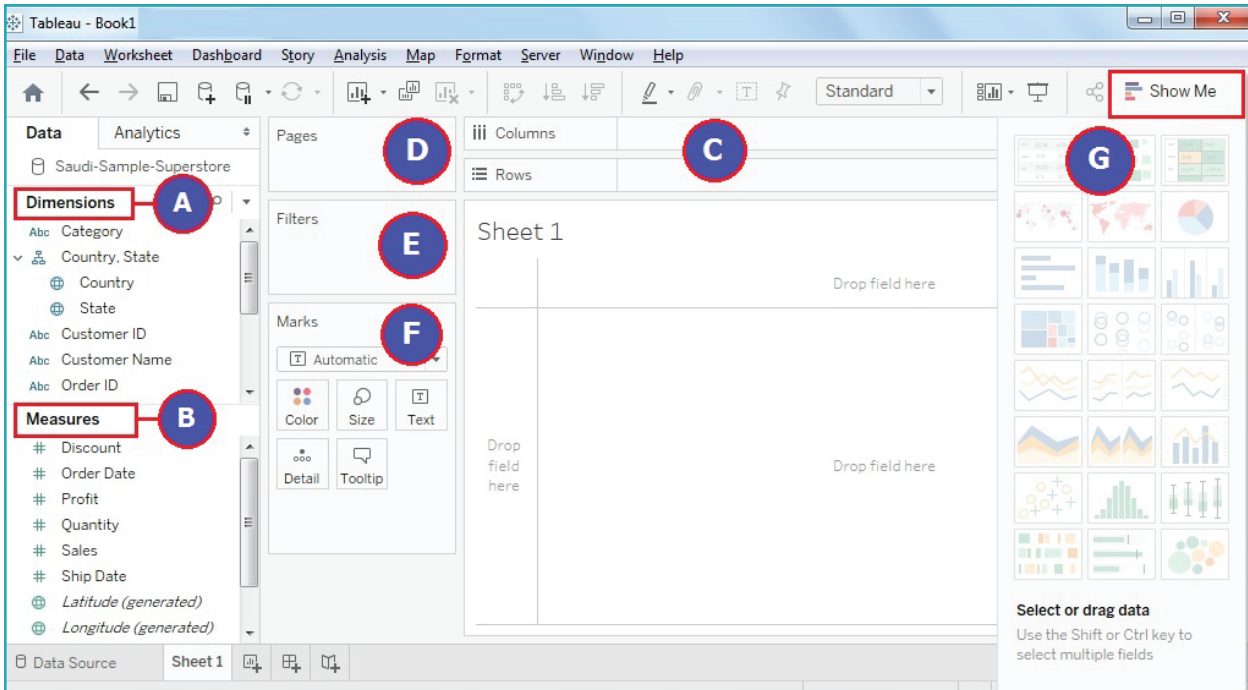


Tableau Interface

Before we go any further, let us learn some user interface elements in the Tableau screen.



A Dimensions

Dimensions are fields that represent the headings of rows or columns. These fields mostly contain a string of discrete values, which cannot be aggregated. An example of a dimension is city name.

B Measures

Unlike dimensions, measures represent fields that contain numerical or continuous values. Thus, you can aggregate and measure the values. Example of measures are sales or profits.

C Rows and Columns shelf

If you intend to draw a chart in Tableau, the rows and columns shelf represents your x and y chart axes.

D Pages shelf

The pages shelf allows you to see dynamic changes in the data by shifting from page to page to represent changes in data values between pages.

E Filters shelf

The filters shelf allows you to see the part of the data that you are interested in. For example, if you are interested in the profit of each branch rather than the total profit of the whole company, you would use this shelf.

F Marks shelf

The marks shelf allows you to annotate your data. For example, you may choose to represent your data using different shapes or sizes.

G Show me

Show me is one of the most important features in Tableau. Once you drag and drop your dimensions and measures, Tableau will create a default graph according to the number and type of dimensions and measures. However, you can change the graph type using the Show me pane.

Create a Map Chart

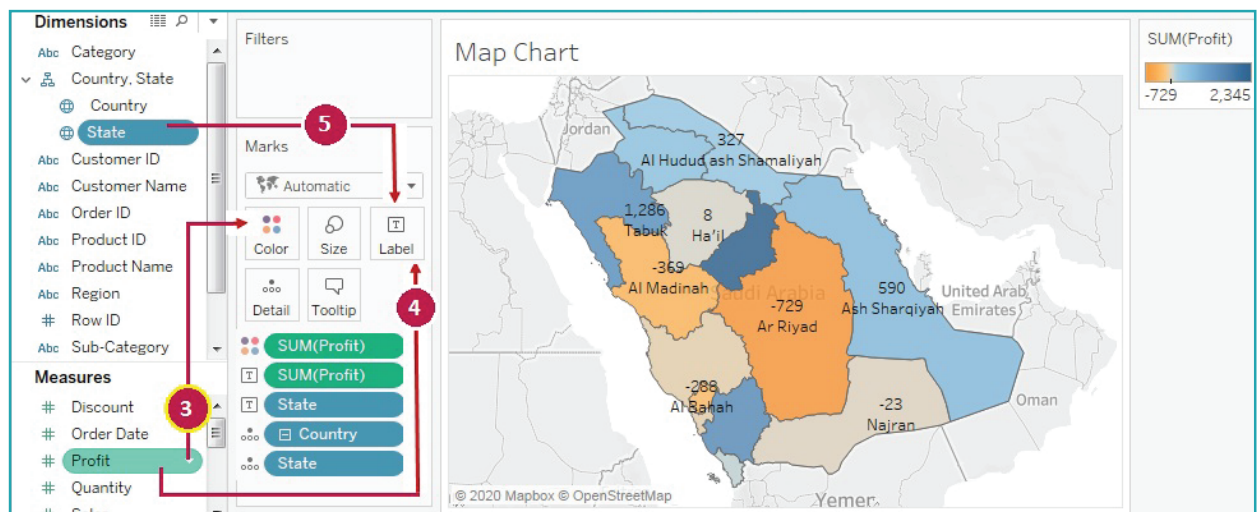
- 1 - Double click on "Sheet 1" in the section "Go to Worksheet" at the bottom left of the screen. Rename as Map Chart.
- 2 - Double Click on the Dimension "State".
- 3 - Drag and drop Measure "Profit" on to "Color" icon
- 4 - Drag and drop Measure "Profit" on to "Label" icon
- 5 - Drag and drop Dimension "State" on to "Label" icon

Note

If Tableau does not recognize the location of the States on the map, you can fix the **unknown** locations by clicking on the "unknown" icon in the bottom right corner of the screen and choosing the first option, "Edit Locations": Change name of the country to "Saudi Arabia" in the dialog box, then click OK:

The screenshot shows the Tableau interface with the following configuration:

- Dimensions:** Country, State
- Measures:** Profit
- Marks:** Automatic
- Map Area:** Empty with "Drop field here" text.

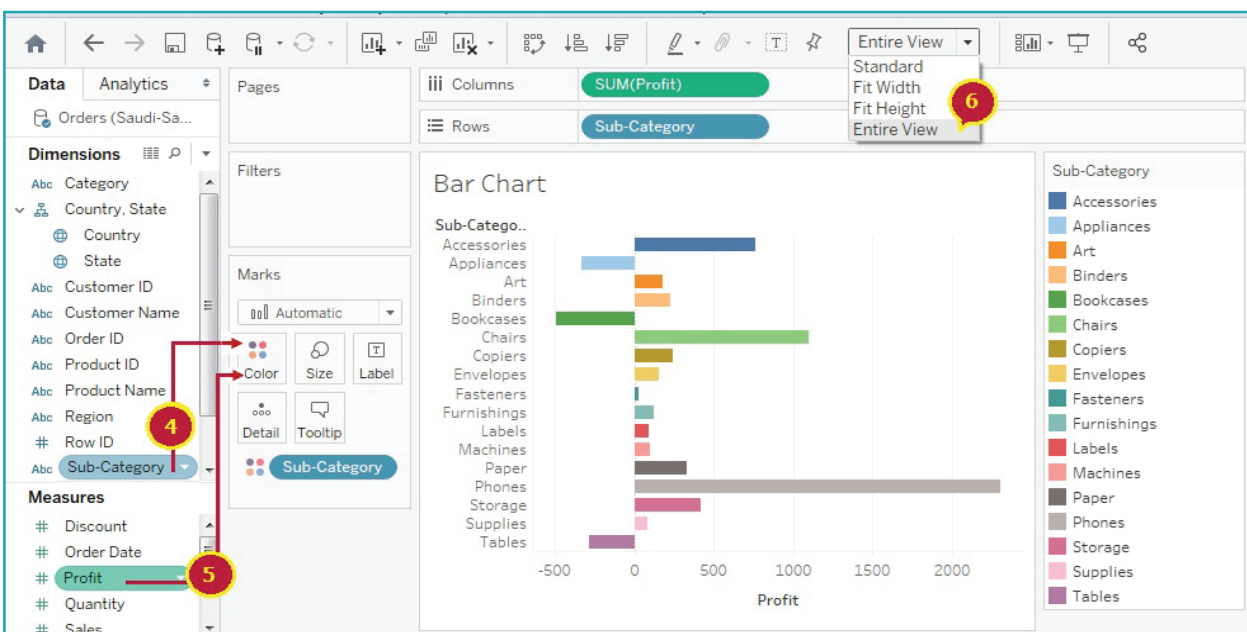
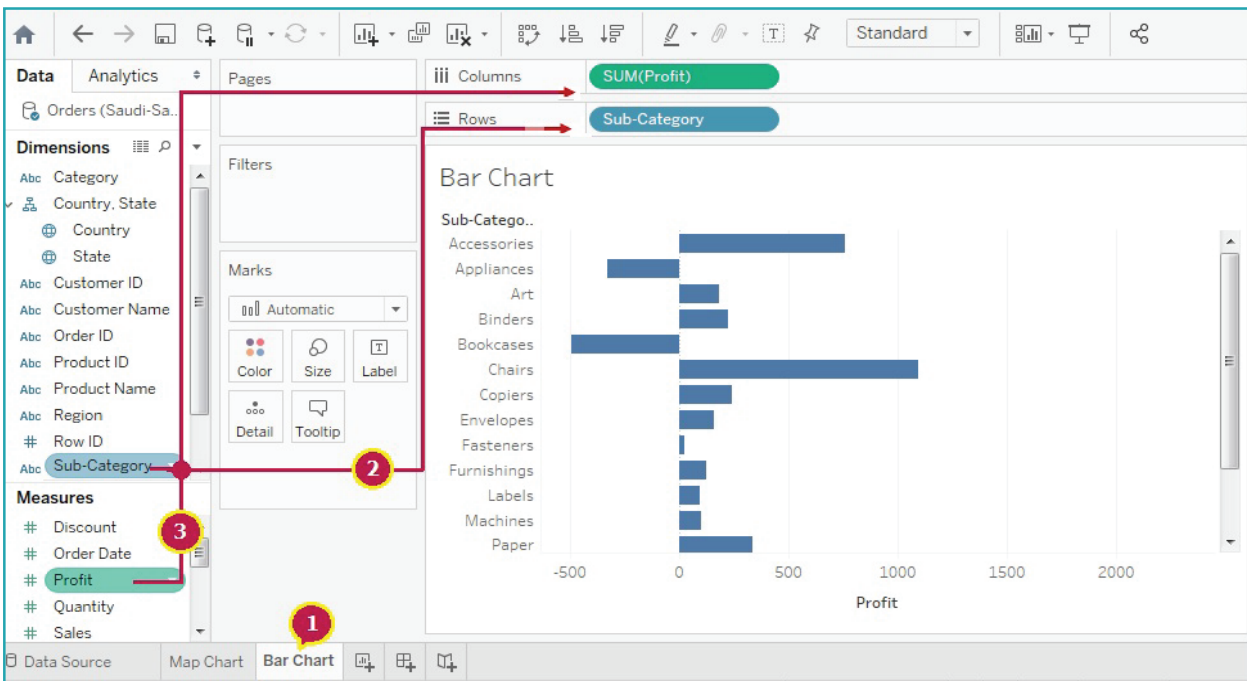


Create a Bar Chart

- 1 - Click on "New Worksheet button" and rename the sheet as "Bar Chart"
- 2 - Drag and drop Dimension "Sub-Category" on to Rows shelf
- 3 - Drag and drop Measure "Profit" on to Columns shelf.

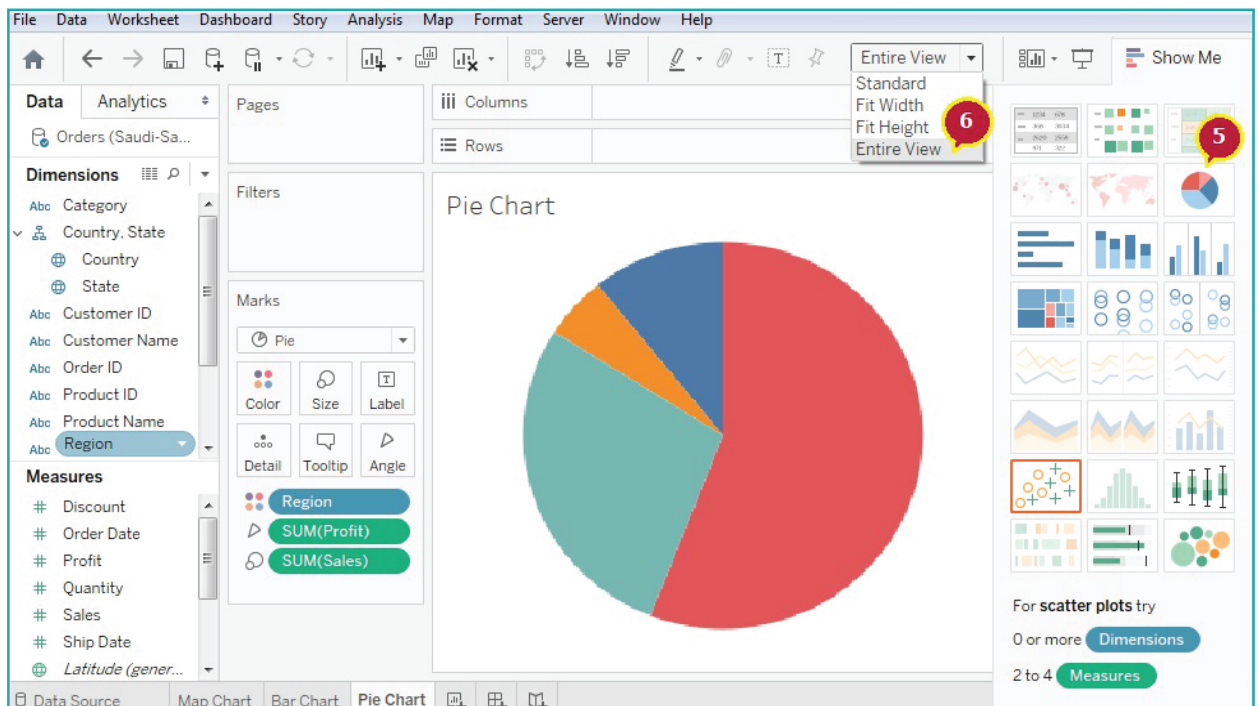
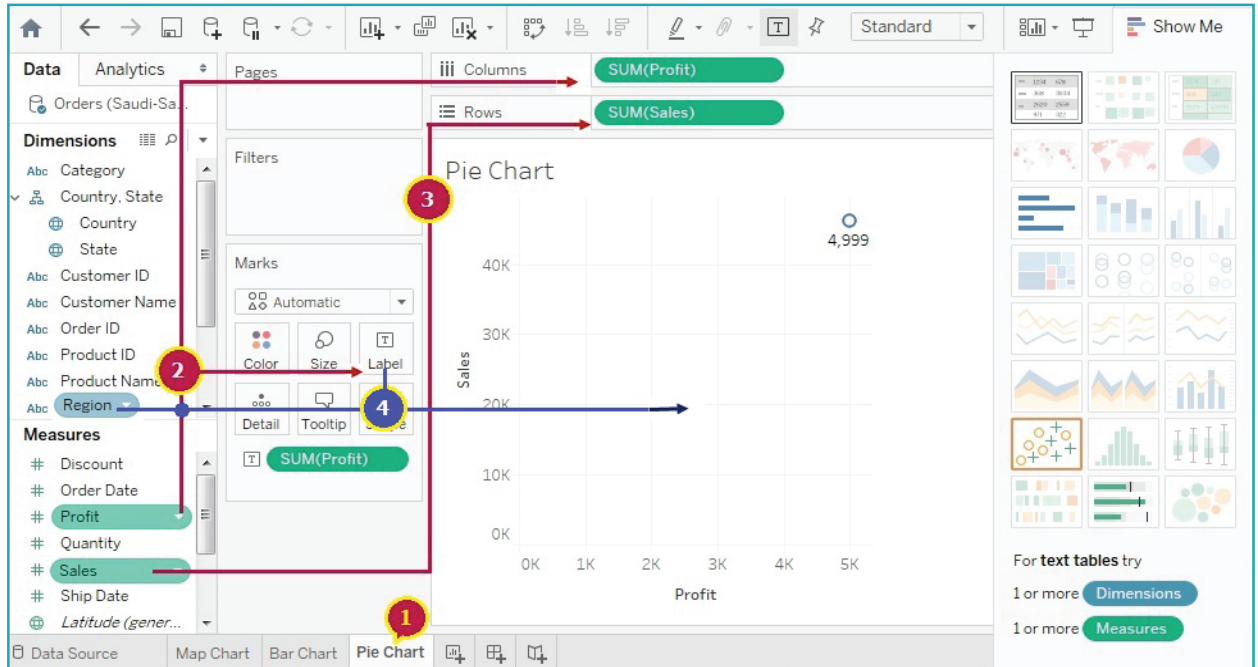
Note To add some customization to the bar chart (Optional)

- 4 - Drag and drop Dimension "Sub-Category" to the Color icon in the Mark shelf.
- 5 - Drag and drop Measure "Profit" to the Color icon in the Mark shelf.
- 6 - To fully visualize the chart, Change the "Standard View" to "Entire View" on the ribbon.



Create a Pie Chart

- 1 - Click on **New Worksheet** button and rename it to "Pie chart."
- 2 - Drag and drop Measure "Profit" onto **Columns** shelf. Again drag and drop Measure "Profit" to **Label** icon.
- 3 - Drag and drop Measure "Sales" onto **Rows** shelf.
- 4 - Drag and drop Dimension "Region" to **central pane**. Again Drag and drop Dimension "Region" to **Label** icon.
- 5 - Select the **Pie Chart** from **Show Me** pane. The pie chart is created.
- 6 - To fully visualize the chart, Change the "Standard View" to "Entire View" on the ribbon.



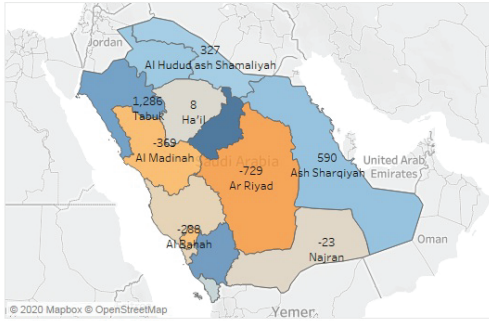
Create and Display Dashboard

- 1 - Click on the "Dashboard" tab at the top of screen and Click on the "New Dashboard" option.
- 2 - Drag and drop the sheets "Map Chart", "Bar Chart" and "Pie Chart" to the central pane
- 3 - Click on the "Presentation Mode" icon to show the dashboard presentation.
- 4 - At the bottom right corner of the screen, use "Previous Sheet/Next Sheet" buttons to navigate between charts.

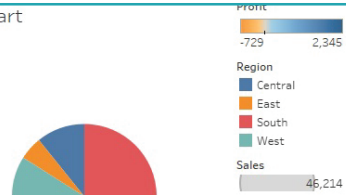
The screenshot shows the software interface with the 'Dashboard' tab selected. A dropdown menu is open, showing options like 'New Dashboard', 'Device Layouts', 'Show Grid', 'Grid Options...', 'Format', 'Copy Image', 'Export Image...', 'Clear', 'Show Title', 'Actions...', 'Auto Update', 'Run Update', 'Add Phone Layouts to Existing Dashboards', and 'Add Phone Layouts to New Dashboards'. A pie chart is visible in the central pane.

The screenshot shows the software interface with a dashboard layout. The dashboard includes a map of Saudi Arabia, a pie chart, and a bar chart. The map is labeled with regions and their sales values: Tabuk (1,286), Ha'il (8), Al Madinah (-369), Al Baha (-288), Ar Riyadh (-729), Ash Sharqiyah (590), and Najran (-23). The pie chart is color-coded by region: Central (blue), East (orange), South (red), and West (green). The bar chart shows sales for various sub-categories, with 'Phones' having the highest sales at 46,214.

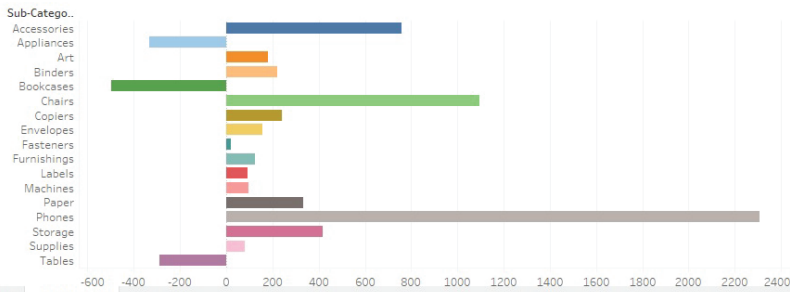
Map Chart



Pie Chart



Bar Chart



Previous Sheet [-]



READING SECTION

Accelerate Your Learning & Gain The skills

The table below compares the different Tableau products against the features of connectivity, distribution, automation, and security. This table is adopted from the book Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau Zen Master by Ryan Sleeper, 2018.

Tableau products	Features			
	Connectivity	Distribution	Automation	Security
Desktop (Personal)	<ul style="list-style-type: none"> - Excel - Text files - Access Statistical files - Shape files - Spatial files - Tableau files 	<ul style="list-style-type: none"> - Offline or - Tableau Public 	Not available	Depends on your computer/ server security
Desktop (Professional)	All possible connections in Tableau	All possible distribution options in Tableau	Not available	Depends on your computer/ server security
Tableau Reader	.twbx files only	Offline	Not available	Depends on your computer/ server security
Tableau Public	<ul style="list-style-type: none"> - Excel - Text files 	Cloud (public)	Not available	Limited; you have control over downloading your file; however, anyone on the web can see it
Tableau Online	Workbooks that have been published to Tableau Online	Cloud	Available via data refresh schedules	Depends on Tableau's third-party host
Tableau Server	Workbooks that have been published to Tableau Server	On-premise or in the cloud	Available via data refresh schedules	Depends on your on-premise or server host's security



Apply

Activity 1

Visit the following website for more online training and videos:
<https://www.tableau.com/learn>

Activity 2

Fill in the blanks with the appropriate words provided below.
(strategic , analytical , operational , KPI)

- a) A _____ is a measurable value that indicates a company's progress towards its goals and how well that company is achieving its key business objectives.
- b) The _____ dashboard is a reporting tool that tracks business metrics and KPIs.
- c) The _____ dashboard is a business intelligence tool that tracks the performance of the business by measuring different KPIs.
- d) The _____ dashboard needs accurate and up-to-date data to provide accurate predictions.

Activity 3

Choose TRUE or FALSE for the following statements.

- | | |
|---|---------------------|
| a) Dashboards can easily identify which parts of the business require immediate attention. | a) True
b) False |
| b) The operational dashboard quickly and frequently changes its data; it is updated from one minute to the next. | a) True
b) False |
| c) The strategic dashboard is a business reporting tool; it tracks the performance of the business by measuring different KPIs. | a) True
b) False |
| d) Dashboards can display large amounts of data which are readable at a glance. | a) True
b) False |



Closing

In this session, we have learned:

- What a dashboard is.
- Types of dashboards.
- Advantages of dashboards.
- Designing dashboards.
- How to create a dashboard using Tableau.

CHAPTER

4

Session 1

Introduction to Algorithms



Goal

In this session, you will learn

about the significance and development of Algorithms and Flowcharts.



Learning objectives

Dear Student

By the end of this session, you should be able to:

Describe the concept and importance of computer algorithms.

Design algorithmic solutions to a range of computational problems from various areas.

List the advantages and usage of flowcharts in developing computer programs.



List the advantages of using algorithms in computer programming.

Describe how computers use algorithms to perform the tasks we expect them to do.

Transform algorithms into flowcharts to understand the flow of information.



Be prepared

You know that Computer is a dumb machine. It works only based on your instructions and cannot do anything on its own. Although it is an accurate machine, if you give it a wrong instruction, it will generate a wrong result. So you need to give it correct and step wise instructions to make it work properly and accurately.

However, with advancement of software engineering and business intelligence application we are able to solve complicated problems using computers.

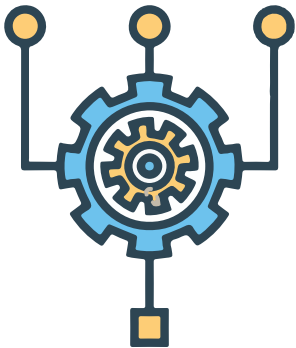
There are big companies worth billions of dollars, like Uber for example, are based on computer application that can be accessed via website and mobile application.





Learn

Algorithms



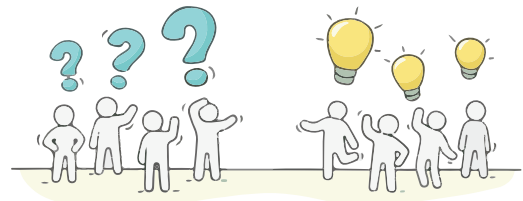
Modern computers can carry out complex tasks. Each task consists of well-defined procedures known as algorithms. **An algorithm is a well-defined procedure that helps us to solve a problem.**

Another way to describe an algorithm is a sequence of unambiguous instructions. The use of the term 'unambiguous' indicates that there is no room for subjective interpretation. Every time you ask your computer to carry out the same algorithm, it will do it in exactly the same manner with exact same result.

You need to write programs in a computer language to solve a problem by a computer. Programs are a set of instructions. Algorithms are written in a step by step procedure before writing the actual computer program. So Algorithms are an important part of program development. In fact, it is difficult to think of a task performed by your computer that does not use algorithms.

Advantages of Algorithms

Algorithms are very useful in solving problems. You can use them to understand the nature of problems in your daily life. Algorithms helps to design the best solutions to your problems



Following are some of the advantages of using Algorithms.



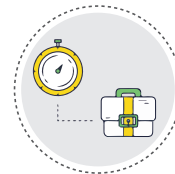
They are easy to understand.



They are easy to implement.



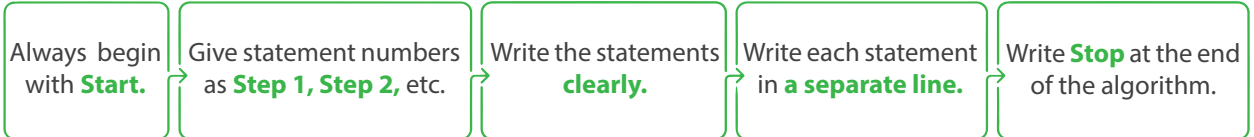
They are easy to modify.



They are not dependent on any particular programming language.

Rules for writing Algorithms

There are some guidelines for writing an algorithm. These guidelines or rules make algorithms self-explanatory. You should follow the given rules while writing an algorithm.



How Algorithms Work?

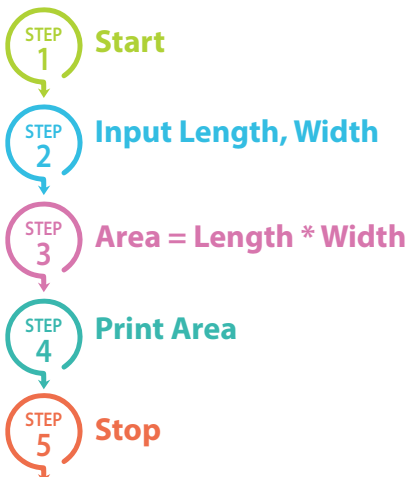


Let's take a closer look at an example. A very simple example of an algorithm would be to calculate the area of a rectangle. A rectangle can be defined as the plain figure with two adjacent sides equal in length. Area of a rectangle is the amount of space occupied by the **rectangle**. **The formula** to calculate the Area of Rectangle is

$$\text{Area} = \text{Length} * \text{Width}$$

Below is what the algorithm could look like. The input will be the length and width of the rectangle. Use the above formula to calculate the Area.

Algorithm to calculate the area of a rectangle



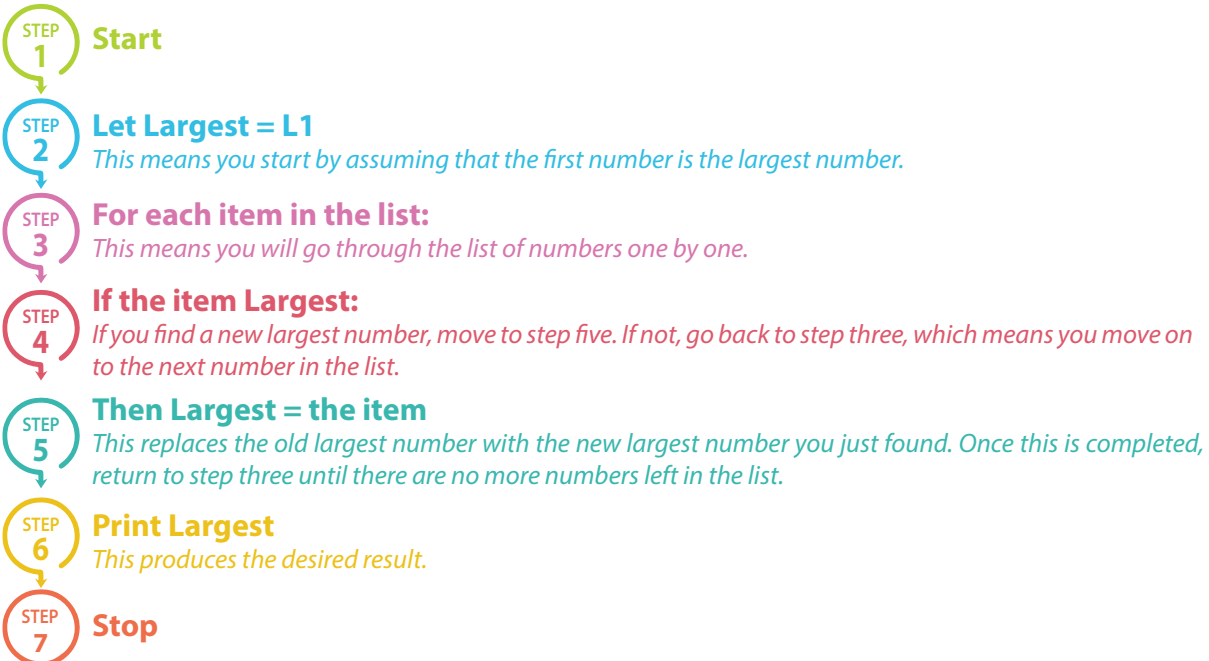
Note

the algorithm is described as a series of logical steps in an English-like language that is easily understood. For a computer to actually use these instructions, they need to be written in a language that a computer can understand, known as a programming language.

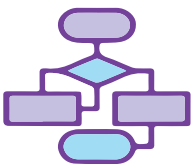
Go through the following example to have a better understanding of algorithms.

Let us look at another example to find the largest number in an unsorted list of numbers. If you were given a list of five different numbers, you could easily figure it out in no time. However, if the list consists of five million different numbers, you definitely need a computer to do this, and a computer needs an algorithm. Below is what the algorithm could look like. Let's say the input consists of a list of numbers, and this list is called L. The number L1 would be the first number in the list, L2 the second number, etc. The list is not sorted. So, the input to the algorithm is a list of numbers, and the output should be the largest number in the list.

Algorithm to find the largest number in an unsorted list of numbers.



Flowcharts



You know how to write the steps of an algorithm to solve a problem. If you put all these step-wise instructions into some predefined symbols, you will get a Flowchart. A Flowchart is a graphical representation of an algorithm. It is a step-by-step process to solve a problem in graphical form. The purpose of making a flowchart is to assist in writing a program. The flowchart lists all the steps to be considered in a program. The steps must be in the correct logical order.

Advantages of Flowcharts

Since the flowchart is the graphical representation of an algorithm, it gives a better understanding of the solution to a problem than words alone. Following are the benefits of using flowcharts in program development.

- 1 A flowchart is independent of the programming language.
- 2 It establishes better communication with the person who is new to computer terminology.
- 3 It is always better to find errors from flowcharts rather than from the programs. Fixing errors at the flowchart stage is much easier and saves your time and effort.
- 4 It is easier to understand the object and logic of the program with the help of the flowcharts.

Standard Symbols of Flowcharts

To design a flowchart, you need several flowchart symbols of different shapes. Each symbol has its own purpose. To connect these symbols, we use lines known as the **flow lines**.

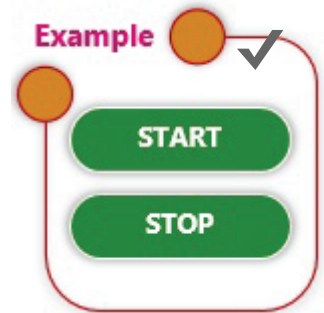
Let us learn about some symbols used to draw a flowchart.



Start / Stop Symbol

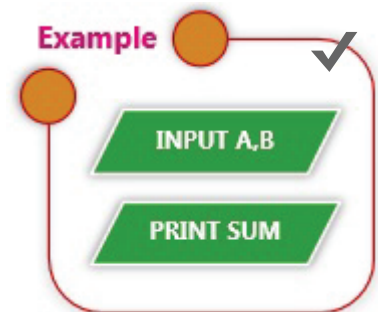
The Start / Stop box is **oval** in shape and is also called the **Terminal box**. The box used in the beginning of the flowchart is called the **Start box**, and the box used at the end of the flowchart is called the **Stop box**.

Start and **Stop** are written inside the boxes.



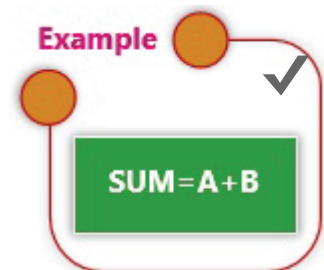
Input / Output Box

A **parallelogram** is used to represent the **Input** or **Read** step, where the data is brought in, and the **Output** or **Print** step, where the results of a problem are shown.



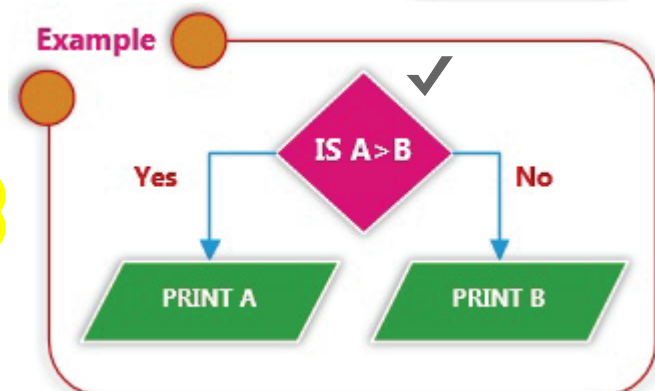
Process Box

A **rectangle** is used to represent the **Process box**. The steps of processing are written inside the box. Whenever you do any **calculation** or **assignment operation**, this symbol is used.



Decision Box

A **diamond-shaped** symbol is used to represent the **Decision box**. It represents a **decision** as well as a **comparison operation**.



For example, the flow chart represents
IF A>B THEN PRINT A ELSE PRINT B



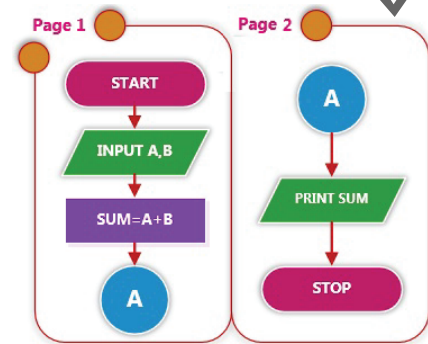
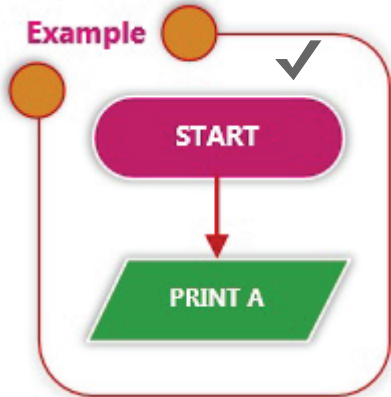
Flow Lines

Lines with arrowheads are used to connect various symbols. These arrows connect the boxes in a flowchart to represent the direction of the flow of data and information.



Connectors

Sometimes the flowcharts are complex or spread over more than one page. This requires connectors to connect two different points of the flowchart to show the flow of main process. A connector is a circle shaped flowchart symbol.

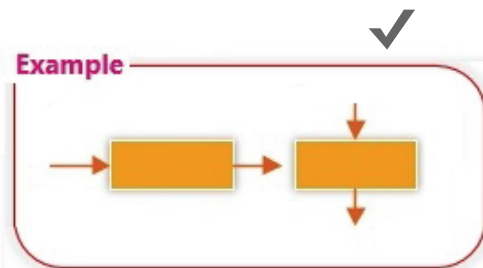


Guidelines for Drawing Flowcharts

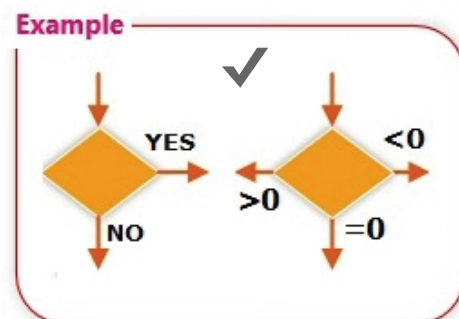
Following are some of the guidelines to draw standard flowcharts.

- (a) In drawing a proper flowchart, all necessary requirements should be listed out in logical order.
- (b) The flowchart should be clear, neat, and easy to follow. There should not be any room for ambiguity in understanding the flowchart.
- (c) The usual direction of the flow of a procedure or system is from left to right or top to bottom.

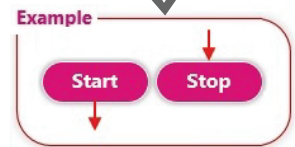
(d) Only one flow line should come out from a process symbol.



(e) Only one flow line should enter a decision symbol, but multiple lines can leave it to denote possible outcomes.



(f) Only one flow line is used in conjunction with a **terminal symbol**.



(g) Write briefly within standard symbols. As necessary, you can use the **annotation symbol** to describe data or computational steps more clearly.

This is to calculate the parameter

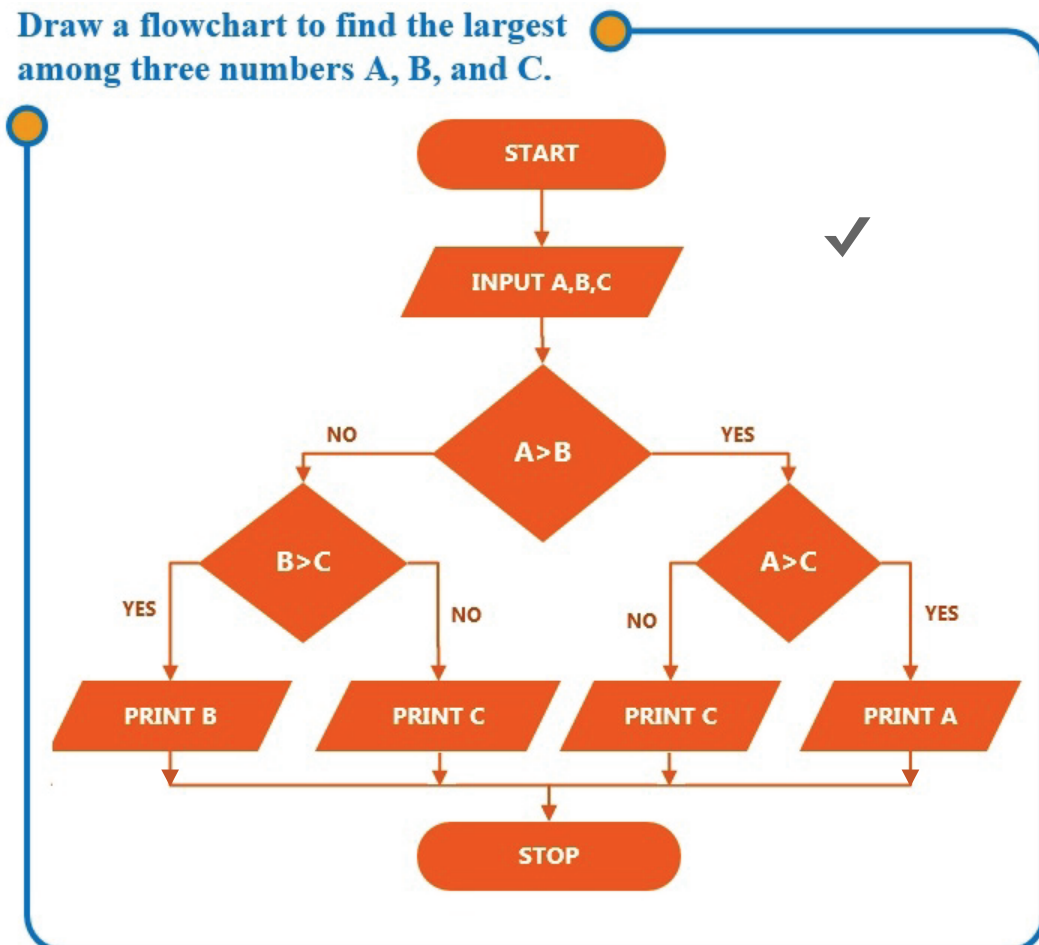
(h) If the flowchart becomes complex, it is better to use **connector** symbols to reduce the number of flow lines. Avoid the intersection of flow lines if you want to make it a more effective method of communication.

(i) Ensure that the flowchart has a **START** and a **STOP**.

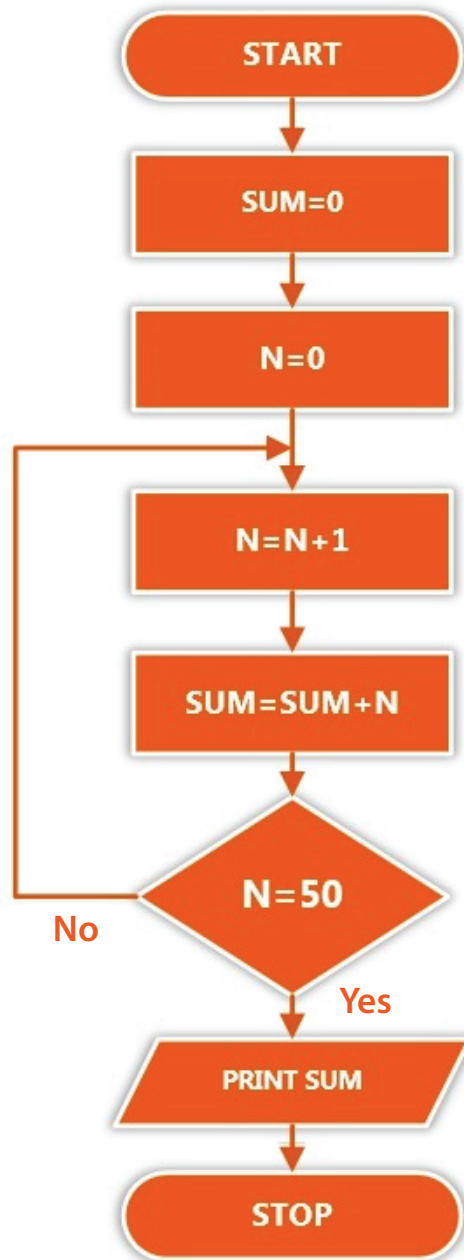
(j) It is useful to test the validity of the flowchart by passing through it with simple test data.

Let's look at few examples.

Draw a flowchart to find the largest among three numbers A, B, and C.



Draw a flowchart to find the sum of the first 50 natural numbers.





Apply

Fill in the blanks.

Flowchart, Algorithms, Diamond, Programming Language

Activity 1

- (a) _____ are written in a step-by-step procedure before writing the actual computer program.
- (b) A _____ is a graphical representation of an algorithm. It is a step-by-step process to solve a problem in graphical or pictorial form.
- (c) A flowchart is independent of the _____.
- (d) A _____ shaped symbol is used to represent the decision box.

Match the following.

Activity 2

- | | |
|---------------------|---|
| (a) A flowchart | is used for input and output steps. |
| (b) An oval | is used to connect two different parts of a flowchart. |
| (c) A parallelogram | is a graphical representation of steps for solution of a problem. |
| (d) A diamond | is used to represent the start step. |
| (e) A circle | is used for a decision-making step. |

Arrange the steps of the algorithm to add two numbers A and B.

Activity 3

- | | |
|--------------------|--------------|
| (a) Stop | Step 1. |
| (b) Print Result | Step 2. |
| (c) Start | Step 3. |
| (d) Read A, B | Step 4. |
| (e) Result = A + B | Step 5. |

Practice the following tasks in your notebook.

Activity 4

1. Write an algorithm to calculate the average of three numbers.
2. Write an algorithm to find the smallest one among five numbers.
3. Prepare a chart of the algorithms to subtract, multiply, and divide two numbers.
4. Draw a flowchart to find the sum of N given numbers.
5. Draw a flowchart to find the sum of the grades of Science and Mathematics for a class of 20 students.



Closing

Dear student, you have just learned the logic of solving problems using algorithms. Although the examples we covered in this session are basic, the concept is the same even for much more complicated scenarios. If you would like to build your own computer application, then you have to practice solving problems using algorithms, then use your flowchart as a guide in the programming phase and write your code with the programming language that you like.

CHAPTER

4

Session

2

Introduction to Programming



Goal

In this session, you will learn

The basic concepts of computer programming using the Python programming language. You will be also familiar with the two main windows that Python uses: the Shell window and the Code window.



Learning objectives

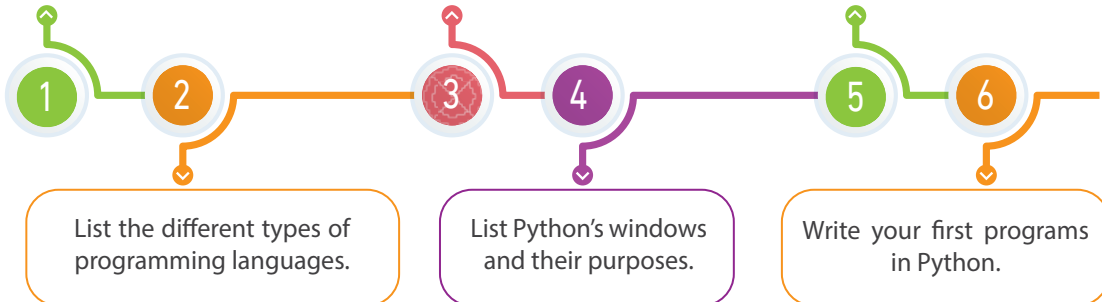
Dear Student

By the end of this session, you should be able to:

Describe Computer programs and Programming.

Differentiate between compilers and interpreters.

List the features that Shell window and the Code window provide.



Be prepared

Every time we use smart devices, there are programs running in the background. Moving a mouse pointer from one part of your computer screen to another may seem like a simple task, but in reality, a computer program just ran. Another simple task like typing letters into Google Docs leads to computer programs being executed in the background. It is all programming everywhere.

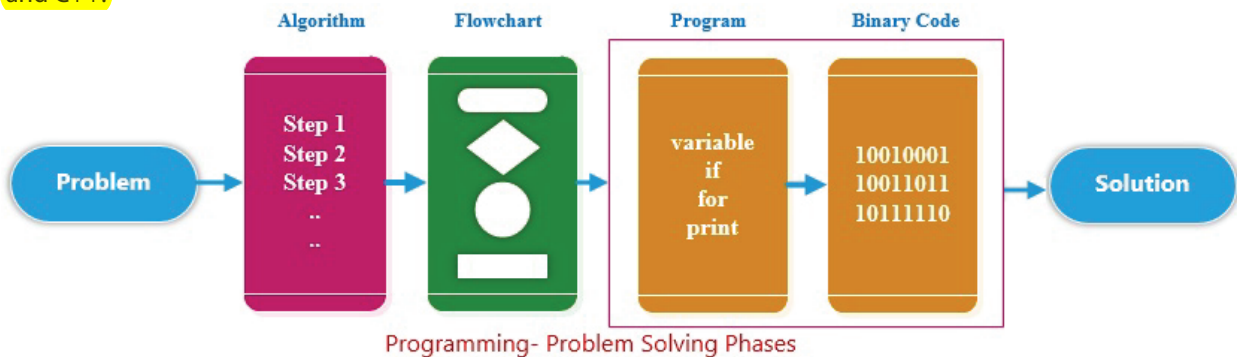




Learn

Programming

Programming is the process of creating a set of instructions that tell a computer how to perform a task. It is often referred to as coding. Programming can be done using a variety of programming languages, such as Java, Python, and C++.



What is a program?



A computer program is a sequence of instructions executed by a computer. Every task of a computer is done using a computer program. We write a program using a programming language. Computer programs are also referred to as code.

Following are some examples of computer softwares, consisting of a group of programs.

- Operating systems like Windows and Android.
- Web browsers like Mozilla Firefox and Apple Safari.
- Office suites like Microsoft Office.
- Video games.



A computer program is stored as a file on the computer's hard drive. When a user runs the program, the file is read by the computer, and the processor reads the data in the file as a list of instructions. Then the computer does what the program tells it to do.

What is a Programming Language?

Machines have their own natural languages, just as humans do. Computers do not understand the human languages. **The natural language of computers is the binary code: 1 and 0.**

These represent two states: on (1) and off (0). That is the natural language of electronic devices. It would be complicated for us as humans to communicate with the computers in binary.

To communicate with machines that understand binary, programming languages were invented. We are now able to communicate with machines using **programming languages** that are close to our own natural languages, such as Arabic or English, but they are more structured and have their own rules and notations.



Types of Programming Languages

Computer programming languages are basically classified into two types

High-Level Programming Languages

High-level programming languages provide strong abstraction and have a simple and easy syntax. Examples include Python, Java, C, C++, Ruby, Perl, Basic, JavaScript, and many more.

Low-level programming languages

Low-level programming languages provide little or no abstraction and have a very complex syntax. These languages mainly consist of numeric terms, while high-level programming languages provide syntax consisting of a complete alphanumeric syntax.

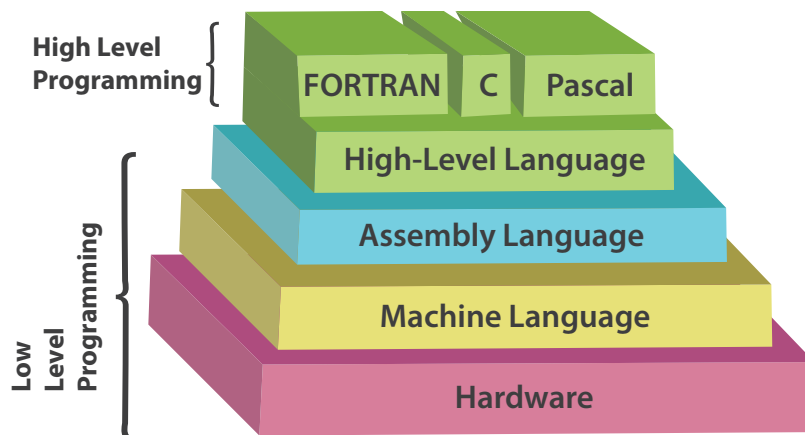
There are two main low-level programming languages

assembly language

Assembly language is a collection of mnemonic codes that specify the operations to be performed by the processor.

machine language

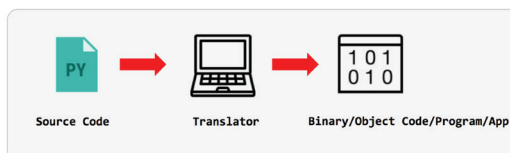
Machine language is a collection of binary digits (0's & 1's) that the computer reads and interprets. It is the language a computer is capable of understanding.



Translators

You know that computers understand only binary, the language of 0's and 1's. On the other hand, the high-level programming languages are closer to our natural languages. Hence, to communicate with computers by using these programming languages, we need special translators.

Translators have the responsibility of converting your source code to binary machine language. We may refer to the binaries as object code, and program as apps, which is a common word today.



Translators

Interpreters

Some languages are interpreted. The translator processes the source code line by line and runs every line in the program or app. This means that interpreted source code starts running until it encounters an error. Then the interpreter stops to report such errors.

Python is a good example of an interpreted programming language.

Compilers

Compilers function differently. They convert the entire source code into the object code via a compilation process. The object code is then executed. If there were errors in the source code, they are detected during the compilation stage and flagged. This interrupts the compilation process, and no object code is generated.

C and C++ are good examples of compiled programming languages.

To summarize

Interpreters translate line by line and execute each line before going on to the next line.

Compilers translate the entire program to a file (binary) and execute the whole file.

Getting Started with Python



There are several powerful languages that are easy to set up and learn, and Python is one of them. Python is an interpreted, interactive, object-oriented programming language. Object-oriented programming language is a programming language model organized around objects rather than "actions" and data rather than logic. Python's flexibility allows it to do many things, both big and small. Python can be used to write simple programs, but it also possesses the full power required to create complex, large-scale enterprise solutions.

Applications of Python



Desktop graphical application development, including games.



Mathematical and scientific analysis of data.



Web and internet development.

Python's applications in the world of computer programming are obvious.

For example, Python is used in some of the largest internet sites — like **Reddit, Dropbox, and YouTube**.

The popular **Python** web framework Django powers both **Instagram** and **Pinterest**. However, that doesn't mean it is complicated or hard to learn.

Source Code

Source code is basically a file, just like a Microsoft **Word (.docx)** file, but slightly different. It is a raw text file, written on very simple editors, like **the Windows Notepad**. You will recall from the previous section that you need either interpreters or compilers to convert your source code to binary. The source code must be saved in a file, which is passed as an input to the translator.



Depending on the language you are writing in, there are designated extensions for saving your source code file.

Python file extension is



Java file extension is



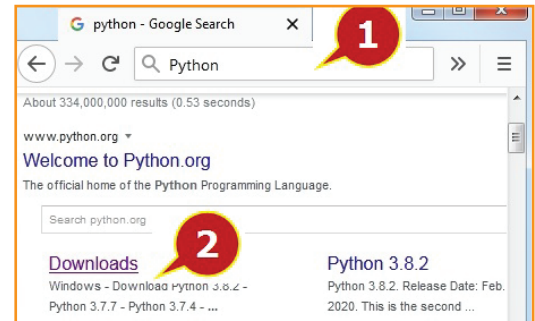
Php file extension is



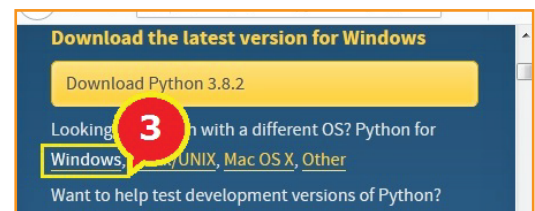
How to install Python on your computer

1 - Type "Python" in your browser and search the internet.

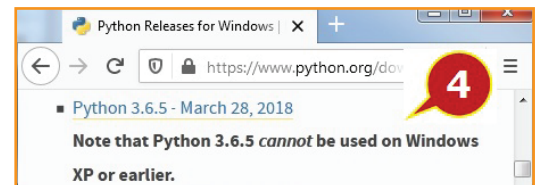
2 - Scroll down and click on Download Python from the official Python website (www.python.org).



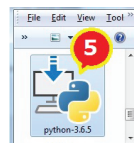
3 - Select the right operating system that you have (e.g. Windows or Mac).



4 - Now you should select the version of Python you are interested in or choose the latest version. In this book, we used version 3.6.5.



5 - Once the file is downloaded, double click on it and follow the wizard to install python.



Now python has been installed in your computer and you can start using it.

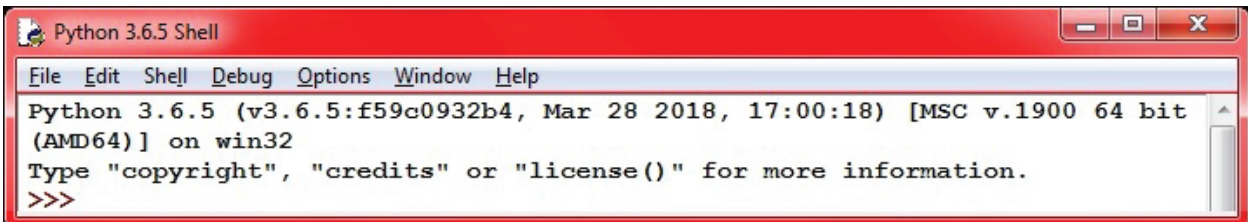
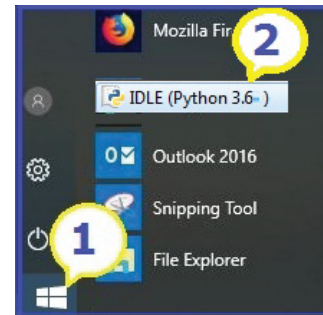
Starting the Python Shell

As you know, we use the Python interpreter shell to run the programs.

1. Click on the Start button to open the Start menu.
- 2- Click on IDLE (Python 3.6) from the All Apps list.

The interpreter shell of Python 3.6 appears.

IDLE stands for **Integrated Development and Learning Environment** which is a bundled set of software tools for program development.

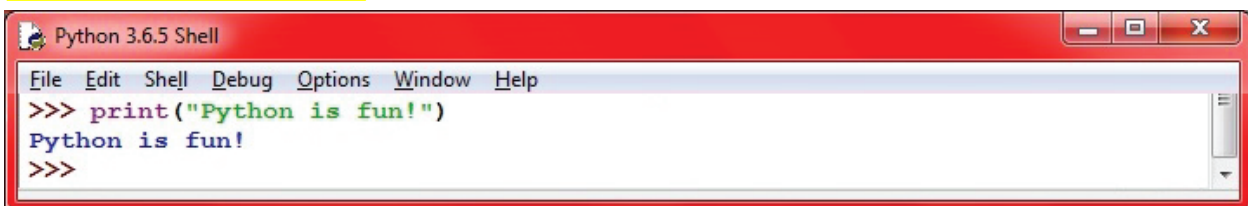


The Python shell provides the very useful ability to execute in interactive mode where you execute a single Python command and get the result. Interacting with the shell is much like using a calculator, except that, instead of being limited to the operations built into a calculator (addition, subtraction, etc.), For example, the expression $2 + 3$ is entered at the shell prompt (`>>>`), and the program immediately responds with the result 5.



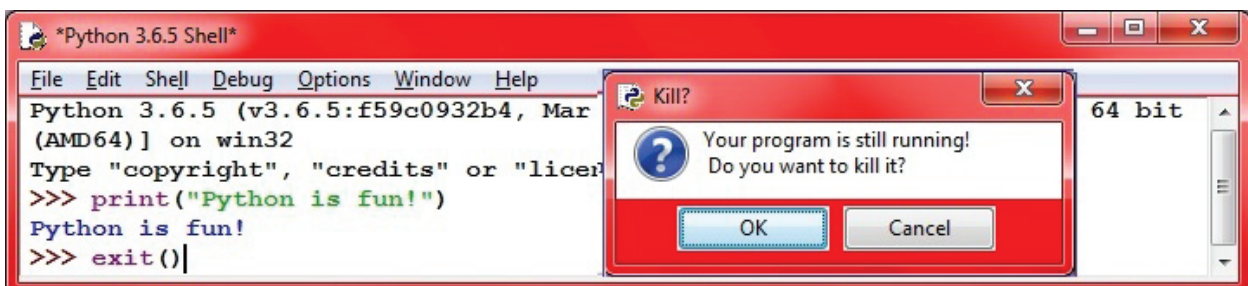
Type the code `print("Python is fun!")` in the shell. You should see the result as: Python is fun!

The Python's `print()` function is a built-in function used to print the given string as an output i.e. it prints the specified message to the screen.



And that's it. You've just made your first programs in Python. Each time you entered a line, Python immediately executed the statement.

Once done, you can use `exit()` or **Ctrl D** to leave the python shell. A dialog box appears. Click Ok to close the python shell



Shell Window



The Shell window opens automatically when you open IDLE (Python's Integrated Development and Learning Environment) or when you run a program from the Code window. The main purpose of it is to run codes. You may use it to write and test small pieces of code. However, you cannot save the code through the Shell window. Below is a screenshot of the main Shell window.

The first bar of the Shell window illustrates the name and version of Python. It is also printed in the first line of the Shell window, with more information about the version (inside the orange box).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

The three arrows, known as the command prompt, appear when the Python is ready for you to type your code. Once you write your code, press enter to run the code. The code runs immediately and the output appears in the next line unless there is an error (bug).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("I am learning Python in King Saud University')
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("I am learning Python in King Saud University"))?
>>> print("I am learning Python in King Saud University")
I am learning Python in King Saud University
>>>
```

As you can see in the screenshot above, we wrote the first command with a missing left bracket and pressed enter. The output was a SyntaxError, and we have been advised how to fix it. In the second command, we debugged the code and ran it again. The output was printed on the next line. You may have noticed that there are different colors of the code. **What do these different colors mean?**

Purple Means Built in Function

Green Means a String

Blue Means Function Definition or Output

Orange Means a Keyword

Red Means an Error

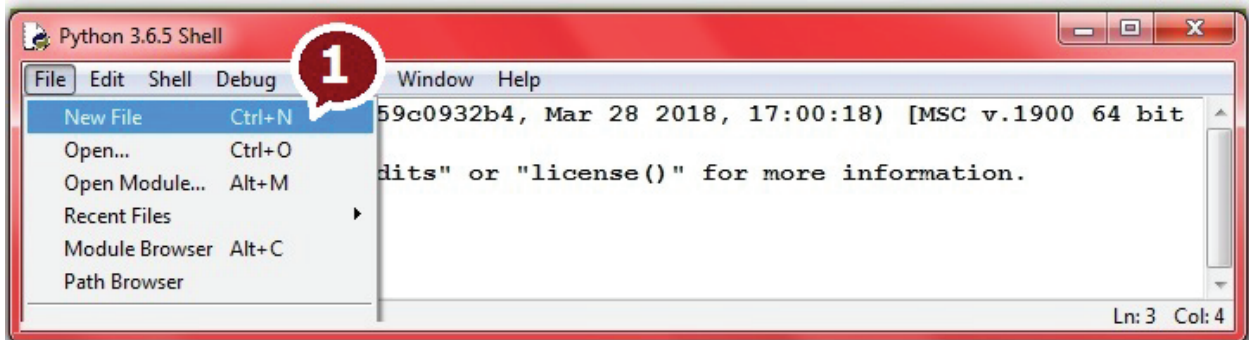
Black is For All Other Commands

Code Window

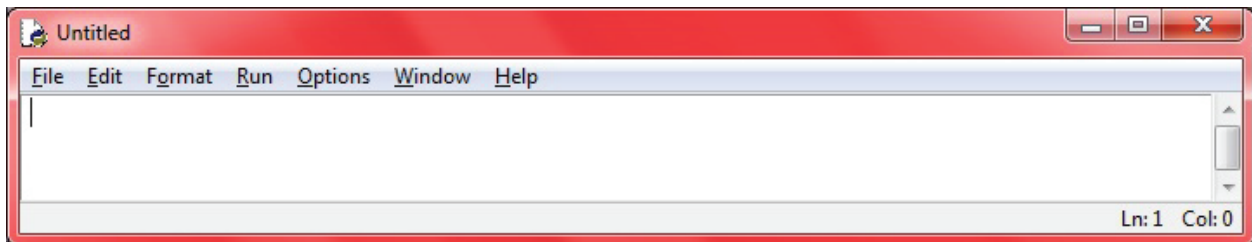


The code window is used to write your code, especially when you are writing a long program. It allows you to save and edit your code in a more practical way.

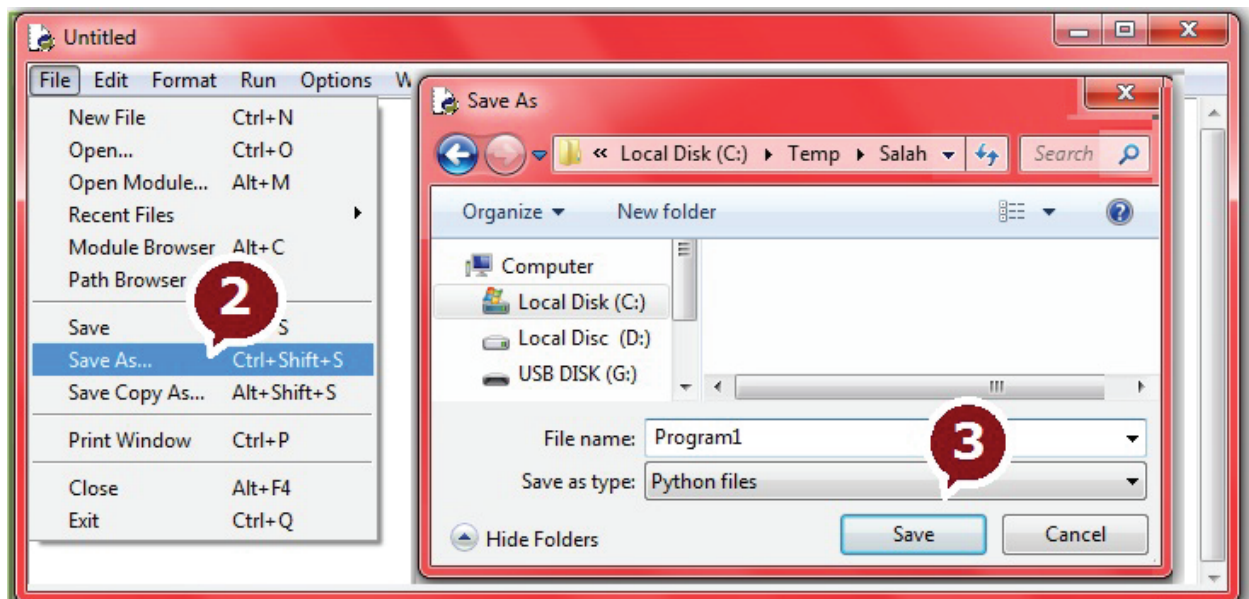
1 - To start a new code window, click on the File menu from the Shell window, then click on "New file".



2 - A new untitled window will appear. This window is ready for you to write your code in it. However, it is good practice to save the file before using it. Click on File then Save to save the file.

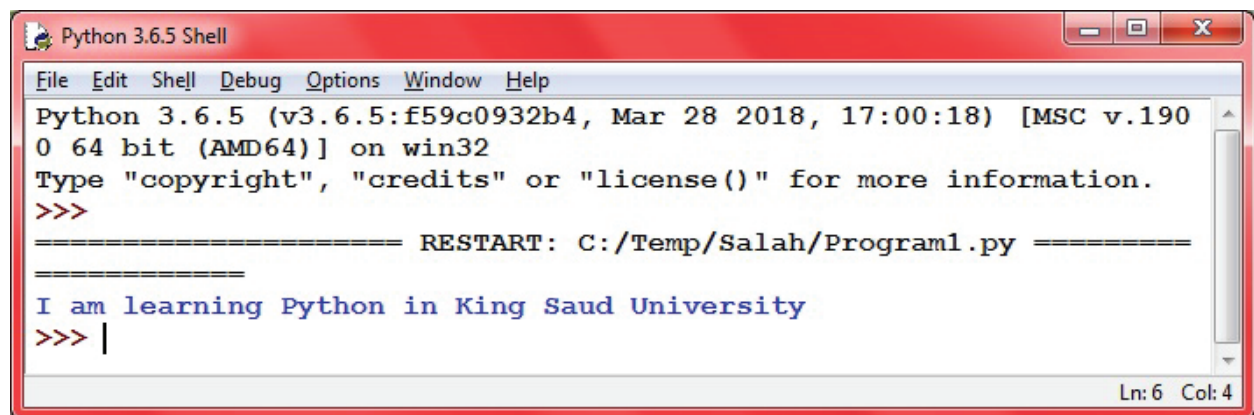
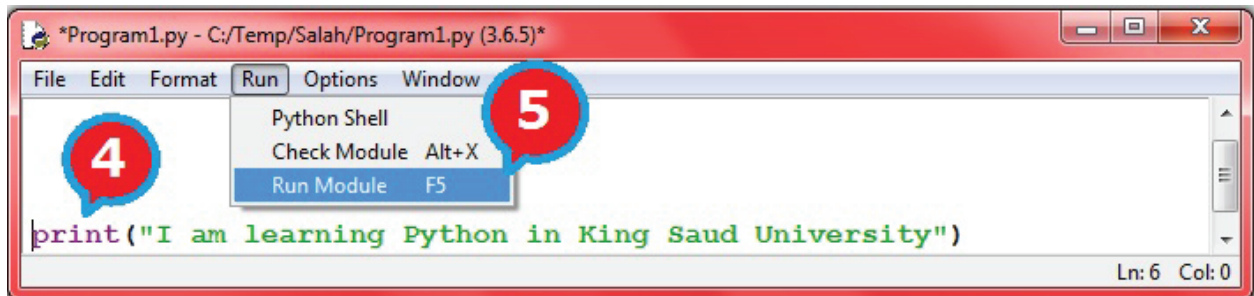


3 - The following window will appear. Select the location and folder that you would like to save your file in, then type the file name and save it as "Python Files".



4 - Now your file is saved and you may notice that the file name on the title bar changed from Untitled to the new name along with the new location that you selected. You can type your code now and save it frequently by clicking File then Save or (Ctrl + S) as a shortcut.

5 - Once you are finished writing your code, you can execute it by saving the file and then clicking on the Run menu, then on Run Module. Alternatively, you may **press F5 on your keyboard and that will execute your code.**



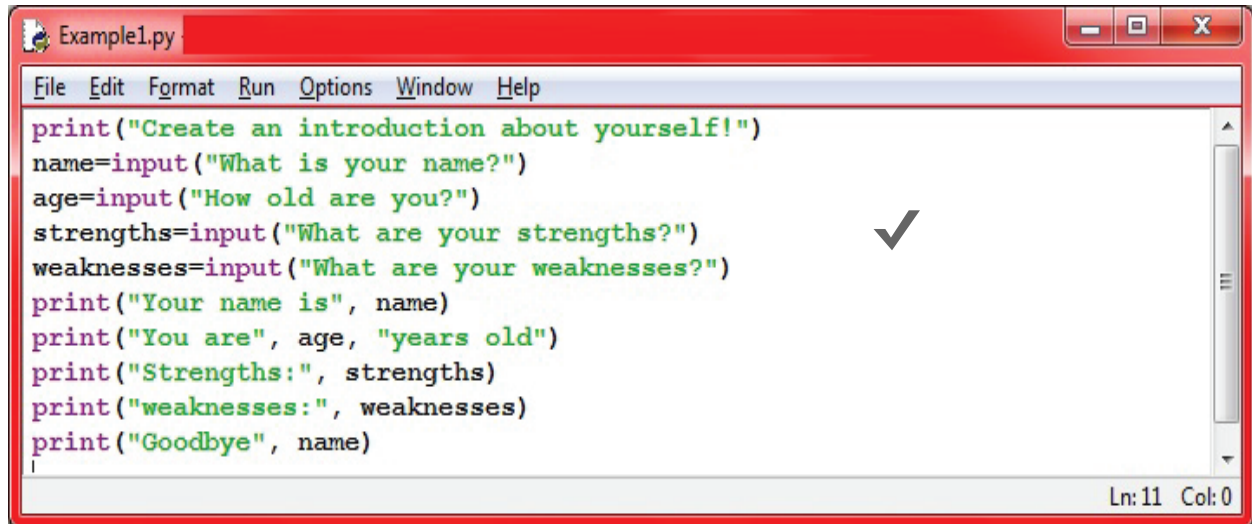
As you can see in the previous screenshot, the code was executed and the output was displayed in the Shell window. The first line is the file name and its location, followed by the program output.

Program to Print Your Introduction

Before we finish with this session, let us create one more interactive program.

We will write a program that asks you a few questions, then prints out an introduction about yourself.

From the same interpreter shell of Python 3.6 that you used in the last examples, click on File and then New file, then write the following code:



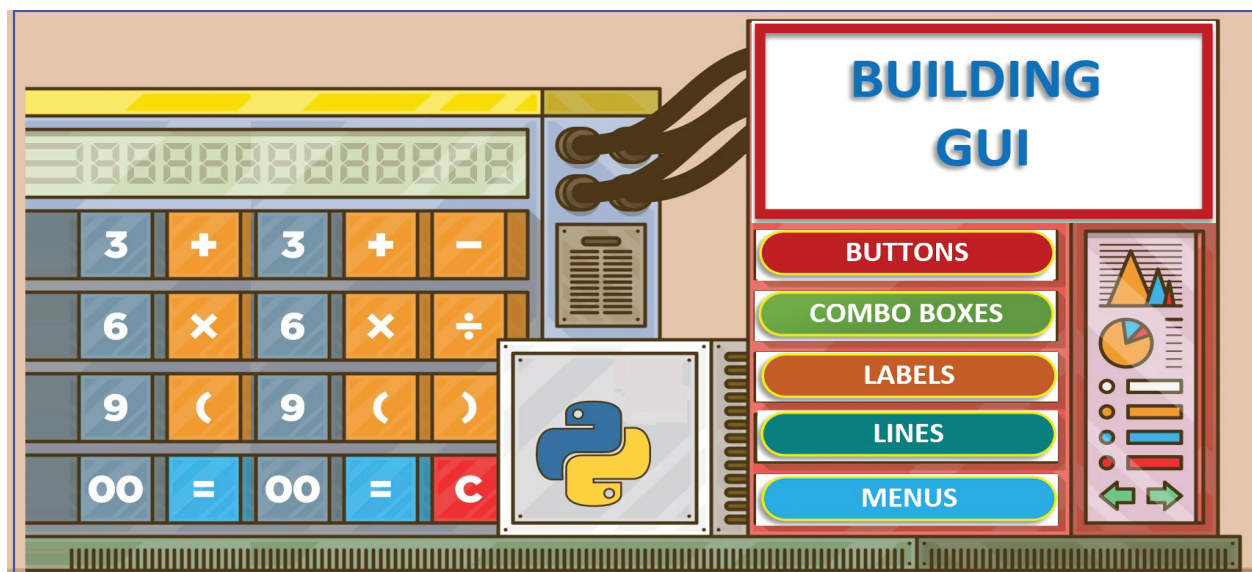
```

File Edit Format Run Options Window Help
print("Create an introduction about yourself!")
name=input("What is your name?")
age=input("How old are you?")
strengths=input("What are your strengths?")
weaknesses=input("What are your weaknesses?")
print("Your name is", name)
print("You are", age, "years old")
print("Strengths:", strengths)
print("weaknesses:", weaknesses)
print("Goodbye", name)
Ln: 11 Col: 0
  
```

In the above program we have used `input()`, which is an in build function of Python. The `input()` function prompts message to a user regarding the program input. It lets a user to provide input to the program through an input device such as Keyboard.

Graphical User Interface (GUI) in Python

Graphical user interfaces (GUIs) are becoming an important part of every application. They allow users to interact with the application through graphical icons and visual indicators instead of text-based user interfaces. You may have noticed that the programs that you built in this session were not attractive enough, as they were text-based applications. GUIs were introduced to make applications more attractive and easier to use.





Apply

Activity 1

Fill in the blanks.

Coding, Binary Code, Integrated Development and Learning Environment, Computer Program

- Programming is also referred to as _____.
- A _____ is a sequence of instructions executed by computer.
- _____ is the natural language of electronic devices.
- An _____ is a bundled set of software tools for program development.

Activity 2

Match the following.

- | | |
|-----------------|---|
| (a) Programming | is the natural language of electronic devices |
| (b) Binary code | is the extension of Python. |
| (c) .py | translate the entire program to a file and execute the whole file |
| (d) Source code | is often referred to as coding. |
| (e) Compilers | is a raw text file, written on very simple editors. |

Activity 3

Practice the following tasks in your notebook.

- Write a program in Python to print the message "I Love Python."
- Write a program in Python to display your name, address, and profession.
- Write a program in Python to subtract, multiply, and divide two numbers.

Activity 4

What do the green and red text mean in the following window? How can you remove this red text?

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print "I am learning Python in King Saud University"
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("I am learning Python in King Saud University")?
>>> print ("I am learning Python in King Saud University")
I am learning Python in King Saud University
>>>
```



Closing

Dear student, you have just learned that anything a computer does must be through a program. A program is a sequence of instructions that is written in a programming language. A programming language is a language that allows us, as humans, to communicate with machines and pass our commands through a code. The code is then translated, either interpreted or compiled (based on the programming language that you used), to be a binary code. The machine then understands the program and executes the commands. You also became familiar with a powerful interpreted object-oriented programming language, Python. You learned the functionality of two main windows that Python uses: the Shell and Code windows. Although you only executed a few simple queries, Python is used for more complicated applications, including business intelligence and machine learning.

CHAPTER

4

Session

3

Building your First Program



Goal

In this session, you will learn

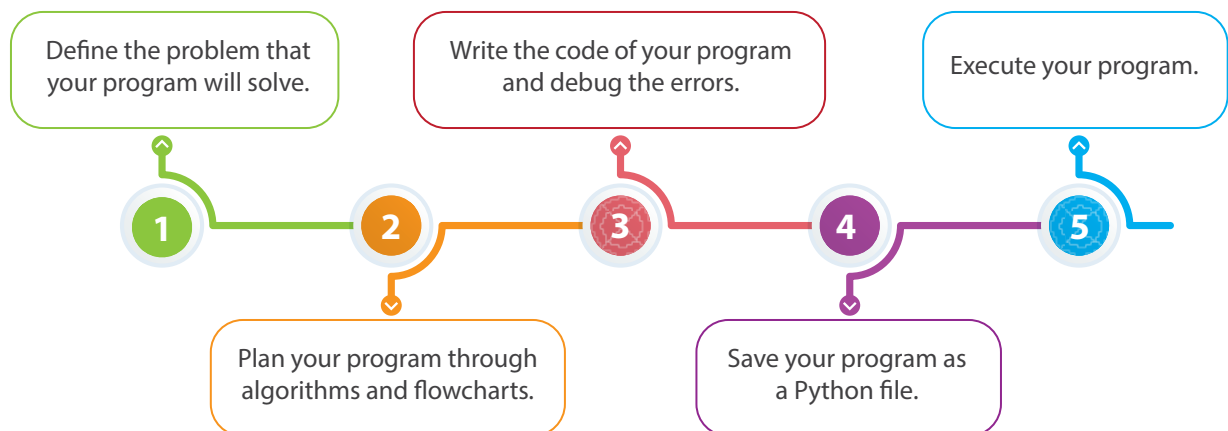
to build and run a complete program that reads, processes, and delivers a solution.



Learning objective

Dear Student

By the end of this session, you should be able to:



Be prepared

Any computer program that you have come across was built to solve a problem or provide a service. If you have a problem and you would like to utilize the power of a computer to solve it, then you should know what to do by now.

To build a computer program, we usually start by understanding the problem, planning the solution, and then start coding. If we don't truly understand the problem, the solution might be defective, giving wrong output or operating inefficiently.





Learn

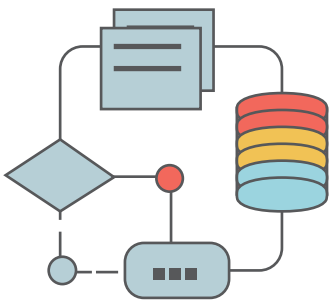
Building your first program

In this session, you will transform a physical game into a computer-based game.

The game is about a person in a jail. There are three doors in front of him. One door will let him go free from the jail, while the other two will take him to the jailors. Now he needs to guess which door is the escape door. If he finds it, then he wins the game. Otherwise, he loses the game and it is over.



Planning the Program



As you understood in session one of this chapter, an algorithm is a useful tool that can be used to plan your program before you start the actual coding. It helps you to write a step-by-step procedure that covers the big picture of your program. Writing an algorithm about your program saves your time, helps you optimize your solution, and makes the implementation process much easier. In this section, we will write the algorithm that solves the Escape Door game's problem, followed by the flowchart.

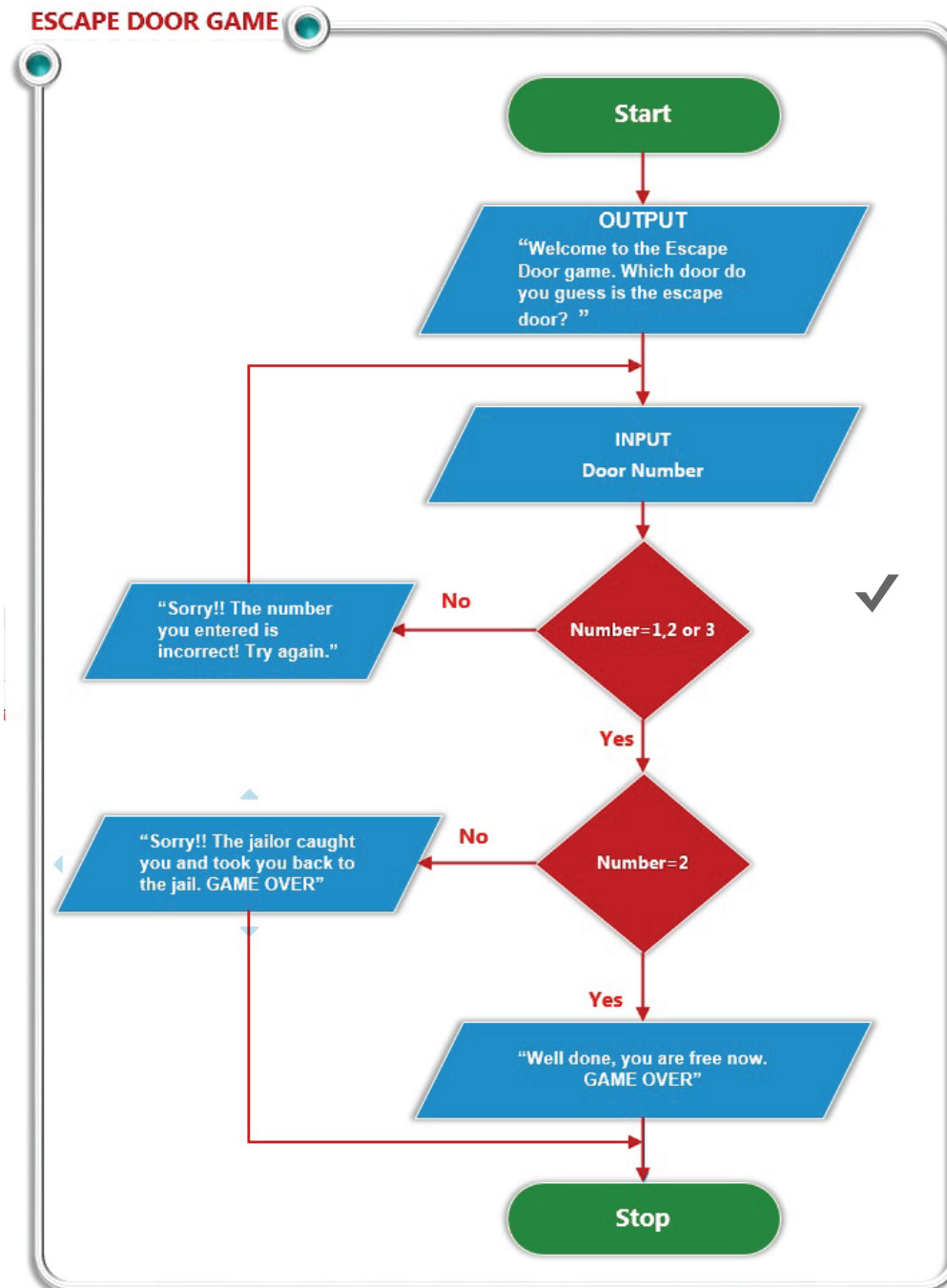
Algorithm: Escape Door Game



Flowchart: Escape Door Game

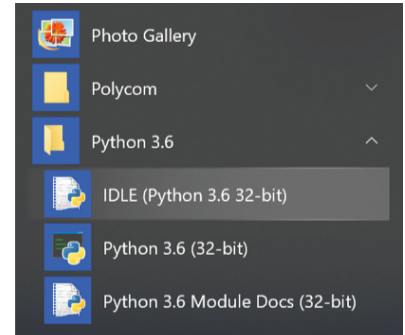
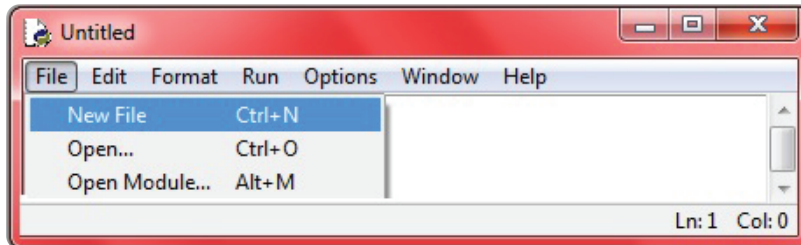
Draw a flowchart of the Escape Door game.

Now you have written an algorithm that explains the solution in a clear and simple way. To make your program even more understandable, we are going to draw a flowchart together. The flowchart, as we learned in session one, will make it easy for anyone to understand the objective and logic of your program.

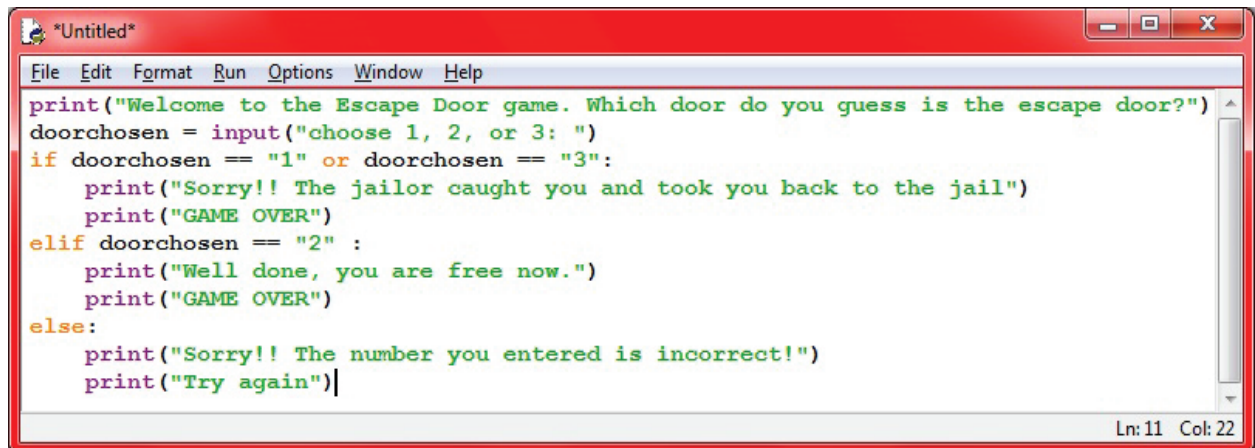


Program code: Escape Door Game

Now it is time to make the game real. You need to open IDLE, as you learned in session three. Once you open it, click the File menu, then choose New file. Save the file with a name that represents the game. Carefully type in the code shown below.



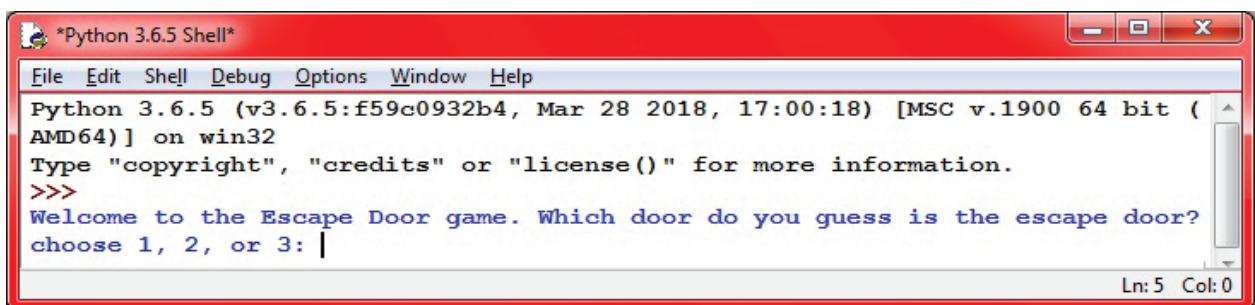
Your program should now look like the following



```
print("Welcome to the Escape Door game. Which door do you guess is the escape door?")
doorchosen = input("choose 1, 2, or 3: ")
if doorchosen == "1" or doorchosen == "3":
    print("Sorry!! The jailor caught you and took you back to the jail")
    print("GAME OVER")
elif doorchosen == "2" :
    print("Well done, you are free now.")
    print("GAME OVER")
else:
    print("Sorry!! The number you entered is incorrect!")
    print("Try again")
```

Ln: 11 Col: 22

Save your file, then click on the Run menu, and click on Run Module. Alternatively, you could press F5 on your keyboard and it would directly run your program. You should get the following window.



```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (
AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
Welcome to the Escape Door game. Which door do you guess is the escape door?
choose 1, 2, or 3: |
```

Ln: 5 Col: 0

If you get an error message (known as a bug), then you have to debug your code by carefully checking the code again. Such errors might be:

Spelling errors

Missing quote marks or brackets

Missing indents

Using = instead of == in an if statement

Not using : after if, elif, and else



Apply

Activity 1

Let us go back to the Escape Door game.

If we have four doors instead of three and only one of them is the escape door, what changes need to be made to the algorithm, flowchart, and code? Can you rewrite them to accommodate the new requirement?

Activity 2

Find errors in the given code and correct them.

```
File Edit Format Run Options Window Help
print("Welcome to the Escape door game, which door you guess is the escape door?")
doorchosen = input("choose 1, 2 or 3")
if doorchosen == "1" or doorchosen == "3":
    print("Sorry!! The jailor caught you and took you back to the prison")
    print("GAME OVER")
elif doorchosen == "2":
    print("Well done, you are free now.")
    print("GAME OVER")
else:
    print("Sorry, you did not enter 1, 2 or 3 !!")
    print("Run the game again to have another go.")
```

Activity 3

Find errors in the given code and correct them.

```
File Edit Format Run Options Window Help
print("Welcome to the Escape door game, which door you guess is the escape door?")
doorchosen = input("choose 1, 2 or 3")
if doorchosen == "1" or doorchosen == "3":
    print("Sorry!! The jailor caught you and took you back to the prison")
    print("GAME OVER")
elif doorchosen == "2":
    print("Well done, you are free now.")
    print("GAME OVER")
else:
    print("Sorry, you did not enter 1, 2 or 3 !!")
    print("Run the game again to have another go.")
```

Activity 4

Create an algorithm and flowchart to add two different numbers (using three variables)



Closing

Dear student, you have just learned how to solve a problem or provide a service through a computer application. Although the program that you built is a simple and small one, the concepts and procedures remain the same even with more complicated ones. It is always a good practice to split a big program into small subsets of functions, then work on each one individually. Once the small subsets are completed and tested, then it is much easier to combine them and run the whole program. There are many techniques and practices in how to code your program, and you need to define your own way and be precise and consistent.

CHAPTER

4

Session

4

Variables, Statements, and Operators



Goal

In this session, you will learn

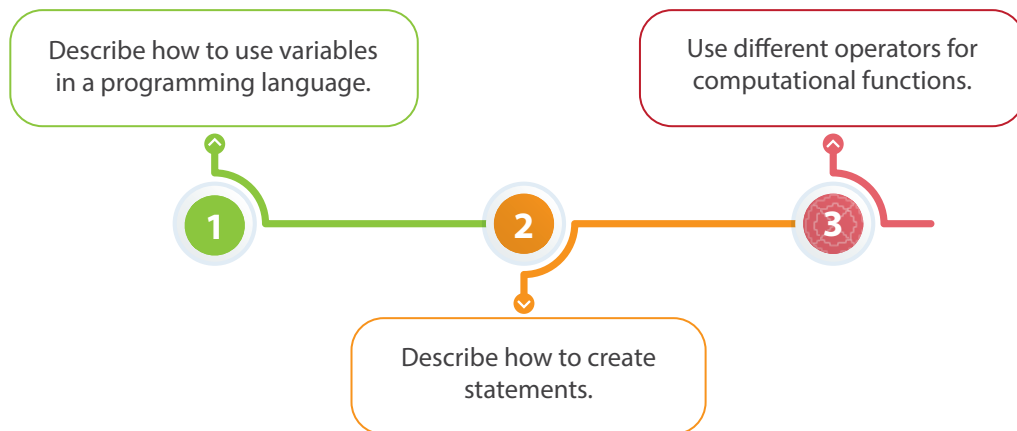
why and how to use variables, operators, and statements to write your Python code.



Learning objectives

Dear Student

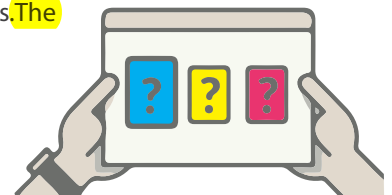
By the end of this session, you should be able to:



Be prepared

One of the most powerful features of programming is the ability to use variables. The process of assigning a value to a variable is an example of a **“statement.”**

A statement is a unit of code that the Python interpreter can execute. In this session, you will learn how to use variables and create executable statements.





Learn

Variables



Before explaining the variables in Python, let us review the values and their types. **A value is a basic element that a program works with, such as a number or a letter.** In the previous sessions, we dealt with different values including 1, 2, 3 and "Python is fun!". These values belong to different types. Here, 1, 2, 3 are integers and "Python is fun!" is a "string," as it contains a string of letters. The string is always enclosed in quotation marks.

Variables are reserved memory locations to store values. They act as containers for storing data values.

If you are not sure what type of variable it is, then you can use the command "type()" function and the interpreter will tell you the type, as shown in the following example.

Python displays the variable type based on the value assigned to it. This means Python will change the variable type if we assign a value of a different type. For example: if you assign ' Python is fun!' to a variable var, then the type will be string. If you reassign another value, 5, to var, then it will be integer.

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> type('Python is fun!')
<class 'str'>
>>>
```

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> var='Python is fun!' #This will create a string type.
>>> type(var)
<class 'str'>
>>>
```

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> var=5 # the 'var' variable is now a number(integer type)
>>> type(var)
<class 'int'>
>>>
```

some basic variable types in Python.

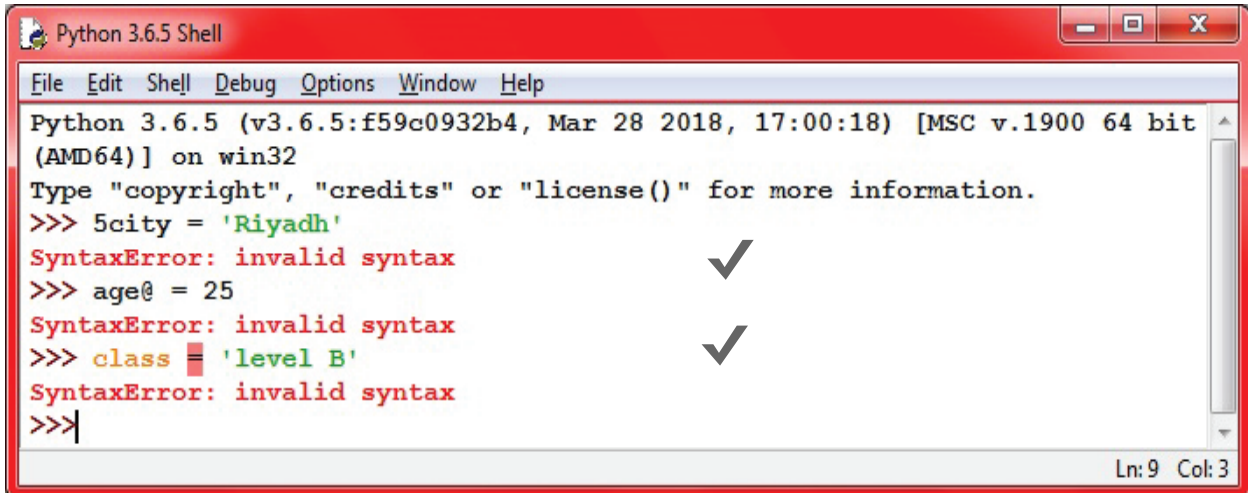
Type	Examples	Description
integer	123 0 -1234	Any positive, negative, or zero value; no limit on size
float	3.15 6.02E 2.5	Real numbers expressed in decimal or scientific notation; limits on scale and precision
boolean	True False	Logical true/false quantities
string	"Tuesday" "" 'Luigis'	Textual snippets comprising any number of characters; delimited by single or double quotes

Rules for Creating Variables

Programmers choose names for their variables that are meaningful and describe what the variable is used for. If you give a variable an illegal name, you get a syntax error:

- 1 Variable names can contain both letters and numbers, but they cannot start with a number.
- 2 It is legal to use uppercase & lowercase letters, but it is a good idea to begin variable names with a lowercase letter.
- 3 Special characters cannot be used in variable name except underscore which is used for multiple words, like `student_id`.
- 4 Python keywords cannot be used as a variable name. A variable name cannot contain space.

In the below examples, it is clear that the first and second examples are wrong. However, what is wrong with the third example? You may have noticed that the variable name in the third example is `class`, which is one of Python's keywords. Keywords cannot be used as variable names.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 5city = 'Riyadh'
SyntaxError: invalid syntax ✓
>>> age@ = 25
SyntaxError: invalid syntax ✓
>>> class = 'level B'
SyntaxError: invalid syntax
>>>
```

Python Keywords

Python has 33 keywords reserved that cannot be used as a variable name. They are:

List of Python Keywords

and

del

raise

def

for

as

try

assert

from

none

true

break

or

if

else

not

while

pass

is

with

class

false

in

continue

elif

finally

yield

except

lambda

return

import

global

nonlocal

Statements



The code for an application is usually contained in a sequence of statements. A statement is a unit of code that the Python interpreter can execute. So far, we have seen two kinds of statements: print statement and assignment. When you type a statement, the interpreter executes it and displays the result, if there is one.

If there is more than one statement, the results appear one at a time as the statements are executed, as explained in the following example:

```
print("King Saud University")
x = 2019
print(x)
```

The output will be

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('King Saud University')
King Saud University
>>> x=2019
>>> print(x)
2019
>>> |
Ln: 8 Col: 4
```

Operators



Python has special symbols called operators that represent computations like addition, subtraction, multiplication and division. The values the operator is applied to are called operands. The operators can be summarized in the following table.

Math symbol	Python operator	Operation	Example	Result
÷	/	Division	4 / 2	2.0
+	+	Addition	4 + 2	6
-	-	Subtraction	4 - 2	2
×	*	Multiplication	4 * 2	8
4 ²	**	Exponent or power	4 ** 2	16
()	()	Parentheses (grouping)	(4 + 2) * 3	18
mod or %	%	Modulus	5%2	1
=	==	Equal	2==2	True
<	<	Less Than	8<5	False
>	>	Greater Than	6>10	False

Now you can write some expressions that are built by the combination of values, variables, and operators. If you type an expression in interactive mode, the interpreter evaluates it and displays the result:

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+1
2
>>>
```

You can start by creating variables and assign values to them, then do some calculations, as in the following example:

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> A=5
>>> B=8
>>> C=A*B
>>> print(A,B,C)
5 8 40
>>>
```

Python Operator Precedence

When more than one operator appears in an expression, the order of evaluation depends on the rules of precedence. For mathematical operators, Python follows mathematical convention as follows:

1	2	3	4
()	**	*, /	+, -
Parentheses have the highest precedence.	Exponentiation has the next highest precedence.	Multiplication and Division have the same precedence, which is higher than Addition and Subtraction, which also have the same precedence.	Operators with the same precedence are evaluated from left to right.

Let us take $5*(4+1)+3**2$ as an example and calculate the result.

We will start with the parentheses: $5*5+3**2$

Second is the exponentiation: $5*5+9$

Followed by multiplication: $25+9$

And finally addition: 34

To try that in Python, you can write the following statement and execute it:

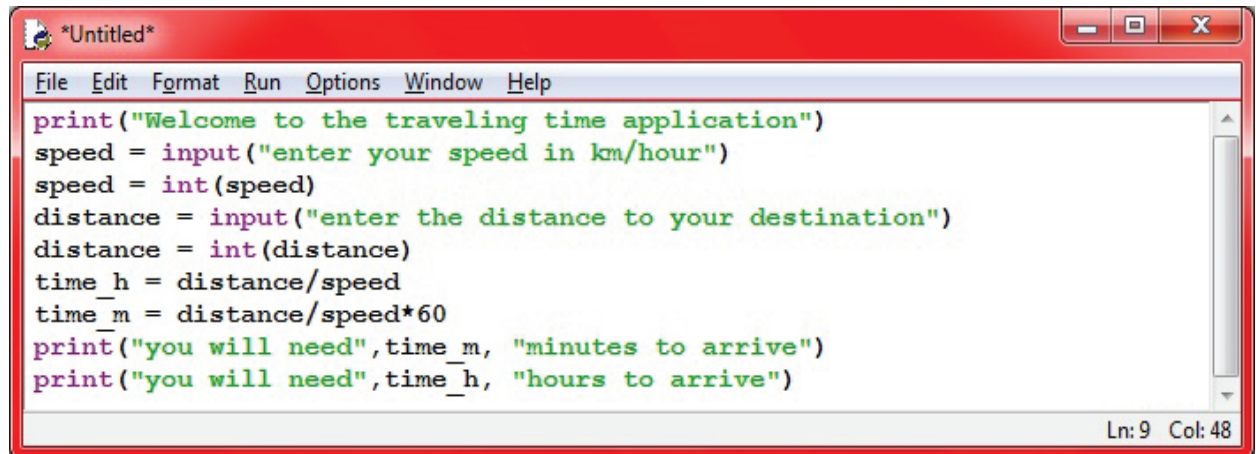
```
>>> 5*(4+1)+3**2
```

```
34
```

Program to Calculate Traveling Time

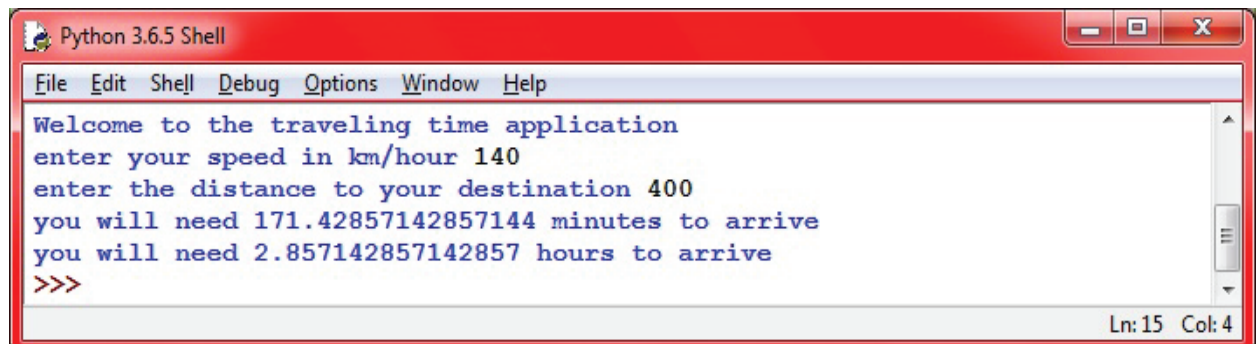
Now let us build a more meaningful application that reads two values, which are your car speed and the distance to your destination; then the application tells you how many minutes you need to arrive.

We will start in a new Code window as you have learned in session 4. Let us save the file as `Traveling_Time`



```
File Edit Format Run Options Window Help
print("Welcome to the traveling time application")
speed = input("enter your speed in km/hour")
speed = int(speed)
distance = input("enter the distance to your destination")
distance = int(distance)
time_h = distance/speed
time_m = distance/speed*60
print("you will need",time_m, "minutes to arrive")
print("you will need",time_h, "hours to arrive")
Ln: 9 Col: 48
```

And when you run the program you will get an output similar to the following.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Welcome to the traveling time application
enter your speed in km/hour 140
enter the distance to your destination 400
you will need 171.42857142857144 minutes to arrive
you will need 2.857142857142857 hours to arrive
>>>
Ln: 15 Col: 4
```

The above program uses type conversion `int` (`speed`). You can explicitly cast, or convert, a variable from one type to another.

Examples:

`int()` : Used to explicitly convert a data type(`integer,float,string`) to integer type.

```
x = int(5)           # x will be 5
y = int(7.8)         # y will be 7
z = int("9")         # z will be 9
```



`float()`: Used to explicitly convert a data type(`integer,float,string`) to float type.

```
x = float(5)         # x will be 5.0
y = float(7.8)       # y will be 7.8
z = float("9")       # z will be 9.0
w = float("6.2")     # w will be 6.2
```





Apply

Activity 1

Create two integer-type variables and calculate the addition of them in a third variable.

Activity 2

Print "I am studying at King Saud University" using an appropriate variable.

Activity 3

What will be the result for the following statement?
`>>> 6*5*(9+2)+3**2+(5+1)?`



Closing

Dear student, you have just learned how to use variables, operators, and statements to enhance your programming and produce solutions for more complicated problems. We mentioned in this session four famous types of variables, and there are many more you may need to learn in your future studies. We also explained the basic operators that allow you to do computations. However, Python has many built-in functions that you can use, and you can also create your own functions. You can now refer to the Python website and explore more advanced variable types and functions to improve your previous programs.

CHAPTER

4

Session

5

Conditional Logic



Goal

In this session, you will learn

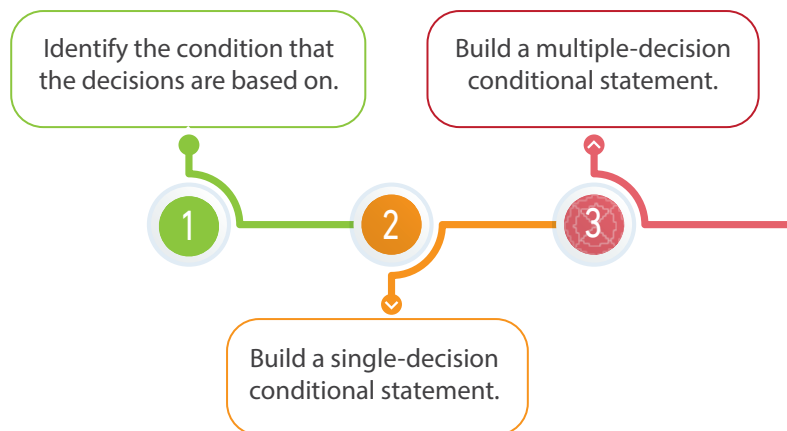
why and how to use conditional statements in your Python code.



Learning objectives

Dear Student

By the end of this session, you should be able to:



Be prepared

One of the most powerful features of programming is the ability to use conditional logic. In programming, there are some situations where you need to make decisions, and based on these decisions, the program will execute the next block of code. In other words, conditional logic here means we are given different paths from which we have to choose one. Python supports the usual logical conditions from mathematics. In this session, you will learn how to use different conditional statements.



Learn

Decision Making



As a **programmer**, you will definitely need to distinguish between many decisions and perform different instructions based on the conditions output. To do so, Python provides you with two main features: Boolean expression and conditional statements.

Boolean Expression

Boolean expression is an expression that is either true or false. The following are examples of Boolean expressions that use the operator `==` to compare two values and produce **True** if they are equal and **False** otherwise.

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
>>> 3==3
True
>>> 3==6
False
>>>
```

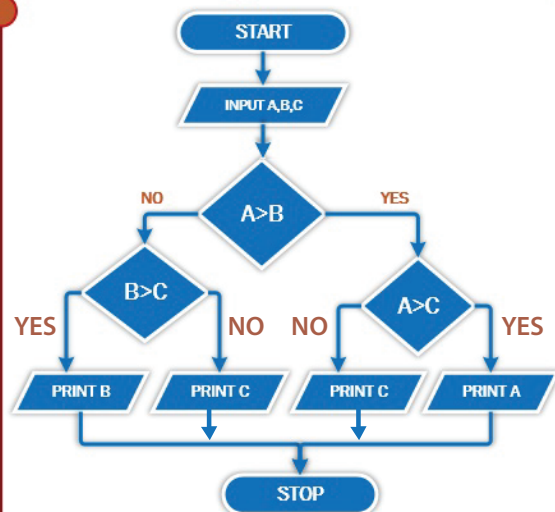
Conditional Statements

Conditional statements are used when we have two or more outputs for a condition, with a specific action for each condition output. The flowchart from session 1 is an example of a decision-making structure. Here, we have three conditions, and each condition has two outputs.

Decision structures evaluate multiple expressions and produce TRUE or FALSE as the outcome. You need to determine which actions to take and which statements to execute if the outcome is TRUE or FALSE. In this session you will learn about following conditional statements used in Python programming.



Decision Making



If Statement



An "if statement" allows your program to run a group of instructions, based on a condition or set of conditions. With an if statement, your program can make a choice. The syntax of writing an if statement in Python is as follows:

```
if condition:  
    indented statement(s)
```

As we mentioned, the condition we are testing is usually a Boolean expression with a true/false output. If the expression is true, the program will run the indented statement(s), but if it's false, the program will skip them and continue with the rest of the program at the next unindented line.

Let us now practice using if statements in the following scenario. build a program that asks students to enter the pass mark for the IT140 course, asks them to enter their accumulated marks until now, and tells them if they have already passed or how many marks they need to pass the course.

Example: if statement

As you learned, you should start by drawing a flowchart before you start coding your program. Make the flowchart now, then proceed to the commands below.

```
*Untitled*  
File Edit Format Run Options Window Help  
pass_marks = eval(input("What is the pass mark for your IT140 course? "))  
your_marks = eval(input("How many marks do you have until now? "))  
if your_marks >= pass_marks:  
    print("Well done, you have already passed!!")  
if your_marks < pass_marks:  
    print("Not yet, but you can do it. You need", pass_marks - your_marks,  
          "more marks to pass!!")  
Ln: 7 Col: 33
```

In the above program we have used the eval() function.

The eval() function evaluates the "String" like a python expression and returns the result as an integer. For example

```
a=eval('3+2+3')
```

```
print(a)
```

```
Output : 8
```


Else Statement



In many scenarios, we want our program to execute a set of instructions if a condition output is **true** and a different set of instructions if the condition output is **false**. Python provides you with a solution for that: the **else** statement. It allows you to test if the condition is **true** without having to perform another test to see if it is false. The **else** statement can only be used after an **if** statement, so we sometimes refer to the two together as an **if-else** statement. The syntax of writing an if-else statement in Python is as follows:

```
if condition:  
    indented statement(s)  
else:  
    other indented statement(s)
```

- If the condition in an if statement is **true**, the indented statements under the if are executed, and the else and all its statements are skipped.
- If the condition in the if statement is **false**, the program skips directly to execute all indented statements in the else part.

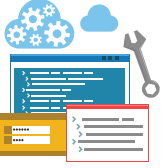
Example: else statement

Let us now rewrite the previous example (the if statement example) with an if-else statement instead.

```
*Untitled*  
File Edit Format Run Options Window Help  
pass_marks = eval(input("Enter the pass mark for your IT140 course? "))  
your_marks = eval(input("How many marks do you have"))  
if your_marks >= pass_marks:  
    print("Well done, you have already passed!!")  
else:  
    print("Not yet, but you can do it. You need", pass_marks - your_marks,  
          "more marks to pass!!")  
Ln: 7 Col: 33
```

An if-else statement makes the code shorter and easier to read since the condition must be either true or false. It also helps prevent coding errors in the two conditions as you may forget some outputs, such as if your mark is equal to the pass mark.

Elif Statement



In other scenarios, if we have more than two possibilities for a condition, we need more than two groups of instructions.

```
if condition:  
    indented statement(s)  
elif condition:  
    indented statement(s)  
else:  
    indented statement(s)
```

Elif is an abbreviation of "else if." Again, exactly one branch will be executed. There is no limit on the number of elif statements. If there is an else clause, it has to be at the end, but there does not have to be one.

Example: elif statement

To practice the elif statement, let us write a program that reads a student's marks and display the grade. The conditions are given as:

95 to 100	Grade A+,
90 to <95	Grade A,
85 to <90	Grade B+,
80 to <85	Grade B,
75 to <80	Grade C+,
70 to <75	Grade C,
65 to <70	Grade D+,
60 to <65	Grade D.

Otherwise, the grade is F,

If the entered marks are above 100 or less than 0; the error message "**Sorry, your_marks,Invalid Marks!**" should be displayed.

```
*Untitled*  
File Edit Format Run Options Window Help  
your_mark = eval(input("Enter your marks?"))  
if your_mark >= 95 and your_mark <= 100:  
    print("Your marks:",your_mark,";Grade:A+")  
elif your_mark >= 90 and your_mark <95:  
    print("Your marks:",your_mark,";Grade:A")  
elif your_mark >= 85 and your_mark <90:  
    print("Your marks:",your_mark,";Grade:B+")  
elif your_mark >= 80 and your_mark <85:  
    print("Your marks:",your_mark,";Grade:B")  
elif your_mark >= 75 and your_mark <80:  
    print("Your marks:",your_mark,";Grade:C+")  
elif your_mark >= 70 and your_mark <75:  
    print("Your marks:",your_mark,";Grade:C")  
elif your_mark >= 65 and your_mark <70:  
    print("Your marks:",your_mark,";Grade:D+")  
elif your_mark >= 60 and your_mark <65:  
    print("Your marks:",your_mark,";Grade:D")  
elif your_mark >= 0 and your_mark <60:  
    print("Your marks:",your_mark,";Grade:F")  
else:  
    print("Sorry",your_mark,";Invalid Marks!")  
Ln: 21 Col: 46
```

It is obvious that the decision is based on the student's marks. Therefore, all conditions are directly related to the marks. You may also have noticed that we have multiple conditions, where each condition has an output, and as a result we use d an elif statement to execute this conditional logic.



Apply

Activity
1

Write a program to compare two variables (a and b) and print the value that is larger along with the variable name.

Activity
2

Write a program to check whether a number is even or odd.

Activity
3

Write a program that reads three numbers, then finds and prints the largest one.



Closing

Dear student, you have just learned how to use different decision-making statements to enhance your programming and produce solutions for more complicated problems. We mentioned in this session three famous types of conditional logic. You can now refer to the Python website and explore more details and more examples where you can use different decision-making statements and improve your previous programs.

CHAPTER

4

Session

6

Loops and Functions



Goal

In this session, you will learn

why and how to use different loops and functions in your Python code.



Learning objectives

Dear Student

By the end of this session, you should be able to:

Use loop statements in a programming language.

Create your own functions.

1

2

3

Use Python functions to enhance programming.



Be prepared

Programming is a useful tool that allows us to perform complicated tasks in an easy and accurate way.

In this session, you will become familiar with two programming features that can execute complicated tasks more efficiently: loop and function. Loop allows you to repeat a statement or group of statements hundreds or thousands of times without typing them again and again. Yes, you only need to write a loop statement with a counter or a condition, and the program will take care of all the repetition.

On the other hand, a function is a block of organized, reusable code that is used to perform a single, related action. There are built-in functions like `print()`, and you can also create your own functions.





Learn

Python Loops



When we need to do something over and over again in a program, loops allow us to repeat those steps without having to type each one separately. To build a loop,

- we first need to identify the repeated steps.
- Second, we need to figure out how many times to repeat those steps. Now that we know the repeated instructions and how many times, it's time to build our loop. There are two famous loop statements in Python, which we will discuss in this session: the **for** loop and the **while** loop.

for loop

A **for** loop statement in Python iterates over a list of items and repeats one step or a group of steps for each item in a list, like the numbers 1 through 20, or names = ['ahmad', 'saad', 'sarah'].

Have a look at following two examples

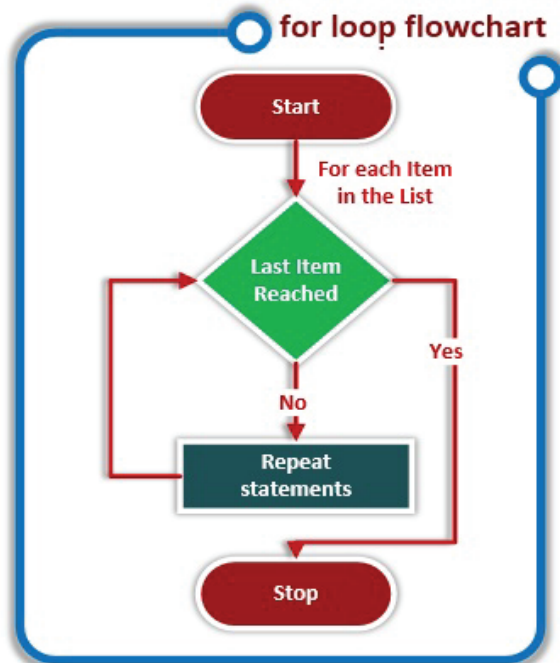
Example 1

```
*Untitled*
File Edit Format Run Options Window Help
fruits=["Apple", "Banana", "Orange"]
for n in fruits:
    print(n)
Ln: 3 Col: 12
```

Example 2

```
*Untitled*
File Edit Format Run Options Window Help
for n in range(10):
    print(n)
Ln: 2 Col: 12
```

for loop flowchart



Note that example 1 uses a List Sequence and example 2 uses a range() function.

List is a collection of values(items) which is ordered and changeable. A list is created by putting different comma-separated values between square brackets. For example -

```
fruits = ["Apple", "Banana", "Orange"]
```

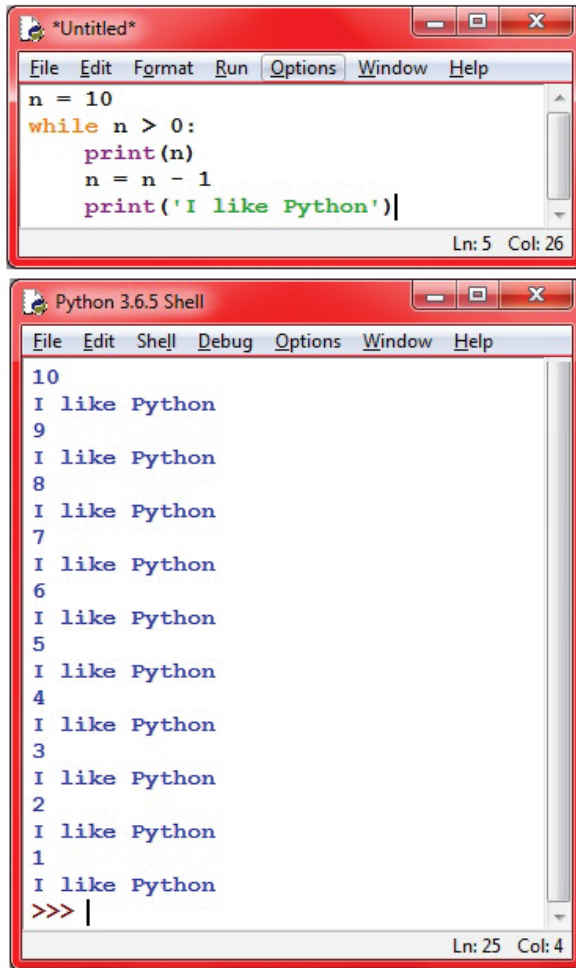
The above program in example 1 will begin to print the items of the list **fruits** and the loop continues until we reach the last item in the list i.e. **Orange**.

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. The range(10) in the above program will generate numbers from 0 to 9 (10 numbers) and are printed in sequence using for loop. i.e. **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.

while loop

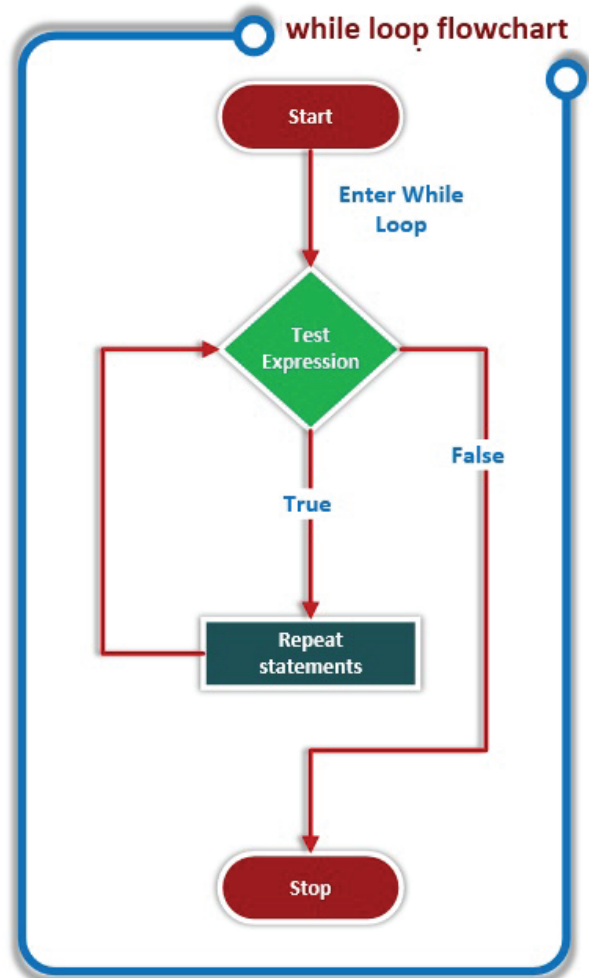
A **while** loop statement repeats a step or group of steps while a given condition is true. It checks the condition before executing the loop body and then executes the steps, which makes it different from the **for** statement.

The following example shows you how to count down from 10 and then print "I like Python":



```
*Untitled*
File Edit Format Run Options Window Help
n = 10
while n > 0:
    print(n)
    n = n - 1
    print('I like Python')
Ln: 5 Col: 26

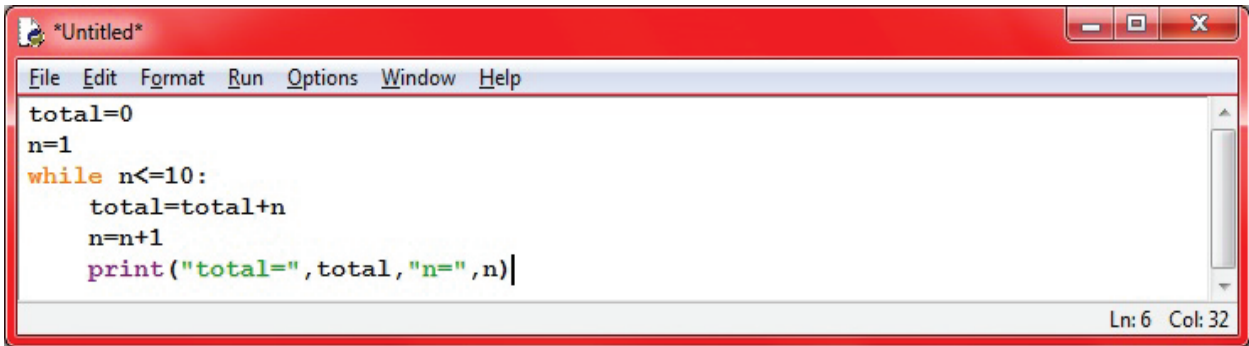
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
10
I like Python
9
I like Python
8
I like Python
7
I like Python
6
I like Python
5
I like Python
4
I like Python
3
I like Python
2
I like Python
1
I like Python
>>> |
Ln: 25 Col: 4
```



As you may notice, we started by having a counter **n** and assigned it a value of **10**. Then we started a **while** statement with a condition **n > 0**. While the condition is true, the while statement body (the intended steps) will be executed.

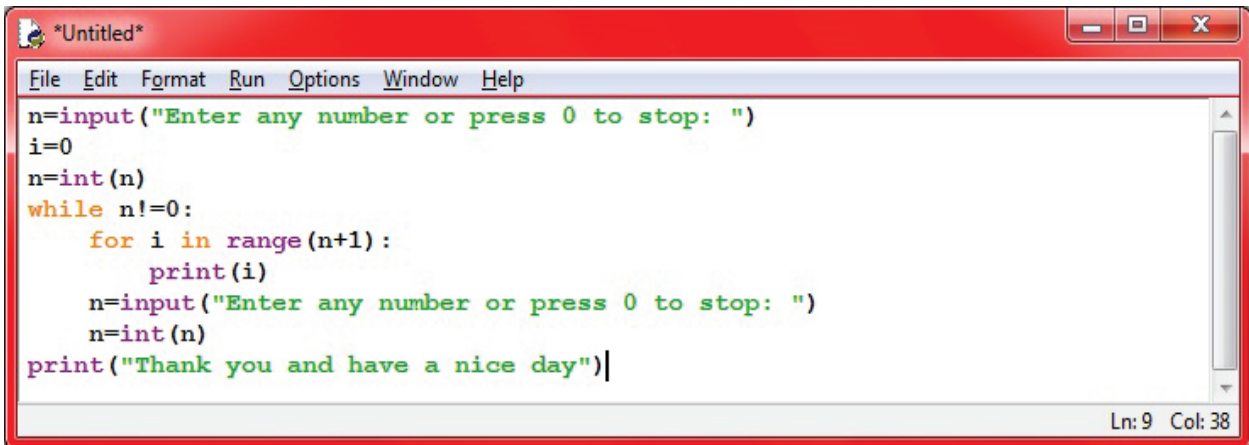
In this example, we have two steps: print **n** and update **n** with **n-1**. The body will be repeated 10 times. Each time we execute the body of the loop, that is called an "iteration". When **n=0**, the condition becomes false. At this stage, the **while** loop statement is completed and the interpreter exits the **while** statement and continues the program. We have one more statement remaining in the program, which is printing "I like Python"; then the program is completed.

Have a look at the following example and predict the outputs.



```
*Untitled*
File Edit Format Run Options Window Help
total=0
n=1
while n<=10:
    total=total+n
    n=n+1
    print("total=",total,"n=",n)|
Ln: 6 Col: 32
```

Let us now practice **while** and **for** statements using a more detailed example. You have been asked to write a program that reads a number and prints all values from zero to that number. The program then asks for a new number and repeats that same process until 0 is entered. Once 0 is entered, the program prints "Thank you and have a nice day" and stops.



```
*Untitled*
File Edit Format Run Options Window Help
n=input("Enter any number or press 0 to stop: ")
i=0
n=int(n)
while n!=0:
    for i in range(n+1):
        print(i)
    n=input("Enter any number or press 0 to stop: ")
    n=int(n)
print("Thank you and have a nice day")|
Ln: 9 Col: 38
```

Functions



A function is a block of code that only runs when it is called. It is a block of organized and reusable code that is used to perform a single task. Functions are very useful in many cases where you need to run the function every now and then.

To use a built-in function or a function that you defined, you only need to call the function and pass the parameters. That will execute the function using the parameters and return an expression if there is any.

Built-in functions

Python has many built-in functions that you can use to save time and shorten your code.

You will find many built-in functions in the following table; some of them you already used in this chapter.

Function	Description
<code>abs()</code>	Returns absolute value of a number
<code>all()</code>	Returns true when all elements in an iterable are true
<code>any()</code>	Checks if any element of an iterable is true
<code>ascii()</code>	Returns string containing printable representation
<code>bin()</code>	Converts integer to binary string
<code>chr()</code>	Returns a character (a string) from an integer
<code>len()</code>	Returns the number of items in an object or the number of characters in a string
<code>pow()</code>	Returns x to the power of y
<code>print()</code>	Prints the given object
<code>range()</code>	Returns sequence of integers between start and stop
<code>repr()</code>	Returns printable representation of an object
<code>reversed()</code>	Returns reversed iterator of a sequence
<code>round()</code>	Rounds a floating point number to n places.
<code>set()</code>	Returns a Python set
<code>sum()</code>	Adds items of an iterable

Let us practice the function `len()`, which returns the number of characters in a string. Have a look at the following example

```
*Untitled*
File Edit Format Run Options Window Help
names=['Sami', 'Ahmad', 'Abdullah']
for i in names:
    print(i, len(i))
Ln: 3 Col: 20
```

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Sami 4
Ahmad 5
Abdullah 8
>>>
```

User Defined Functions

If the built-in functions are not enough for you, then you can create your own functions. These functions are called user-defined functions.

You can define functions to provide the required functionality. Here are simple steps to define a function in Python.

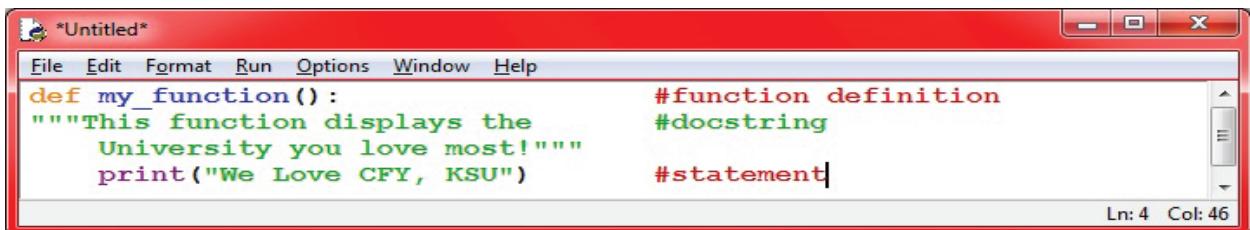
Syntax of Function

```
def function_name(parameters):
    """docstring"""
    statement(s)
    return
```

The function definition consists of following components.

- 1 - Keyword def marks the start of function header.
- 2 - A function name to uniquely identify it. Function naming follows the same rules of writing identifiers in Python.
- 3 - Parameters (arguments) through which we pass values to a function. They are optional.
- 4 - A colon (:) to mark the end of function header.
- 5 - Optional documentation string (docstring) to describe what the function performs.
- 6 - One or more valid python statements that make up the function body. Statements must have same indentation level.
- 7 - An optional return statement to return a value from the function.

Example:



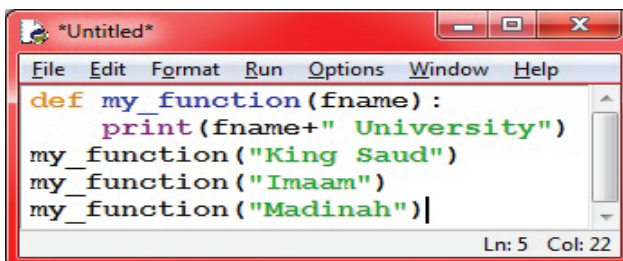
```
*Untitled*
File Edit Format Run Options Window Help
def my_function():
    """This function displays the
       University you love most!"""
    print("We Love CFY, KSU")
#function definition
#docstring
#statement
Ln: 4 Col: 46
```

Function Call

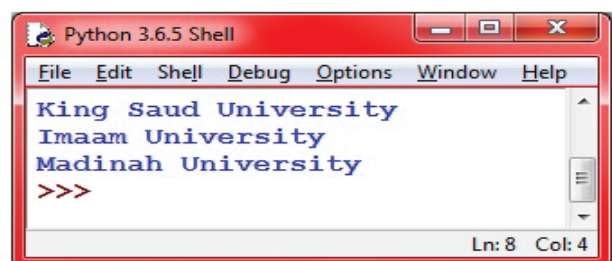
Calling the function actually performs the specified actions. To call a function, use the function name followed by parenthesis. We will now call the above defined function `def my_function()`:



```
*Untitled*
File Edit Format Run Options Window Help
def my_function():
    """This function displays the
       University you love most!"""
    print("We Love CFY, KSU")
my_function()
#function call
Ln: 5 Col: 50
```



```
*Untitled*
File Edit Format Run Options Window Help
def my_function(fname):
    print(fname+" University")
my_function("King Saud")
my_function("Imaam")
my_function("Madinah")
Ln: 5 Col: 22
```



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
King Saud University
Imaam University
Madinah University
>>>
Ln: 8 Col: 4
```

`my_function` is called three times by passing different arguments to parameter `fname`. The passed argument is concatenated with string "University" using the "+" operator in the print function.



Apply

Activity 1

Write a program to print all natural numbers in reverse (from n to 1), using a for loop)

Activity 2

Write a program to print all even numbers between 1 and 100, using a while loop.

Activity 3

Write a program to find the maximum and minimum among three numbers, using functions.



Closing

Dear student, you have just learned how to use different loops and different types of functions to enhance your programming and produce solutions for more complicated problems. We mentioned in this session two famous types of loops and two ways of using functions. You can now refer to the Python website and explore more examples where we can use different loops and functions and improve your previous programs.

CHAPTER

4

Session

7

File Input/Output (File I/O)



Goal

In this session, you will learn

how to read data from text files, use the data in your program, and write data to a text file.



Learning objectives

Dear Student

By the end of this session, you should be able to:

1 Read from and write to files in a programming language.

3 List different types of files.

1

2

3

2 Use files in different modes.



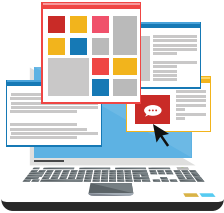
Be prepared

You have learned many programming features of Python that allow you to write and execute many programs efficiently. That is really great. In this session, you will learn an important feature that allows you to import data from a text file and store the program output to a text file. In many cases, you do not want to enter every single variable that your program requires by typing it, as you may already have it stored in a file. In other cases, you need to store the program outputs in a file for documentation and sharing purposes. In such scenarios, using file I/O is very helpful. In this session, you will learn how to read data from a text file, manipulate it, and then write data to a file and store it.



Learn

Files in Python

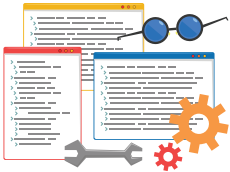


File is collection of data or information kept on a computer storage device and has a name, called the filename. You already know about different kinds of files, like your music files, video files, text files etc. Python gives you easy ways to manipulate the files. Python has two categories of files:

1- Text file

2- Binary file

Text File



A text file is structured as a sequence of lines, where each line includes a sequence of characters. Each line is terminated with a special character, called the EOL or End of Line character. The most common EOL is the comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. A backslash character can also be used to tell the interpreter that the next character – following the slash – should be treated as a new line. On the other hand, a binary file is any type

Binary File

On the other hand, a binary file is any type of file that is not a text file. Because of their nature, binary files can only be processed by an application that knows or understands the file's structure. In other words, there must be an application that can read and interpret binary files.

Python File Handling

When we want to read from or write to a file we need to open it first. When we are done, it needs to be closed. Hence, in Python, a file handling takes place in the following order.

1

Open a File

2

Read or Write a File

3

Close a File

Open a File

Python provides basic functions and methods necessary to read data from files using the **file object**. It is handled natively in Python, so you do not need to import a library. The first thing you will need to do is use Python's built-in **open** function to get a **file object**. As the **open ()** function will return a file object, it is most commonly used with two arguments. An argument is nothing but a value that has been provided to a function, which is relayed when you call it. For example, if we declare the name of a file as "student", that name would be considered as an argument. The syntax to open a file object in Python is:

```
file_object = open("filename", "mode")
```

Where **file_object** is the variable to add the file object, and **mode** tells the interpreter and developer which way the file will be used.

File Opening Modes

We can specify the mode while opening a file. In mode, we specify whether we want to read 'r', write 'w' or append 'a' to the file. We also specify if we want to open the file in text mode or binary mode.

Here is a list of some basic modes that can be used while opening a text file. A list of binary file modes is given in the reading section at the end of the session.

Modes	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer is placed at the beginning of the file.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.

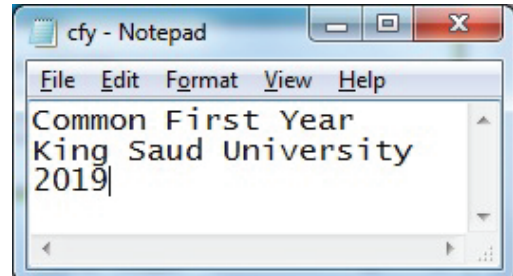
Before we take an example of opening and reading file, let us explore some basic file attributes.

Attribute	Description
file.closed	Returns true if file is closed, false otherwise.
file.mode	Returns access mode with which file was opened.
file.name	Returns name of the file.

Reading Data from File

Suppose that you have created a text file "cfy.txt" with some lines of text and saved it in the folder; "C:\Temp". If you have your text file stored in another location, you can use it after changing the current directory in the Python shell. Let us now change the current directory to Temp.

```
>>> import os, sys
>>> os.chdir("C:\Temp")
To check your current directory, type:
>>> os.getcwd()
```



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import os, sys           # import os & sys modules
>>> os.chdir("c:/Temp")     # change the current directory to Temp
>>> os.getcwd()            # to check your current directory
'c:\\Temp'
>>> |
```

Now we are ready to open the file and read the data.

To open the file in Read mode type

```
>>> mycfy = open("cfy.txt", "r")
```

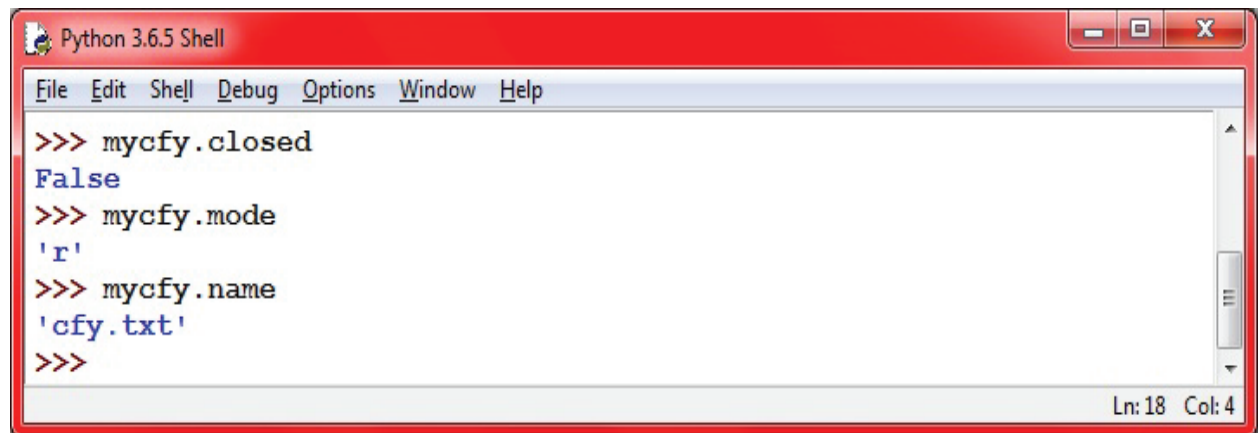
To read the whole file "cfy.txt" at once use the `read()` function with the file object "mycfy". To display the text inside the file object "cfy.txt" use a print statement

```
>>> print("The text is:", mycfy.read())
```

Let us check the output

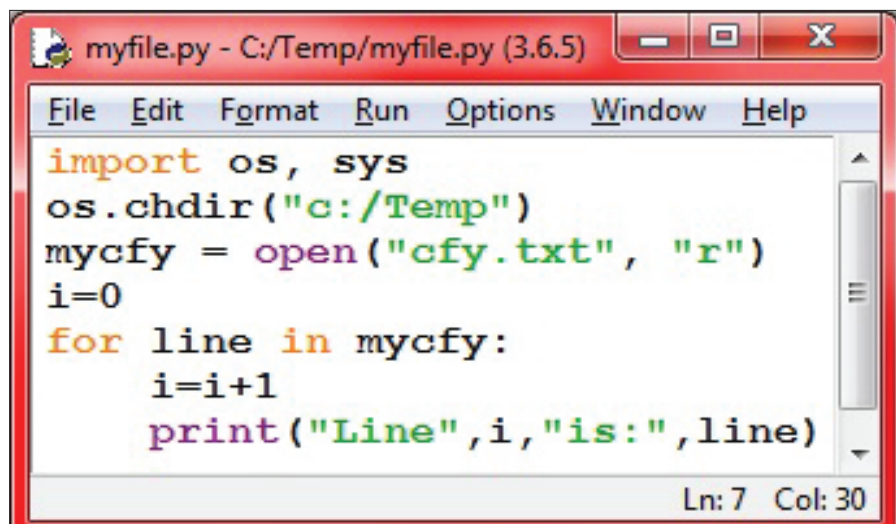
```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
>>> mycfy = open("cfy.txt", "r")
>>> print("The text is:", mycfy.read())
The text is: Common First Year
King Saud University
2019
>>>
```

Let us now find out some attributes of the file "mycfy"

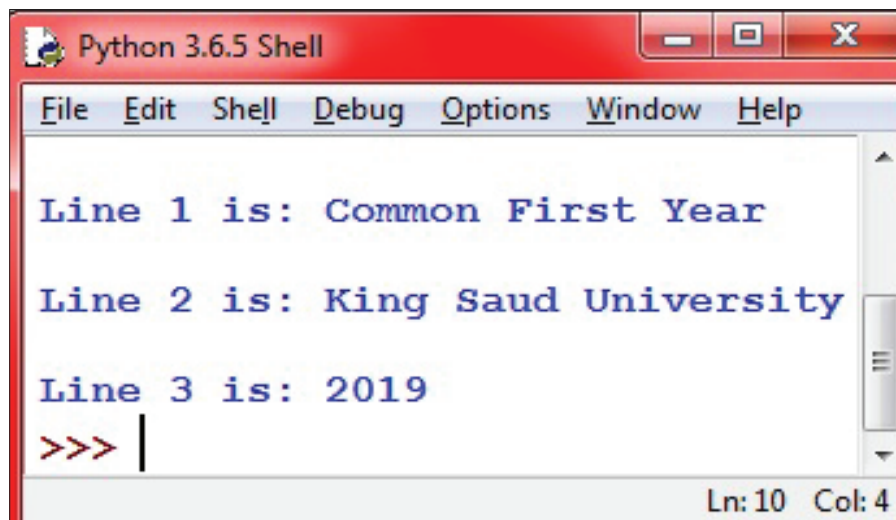


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
>>> mycfy.closed
False
>>> mycfy.mode
'r'
>>> mycfy.name
'cfy.txt'
>>>
Ln: 18 Col: 4
```

Now it is time to practice more advanced program. You are asked to build a program that reads `cfy.txt` file and prints it line by line with the line number. Here you need to use for loop from previous session.



```
myfile.py - C:/Temp/myfile.py (3.6.5)
File Edit Format Run Options Window Help
import os, sys
os.chdir("c:/Temp")
mycfy = open("cfy.txt", "r")
i=0
for line in mycfy:
    i=i+1
    print("Line",i,"is:",line)
Ln: 7 Col: 30
```



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Line 1 is: Common First Year
Line 2 is: King Saud University
Line 3 is: 2019
>>> |
Ln: 10 Col: 4
```

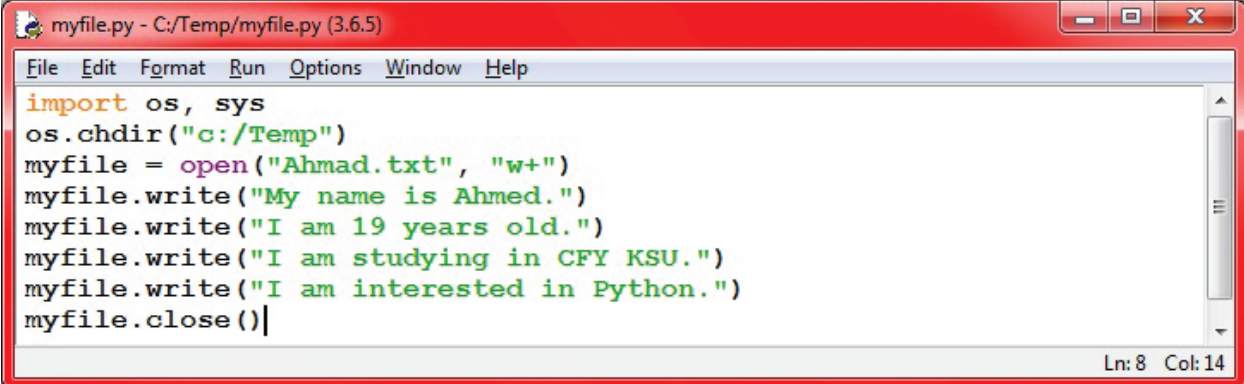
Writing Data to File

Before we write to a file, let's create our own file and name it. Using the open file statement `open("filename", "mode")` with mode (W+) which opens a file for both writing and reading. It overwrites the existing file if the file exists and creates a new file for reading and writing if the file does not exist.

Once you have your file ready then you can write to it using the `write()` function which writes any string to an open file. It is important to note that Python strings can have binary data and not just text. The `write()` function does not add a newline character (`'\n'`) to the end of the string. The syntax of write function as follows:

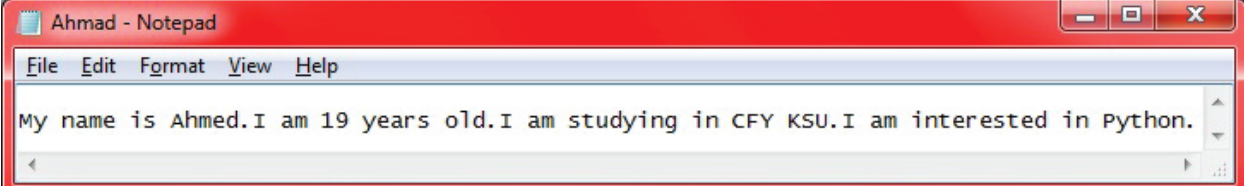
```
fileObject.write(string)
```

Let us create a file with your name and write your details in it.



```
myfile.py - C:/Temp/myfile.py (3.6.5)
File Edit Format Run Options Window Help
import os, sys
os.chdir("c:/Temp")
myfile = open("Ahmad.txt", "w+")
myfile.write("My name is Ahmed.")
myfile.write("I am 19 years old.")
myfile.write("I am studying in CFY KSU.")
myfile.write("I am interested in Python.")
myfile.close()
```

Now run the above program and it should create a file called "Ahmad.txt" with your information stored in it. To open the file, go to C:\Temp and double click on the file to open it.



```
Ahmad - Notepad
File Edit Format View Help
My name is Ahmed.I am 19 years old.I am studying in CFY KSU.I am interested in Python.
```

Closing a File

When you are finished working on your file, you can use the `close()` function to close it. This will close the file completely and terminate resources in use. It is important to understand that when you use the `close()` function, any further attempts to use the file object will fail. The syntax of closing a file is:

```
fileObject.close()
```

We have already closed the file "Ahmad.txt" in the above program with the following:

```
myfile.close()
```



READING SECTION

Accelerate Your Learning & Gain The skills

Here is a list of some binary file modes that can be used while working with binary files.

Modes	Description
rb	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is the default mode.
rb+	Opens a file for both reading and writing. The file pointer is placed at the beginning of the file.
wb	Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
wb+	Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
ab	Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
ab+	Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.



Apply

Activity
1

Write a program to create a file with the name "KSU.txt" and add three things which you like in the KSU.

Activity
2

Write a program to display contents of a file on the screen.

Activity
3

Write a program to read a file and display its contents with line numbers on a screen.

Activity
4

Write a program to copy one file to another.



Closing

Dear student, you have just learned how to use files in programming languages. We mentioned in this session some famous file operations that we can apply to a file. You can now refer to the Python website and explore more examples where we can use files and practice some more file modes and improve your previous programs.

CHAPTER

5

Session

1

AN OVERVIEW OF INFORMATION SECURITY



Goal

This session provides

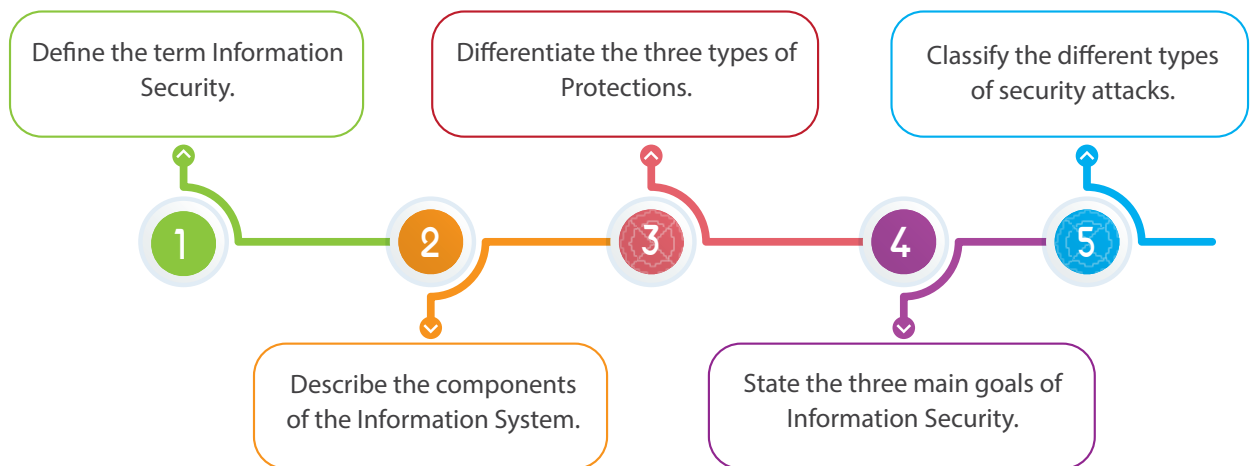
an overview of the field of Information Security, its main goals, and different types of Security Attacks.



Learning objective

Dear Student

By the end of this Session, you should be able to:



Be prepared

Discuss, in pairs, the following questions:

- 1 - What is Information Security all about? What are its advantages?
- 2 - What are the consequences of not following the practices of information security?
- 3 - Have you ever been a victim of a Security Attack? If yes,
 - A - What have you lost or suffered from?
 - B - What you could have done to prevent such an attack from happening?



List of Terms

Term	Explanation
Attack	Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself.
Authorized user	A person who possesses the authority to perform an action or set of activities.
Unauthorized user	A person who does not possess the authority to perform an action or set of activities.
Unauthorized disclosure	Exposure of information to user(s) not authorized to access the information.
Unauthorized modification	Modification of information by user(s) not authorized to modify the information.
Unauthorized disruption	Unauthorized user interrupts or prevents the correct operation of system services and functions in order to make information unavailable.
Unauthorized destruction	Unauthorized user destroys information in order to make it unavailable.
Encryption	Converting the data (Scramble) into a form that is not understandable .



Learn

Introduction to Information Security



Information security is a concept that has become of great importance in our lives. This is because our adoption of computing technology has become universal. We work daily on computers, go to school online, buy goods from merchants on the internet, play on computers at home, work for our employers, carry our smartphones around with us and use them to check our bank balances, send and receive emails, track our exercise with sensors in our shoes, and so on.

While the technology has increased our productivity and simplified our access to vast information, it raises various security issues. For example, if an attacker is able to access our bank information, the consequences can be severe. About 30 years ago, such security issues almost did not exist. This is because technologies were not implemented as widely as nowadays, and fewer people were using the technology at that time.



Information Security is a field that involves the study of practices of protecting our information wherever it is (in the Information System) from those who would seek to misuse it.



The term "Information Security" is defined as protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction.

Information Security is not only concerned with protecting the information itself, but also protecting each component of the Information System.

So let's understand the different components of the Information System that need to be protected.

Components of an Information System

As depicted in figure below, the different components of the Information System (IS) are :

Procedures

People

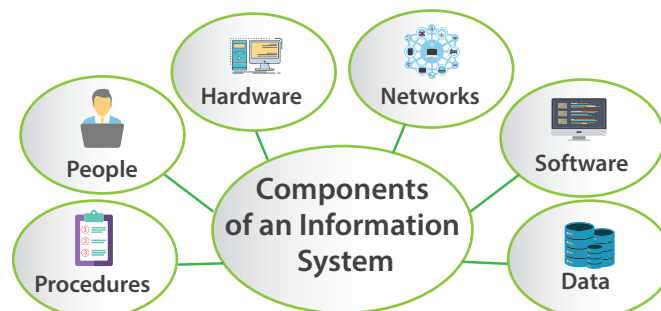
Hardware

Networks

Software

Data

All these components play an important role in producing the desired information and presenting, storing, or transferring it.



Procedures



Procedures are another component of the IS that also is often neglected in information security consideration. Procedures are written instructions to accomplish a specific task in an Information System. Organizations usually distribute procedures to their employees so that they can access the information system, but without proper protection of procedures. Protection of Procedures is of great importance. For example, similar to all critical information, Procedures should be distributed to employees or users on a need-to-know basis.

People



People, the users of the IS, are also an important component of the IS, which is often neglected in information security considerations. People always constitute a threat to information security, and they are considered the weakest link in an organization's security program. There must be security policies that prevent people from accidentally or intentionally damaging or losing information.

Hardware



Hardware consists of the physical technologies installed in the IS that host and execute the software, store and transport the data, and provide interfaces for entering or removing the information in the system. This component of the IS is considered as physical assets which must be protected from harm, theft, and unauthorized access or interaction. The physical location of computers and the computers themselves must be secured because a breach of physical security can result in a loss of information.

Networks



Information Systems are often connected together to form Networks, such as a Local Area Network (LAN). This component of the IS has increased the necessity of information security. Threats to the security of Information and Information Systems increase when organizations work on LANs, and they become worse when these LANs are connected to the Internet. Extreme focus is required to secure this component of the IS.

Software



Software consists of the applications, operating systems, and command utilities that are being used in the Information System. This component of the IS is considered the most difficult to secure.

Data



Data consists of the facts and figures that are stored, processed, or transmitted in and out by the Information System. This component of the IS is considered the most valuable to an organization and the main target for intentional attacks. Therefore, it must be highly protected.

Thus, in order to protect information, each component of the information system must be protected as well. These components are different from each other, and each of them requires a different kind of protection.

Types of Protections

There are different types of protections that can be applied for protecting the different components of the IS. Following are the three main types of Protections in the Information Security context:

Physical Protection

The Hardware component of the IS requires Physical Protection.

Physical Protection involves the use of physical security policies, such as locks and keys that restrict access to the area where the hardware components are located.

Awareness-Based Protection

The People and Procedures components of the IS are protected by Awareness-Based Protection.

Awareness-Based Protection involves education and training to prevent users from accidental or intentional misuse of information or procedures.

Technical Protection

Software, Data, and Networks are protected by Technical Protection.

Technical Protection involves the use of security technologies to manage access to, and usage of, a system or information by different users.

Examples of such security technologies are Access Control systems, Cryptography, firewalls, etc.

Two main security technologies that we will be focusing on in this chapter are:

Access Control



Access Controls involve processes that allow us to limit access to a system or information by the users and to control and manage it. Users must provide User IDs, Passwords, biometric identifiers, etc., to get access to a system or information. You will learn more about Access Controls in Session 2 of this Chapter.

Cryptography



Cryptography allows us to keep information secret. It involves generating codes to convert (scramble) data into an unreadable format so that an unauthorized user can't understand or misuse it. To recover the data, the authorized user requires a key. You will learn more about Cryptography in Sessions 3 and 4 of this Chapter.

TIP

Technical Protection can also be additionally applied for the protection of People and Procedures to prevent individuals from intentionally misusing knowledge in the Procedures.

Main Goals of Information Security

The main goals that underlie information security are to achieve Confidentiality, Integrity, and Availability of information, commonly known as the CIA triad.



1 - Confidentiality



“**Confidentiality is the ability to prevent unauthorized disclosure of information so that sensitive information is accessible only to authorized users.**”

Every one of us has sensitive information that we want to keep confidential and not have disclosed to unauthorized users.

For example

A patient's record in a hospital is a kind of sensitive information that should be confidential. Providing such information to third parties is unacceptable.

2 - Integrity



“**Integrity is the ability to prevent unauthorized modification of information in order to maintain its accuracy, consistency, and trustworthiness over its entire lifecycle.**”

In other words, Integrity is a property that is concerned with allowing sensitive information to be modified only by authorized users.

For example

In a hospital, only the doctors should be able to modify records of the assigned patients. Modification of such records by unauthorized users might lead to wrong treatment for the patient.

There are times or places where Integrity is more important than Confidentiality.

3 - Availability



“**Availability is the ability to prevent unauthorized withholding of information so that it is available when needed by the authorized users.**”

Although it might seem less important than Confidentiality and Integrity, there are situations where the Availability of information is of great importance.

For example

A patient's record should be available to the authorized users at all times in a hospital. In emergency situations, getting access to such information as quickly as possible is very important; if the information is not available on time, there could be loss of lives due to delayed treatment.

Types of Security Attacks.

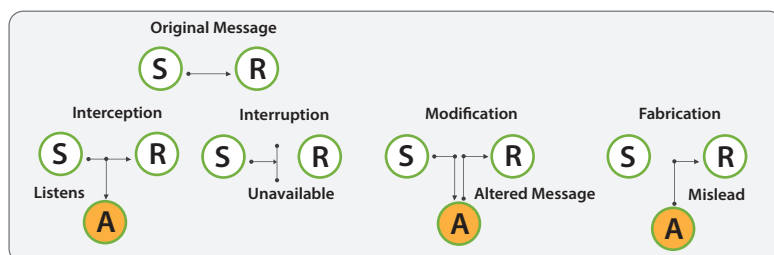
A security attack is one of the main concerns in information security. It can come in different forms. When performed by unauthorized users (outsiders), it is known as an **External Attack**; when performed by authorized users (insiders), it is known as an **Insider Attack**. It is more difficult to detect or prevent insider attacks than external attacks, because insiders are authorized users who operate within their privileges.

Insider and external attacks can be further classified into four categories:

- 1 - Interception Attacks
- 2 - Interruption Attacks
- 3 - Modification Attacks
- 4 - Fabrication Attacks



Each type of attack can affect the Confidentiality, Integrity, or Availability of the CIA triad, as depicted in Figure aside.



Confidentiality	Interception
Integrity	Interruption
	Modification
	Fabrication
Availability	Interruption
	Modification
	Fabrication

1 - Interception Attacks

Interception attacks allow an attacker to gain access to sensitive information. Interception might take the form of unauthorized file viewing or copying, eavesdropping on phone conversations, or reading email. This type of attack affects the confidentiality of the information.

2 - Interruption Attacks

Interruption attacks allow an attacker to make sensitive information unusable or unavailable to authorized users. This type of attack affects the availability of the information. It might affect the integrity of the information as well.

For example, if an attacker overwrites files in a file system in order to prevent access to the data they contain, we might consider this an integrity attack, due to the possible corruption of data.

3 - Modification Attacks

Modification attacks affect the integrity of the information. They allow an attacker to tamper with sensitive information. This type of attack might affect the availability of the information as well.

For example, if an attacker tampers with a configuration file that manages how a particular service behaves, this might affect the availability of that service by changing the contents of the file.

4 - Fabrication Attacks

Fabrication affects the integrity of the information. It is a type of attack in which the attacker inserts forged objects or information into the system, as if it was generated by an authorized user. This forged information misleads authorized users when they use it. In addition, fabricated data may reduce confidence in genuine data within the affected system. This type of attack might affect the availability of the information as well.

For example, if an attacker generates enough network traffic, email, or nearly anything else that consumes resources, this can potentially render the service that handles such traffic unavailable to authorized users of the system.



Apply

Read the following cases and note down your responses in the boxes provided below.

Activity
1

A - Consider that you are a Security Officer in a Bank and you have a flash drive (USB) containing secret financial details and security codes of the banks. You want to protect this flash drive from theft or misplacing.

1 - Note down the possible security measures you will follow to protect your flash drive (USB).

2 - Match your security measure with the appropriate type of Protection you have learned in this Session.

3 - Consider that you are using your laptop on a coffee shop, which provides free open Wi-Fi. Will you use that flash drive (USB) on your laptop? Why or why not?

B - You are planning to buy a car on loan from a bank. An employee of the bank has asked you to fill out a form (PDF) and send it to him via email. In the PDF form, it asks details about your financial sources and also about your family. You filled out the form correctly, but while sending the email, you mistakenly sent it to a wrong person.

4 - Which of the goals of Information Security is affected in this case?

5 - How can you avoid this kind of mistake in the future? List your suggestions.

C - When you start chatting with someone through an application on your smartphone, sometimes it gives a message like "This chat is secured with end-to-end encryption."

6 - What do you understand from this message? And which goal(s) of Information Security are achieved with this?

**Activity
2**

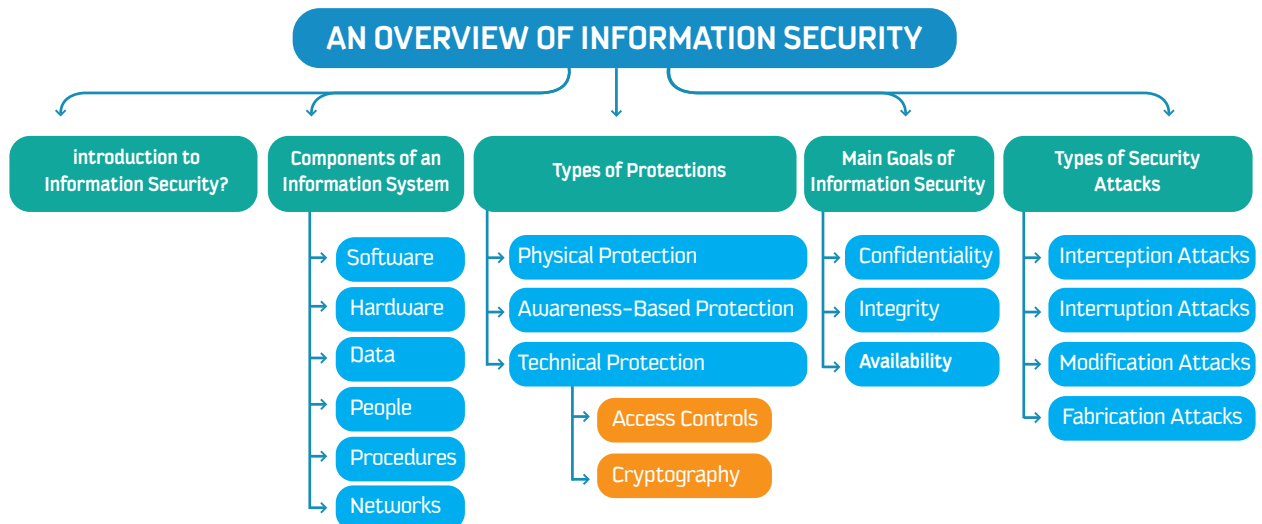
Choose the correct option for the blanks in the following statements.

- | | |
|---|---|
| a - _____ is/are the main goal(s) of Information Security. | a) Availability
b) Confidentiality
c) Integrity
d) All of the above |
| b - _____ attacks insert fake objects or information into the system. | a) Modification
b) Interception
c) Fabrication
d) All of the above. |
| c - _____ is the assurance that the information is trustworthy and accurate. | a) Integrity
b) Awareness-Based Protection
c) Confidentiality
d) Information Security |
| d - Interception attacks are generally designed to target and challenge the _____ pillar(s) of the CIA Triad. | a) Confidentiality
b) Availability & Confidentiality
c) Integrity & Availability
d) All of the above |
| e - _____ involves providing Education and Training to employees in an organization. | a) Professional Development
b) Integrity
c) Awareness-Based Protection
d) None of the above |



Closing

Dear student, Following is a mind map that summarizes the concepts and knowledge you have learned in this session.



CHAPTER

5

Session

2

ACCESS CONTROL



Goal

This session provides

students with an introduction to Access Control as one of the Technical Protection Methods in Information Security.



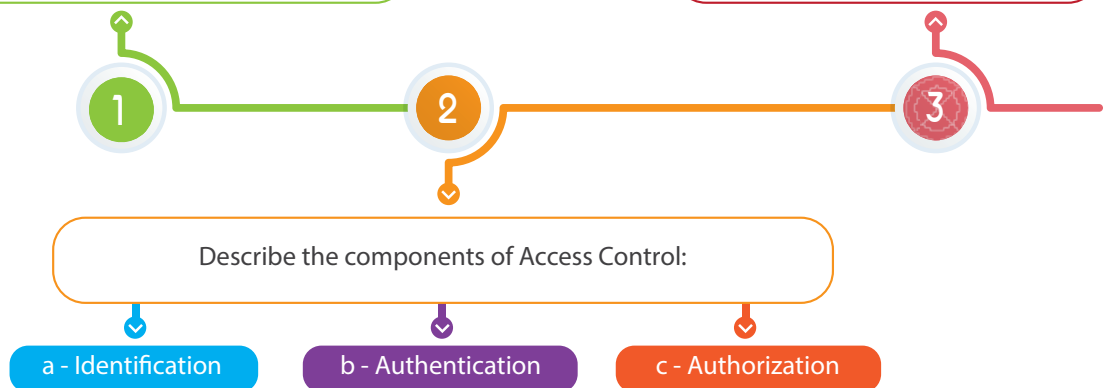
Learning objective

Dear Student

By the end of this Session, you should be able to:

Understand the concept of Access Control and its importance in Information Security.

Understand the different Access Control Models.



Be prepared

Discuss, in pairs, the following questions:

1 - Do you come across situations in everyday life where you need to prove your identity? The People who ask for your identity—what do they do after that, and how do they manage their work?

2 - Have you ever learned about / thought about / experienced managing and controlling access to a resource?

For example, access to your car by many drivers. How do you manage or control it?

List of Terms

Term	Explanation
Subject	A user (or program of a user) that needs access to data or resources.
Object	The data or resources of the system.
Pass-phrase	Usually much longer than a password, an actual phrase of some type; easier for the user to remember.
Cognitive password	Questions about facts or predefined responses that only a user knows the answers to.
Token device	Handheld device with an LCD display.
OTP	One-Time Password.
MAC	Mandatory Access Control
DAC	Discretionary Access Control
RBAC	Role-Based Access Control
ACL	Access Control List



Learn

Introduction to Access Control



Information security aims to achieve Confidentiality, Integrity, and Availability of information. This can be achieved by allowing only authorized people to perform actions while disallowing the actions of unauthorized people.

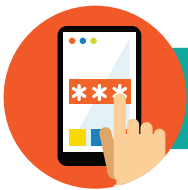
For instance,

confidentiality is met if only authorized people can view or access the information;

integrity is met if only authorized people can modify the information;

and **availability** is met if only authorized people can withhold the information.

Thus, there must be a method to differentiate authorized from unauthorized people. This is what Access Control does.



The major function of Access Control is to manage access rights of users when fully sharing the system resources, and to ensure that illegal use of and access to the resources cannot occur.

Access Control limits the access of the subject (**user**) to the object (**data**). It controls the subject's access according to his/her authentication.

Access Control is one of the central themes of information security; it helps to achieve confidentiality and integrity of data.

Components of Access Control

Access Control involves performing three steps as follows:



- 1 **Firstly**, each person must have a unique identity to identify him/herself to the system; this step is known as **Identification**.
- 2 **Secondly**, each person must prove to the system that he/she is really who they claim to be; this step is known as **Authentication**.
- 3 **Finally**, the system must define what the authenticated person can or cannot do with the information; this step is known as **Authorization**.

Thus, Access Control comprises three components; **Identification**, **Authentication**, and **Authorization**. Each complements the other, and they must be implemented in order. Let's discuss each one in detail.

Identification

The first step in access control is identification.

“**Identification is the process by which a user agrees an identity and responsibility is initiated.**”



Hence, the identification process is established when a user provides a username, a log-in ID, a personal identification number (PIN), or a smart card. Once a user has identified himself, his identity will be responsible for further actions undertaken by him. Computers are not able to differentiate humans from each other but rather are able to differentiate your user account from all other user accounts.

Identification is about providing a public piece of information (username, account number), and it might be known by a subject's friends or family. Thus, identification does not play an important role in making the system more or less secure. However, it is only the first step and the starting point that facilitates, introduces, and is relied on by the two most important steps: **Authentication** and **Authorization**.

Authentication

The second step in access control, Authentication, relies on the identification step. While identification is about providing a public piece of information, authentication is about providing a private piece of information that is known solely by a certain user.

“**Authentication is defined as the process of verifying or testing that a claimed identity is valid.**”



In the authentication step, additional information is needed from a user, and it must correspond exactly to the identity declared. **A well-known example of authentication is a password.**

Authentication techniques can be classified into three types based on their characteristics, as follows:

- a) **Something the user Knows**
- b) **Something the user Has**
- c) **Something the user Is**

Something the user Knows

Authentication Techniques such as *Password, Pass-phrase and Cognitive Password* belong to the category of “Something the user Knows.”

Password is the most commonly used technique for authentication processes; it is considered the weakest.

Pass-phrase, which is a string of characters usually much longer than a password, is more efficient and effective than passwords. A pass-phrase is easier for the user to remember than passwords, as it is an actual phrase of some type. Pass-phrases also provide a higher level of protection than passwords, as they are longer and harder for an attacker to guess or break.

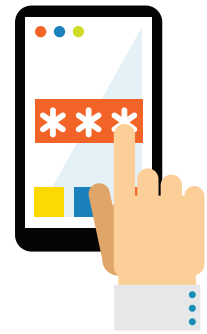
Cognitive password, which is a number of questions about facts or predefined responses that only a user should know, is efficient.

For example

Examples of cognitive password are “What is your birth date?”, “What is your mother's maiden name?”, and “What is the name of your division manager?”

An effective cognitive password system asks a different set of questions each time. Cognitive passwords are more difficult for an attacker to break, and they are usually easier for a user to remember.

Although making a long, complicated password is preferred to survive attacks and make this type of authentication stronger, humans are not able to remember long, complicated passwords that consist of different alphanumerical characters. Even if a user is forced to create such passwords, the user would write them down somewhere, as he/she will not be able to remember them; hence, they are susceptible to physical theft.



Thus, this type of authentications is not efficient to be used alone, and it is usually used in combination with type 2 (something the user has) or type 3 (something the user is).

Something the user Has

There are several authentication techniques that can be classified in the category of “Something the user Has,” such as One-Time Password (OTP), Memory card, and Smart card.

One-Time Password (OTP)

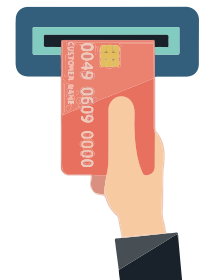
One-Time Password is dynamically generated and it is valid only once. The advantage of using One-Time Passwords is that when the password is known by an attacker, it is useless; the attacker cannot use it. In order for a user to obtain the one-time password, he/she must possess a handheld device with an LCD display, referred to as a Token device. For the authentication process to be performed, the user first reads the generated password from the Token device and then enters it, along with a username, into a system for the authentication process. This generated password can only be used once, and if the user wants to go through the authentication process again, another password is generated dynamically for the user to perform the authentication process again.



Memory card

Memory card is another technique to use in this type. Memory cards contain information within a magnetic strip and rely on a reader of some sort to process the information. For a user to be authenticated in this technique, he/she first inserts the card into the reader and then enters a set of credentials to be authenticated.

A well-known example of Memory card is an Automated Teller Machine (ATM). The user inserts the ATM card into the ATM and then enters his/her PIN number. The card provides the account number (user information) while the user provides the secret code (PIN); together, they provide a credential set.



Smart card

Smart card is different from a Memory card. It has a microprocessor and integrated circuits, which make the smart card able to process information rather than only containing the information as with a Memory card. While using the smart card, the user must insert it into a reader which has electrical connections that switch on and power the smart card processor. The user then enters a PIN value, which unlocks the information and processing power contained on the smart card. Smart cards are considered more tamper-proof than memory cards as a smart card locks itself if the user has entered

Brainstorming

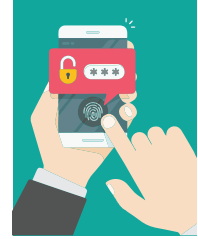
It is also possible to authenticate based on something you have, that is, something in your possession. Give an example of authentication based on something you have.

TIP

Cryptographic keys can be used for the Authentication process; this can be classified under the “Something that user Has” type of Authentication Techniques. You will learn about Cryptographic Keys in Session 3 of this chapter.

Something the user Is

The authentication process based on “Something the user Is” is considered the strongest and most efficient. It is suitable for organizations that require a very high level of security. This type of authentication is usually referred to as Biometric Authentication. Biometrics measure unique physical or behavioral characteristics of a user to recognize or authenticate his/her identity. Physical biometrics include fingerprint, hand or palm geometry, retina, iris, and facial characteristics. Behavioral biometrics include signature, voice, keystroke pattern, and gait.



The acceptance of biometric authentication by society is very low, and the implementation is very expensive compared to other authentication types.

People do not prefer biometric authentication because of the associated inconvenient processes that a user must go through to be authenticated, and also because it reveals personal information about them. As a result, it can only be seen in organizations that require a very high level of security, where confidentiality is more important than people’s convenience and preference.

Multi-Factor Authentications

The three types of authentication techniques provide different levels of security; each type is stronger than the one preceding it. In other words, type 3 of the authentication techniques is the strongest type because it is the most difficult to breach, compared to the other two types.

Deciding which Authentication technique to use depends on the level of security desired by a certain organization: whether it is a low, moderate, or high level.

Brainstorming

When you want to authenticate yourself to your computer, most likely you type in your username and password. The username is considered public knowledge, so it is the password that authenticates you. Your password is something you know.

For example

Authentication techniques like Token devices are used by financially oriented organizations, while Biometrics are more suitable for military-oriented organizations where the data is top secret.

It is also possible to authenticate based on something you are, that is, a physical characteristic. Such a characteristic is known as a biometric. **What is an example of biometric-based authentication?**

To satisfy the levels of security desired by different organizations, one or two or all three types of Authentication techniques can be used at a time.

When only one of the three types of Authentication techniques are used, it is referred to as

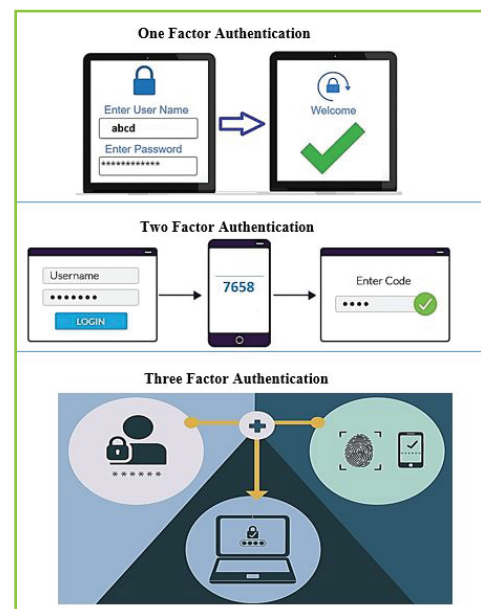
One-factor authentication.

When two of the three types of Authentication techniques are used, it is referred to as

Two-factor authentication.

Whereas if all three types of Authentication techniques are utilized, it is referred to as

Three-factor authentication.



In order to have a strong authentication process, the authentication must utilize two-factor or more. An example of utilizing two-factor is when a subject/user uses a debit card at a shop. He must swipe the card (something he has) and enter a PIN (something he knows) to complete the transaction.



Brainstorming

Two-factor authentication requires that two of the three authentication methods (something you know, something you have, something you are) be used. Give an example from everyday life where two-factor authentication is used. Which two of the three are used?

Authorization

Authorization is the third and final step of access control. It is performed after a user has been identified and authenticated.

“Authorization is the process of assigning access and the right to carry out specific operations to authenticated users, depending upon their preconfigured access rights and permissions outlined in access criteria.”



It must be noted that not every identified and authenticated user can use all resources. Instead, after a user is identified and authenticated, a check must be performed to find out what accesses and operations he/she can perform. In other words, by performing the Authorization step, we can determine what an identified and authenticated user can actually access and what operations he/she can carry out.

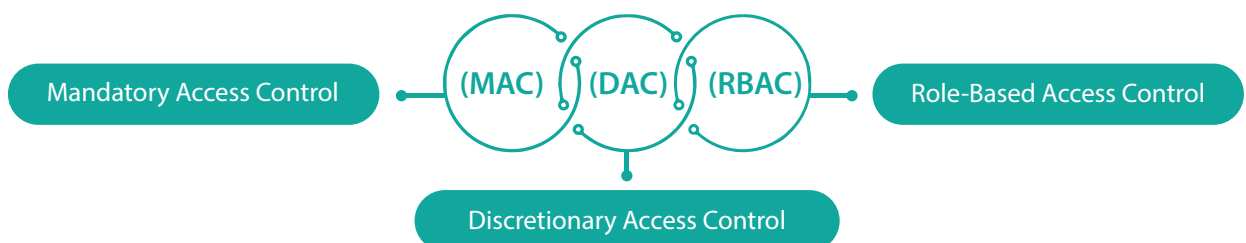
Authorization is provided by a system through Access Control Models, which manage the type and level of each user's access to objects. A wide variety of Access Control Models exist, each of which provides a different approach to Authorization. The following section gives a brief description of the major types of Access Control Models.

Access Control Models

“An Access Control Model is a framework that dictates access control using various access control technologies.”

The main objective of an Access Control Model is to enforce the rules and objectives of certain security policies. In other words, it dictates how objects must be accessed by subjects/users.

There are different Access Control Models that focus on different goals of Information Security (the CIA triad). Some Access Control Models focus more on Confidentiality of data, whereas others focus more on data Integrity. Each Model has its own advantages and disadvantages and serves different organizations' needs, culture, nature of business, security policy, etc. **Traditionally, there are three widely accepted Access Control Models:**

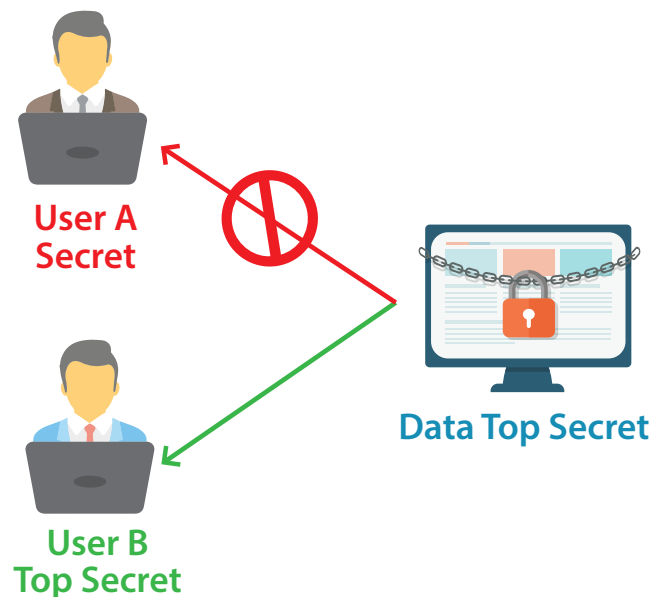


1 - Mandatory Access Control (MAC) Model

In Mandatory Access Control, each object is attached to a security label and each user is assigned to a security clearance.

Access restriction in the Mandatory Access Control Model is based on the security clearance of users and security labels of objects.

In order for a subject/user to access an object, the subject's clearance level must be equal to or greater than the object's label level. For instance, if an object is very confidential in an organization and has been assigned a "Top Secret" security level, a user who has been assigned the lower security level "Secret" cannot access the object, as his/her security level does not match the object's.



As this model is named mandatory, the security levels are enforced by the system and only the administrators of the system can modify them. In other words, it does not allow a user or a program of the user to modify the security levels. That makes this model stricter and more secure, suitable for organizations that require a high level of security, such as the military.

But this model is neither flexible nor scalable.

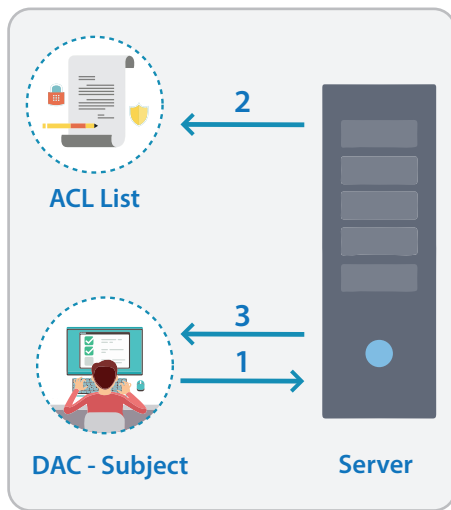
2 - Discretionary Access Control (DAC) Model

In contrast to MAC, Discretionary Access Control Model gives the owner or the creator of an object the freedom to specify who can access the object and what operations can be performed on the object.

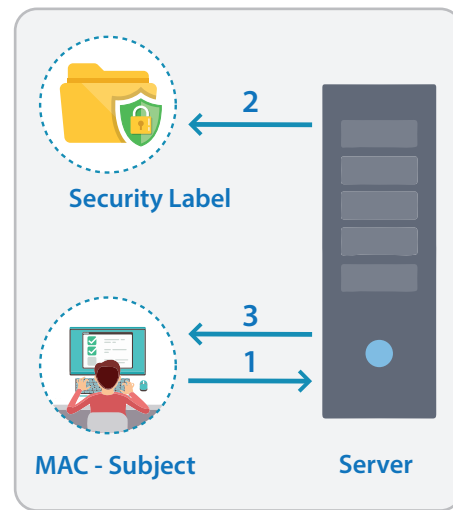
For instance, if a subject creates a file on his system and wants to share it with other subjects, the subject can control and specify who can access the file. In other words, the access control is based on the discretion or the decision of the owner.

DAC is often implemented using an access control list (ACL) of objects, where each ACL defines the types of access granted or restricted to an individual or a group of subjects.

Unlike MAC, DAC is applicable for organizations that do not require a high level of protection.



DAC Model



MAC Model

In Discretionary Access Control, a subject is permitted to access an object based on the identity of the subject, and some subjects can also delegate their own access authorities to other subjects.

However, this has led this model to be flexible but also complex.

Example of MAC and DAC

In building operating systems, the decision to select MAC or DAC depends on what functionality an operating system is intended to provide. For instance, Windows-based platforms provide DAC access structure rather than MAC. However, specially developed operating systems, such as those created for government agencies and the military, provide a MAC access structure to enforce the level of security needed. Many operating systems, such as Linux, Unix and Windows NT/Server, use DAC access structure.

3 - Role-Based Access Control (RBAC) Model

The Role-Based Access Control Model was proposed to overcome the complexity problem associated with the MAC and DAC models.

In the Role-Based Control Model, access to objects is restricted based on the business function or role that subjects/users are performing.

Unlike DAC, access permissions in RBAC are assigned to roles rather than to subjects' identifiers. In this model, groups of users are created by the administrators, who then assign access rights and permissions to the groups. A user in a group will be able to utilize the access rights and permissions of the group.

This model is more appropriate for large organizations that are required to change access rights and permissions more often. This is because this model allows the administrators to add a subject, add an object, or change access rights and permissions very easily by changing centralized roles without having to manipulate any subject or object in the system. For instance, in a company, the administrator can add a new employee to a role rather than creating access rights and permissions for every person who joins the company.



Apply

Activity 1

Do the following Access Control Exercises by using your Google Drive account.

- 1 - Authorize your class friend(s) to only be able to view a Word file stored in your Google Drive account.
- 2 - Authorize your class friend(s) to view and edit a Word file stored in your Google Drive account.
- 3 - Share a Word file stored in your Google Drive account by using the sharable link feature.

Activity 2

Choose the correct option for the blanks in the following statements.

1. _____ is a number of questions about facts or predefined responses that only the user should know.
 - a) Password
 - b) Pass-phrase
 - c) Cognitive Password
 - d) Subject

2. When one of three types of Authentication techniques is used, it is referred to as _____.
 - a) Two-factor authentication
 - b) One-factor authentication
 - c) Three-factor authentication
 - d) None of the above

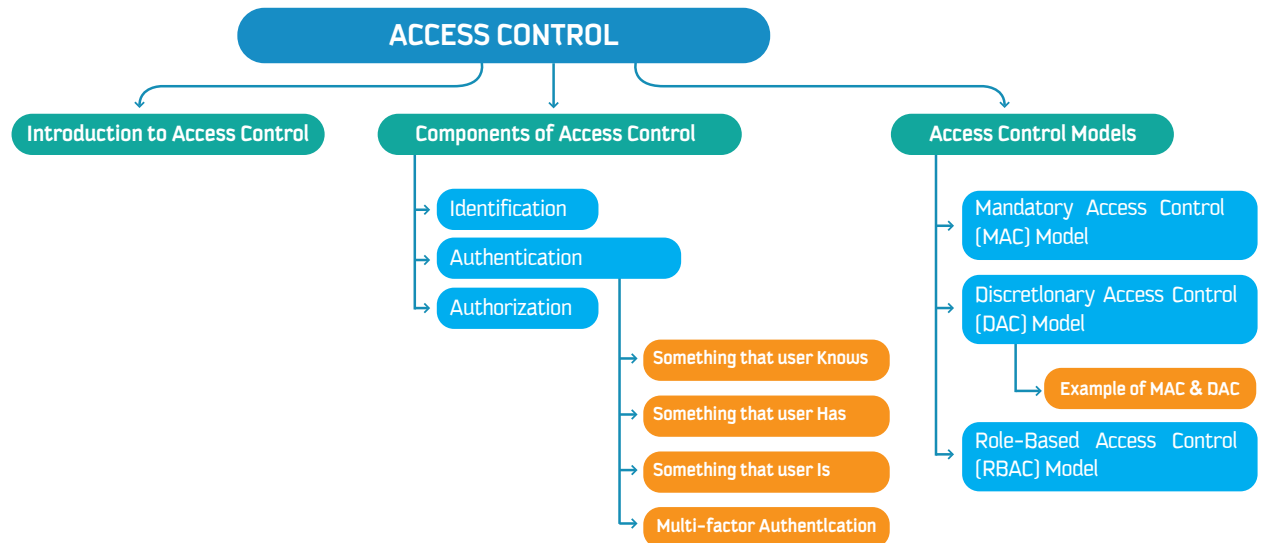
3. _____ is the process of assigning access and the right to carry out specific operations to the authenticated users.
 - a) Identification
 - b) Password Protection
 - c) Role-Based Access Control
 - d) Authorization

4. OTP stands for _____.
 - a) One-Time Password
 - b) One-Time Protocol
 - c) One-Way Protocol
 - d) None of the above



Closing

Dear student, Following is a mind map that summarizes the concepts and knowledge you have learned in this session.



Session

CHAPTER

5

Session

3

Introduction to Cryptography
Part-I



Goal

This session provides

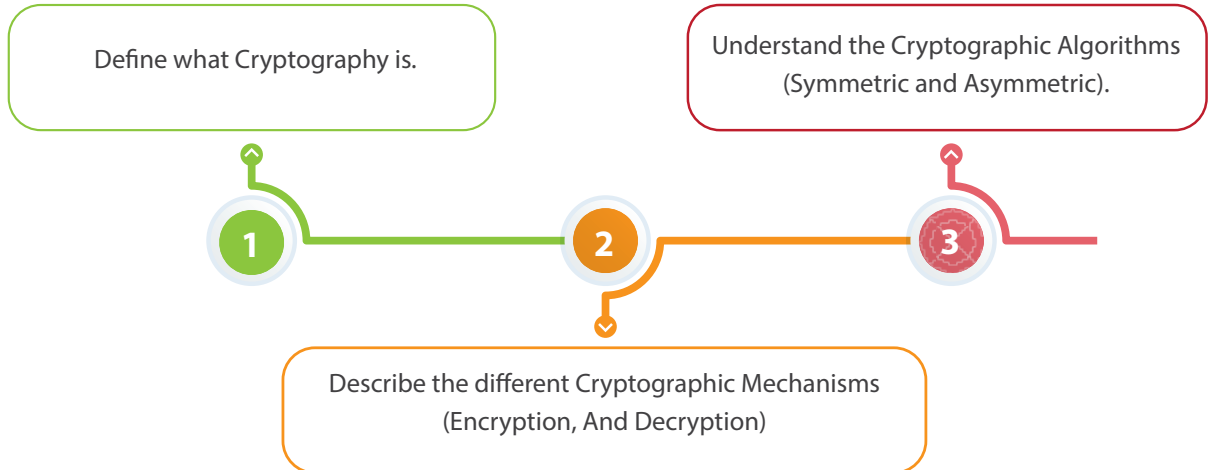
students with an introduction to Cryptography, its importance in the Information Security context, and Cryptographic Mechanisms and Algorithms .



Learning objective

Dear Student

By the end of this Session, you should be able to:





Be prepared

Dear students, discuss in pairs the following questions:

1 - How secure is our digital information? Is our digital information protected from natural disasters, computer hackers, and human errors?

2 - When checking your email over a secure connection, or making a purchase from an online retailer, have you ever wondered how your private information or credit card data is kept secure?

3 - In ancient times, how did people share secret messages during world wars?

List of Terms

Term	Explanation
Plain Text	The original message (or) an original readable textual material message or data.
Cipher Text	The coded message (or) a non-readable form of the original text.
Cipher	An algorithm for transforming plain text to cipher text.
Encryption	Process of converting plain text to cipher text or process of coding information into a form that is unreadable.
Decryption	Restoring the original plain text from the cipher text.
Key	A value that is used to encrypt or decrypt a message. It can include numeric and alphanumeric text and special symbols.
Algorithm	In the context of encryption, the mathematical formula used to scramble and unscramble data.
Eavesdropper	Unauthorized listener (spy).

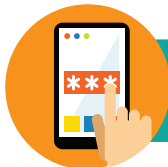


Learn

Cryptography



Cryptography is a science of secret writing and is an ancient art. The aim is to hide the meaning of a message so that it is unreadable by eavesdroppers. Cryptography has a definitive existence in everyday life. There are a range of situations where we want our data to be undisclosed. Everything from our smartphone to our banking relies heavily on cryptography to keep our information safe and our life secure.

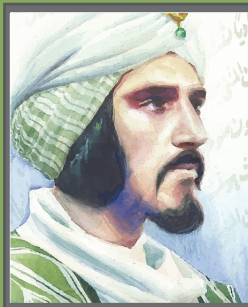


Cryptography is the study of methods used to transmit information securely between two or more participants, so that it is unreadable by unauthorized users.

Security of information in communications, computer systems, electronic commerce, and the emerging information society depends on cryptography. Cryptography can protect communications that traverse untrusted networks. Cryptography is an essential tool to achieve the basic goals of Information Security: Confidentiality, Integrity, and Availability.

Cryptanalysis is the science of breaking cipher texts without knowledge of the key. Cryptography and Cryptanalysis are known as Cryptology.

The Arabic Origins of Cryptography



During the golden age of the Islamic Civilizations, the Arabs translated many branches of knowledge from other languages and considerably added and enriched them. However, the Arabs were the first to formalize Encryption and Cryptanalysis. David Kahn, a prominent historian of Cryptography, wrote "Cryptology was born among the Arabs. They were the first to discover and write down the methods of cryptanalysis." [DK] Many Arabic manuscripts were discovered in 1979 that show the earliest work on Encryption and Cryptanalysis was done by Ya'qub al-Kindi in his Treatise "Risalah fi Istikhraj al Mu'amma" (Treatise on Decrypting Cryptographic Messages), written in the 9th century AD, which is about 1200 years ago. [MAT] Many Arab scholars have written original works on Encryption and Cryptanalysis after al-Kindi, and possibly before him.

Cryptographic Mechanisms

In Cryptography, a variety of cryptographic mechanisms exist to support different security services: data confidentiality, data integrity, etc. **Some of the popular cryptographic mechanisms are as follows:**

- 1- Encryption & Decryption**
- 2- Hash Functions**
- 3- Digital Signatures**
- 4- Digital Certificates (Digital IDs)**

Encryption and Decryption Mechanisms are discussed in this session and other mechanisms (Hash Functions, Digital Signatures and Digital Certificates) are discussed in next session.

Encryption & Decryption – A Core Cryptographic Mechanism

Cryptography, prior to the modern age, was effectively synonymous with encryption, which means simply hiding or converting data.

“**Encryption** is a process of converting the data (referred to as “plain text”) into a form that is not understandable (referred to as “cipher text”).

Decryption is a process of converting the coded text (referred as “cipher text”) into a form that is understandable (referred to as “plain text”).

The conversion and subsequent recovery of the data depend on a code or a key, which can be a number or a string of characters.

For example, you want to send a coded message to your friend.

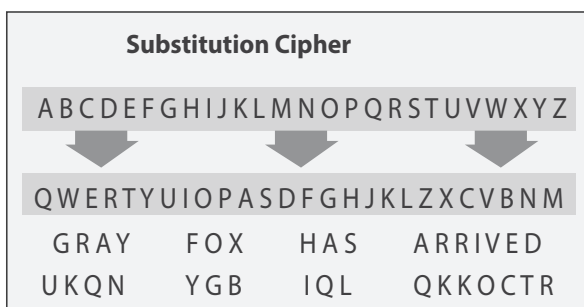
- You can substitute each letter of your message with the letter that is two down from it in the alphabet order. So “A” becomes “C,” “B” becomes “D,” and so on.
- You need to inform your trusted friend that the code (or the Key) is “Shift by 2.”
- Your friend gets the message and decodes it properly by replacing each letter with the letter that is two up in the alphabetic order.
- Anyone else who sees the message will see only distorted text.

The fundamental requirement is that no information be lost (that is, all operations are reversible).

Thus, Encryption is a cryptographic mechanism that helps to achieve the security service Confidentiality, whether the information is in storage or in transit.

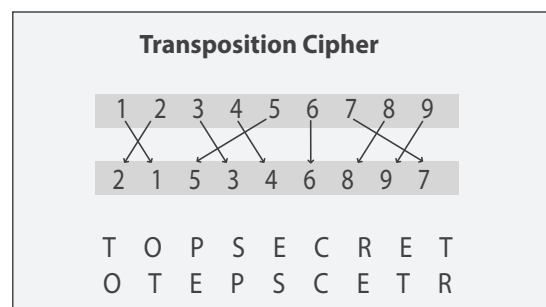
There are a range of techniques of encryptions. In other words, there are different ways to transform the “plain text” into the “cipher text.” Two of the techniques are basic:

Substitution Ciphers



In substitution, each element in the plain text (bit, letter, group of bits or letters) is mapped into another element.

Transposition Ciphers



Transposition Ciphers are a bit different from Substitution Ciphers. In transposition, elements in the plain text are rearranged or plain text characters are shifted in some regular pattern to form cipher text.

Practice on Basic Encryption – Substitution Ciphers

You have learned in how a message can be encrypted by using the Substitution cipher technique. Use the [CFY-KSU Cryptography Tool](#) and do the following.



Encrypt a message with the Substitution cipher technique by using the CFY-KSU Cryptography Tool. Send the encrypted message to your class friend through your KSU email.

(Hint: Use the option “Shift by 3”)

- 1 In the CFY-KSU Cryptography Tool, choose the “Basic Encryption” tab.
- 2 Enter your message in the “Plain Text” box.
- 3 Choose the “Substitution Cipher” option.
- 4 Choose the option “Shift by 3.”
- 5 Encrypt the message.
- 6 The Encrypted message will be shown in the “Cipher Text” box.
- 7 Copy the encrypted message and send it through your KSU email to one of your class friends.
- 8 Inform your friend of the encryption technique (Substitution cipher) that was used so that only he will be able to decrypt the message correctly.



Decrypt an encrypted message that you have received in your KSU email from your class friend by using the CFY-KSU Cryptography Tool.

Your friend has informed you that the Encryption technique Substitution cipher with “Shift by 3” was used while encrypting the message.

- 1 In the CFY-KSU Cryptography Tool, choose the “Basic Encryption” tab.
- 2 Copy the encrypted message from your email and paste it into the “Cipher Text” box.
- 3 As per the information from your friend (the sender), choose the “Substitution” option and choose the “Shift by 3” Key.
- 4 Decrypt and read the message from your friend.

Practices on Basic Encryption –Transposition Ciphers

You have learned in Session 3 how a message can be encrypted by using the Transposition cipher technique. Use the **CFY-KSU Cryptography Tool** and do the following.

1

Encrypt a message with Transposition cipher technique by using the CFY-KSU Cryptography Tool. Send the encrypted message to your class friend through your KSU email.

- 1 In the CFY-KSU Cryptography Tool, choose the “Basic Encryption” tab.
- 2 Enter your message in the “Plain Text” box.
- 3 Choose the “Transposition Cipher” option.
- 4 Encrypt the message.
- 5 The Encrypted message will be shown in the “Cipher Text” box.
- 6 Copy the encrypted message and send it through your KSU email to your class friend.
- 7 Inform your friend of the encryption technique (Transposition) that was used so that only he will be able to decrypt the message correctly.

NOTE

CFY-KSU Cryptography Tool uses the “Rail Fence” technique of Transposition Cipher.

2

Decrypt an encrypted message that you have received in your KSU email from your class friend by using the CFY-KSU Cryptography Tool.

Your friend has informed you that the Encryption technique Transposition cipher was used while encrypting the message.

- 1 In the CFY-KSU Cryptography Tool, choose the “Basic Encryption” tab.
- 2 Copy the encrypted message from your email and paste it into the “Cipher Text” box.
- 3 As per the information from your friend (the sender), choose the “Transposition” option.
- 4 Decrypt and read the message from your friend.

Cryptographic Algorithms

Cryptography involves the combination of mathematical and computational strategies, keys, algorithms, etc., to securely and reliably prevent unauthorized access to sensitive information and verify each component in a communication.

“A mathematical function/procedure for performing encryption and decryption on data is often known as a Cryptographic Algorithm.”

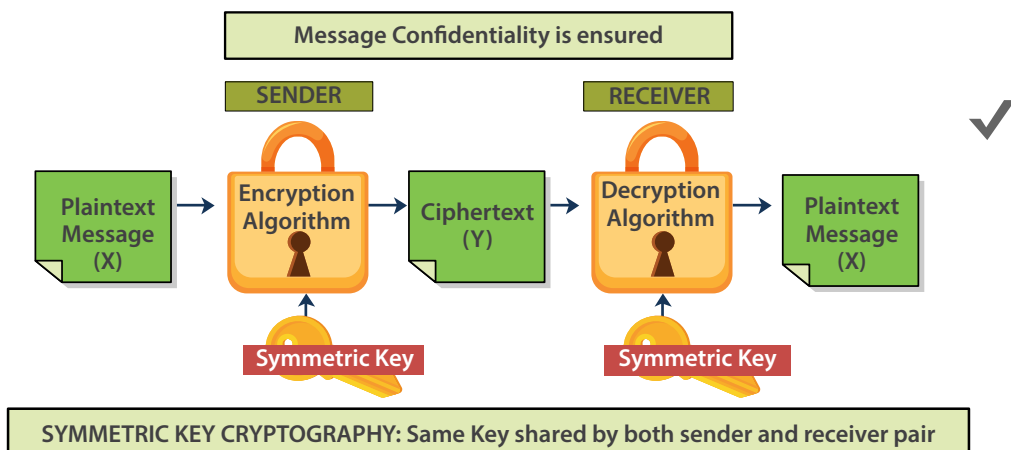
Through the use of an algorithm, information is made into meaningless cipher text; a key is required to transform the data back into its original form. The security of encrypted data is totally dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

Fundamentally, Cryptographic Algorithms are of two types:

- 1) Symmetric Cryptography
- 2) Asymmetric Cryptography

1 - Symmetric Key Cryptography

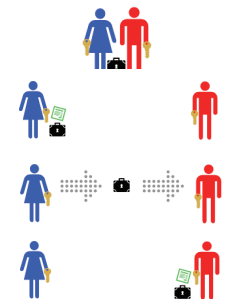
Symmetric algorithms encrypt and decrypt the data using the same key. The sender and receiver must have the same key in order to share confidential information. Because a single key is used for both encryption and decryption, this method is called symmetric cryptography.



For example, two users, Alice and Bob, want to exchange messages privately and securely.

In the Symmetric Key Cryptography setting, the following steps will occur:

- 1 - Alice buys a lockbox and two identical keys to open it.
- 2 - Alice arranges to meet Bob to give him one of the keys. (It's inconvenient, but she only has to do it once.)
- 3 - Alice uses her key to lock her message into the lockbox and sends the lockbox to Bob.
- 4 - Bob can use his identical key to unlock the lockbox and read the message. Bob can use the same method to securely reply.



Challenges in symmetric key cryptography

- **Distribution of key** – Secret key sharing among senders and receivers is a challenge. It gets even more inconvenient when the users are on opposite sides of an ocean.

If multiple parties are involved in the communication, then multiple keys have to be distributed. Managing many keys is difficult.

- **Authenticity** – Trust and Authenticity of the two parties is also a concern.

Symmetric key cryptography is more useful for encryption of files or file systems in local machines and less useful for communication between two systems in a network because of Key Distribution challenges.

Practice on Symmetric Key Cryptography

As you know in Symmetric Key Cryptography algorithm, the sender and receiver share the same key, which is used for both encryption and decryption. Use the [CFY-KSU Cryptography Tool](#) and practice Symmetric Key Cryptography as follows.

1

Generate a symmetric key using the CFY-KSU Cryptography Tool and share it with your class friend.

- 1 In the CFY-KSU Cryptography Tool, choose the “Symmetric Key Cryptography” tab.
- 2 Generate a key and download it onto the desktop.
- 3 Share this key with your class friend.

(Hint: You may use the Flash drive, because you need to share it privately.)

2

Encrypt a file with the Symmetric Key and send it to your class friend through your KSU email.

- 1 In the CFY-KSU Cryptography Tool, choose the “Symmetric Key Cryptography” tab.
- 2 In the Encrypt Section, upload the file you want to encrypt.
- 3 Upload the Symmetric Key.
- 4 Encrypt the file and download it onto the desktop.
- 5 Send this encrypted file to your friend through your KSU mail. Ask him to decrypt the file by using the pre-shared key in the CFY-KSU Cryptography Tool.

3

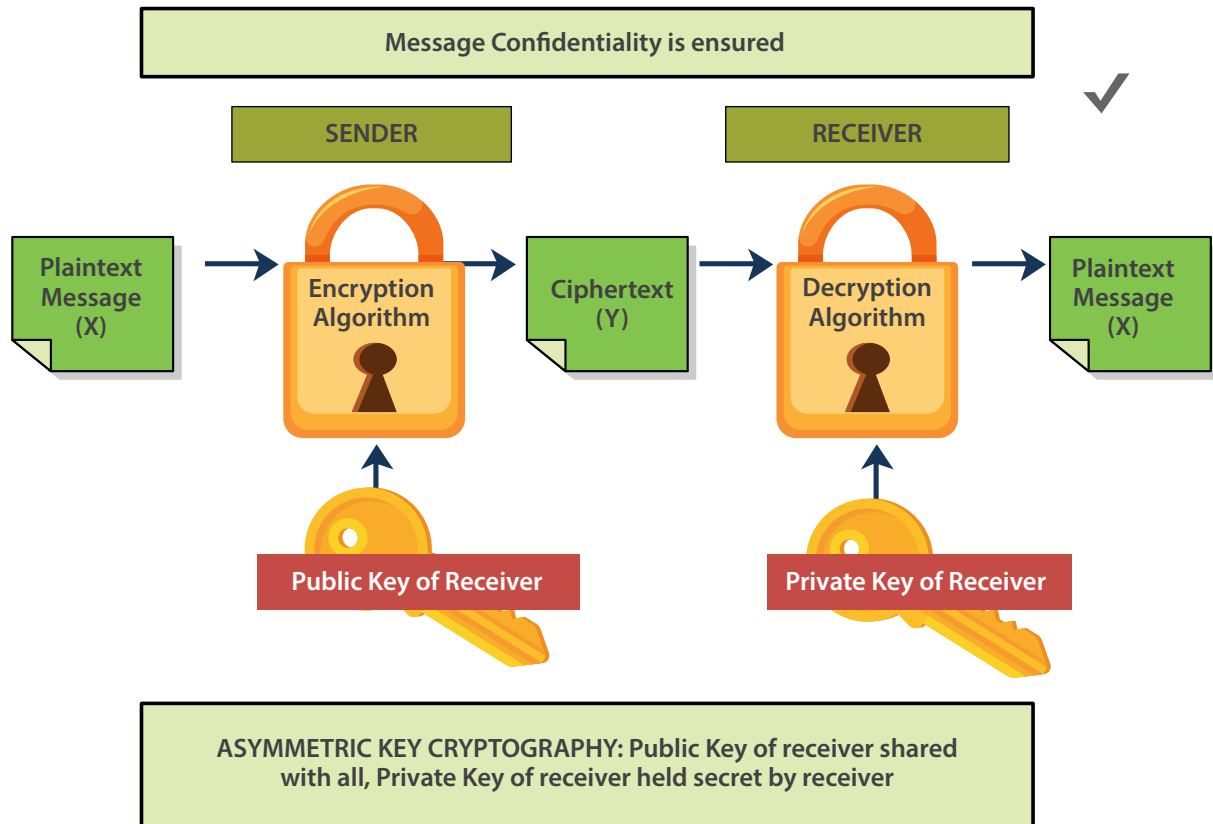
Decrypt the encrypted file that you have received in your KSU email with a key shared by your friend.

- 1 Download the encrypted file sent by your class friend in your KSU email and save it onto the desktop.
- 2 In the CFY-KSU Cryptography Tool, choose the “Symmetric Key Cryptography” tab.
- 3 Upload the encrypted file in the Decrypt section.
- 4 Upload the Symmetric Key, which was shared by your class friend separately.
- 5 Decrypt and download the decrypted file onto the desktop.
- 6 Open and check the original file.

2 - Asymmetric Key Cryptography

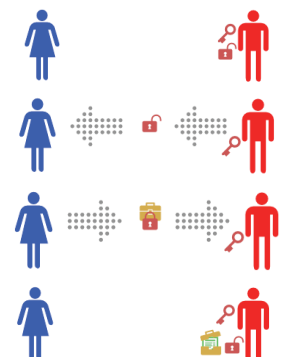
In asymmetric cryptography, two different keys (Public and Private keys) are used for encryption and decryption of data. Using the public key, anyone can encrypt the data. The private key is used to decrypt the data and is kept secret at all times, but the public key may be freely distributed. In this approach, both sender and receiver have the ability to generate both keys (using a computer system) together. Hence, the problem of key distribution never arises.

Both message confidentiality and authenticity are ensured through asymmetric key cryptography. It is also commonly known as Public Key Cryptography.



For example, two users, Alice and Bob, want to exchange messages privately and securely. In the Asymmetric Key Cryptography setting, the following steps will occur:

- STEP 1** Bob buys a padlock and matching key.
- STEP 2** Bob sends the (unlocked) padlock (Public Key) to Alice, keeping the key (Private Key) safe with himself.
- STEP 3** Alice puts her message in a simple box, locks it with Bob's padlock (Public Key), and sends it to Bob.
- STEP 4** Bob receives the locked box, which he can open with his key (Private Key). He can read the message. Alice can follow Steps 1 and 2 to get a reply from Bob securely.



Practice on Asymmetric Key Cryptography

As you know, in the Asymmetric Key Cryptography algorithm, the sender uses a Public key of the Receiver for the encryption and the Receiver uses his Private key (which is paired with the Public key) for the decryption. Thus, a key pair is required in the Asymmetric Key Cryptography algorithm.

Use the **CFY-KSU Cryptography Tool** and practice the Asymmetric Key Cryptography as follows.

1

Generate your Public and Private keys using the CFY-KSU Cryptography Tool and share the Public key with your class friend.

- 1 In the CFY-KSU Cryptography Tool, choose the "Asymmetric Key Cryptography" tab.
- 2 Generate the Public and Private Keys and download them onto the desktop.
- 3 Share the Public key with your class friend.

(Hint: You can share it publicly or send it through your KSU email.)

2

Encrypt a file with the shared Public Key of your class friend by using the CFY-KSU Cryptography Tool and send the encrypted file to him through your KSU email.

- 1 In the CFY-KSU Cryptography Tool, choose the "Asymmetric Key Cryptography" tab.
- 2 In the Encrypt section, upload the file.
- 3 Upload the shared Public Key of your class friend.
- 4 Encrypt the file and download it onto the desktop.
- 5 Send this encrypted file to your friend through your KSU mail. Ask him to decrypt the file by using the pre-shared key in the CFY-KSU Cryptography Tool.

3

Decrypt the encrypted file that you have received in your KSU email by using your Private Key.

- 1 Download the encrypted file from your KSU email to the desktop.
- 2 In the CFY-KSU Cryptography Tool, choose the "Asymmetric Key Cryptography" tab.
- 3 In the Decrypt section, upload the encrypted file.
- 4 Upload your Private Key.
- 5 Decrypt and download the file onto the desktop.
- 6 Open and check the original file.



Apply

Activity 1

1 - Using a transposition cipher, reverse the order of the plain text "ELPMAXE ELPMIS A"

2 - Convert this plain text to cipher text with the shift parameter of 3.

PLAIN TEXT: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Activity 2

Choose the correct option for the blanks or select True or False in the following statements.

1 - _____ Cryptography uses the same key for encryption and decryption of data.

- a) Public Key
- b) Asymmetric Key
- c) Symmetric Key
- d) None of the Above

2 - The _____ is the original message before transformation.

- a) Cipher text
- b) Secret text
- c) Plain text
- d) None of these

3 - A Key is a value that is used to encrypt or decrypt a message.

- a) True
- b) False

4 - _____ is the process of converting plain text to cipher text.

- a) Plain text
- b) Decryption
- c) Encryption
- d) None of these

5 - Cryptography can ensure the confidentiality and integrity of both data in transit and data at rest.

- a) True
- b) False

Activity
3

As Asymmetric Cryptography is slower and Symmetric is faster, it's a common practice to use Asymmetric Cryptography only to share the Symmetric Key and to use Symmetric Cryptography at both the ends.

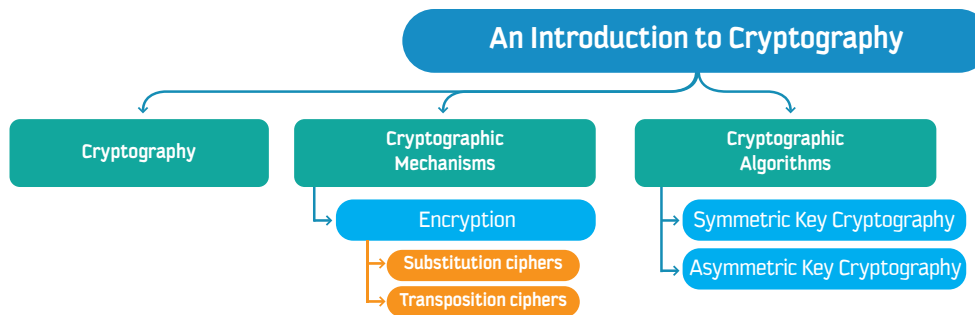
Use the CFY-KSU Cryptography Tool to send an Encrypted "Symmetric Key" by using Asymmetric Cryptography.

- a - Encrypt a file with Symmetric Key Encryption by using the CFY-KSU Cryptography Tool.
- b - Encrypt the generated "Symmetric Key" with Asymmetric Cryptography. Use the Receiver's "Public Key" for this encryption.
- c - Send the encrypted "Symmetric Key" to the Receiver along with the encrypted file. It can be simply sent by using email.
- d - The receiver needs to Decrypt the "encrypted Symmetric Key" received with his "Private Key." He gets the "Symmetric Key," with which he can decrypt the file easily.



Closing

Dear student, Following is a mind map that summarizes the concepts and knowledge you have learned in this session.



CHAPTER

5

Session

4

**Introduction to Cryptography
Part-II**



Goal

This session provides

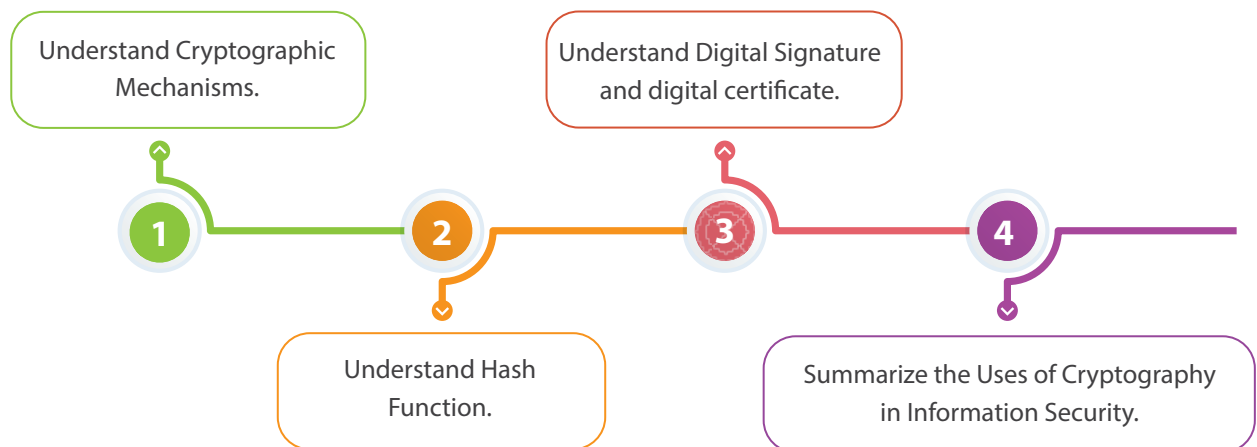
Students with an Cryptographic mechanisms like Hash Function, Digital signature, digital certificate and use of cryptography in information security.



Learning objective

Dear Student

By the end of this Session, you should be able to:



List of Terms

Term	Explanation
Hash Function	Cryptographic mechanism that helps to achieve the security service Data Integrity
Message Digest	Values returned by a hash function.
Digital Signature	Signature in digital form for an electronic document
Certificate Authority	A CA is a trusted entity that handles digital certificate



Learn

In the previous session we have discussed about cryptographic mechanisms like encryption, decryption and cryptographic algorithms. Now in this session we will discuss about hash function, digital signature and digital certificate.

Hash Functions

A Hash function is a one-way mathematical function that generates a fixed-length hash value based on the plain text.

Values returned by a hash function are called **message digest** or simply **hashes**.

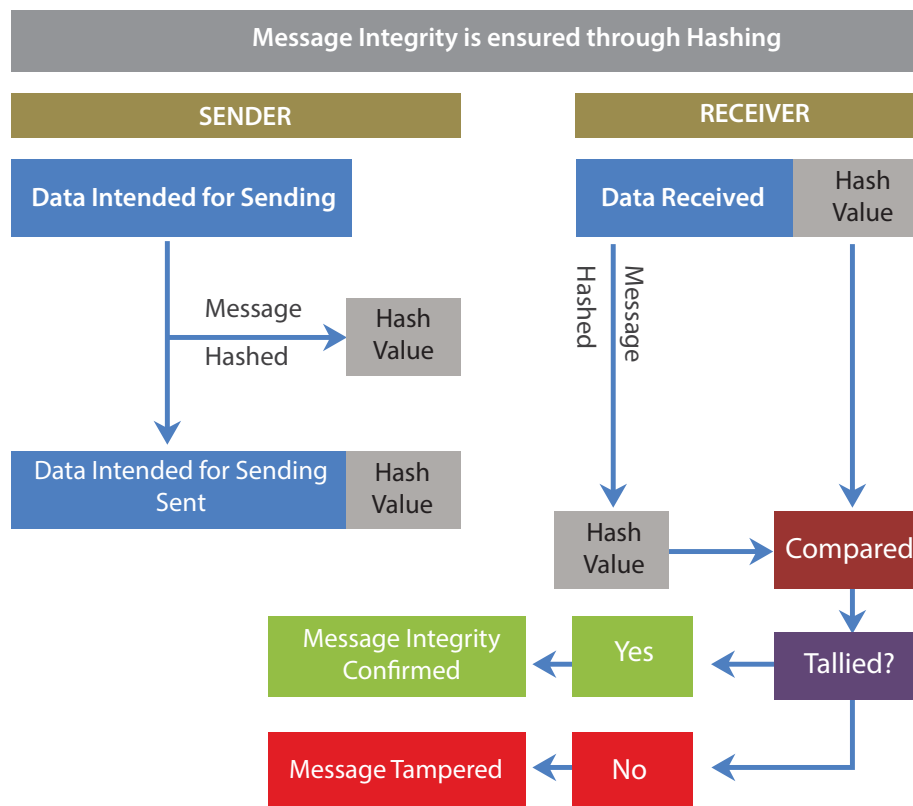
The hashes can be sent with the message. The receiver simply hashes the message again using the same algorithm, then compares the two hashes. If the hashes match, the message has not changed. If they do not match, the message has been altered.

It is impossible for someone to change the data without affecting the hash value, thus securing the data from intruders.

Hashing functions are easy to compute but hard to reverse.

A hash function $H()$, applied on input (x) , returns a fixed string, hs . Mathematically, it is written as $hs = H(x)$.

Thus, the Hash function is a cryptographic mechanism that helps to achieve the security service Data Integrity.



Practice on Hash Functions

As you know, in the Hash Function mechanism, the sender generates a Hash value which is sent along with the message (or a file). The receiver generates the Hash value again and compares the two Hash Values. If they are not the same, the message/file is compromised. Use the [CFY-KSU Cryptography Tool](#) and practice the Hash Functions as follows.



Generate a Hash Value for a file that you want to send safely to your friend by using the CFY-KSU Cryptography Tool.

Send the Hash Value along with the file to your class friend through the KSU email.

- 1 In the CFY-KSU Cryptography Tool, choose the “Hash Functions” tab.
- 2 Upload your file in the Generate Hash Value section.
- 3 Generate the Hash Value and download it onto the desktop. Save it with the name “H1”.
- 4 Send this Hash Value along with the file to your class friend through your KSU email.



You have received a file and a Hash Value in your KSU email. Generate the Hash Value again and compare it with the Hash Value received by using the CFY-KSU Cryptography Tool.

- 1 Download the file and the Hash Value from your KSU email and save it onto the desktop.
- 2 In the CFY-KSU Cryptography Tool, choose the “Hash Functions” tab.
- 3 Upload the file in the Generate Hash Value section.
- 4 Generate the Hash Value and download it onto the desktop. Save it with the name “H2”.
- 5 Now, in the Compare Hash Values section, upload the Hash Value “H1”, which was received in your KSU email.
- 6 Upload the Hash Value “H2”, which you have generated.
- 7 Compare the two Hash Values and check the result.

Digital Signature

In situations where there is not complete trust between sender and receiver, we need Digital Signatures to authenticate the sender of the message.

A digital signature is like a handwritten signature, but it is in digital form for an electronic document.



To digitally sign a message, the sender generates a hash of the message, then uses a key to encrypt the hash, thus generating a digital signature. (The key used is often a private key)

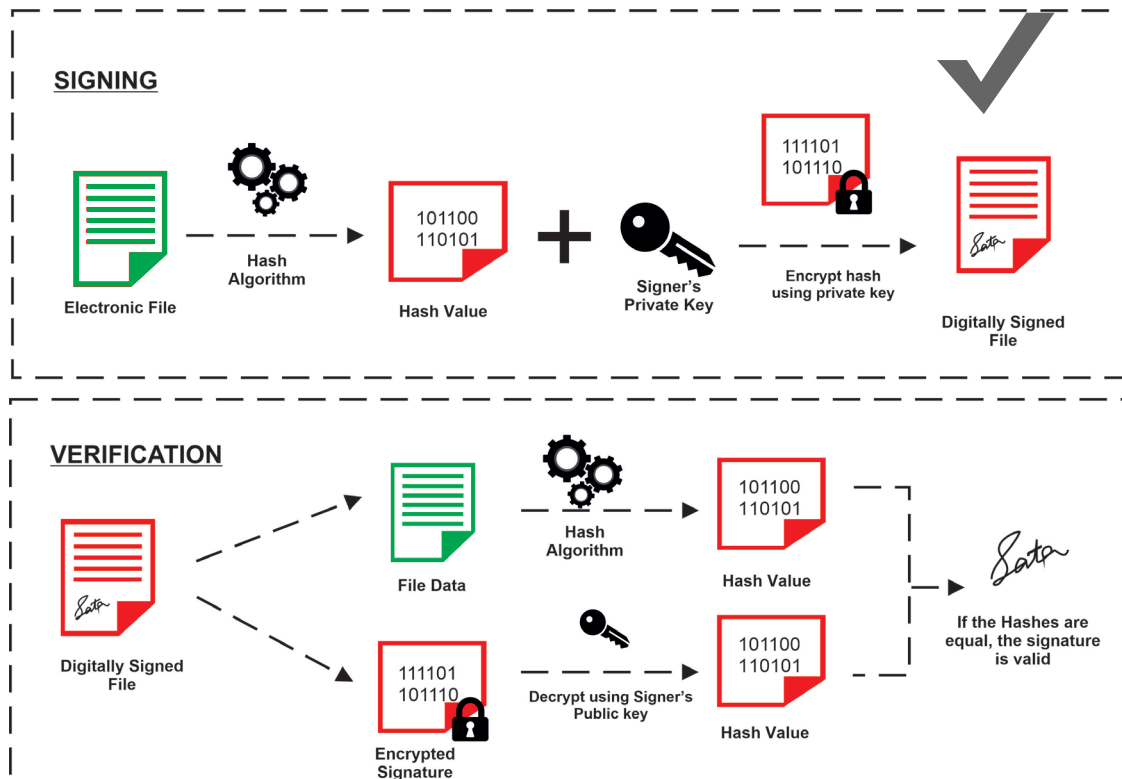
The sender then sends the digital signature along with the message, usually by appending it to the message itself.

When the message arrives at the receiving end, the receiver uses a key shared by the sender to decrypt the digital signature, thus restoring the original hash of the message. (The key shared is often a public key)

The receiver can then verify the integrity of the message by hashing the message again and comparing the two hashes.

Although this may sound like a considerable amount of work to verify the integrity of the message, it is often done by a software application of some kind, and the process typically is largely invisible to the end user.

A digital signature is considered legally binding, and if it is lost or stolen, it must be revoked.



Practice on Digital Signatures

As you learned about Digital Signatures in Session 3, the sender can generate a Digital Signature for a file by first generating the Hash Value for it and then encrypting the Hash Value with a Private Key. Use the [CFY-KSU Cryptography Tool](#) and practice the Digital Signatures as follows.

1

Generate a Private and a Public key using the CFY-KSU Cryptography Tool and share the Public with your class friend.

- 1 In the CFY-KSU Cryptography Tool, choose the “Asymmetric Key Cryptography” tab.
- 2 Generate the Public and Private Keys and download them onto the desktop.
- 3 Share the Public key with your class friend.

(Hint: You can share it publicly or send it through your KSU email.)

2

Generate a Digital Signature for your file using the CFY-KSU Cryptography Tool and send it along with the file to your class friend.

- 1 From “Hash Functions” tab of the CFY-KSU Cryptography Tool, generate the Hash Value for the file. Download and save this Hash Value onto the desktop.
- 2 Choose the “Digital Signatures” tab in the CFY-KSU Cryptography Tool.
- 3 In the Generate Digital Signature section, upload the Hash Value for the file.
- 4 Upload your Private Key.
- 5 Generate the Digital Signature and save it onto the desktop.
- 6 Send this Digital Signature along with your file to your class friend through your KSU email.

3

Verify a Digitally signed file that you have received in your KSU email from your class friend, by using the CFY-KSU Cryptography Tool.

- 1 First, generate the Hash Value for the received file. Use the “Hash Functions” tab of the CFY-KSU Cryptography Tool. Download and save this Hash Value onto the desktop.
- 2 Choose the “Digital Signatures” tab in the CFY-KSU Cryptography Tool.
- 3 In the Verify section, upload the Hash Value generated.
- 4 Upload the Digital Signature received.
- 5 Upload the Public Key shared by your class friend.
- 6 Verify the Digital Signature and check the result.

Digital Certificate

Message signing can be scaled up by using Digital Certificates.

“A Digital Certificate is an electronic equivalent of an individual's identity card; it can be used to verify the identity of someone you don't know.”



A digital certificate is typically formed by taking a key (public key) and identifying information, such as a name and address, and having them signed by a certificate authority (CA).

- A CA is a trusted entity that handles digital certificates.
- One well-known CA, at present, is VeriSign.



The advantage of having a certificate is that it allows us to verify that a key (public key) is truly associated with a particular individual. In the case of the digital signature we discussed in the preceding section, it might be possible that someone had falsified the keys being used to sign the message and that the keys did not actually belong to the original sender. If we have a digital certificate for the sender, we can easily check with the CA to ensure that the public key for the sender is legitimate.

Main Security Services achieved through Cryptography

Cryptography in the digital world involves many techniques and mechanisms that support the different security services. Different cryptographic mechanisms, such as Encryption, Hash Functions, and Digital Signatures, work as tools to achieve the different security services. This section outlines how the security services are achieved through different cryptographic mechanisms.

Confidentiality



By using Encryption, which is one of the core cryptographic mechanisms, we achieve the security service “Confidentiality.” Before you send sensitive data over a network or store it on a disk, you can encrypt it, which turns it unreadable. The intended receiver needs to decrypt it to get the readable data back.

Integrity



Cryptographic mechanisms such as Digital Signatures and Hash Functions are used to ensure the security service “Integrity.” One can easily forge email, but it's very hard to forge a digitally signed email message. Similarly, it's hard for someone to modify or manipulate a message that is digitally signed.

Cryptographic mechanisms such as Digital Certificates (Digital IDs) are also used to achieve the access control component “Authentication.”

Non-Repudiation



The security service “Non-Repudiation” prevents the sender from denying that he or she sent the message. The cryptographic mechanism Digital Signatures is used to achieve this service.



Apply

Activity 1

We use Hash Functions for Data Integrity. What if, in the middle of communication, someone stops the file, modifies it, generates a new Hash value, and sends you the new ones? For this reason, Hash Values are often sent encrypted.

Use the CFY-KSU Cryptography Tool to send an Encrypted "Hash Value" to your class friend by using Asymmetric Cryptography.

- a - Generate the Hash Value for a file.
- b - Encrypt the Hash Value by using Asymmetric Cryptography.
- c - Send the file and the encrypted Hash Value to your friend by using your ksu email.
- d - Your friend should Decrypt the received encrypted Hash Value by using the Private Key.
- e - He should generate the Hash Value again for the file at his end.
- f - He should then compare the two Hash Values to ensure complete data integrity.