

SAVITRIBAIPHULE PUNEUNIVERSITY

LABCOURSE III

SECTION I

Web Technologies I

Course Type: DSEC
(COURSECODE:CS-358)

T.Y.B.SC.(COMPUTERSCIENCE)

SEMESTER-I

Name_____

College Name_____

Roll No._____Division_____

Academic Year_____

Internal Examiner:-----External Examiner:-----

Co-ordinators

- **Dr.Mulay P. P.**, Annassaheb Magar College, Hadapsar, Pune
- **Dr. A.B.Nimbalkar**, Annassaheb Magar College, Hadapsar, Pune

Editor:

Dr. A.B.Nimbalkar, Annassaheb Magar College, Hadapsar, Pune

PREPARED BY

Prof. Shaikh A.M.	H.P.T. Arts & R.Y.K. Science College, Nashik
Prof. Kadam S.A.	BaburaojiGholap College , Sangvi, Pune
Prof. Byagar S.	Indira College of Commerce and Science, Pune
Prof. Malani P.S.	K.K.Wagh Arts,Commerce,Science & Computer Science College, Nashik

About The WorkBook

Objectives –

1. The scope of the course.
2. Bringing uniformity in the way course is conducted across different Colleges.
3. Continuous assessment of the students.
4. Providing ready references for students while working in the lab.

How to use this book?

This book is mandatory for the completion of the laboratory course. It is a Measure of the performance of the student in the laboratory for the entire duration of the course.

Instructions to the students

- 1) Students should carry this book during practical sessions of Computer Science.
- 2) Students should maintain separate journal for the source code and outputs.
- 3) Students should read the topics mentioned in reading section of this Book before coming for practical.
- 4) Students should solve all exercises which are selected by Practical in-charge as a part of journal activity.
- 5) Students will be assessed for each exercise on a scale of 5

1	Note done	0
2	Incomplete	1
3	Late complete	2
4	Needs improvement	3
5	Complete	4
6	Well-done	5

Instructions to the practical in-charge

1. Explain the assignment and related concepts in around ten minutes using whiteboard if required or by demonstrating the software.
2. Choose appropriate problems to be solved by student.
3. After a student completes a specific set, the instructor has to verify the outputs and sign in the provided space after the activity.
4. Ensure that the students use good programming practices.
5. You should evaluate each assignment carried out by a student on a scale of 5 as specified above ticking appropriate box.
6. The value should also be entered on assignment completion page of respected lab course.

PHP Semester – 1
Assignment Completion Sheet

Sr. No.	Assignment Name	Marks(out of 5)	Sign
1	Assignment Using HTML and CSS		
2	Assignment Using Boot Strap		
3	Assignment Using Functions and Strings		
4	Assignment Using Arrays		
5	Assignment Using Files and DATABASE(PostgreSQL)		
	Total out of 25		
	Total out of 05		

Assignment 1: TO STUDY HTML, HTML5 & CSS

You should read following topics before starting this exercise

1. Internet and web
2. Web browsers and web servers
3. HTML tags

Internet and the Web

The internet is a collection of connected computers that communicate with each other. The Web is a collection of protocols (rules) and software's that support such communication.

In most situations when two computers communicate, one acts as a server and the other as a client, known as client-server configuration.

Browsers running on client machines request documents provided by servers. Browsers are so called because they allow the user to browse through the documents available on the webservers. A browser initiates the communication with a server, requesting for a document. The server that is continuously waiting for a request, locates the requested document and sends it to the browser, which displays it to the user.

The most common protocol on the web is Hyper Text Transfer protocol (HTTP)

The most commonly used browsers are Microsoft Internet Explorer (IE). Netscape browser and Mozilla. The most commonly used web servers are Apache and Microsoft Internet Information server (IIS).

HTML Basics

Hyper Text Markup Language is a simple markup language used to create platform-independent hypertext documents on the World Wide Web. Most hypertext documents on the web are written in HTML.

You will need a simple text editor to write html codes. For example you can use notepad in windows and Vi editor in Linux operating system. You will need a browser to view thehtml code, you can use IE on windows and Mozilla on Linux operating system.

HTML tags are somewhat like commands in programming languages. Tags are not themselves displayed, but tell the browser how to display the document's contents.

Every HTML tag is made up of a tag *name*, sometimes followed by an optional list of attributes, all of which appears between angle brackets < >. Nothing within the brackets will be displayed in the browser. The tag name is generally an abbreviation of the tag's function.

Attributes are properties that extend or refine the tag's function. The name and attributes within a tag are not case sensitive. Tag attributes, if any, belong after the tag name, each separated by one or more spaces. Their order of appearance is not important. Most attributes take *values*, which follow an equal sign (=) after the attribute's name. Values are limited to 1024 characters in length and may be case sensitive. Sometimes the value needs to appear in quotation marks (double or single).

Most HTML tags are containers, meaning they have a beginning start tag and an end tag. An end tag contains the same name as the start tag, but it is preceded by a slash (/). Few tags do not have end tags.

HTML5

HTML5 will be the new standard for HTML.

HTML5 is still a work in progress. However, the major browsers support many of the new HTML5 elements and APIs.

HTML5 - New Features

New features of HTML5 are based on HTML, CSS, DOM, and JavaScript. To better handle today's internet use, HTML5 also includes new elements for drawing graphics, adding media content, better page structure, better form handling, and several APIs to drag/drop elements, find Geolocation, include web storage, application cache, web workers, etc. Some of the most interesting new features in HTML5 are:

- The <canvas> element for 2D drawing
- The <video> and <audio> elements for media playback
- Support for local storage
- New content-specific elements, like <article>, <footer>, <header>, <nav>, <section>
- New form controls, like calendar, date, time, email, url, search

Browser Support for HTML5

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

The HTML5 <!DOCTYPE>

In HTML5 there is only one <!doctype> declaration, and it is very simple:

```
<!DOCTYPE html>
```

The <!DOCTYPE> declaration helps the browser to display a web page correctly. There are many different documents on the web, and a browser can only display an HTML page 100% correctly if it knows the HTML type and version used.

Header and Footer:

The <header> tag specifies a header for a document or section. The <header> element should be used as a container for introductory content or set of navigational links. You can have several <header> elements in one document.

Note: A <header> tag cannot be placed within a <footer>, <address> or another <header> element.

The <footer> tag defines a footer for a document or section. A <footer> element should contain information about its containing element. A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

You can have several <footer> elements in one document.

Some HTML tags required to design simple web pages are given below

Tag	Description
<!DOCTYPE>	Defines the document type
<!--...-->	Allows one to insert a line of browser-invisible comments In the document
<HTML> </HTML>	<HTML> tag tells the browser that this is start of the HTML and </HTML> marks its end.
<HEAD> </HEAD>	Every html page must have a header. <Head> tag defines the Head Segment of an html document
<TITLE> </TITLE>	One of the most important parts of a header is title. Title is the small text that appears in title bar of viewer's browser.
<BODY> </BODY>	Every web page needs a body in which one can enter web page content
 	A single tag used to break lines
<p>	A single tag used to break text. Breaking text with the <p> tag adds vertical spacing
	To make text appear bold
<U></U>	To make text appear underlined
<I></I>	To make text appear italic

<CENTER> </CENTER>	Centers enclosed text
<BIG></BIG>	Sets the type one font size larger than the surrounding text
<SMALL> </SMALL>	Sets the type one font size smaller than the surrounding text
	Formats enclosed text as subscript.
	Formats enclosed text as superscript.
<MARQUEE> </MARQUEE>	Creates a scrolling-text marquee area.
	Loads an inline image
<Header>	The <header> tag specifies a header for a document or section. The <header> element should be used as a container for introductory content or set of navigational links. You can have several <header>elements in one document. Note:A<header> tag cannot be placed within a <footer>, <address>or another <header>element
<footer>	The<footer>tag defines footer for a document or section. A<footer>element should contain information about its containing element. A footer typically contains the author of the document,copy right information, links to terms of use, contact information, etc. You can have several<footer>elements in one document.

An HTML document is divided into two major portions: the head and the body. The head contains information about the document, such as its title and “meta” information describing the contents. The body contains the actual contents of the document (the part that is displayed in the browser window).

A sample HTML document is given below

```

<!--MyfirstProgramme--!>
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>

```

A sample HTML5 document is given below

```
<!--Starting my first web page assignment--!>
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>MyWebpage</TITLE>
</HEAD>
<BODY BACKGROUND="myimage.jpg" text="#FF0000">
The<FONTsize=6>Fontsize</FONT>canbechanged<Br>aswellas<FONTcolor="#00
```

Example of All basic HTML Tags

```
<!DOCTYPE html>
<html>
<body>
<b>Bold Text</b><br>
<i>Italic Text</i><br>
<sup>superscripted Text</sup><br>
<sub>subscripted Text</sub><br>
<big> Big Text<><br>
<strong> String Text</strong><br>
<u>Underlined Text</u>
<font face="Arial" size="10" color="blue">Formatted Text</font>
</body></html>
```

Lists: Lists are a great way to provide information in a structured and easy to read format.

There are two types of lists:

1] Numbered List(Ordered List)

An ordered list is used when sequence of list items is important.

2] Bulleted List(Unordered List)

An unordered list is a collection of related items that have no special order or sequence.

Tags used to create lists are given in the following table.

Tag	Description	Attributes	Example
	Specify the list item.		
	Thetag formats the contents of an ordered list with numbers. The numbering starts at 1.It is incremented by one for each successive ordered list item tagged with 	Type=a/A/i/I/1 Sets the numbering style to a,A,i,I,1 default 1 start="A" Specifies the number or letter with which the list should start.	<!DOCTYPEhtml> <html> <body bgcolor="pink"> > <u> List of Cities.... </u> <ol type="A" start="A"> Mumbai Pune Nashik Nagpur </body> </html>
 	 tag defines the unordered list of items	Type = disc/square/circle Specifies the bullet type.	<!DOCTYPEhtml> <html> <body bg color= "sky blue" text="yellow"> <i><u> List of Fruits </i></u> <ul type="square"> Apple Pinapple Mango Guava </body> </html>

Tables: A table is a two dimensional matrix, consisting of rows and columns. HTML tables are intended for displaying data in columns on a web page. Tables contains information such as text, images, forms, hyperlinks etc.

Tags used to create table are given in the following table.

Tag	Description	Attributes
<TABLE> </TABLE>	Create a Table	Border=number Draws an outline around the table rows and cells of width equal to number. By default table have no borders number =0. Width=number Defines width of the table. Cell spacing=number Sets the amount of cell space between table cells. Default value is 2 Cell padding=number Sets the amount of cell space, in number of pixels between the cell border and its contents. Default is 2 Bgcolor=" #rrggbb" sets background color of the table Bordercolor=" #rrggbb" sets border color of the table align=left right center Aligns the table. The default alignment is left frame=void above below hsides lhs rhs vsides box border Tells the browser where to draw borders around the table
<TR> </TR>	Creates a row in the table	
<TH> </TH>	Cells are inserted in a row of the table for heading	
<TD> </TD>	Data cells are inserted in a row of The table	

A sample HTML document for creating table is given below

```
<!DOCTYPEhtml>
<html>
<head>
</head>
<body>
<table border=2cellspacing=4cellpadding=4 border color dark="red" border color light="blue"
      align="center">
<caption>List of Books</caption>
<tr>
<th row span=2align="center">ItemNo</th>
<th row span=2align="center">ItemName</th>
<th align="center"colspan=2>Price</th>
</tr>
<tr>
<th align="center">Rs.</th>
<th align="center">Paise</th>
</tr>
```

Frames : Using frames, one can divide the screen into multiple scrolling sections, each of which can display a different web page in to it. It allows multiple HTML documents to be seen concurrently

Inline frames: It is a new frame tag introduced in HTML5. It is having same properties and attribute options as in<FRAME>tag. An<iframe>tag is used to display a web page within a web page. Inline frames can be included within the text block in HTML5 document.

Syntax of inline frame is:

```
<iframesrc="URL"></iframe>
```

Tags used to add frames are given in the following table

Tag	Description	Attributes	Example
<FRAMESET> </FRAMESET>	Splits browser screen into frames. Frameset is deprecated in html5, but it still works!	Rows=number helps in dividing the browser screen into horizontal sections or frames. Cols=number divides the screen into vertical sections or frames. The number written in the rows and cols attribute can be given as absolute numbers or percentage value or an asterisk can be used to indicate the remaining space.	<frameset rows="20%,30%,*">
<FRAME> </FRAME>	used to define a single frame in a <frameset>	name= <i>text</i> Assigns a name to the frame. No resize Prevents users from resizing the frame. src= <i>url</i> Specifies the location of the initial HTML file to be displayed	<html> <frameset rows="50%,*"> <frameset cols="50%,*"> <frame src="success.html" name="frm1">
		By the frame . Border color=" #rrggbb " or <i>color name</i> Sets the color for frame's borders	<frame src="welcome.html"> </frameset> <frame src="failure.html"> </frameset> </html>

<p><IFRAME> </IFRAME></p>	<p><iframe></p> <p>Tag is used to specify an inline frame.</p> <p>An inline frame Allows you to embed another document within the current HTML document. The HTML5 Specification refers to this as a "nested browsing context".</p>	<p>Iframe Attribute</p> <p>Attribute Specifications to Adjust Appearance and Behavior</p> <ul style="list-style-type: none"> ○ src="(URL of initial iframe content)" ○ name="(name of frame ,required for targeting)" ○ long desc="(link to long description)" ○ width=(frame width,% or pixels) ○ height=(frame height,% or pixels) ○ align=[top middle bottom left right center](frame alignment, pick two, use comma) ○ frame border=[1 0](frame border, default is 1) ○ margin width=(margin width, in pixels) ○ margin height=(margin height, in pixels) ○ scrolling=[yes no auto](ability to scroll) 	<p>1.<iframe src="C:\DocumentsandSettings\Desktop\ a7.html"width="150"height="150"></iframe></p> <p>2.<iframe name="inlineframe"src="float.html"frameborder="0"scrolling="auto"width="500"height="180"marginwidth="5"marginheight="5"></iframe></p>
---	---	---	---

What is CSS?

CSS is an acronym for Cascading Style Sheets. CSS is a style language that defines layout of HTML documents. For example, CSS covers fonts, colours, margins, lines, height, width, background images, advanced positions and many other things. Just wait and see!

HTML can be (mis-)used to add layout to websites. But CSS offers more options and is more accurate and sophisticated. CSS is supported by all browsers today.

Difference between CSS and HTML:

HTML is used to structure content. CSS is used for formatting structured content. The language HTML was only used to add structure to text. An author could mark his text by stating "this is a headline" or "this is a paragraph" using HTML tags such as <h1> and <p>.

As the Web gained popularity, designers started looking for possibilities to add layout to online documents. To meet this demand, the browser producers (at that time Netscape and Microsoft) invented new HTML tags such as for example which differed from the original HTML tags by defining layout - and not structure.

This also led to a situation where original structure tags such as <table> were increasingly being misused to layout pages instead of adding structure to text. Many new layout tags such as <blink> were only supported by one type of browser. "You need browser X to view this page" became a common disclaimer on web sites. CSS was invented to remedy this situation by providing web designers with sophisticated layout opportunities supported by all browsers. At the same time, separation of the presentation style of documents from the content of documents, makes site maintenance a lot easier.

Advantages of CSS: CSS was a revolution in the world of web design. The concrete benefits of CSS include:

- control layout of many documents from one single style sheet;
- more precise control of layout;
- Apply different layout to different media-types (screen, print, etc.);
- Numerous advanced and sophisticated techniques.

How does CSS work?

In this assignment you will learn how to make your first style sheet. You will get to know about the basic CSS model and which codes are necessary to use CSS in an HTML document.

Many of the properties used in Cascading Style Sheets (CSS) are similar to those of HTML. Thus, if you are used to use HTML for layout, you will most likely recognize many of the codes. Let us look at a concrete example.

The basic CSS syntax

Let's say we want a nice red color as the background of a webpage: Using HTML we could have done it like this:

```
<body bgcolor="#FF0000">
```

With CSS the same result can be achieved like this:

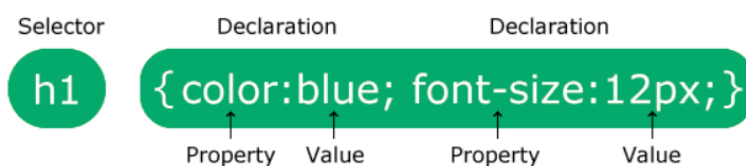
```
body {background-color: #FF0000;}
```

With CSS the same result can be achieved like this:

```
body{background-color:#FF0000;}
```

As you will note, the codes are more or less identical for HTML and CSS. The above example also shows you the fundamental CSS model:

rule=selector + declaration



But where do you put the CSS code? This is exactly what we will go over now.

Applying CSS to an HTML document

There are three ways you can apply CSS to an HTML document. These methods are all outlined below.

1. In-Line Method(the attribute style).
2. Internal Method(the tag style).
3. External Method(link to a style sheet).

Method 1: In-line (the attribute style)

One way to apply CSS to HTML is by using the HTML attribute `style`. Building on the above example with the red background color, it can be applied like this:

:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color:#FF0000;">
    <p>This is a red page</p>
  </body>
</html>
```

Method 2: Internal (the tag style)

Another way is to include the CSS codes using the HTML tag `<style>`. For example like this:

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      {background-color:#FF0000;}
    </style>
```

```
</head>
<body>
  <p>This is a red page</p>
</body>
</html>
```

Samplecode1:

```
<!DOCTYPEhtml>

<html>

<head>

<style>bod
y
{
background-color:#d0e4fe;
}
h1
{
color:orange;
text align:center;
}
p
{
font-
family:"TimesNewRoman";font
-size:20px;
}
</style>

</head>

<body>

<h1>CSSexample!</h1>
```

```
<p>This is a paragraph.</p>

</body>

</html>
```

Method 3

External style sheets are separate files full of CSS instructions (with the file extension .css). When any web page includes an external style sheet, its look and feel will be controlled by this CSS file (unless you decide to override a style using one of these next two types). This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page!

Sample program of External Style:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body></html>
```

Code for mystyle.css

```
body {
background-color: light blue;
}
h1 {
color: navy; margin-left: 20px;
}
```

Creation of forms

You should read following topics before starting this exercise

1. Forms and different types of Input element details

Forms: HTML5 provides better & more extensive support for collecting user inputs through forms. A form can be placed anywhere inside the body of an HTML document.

You can have more than one form in the document.

Tags used to add input forms are given in the following table.

Tag	Description	Attributes	Example
<FORM> </FORM>	Creates a form	action="URL" Gives the URL of the application that is to receive & process the forms data. method="get" or "post" Sets the method by which the browser sends the forms data to the server for processing.	<!doctype html> <head>.. </head> <body> <form action="url" method="post/get"> </form> </body> </html>
<INPUT> </INPUT>	It is used for managing the input controls that will be placed within the tag.	Name=text It is used to name the field. Maxlength=number The maximum number of input characters allowed in the input control. Size=number The width of the input control in pixels. type="(checkbox/hidden/radio/reset/submit/text/image)" value="default value (for text or hidden widget); Value to be submitted with the form(for a check box or radio button); Or label(for Reset or Submit buttons)" src="source file for an image",\checked indicates that checkbox or radio button is checked align="(text top/abs middle /baseline/bottom,)"	<label for>enter your name</label> <input type="text" name="nm" width=20> <input type="radio" name="gender" value="male" checked>Male <input type="radio" name="gender" value="Female">Female <input type="checkbox" name="chess" value="chess">Chess <input type="checkbox" name="Poker" value="Poker">Poker

<SELECT> </SELECT>	Defines and displays a set of optional list items from which the user can select one or more items.	name="(name to be passed to the script as part of name/value pair)" rows="no. of rows" cols="(no. of cols.)"	 Age Between: <select name="age" size=1> </select>
<TEXTAREA> </TEXTAREA>	Multi line text entry widget	name=name of data field size=#of items to display multiple allows multiple selections	<textarea rows=10columns=40> </textarea>
<OPTION>	indicates a possible item within a select widget	selected=default selection value="data submitted if this option is selected"	<select name="age" size=1> <option selected>21-30 <option>31-40 </select>

New input types in HTML5

HTML5 introduces 13 new input types. When viewed in a browser that doesn't support them, these input types fall back to text input.

Input types	Description	Attribute	Example
Label	Define a Label for <input>element		<form> <label for>enter your name</label> <input type="text"></form>
Legend	Define a Caption for field set element	•	<form> <legend>create new account</legend> </form>

Number	For numerical input	<ul style="list-style-type: none"> • value: The initial value. If omitted, the field is initially blank, but the internal value is not consistent across browsers. • step: How much to change the value when you click on the up or down arrows of the control. The default is 1. • min, max: The smallest 	<pre><input type="number" min="0" max="20"step="2" value="10"name="so me-name"/></pre>
		And largest values that can be selected with the up/down arrows.	
Range	For numerical input, but unlike number, the actual is not important.	<ul style="list-style-type: none"> • value: The initial value. The default is half way between the min and the max. • step: How much to change the value when you click on the up or down arrows of the control. The default is 1. • min, max: The smallest and largest values that can be selected. The default form in is 0, and the the default for max is 100 	<pre><input type="range" name="some-name"/></pre>
Date	For entering a date with no time zone.	<ul style="list-style-type: none"> • value: The initial value. The format is "yyyy-mm-dd". • step: The step size in days. The default is 1. • min,max: The smallest and largest dates that can be selected, formatted as date strings of the form "yyyy-mm-dd". 	<pre><input type="date" name="some-name"/></pre>

Time	For entering a time value with hour, minute, seconds, and fractional seconds, but no time zone.		<code><input type="time"/></code>
Datetime-local	For entering a date and Time with no time zone.		<code><input type="datetime-local"/></code>
Color	For choosing color through a	<ul style="list-style-type: none"> • value: The initial value. The intention is that if a browser pops up a color chooser, the initial 	<code><input type="datetime-local"/></code>

	Color well control	Selection will match the current value.	
Email	For entering a single mail address or a list of email addresses.	<ul style="list-style-type: none"> • value: The initial value(a legal email address). • list: The id of a separate " data list" element that defines a list of email addresses for the user to choose among 	<pre><input type="email" name="some-name"/> <input type="email" list="email-choices" name="some-name"/> <datalist id="email-choices"> <option label="Marty Hall" value="hall@coreservlets.com"> <option label="Somebody Else" value="someone@example.com"> <option label="ThirdPerson" value="other@example.com"> ... </data list></pre>
URL	For entering a single URL.	<ul style="list-style-type: none"> • value: The initial value, as an absolute URL. • list: The id of a separate "data list" element that defines a list of URLs for the user to choose among. 	<pre><input type="url" name="some-name"/> <input type="url" list="url-choices" name="some-name"/> <data list id="url-choices"> </datalist></pre>
Tel	For entering A telephone number.	<ul style="list-style-type: none"> • value: The initial value as a phone number 	<pre><input type="tel" name="some-name"/></pre>

Placeholder	Gives the user a hint about what sort of data they should enter.	<ul style="list-style-type: none"> • Placeholder: A small hint. This differs from the "value" attribute in two ways. First, it will usually be rendered differently (e.g., light gray). Second, it will automatically disappear when you click in the text field. • value: The initial value. If you have <i>both</i> placeholder and value, 	Input type="text(<i>or other</i>)"placeholder="some text" name="some-name"/>
-------------	--	--	--

		The value wins, and place holder is ignored.	
Autofocus	Focuses the input on the element when the Page is loaded.	<ul style="list-style-type: none"> value: The initial value TRUE/FALSE 	<code><input id="last name" type="text" autofocus="true"></code>
autocomplete	For specifying that a field should not autocomplete or be pre-filled by the browser based on a user's past entries.	<ul style="list-style-type: none"> value: The initial value ON/OFF 	<code><input id="password confirmation" type="password" name="Password confirmation" autocomplete="off"></code>
List/data list	Represents a set of Option elements that can be used in combination with the new list attribute for input to make dropdown menus.	<ul style="list-style-type: none"> list: The id of a separate "data list" element that defines a list of choices for the user to choose among. The option element(inside "data list") label: Extra information that may be displayed in the autocomplete list. Browsers might show this label or a combination of the label and the value. The label is never part of the value that is inserted in to the text field when the entry is selected. value: The value that should be inserted into the text field when the entry is selected. 	<pre> <input type="text (or other)"list="some- id" name="some- name"/> <data list id="email- choices"> <option label="Display Val1 "value="InsertVal1"> <option label="Display Val2"value="InsertVal2"> <option label="Display Val3"value="InsertVal3"> ... </datalist> </pre>

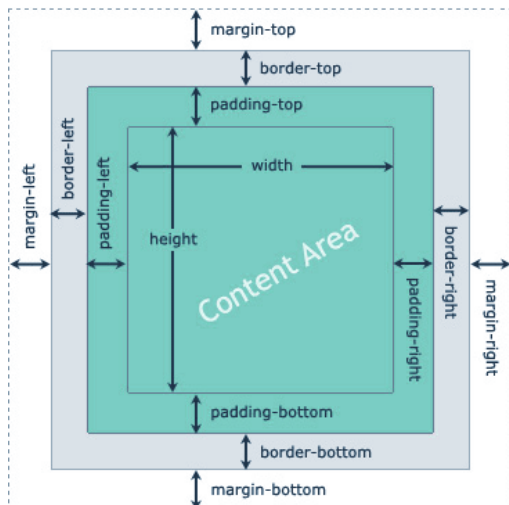
Try this Code

```
<!DOCTYPE html>
<html>
<body>
<form>
<h3>Registration Form</h3>
Enter your Name <input type="text" name="t1">
Your Email Address<input type="email" name="t2">
Your Phone Number<input type="text" name="t3">
Your City
<select name=t4>
<option>Nashik</option>
<option>Pune</option>
<option> Mumbai</option>
</select>
Gender
Male <input type=radio name=r1>
Female<input type=radio name=r1>
<input type=submit value="display">
</form>
</body>
</html>
```

BOX Model in CSS

Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS box model typically describes how these rectangular boxes are laid out on a web page. These boxes can have different properties and can interact with each other in different ways, but every box has a *content area* and optional surrounding *padding*, *border*, and *margin areas*.

The following diagram demonstrates how the width, height, padding, border, and margin CSS properties determines how much space an element can take on a web page.



The specific area that an element box may occupy on a web page is measured as follows-

Size of the box	Properties of CSS
Height	height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom
Width	width + padding-left + padding-right + border-left + border-right + margin-left + margin-right

example

```
div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}
```

CSS Navigation Bar

A Navigation bar or navigation system comes under GUI that helps the visitors in accessing information. It is the UI element on a webpage that includes links for the other sections of the website.

A navigation bar is mostly displayed on the top of the page in the form of a horizontal list of links.

```
<html>  
<head>  
<style>  
#ul-nb {  
list-style: none;  
margin:2px;  
padding:3px;  
}  
#ul-nb li {  
float:left;  
padding:10px;  
background:orange;  
text-align: center;  
margin-left:5px;  
}  
#ul-nbli:hover {  
background:red;  
opacity:0.8;  
color:white;  
}  
</style>  
</head>
```

```

<body>
<ul id="ul-nb">
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Community</a></li>
<li><a href="#">Careers</a></li>
</ul>
</body>
</html>

```

Set-A

Q1) Create a HTML document to display the following screen.

List of Indian States with thier capital

1. Delhi
 - NewDelhi
2. Haryana
 - Chandigarh
3. Gujarat
 - Gandhinagar
4. Rajasthan
 - Jaipur
5. Maharashtra
 - Mumbai
6. Uttarpradesh
 - Lucknow

Q 2. Write a HTML code, which generate the following output

List of Books			
Item No	Item Name	Price	
		Rs.	Paise
1	Programming in Python	500	50
2	Programming in Java	345	00

Q3. Write a HTML script to design the following screen

This is a header.	
Look in the box at the right for some information.	Here is some information.
This is a footer.	

SET B

Q 1. Write the HTML code for generating the form as shown below. Apply the internal CSS to following form change the font size of the heading to 6pt and change the color to red and also change the background color yellow

<p>Operating System Information</p> <p>Enter your name <input type="text"/></p> <p>Password <input type="password"/></p> <p>Which of the following Operating System have you used</p> <p><input checked="" type="checkbox"/> Linux <input checked="" type="checkbox"/> Windows 10 <input type="checkbox"/> Macintosh 8.0</p> <p>Which of the Operating System do you like the best?</p> <p><input checked="" type="checkbox"/> Linux <input type="checkbox"/> Windows 10 <input type="checkbox"/> Macintosh 8.0</p> <p>You have completed the form <input type="button" value="Sign Up"/></p>

Q 2. Write HTML 5 code which generates the following output and display each element of list in different size, color & font. Use external CSS to format the list

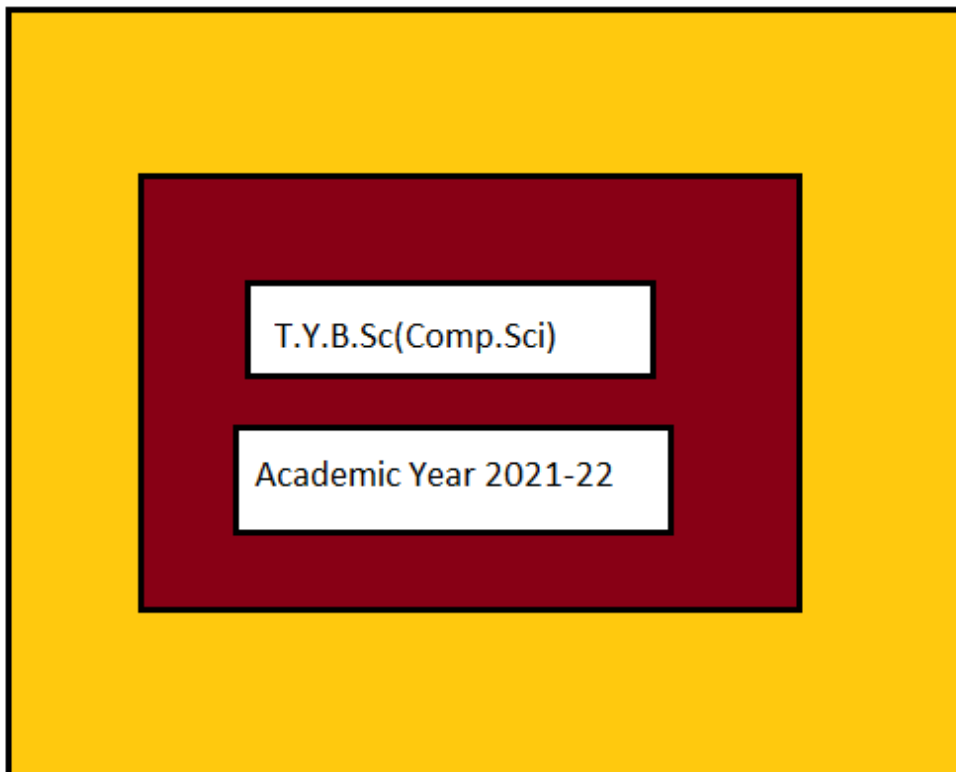
- Non flowering plants
 - Fern
 - Spore
- Flowering plants
 - Lilly
 - Rose
 1. Red Rose
 2. Pink Rose

- Q 3.** Create HTML5 page with following specifications
- i) Title should be about your City.
 - ii) Color the background by Pink color.
 - iii) Place your city name at the top of page in large text and in blue color.
 - iv) Add names of the landmarks in your city, each in different color, style and font
 - v) Add scrolling text about your City.
 - vi) Add any image at the bottom. (Use inline CSS to format the web page)

Set C.

Q 1. Design HTML 5 Page Using CSS which display the following Box
(use Box Model Property in CSS)

CSS Box Model Property



Q 2Design HTML 5 Page Using CSS Which Display the following Navigation Bar



Signature of the instructor: _____ **Date :** _____

Assignment Evaluation

0:Not Done	<input type="text"/>	2:Late Complete	<input type="text"/>	4:Complete	<input type="text"/>
1:Incomplete	<input type="text"/>	3:Needs Improvement	<input type="text"/>	5: Well Done	<input type="text"/>

Assignment: 2 Bootstrap

Prerequisites

Before proceeding with the Bootstrap tutorial, you should have some basic knowledge to implement web applications using HTML, CSS, and JavaScript to understand the bootstrap framework and its components.

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

Bootstrap History

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub!

What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

Why Use Bootstrap?

Advantages of Bootstrap:

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

Bootstrap Installation

Where to Get Bootstrap?

There are two ways to start using Bootstrap on your own web site.

You can:

- Download Bootstrap from getbootstrap.com
- Include Bootstrap from a CDN

Download Bootstrap Files

You can download the latest version of the Bootstrap framework from <https://getbootstrap.com>. The download files will contain the compiled and minified versions of CSS and JavaScript plugins. We can include any of these versions, i.e., either full or minified version based on our requirements.

The **full** (uncompressed) version contains the proper description of each method along with the comments. It is user-friendly and easily debugged but slower to load and heavier than the **minified** version. This should use mainly in the development process.

On the other hand, the **minified** version eliminates any unnecessary spaces and comments, and hence it is not so user-friendly. This loads faster and is much lighter than the **compiled** version. So this version should be used when your project goes live.

Now, add the downloaded files to your application root directory and include those file references in the header (<head>) section by using src attribute of the <script> tag like as shown below.

```
<html lang="en">
<head>
<title>Bootstrap Example</title>

<!-- Latest Bootstrap CSS -->
<script src="~/css/bootstrap.min.css"></script>
<!-- jQuery Library -->
<script src="~/jquery/jquery-3.4.1.min.js"></script>
<!-- Popper JS -->
<script src="~/js/popper.min.js"></script>
<!-- Latest Compiled JavaScript -->
<script src="~/js/bootstrap.min.js"></script>
</head>
<body>
</body>
</html>
```

If you observe the above code, we included jQuery and Popper.js files in the header section because if we want to use compiled bootstrap js file, we need to include jQuery and Popper.js before bootstrap js file. So, you need to download and include those files in your application directory.

Use Bootstrap from CDN

If you don't want to download the bootstrap files, then we can directly add them to our webpage by referencing them from public CDN (Content Delivery Network) like as shown below.

```

<html lang="en">
<head>
<title>Bootstrap Example</title>

<!-- Latest Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.cs
s">
<!-- jQuery Library -->
<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<!-- Latest Compiled JavaScript -->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</head>
<body> </body>
</html>

```

The CDN's will provide a performance benefit by reducing the loading time because the bootstrap files are hosted on multiple servers, and those are spread across the globe, so when a user requests the file, it will be served from the nearest server to them.

When we request a webpage, the CDN files will cache in the browser. If we request the same web page, then the CDN files will load from cache instead of downloading again from CDN.

Example1 :

```

<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<div class="jumbotron text-center">

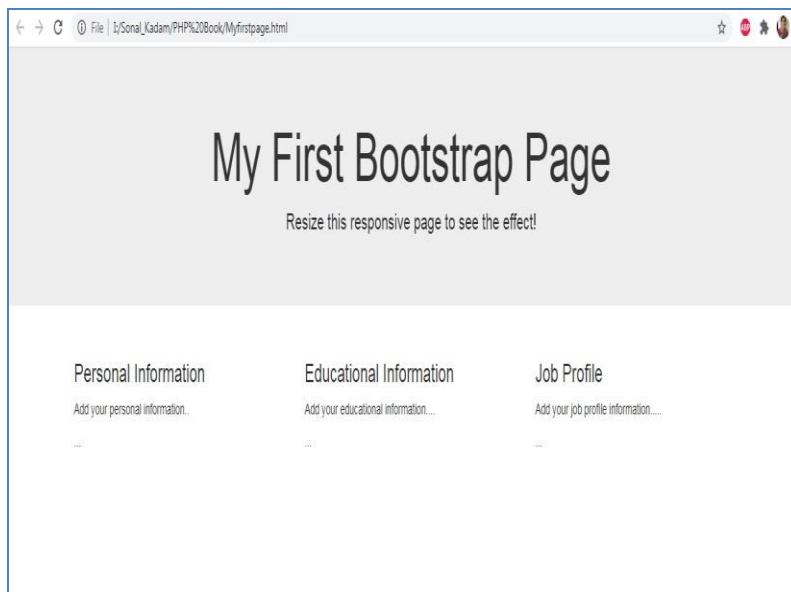
  <h1>My First Bootstrap Page</h1>

```

```
<p>Resize this responsive page to see the effect!</p>
</div>
```

```
<div class="container">
  <div class="row">
    <div class="col-sm-4">
      <h3>Personal Information</h3>
      <p>Add your personal information..</p>
      <p>...</p>
    </div>
    <div class="col-sm-4">
      <h3>Educational Information</h3>
      <p>Add your educational information....</p>
      <p>...</p>
    </div>
    <div class="col-sm-4">
      <h3>Job Profile</h3>
      <p>Add your job profile information.....</p>
      <p>...</p>
    </div>
  </div>
</div>
</body>
</html>
```

Output:



Example 2 :

```
<html lang="en">
```

```
<head>
```

```
  <title>Bootstrap 4 Website Example</title>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

```
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
```

```
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

```
<style>
```

```
.fakeimg {
  height: 200px;
  background: #aaa;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="jumbotron text-center" style="margin-bottom:0">
```

```
<h1>My First Bootstrap 4 Page</h1>
```

```
<p>Resize this responsive page to see the effect!</p>
```

```
</div>
```

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
```

```
<a class="navbar-brand" href="#">Navbar</a>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-  
target="#collapsibleNavbar">
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="collapse navbar-collapse" id="collapsibleNavbar">
```

```
<ul class="navbar-nav">
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#">Home</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#">Page 1</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#">Page 2</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
</nav>
```

```
<div class="container" style="margin-top:30px">
```

```
<div class="row">
```

```
<div class="col-sm-4">
  <h2>About Me</h2>
  <h5>Photo of me:</h5>
  <div class="fakeimg">Your Image</div>
  <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
  <h3>Some Links</h3>
  <p>Lorem ipsum dolor sit ame.</p>
  <ul class="nav nav-pills flex-column">
    <li class="nav-item">
      <a class="nav-link active" href="#">Personal Data</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Educational Info</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Business Profile</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Disabled</a>
    </li>
  </ul>
  <hr class="d-sm-none">
</div>
<div class="col-sm-8">
  <h2>TITLE HEADING</h2>
  <h5>Title description</h5>
  <div class="fakeimg">Add Image</div>
  <p>Add Some text..</p>
```



```
<p>Sunt in culpa qui officia deserunt mollincididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
```

```
<br>
```

```
<h2>TITLE HEADING</h2>
```

```
<h5>Title description</h5>
```

```
<div class="fakeimg">Add Image</div>
```

```
<p>Add Some text.</p>
```

```
<p>Sunt in culpa qui officia deserunt mollit anim id est laborum qua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="jumbotron text-center" style="margin-bottom:0">
```

```
<p>Footer</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Output:

Run above program and check output and do changes as per your webpage requirement

The Carousel(slideshow) Plugin

The Carousel plugin is a component for cycling through elements, like a carousel (slideshow).

How To Create a Carousel ?

The outermost <div>:

Carousels require the use of an id (in this case id="myCarousel") for carousel controls to function properly.

The class="carousel" specifies that this <div> contains a carousel.

The .slide class adds a CSS transition and animation effect, which makes the items slide when showing a new item. Omit this class if you do not want this effect.

The data-ride="carousel" attribute tells Bootstrap to begin animating the carousel immediately when the page loads.

The "Indicators" part:

The indicators are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user is currently viewing).

The indicators are specified in an ordered list with class `.carousel-indicators`.

The `data-target` attribute points to the id of the carousel.

The `data-slide-to` attribute specifies which slide to go to, when clicking on the specific dot.

The "Wrapper for slides" part:

The slides are specified in a `<div>` with class `.carousel-inner`.

The content of each slide is defined in a `<div>` with class `.item`. This can be text or images.

The `.active` class needs to be added to one of the slides. Otherwise, the carousel will not be visible.

The "Left and right controls" part:

This code adds "left" and "right" buttons that allows the user to go back and forth between the slides manually.

The `data-slide` attribute accepts the keywords "prev" or "next", which alters the slide position relative to its current position.

Add Captions to Slides

Add `<div class="carousel-caption">` within each `<div class="item">` to create a caption for each slide:

Example :

```
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
```

```

<body>
<div class="container">
  <h2>Carousel Example</h2>
  <div id="myCarousel" class="carousel slide" data-ride="carousel">
    <!-- Indicators -->
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
    </ol>
    <!-- Wrapper for slides -->
    <div class="carousel-inner">
      <div class="item active">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>LA is always so much fun!</p>
        </div>
      </div>
      <div class="item">
        
        <div class="carousel-caption">
          <h3>Chicago</h3>
          <p>Thank you, Chicago!</p>
        </div>
      </div>
      <div class="item">
        

```

```

<div class="carousel-caption">
  <h3>New York</h3>
  <p>We love the Big Apple!</p>
</div>
</div>
</div>
<!-- Left and right controls -->
<a class="left carousel-control" href="#myCarousel" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-left"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel" data-slide="next">
  <span class="glyphicon glyphicon-chevron-right"></span>
  <span class="sr-only">Next</span>
</a>
</div>
</div>
</body>
</html>

```

Output:

Run above program and check output and change images as per your webpage requirement

Bootstrap Forms

Bootstrap's Default Settings

Form controls automatically receive some global styling with Bootstrap:

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Bootstrap Form Layouts

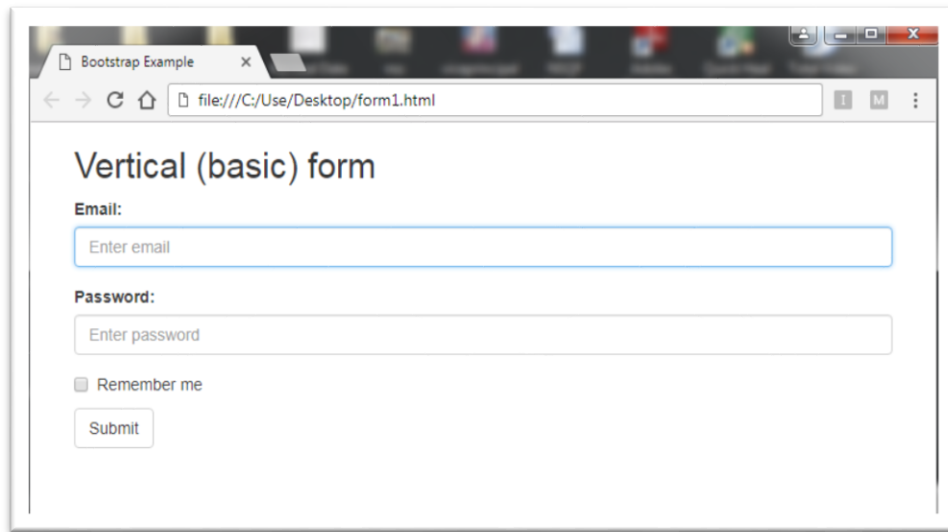
Bootstrap provides three types of form layouts:

- Vertical form (this is default)
- Horizontal form
- Inline form

Standard rules for all three form layouts:

- Wrap labels and form controls in `<div class="form-group">` (needed for optimum spacing)
- Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements

Bootstrap Vertical Form (default)



VerticalForm.html :

```
<html lang="en">
```

```
<head>
```

```
  <title>Bootstrap Example</title>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
```

```
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
  <h2>Basic form</h2>
```

```
  <form action="/action_page.php">
```

```
    <div class="form-group">
```

```
      <label for="email">Email:</label>
```

```

    <input type="email" class="form-control" id="email" placeholder="Enter email"
name="email">
</div>
<div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd" placeholder="Enter password"
name="pwd">
</div>
<div class="checkbox">
    <label><input type="checkbox" name="remember"> Remember me</label>
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
</body>
</html>

```

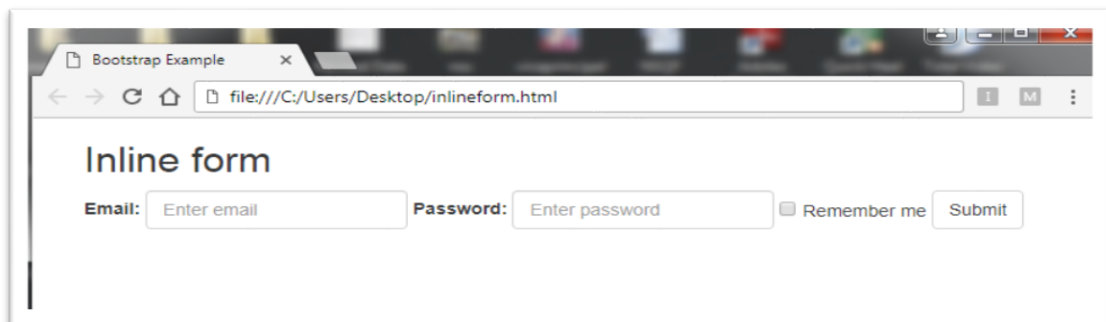
Bootstrap Inline Form

In an inline form, all of the elements are inline, left-aligned, and the labels are alongside.

Note: This only applies to forms within viewports that are at least 768px wide!

Additional rule for an inline form:

- Add class `.form-inline` to the `<form>` element



Inlineform.html

```
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
  <h2>Inline form</h2>
  <form class="form-inline" action="/action_page.php">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email" placeholder="Enter email"
name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd" placeholder="Enter password"
name="pwd">
    </div>
    <div class="checkbox">
      <label><input type="checkbox" name="remember"> Remember me</label>
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
```

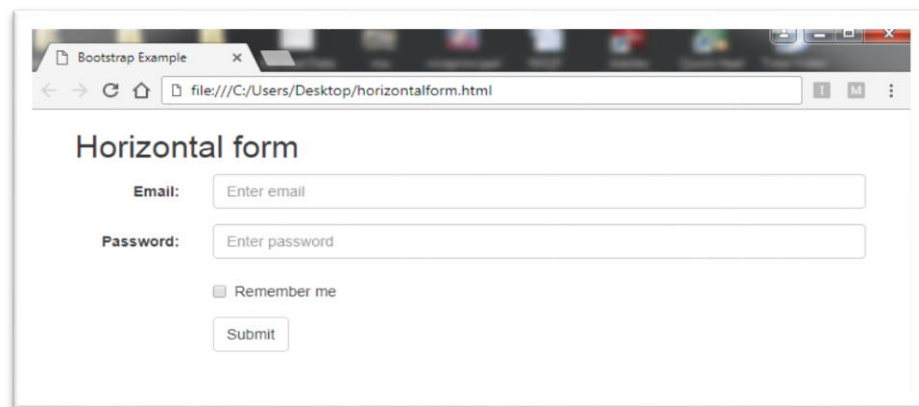
```
</form>
</div>
</body>
</html>
```

Bootstrap Horizontal Form

A horizontal form means that the labels are aligned next to the input field (horizontal) on large and medium screens. On small screens (767px and below), it will transform to a vertical form (labels are placed on top of each input).

Additional rules for a horizontal form:

- Add class `.form-horizontal` to the `<form>` element
- Add class `.control-label` to all `<label>` elements



HorizontalForm.html

```
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
```



```
<body>
```

```
<div class="container">
```

```
<h2>Horizontal form</h2>
```

```
<form class="form-horizontal" action="/action_page.php">
```

```
<div class="form-group">
```

```
<label class="control-label col-sm-2" for="email">Email:</label>
```

```
<div class="col-sm-10">
```

```
<input type="email" class="form-control" id="email" placeholder="Enter email"
name="email">
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label class="control-label col-sm-2" for="pwd">Password:</label>
```

```
<div class="col-sm-10">
```

```
<input type="password" class="form-control" id="pwd" placeholder="Enter password"
name="pwd">
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<div class="col-sm-offset-2 col-sm-10">
```

```
<div class="checkbox">
```

```
<label><input type="checkbox" name="remember"> Remember me</label>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

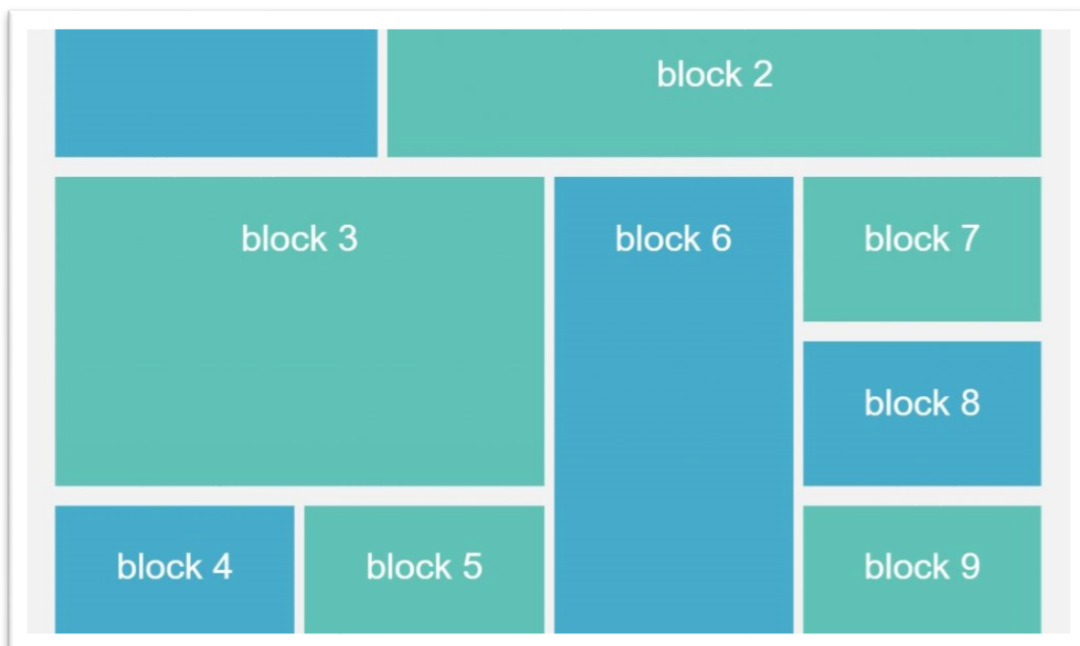
```
<div class="col-sm-offset-2 col-sm-10">
```

```
<button type="submit" class="btn btn-default">Submit</button>
```

```
</div>  
</div>  
</form>  
</div>  
</body>  
</html>
```

Set A:

1. Create following Bootstrap Web Layout Design. There are 9 blocks of the region in the arrangement. You can either place the images in them or the contents.

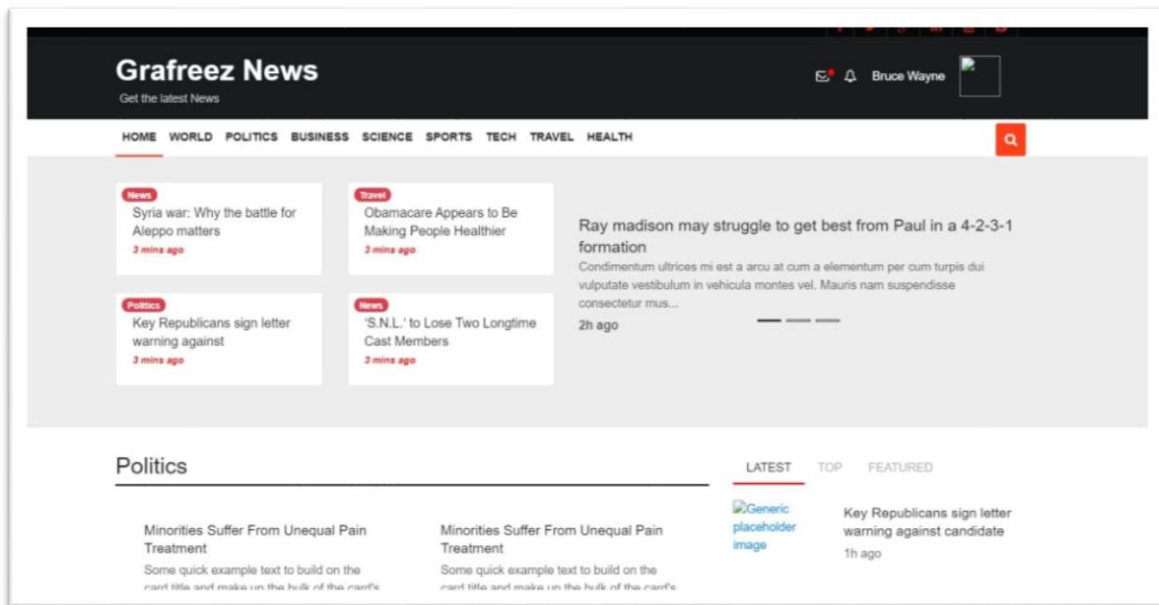


Set B

1. Create following layout using bootstrap




2. Create following News Template Layout using Bootstrap. At the top bar, you can include a climate gadget and helpful links like part login and information exchange. Like in all other free news site formats, mega menu choices are present to give you a chance to sort out the links.





Set C:


1. Create following one page website layout using bootstrap







Xtra Blog


 [Blog Home](#)

 [Single Post](#)

 [About Xtra](#)

 [Contact Us](#)



Xtra Blog is a multi-purpose HTML template from TemplateMo website. Left side is a sticky menu bar. Right side content will scroll up and down.



Simple and useful HTML layout

There is a clickable image with beautiful hover effect and active title link for each post item. Left side is a sticky menu bar. Right side is a blog content that will scroll up and down.

Travel . Events June 24, 2020
 36 comments by Admin Nat



Multi-purpose blog template

Xtra Blog is a multi-purpose HTML CSS template from TemplateMo website. Blog list, single post, about, contact pages are included. Left sidebar fixed width and content area is a fluid full-width.

Creative . Design . Business June 16, 2020
 48 comments by Admin Sam



How can you apply Xtra Blog

You are allowed to convert this template as any kind of CMS theme or template for your custom website builder. You can also use this for your clients. Thank you for choosing us.

Music . Audio June 11, 2020
 24 comments by John Walker



A little restriction to apply

You are not allowed to re-distribute this template as a downloadable ZIP file on any template collection website. This is strongly prohibited as we worked hard for this template. Please contact TemplateMo for more information.

Artworks . Design June 4, 2020
 72 comments by Admin Sam

Assignment No. 3 : To study functions & strings

User-defined functions

A function may be defined using syntax such as the following:

```
function function_name([argument_list...])
{
    [statements]
    [return return_value;]
}
```

Any valid PHP code may appear inside a function, even other functions and class definitions. The variables you use inside a function are, by default, not visible outside that function. In PHP3 functions must be defined, before they are referenced. No such requirement exists in PHP4.

Example 1.

Code	Output
<pre><?php msg("Hello"); // calling a function function msg(\$a) // defining a function { echo \$a; } ?></pre>	Hello

Default parameters

You can give default values to more than one argument, but once you start assigning default values, you have to give them to all arguments that follow as well.

Example 2.

Code	Output
<pre><?php function display(\$greeting, \$message="Good Day") { echo \$greeting; echo "
"; echo \$message; } display("Hello"); ?></pre>	Hello Good Day

Variable parameters

You can set up functions that can take a variable number of arguments. Variable number of arguments can be handled with these functions:

func_num_args : Returns the number of arguments passed

func_get_arg : Returns a single argument

func_get_args : Returns all arguments in an array

Example 3.

Code	Output
<pre><?php echo "Passing 3 arg. to xconcat
"; echo "Result is ..."; xconcat("How","are","you"); function xconcat() { \$ans = ""; \$arg = func_get_args(); for (\$i=0; \$i<func_num_args(); \$i++) { \$ans.= \$arg[\$i]." "; } echo \$ans; } ?></pre>	<p>Passing 3 arg. to xconcat Result is ...How are you</p>

Missing parameters

When using default arguments, any defaults should be on the right side of any non-default arguments, otherwise, things will not work as expected.

Example 4.

Code	Output
<pre><?php function makecoffee (\$type = "Nescafe") { return "Making a cup of \$type
"; } echo makecoffee (); echo makecoffee ("espresso"); ?></pre>	<p>Making a cup of Nescafe. Making a cup of espresso.</p>
<pre><?php function make (\$type = "acidophilus", \$flavour) { return "Making a bowl of \$type \$flavour
"; } echo make ("raspberry"); // won't work ?></pre>	<p>Warning: Missing argument 2 in call to make()..... Making a bowl of raspberry</p>
<pre><?php function make (\$flavour, \$type = "acidophilus") { return "Making a bowl of \$type \$flavour
"; } echo make ("raspberry"); //it works ?></pre>	<p>Making a bowl of acidophilus raspberry.</p>

Variable functions

Assign a variable the name of a function, and then treat that variable as though it is the name of a function.

Example 5.

Code	Output
<pre><?php \$varfun='fun1'; \$varfun(); \$varfun='fun2'; \$varfun(); \$varfun='fun3'; \$varfun(); function fun1() { echo "
Function one"; } function fun2() { echo "
Function two"; } function fun3() { echo "
Function three"; } ?></pre>	Function one Function two Function three

Anonymous functions

The function that does not possess any name are called anonymous functions. Such functions are created using `create_function()` built-in function. Anonymous functions are also called as lambda functions.

Example 6.

Code	Output
<pre><?php \$name=create_function('\$a,\$b', '\$c = \$a + \$b; return \$c;'); echo \$name(10,20); ?></pre>	30

Strings

Strings in PHP

- Single quoted string (few escape characters supported, variable interpolation not possible)
- Double quoted string (many escape characters supported, variable interpolation possible)
- Heredoc

There are functions to print the string, namely print, printf, echo.

The print statement can print only single value, whereas echo and printf can print multiple values. printf requires format specifiers. If echo statement is used like a function, then only one value can be

printed.

Comparing Strings

Example 1.

Code	Output
<pre><?php \$a='amit'; \$b='anil'; if(\$a==\$b) //using operator echo "Both strings are equal
"; else echo "Both strings are not equal
"; if(strcmp(\$a,\$b)>0) //using function { echo "String2 sorts before String1"; } elseif(strcmp(\$a,\$b)==0) { echo "both are equal"; } elseif(strcmp(\$a,\$b)<0) // negative value { echo "String1 sorts before String2"; }?></pre>	Both strings are not equal String1 sorts before String2
<pre><?php \$a=34; \$b='34'; if(\$a=== \$b) //using operator echo "Both strings are equal
"; else echo "Both strings are not equal
"; ?></pre>	Both strings are not equal

Other string comparison functions

strcasecmp()

: case in-sensitive string comparison

strnatcmp()

: string comparison using a “natural order”

algorithm

strnatcasecmp()

: case in-sensitive version of strnatcmp()

String manipulation & searching string

Example 2.

Code	Output
<pre><?php \$small="India"; \$big="India is my country"; \$str=substr(\$big,6,5); echo "
\$str"; \$cnt = substr_count(\$big,"i");</pre>	is my There are 2 i's in India is my country
56	is found at 6 position

<pre>echo "
There are".\$cnt." i's in \$big"; \$pos=strpos(\$big,"is"); echo "
is found at \$pos position"; \$replace=substr_replace(\$big,"Bharat",0,5); echo "
before replacement->\$big"; echo "
after replacement ->\$replace"; ?></pre>	<p>before replacement->India is my country after replacement ->Bharat is my country</p>
---	---

Regular Expressions

Two types of regular expressions

POSIX – style

PERL – compatible

Purpose of using regular expressions

Matching

Substituting

Splitting

Example 3.

Code	Output
<pre><?php \$big=<<< paragraph India is my country. I am proud of it. I live in Maharashtra. paragraph; echo "
"; \$found=preg_match('/am/i',\$big); if(\$found) echo "
am found in \"\$big\""; \$replace=preg_replace('/India/','Bharat',\$big); echo "
\$replace"; \$split=preg_split('/ /',\$big); foreach(\$split as \$elem) { echo "
\$elem";} ?></pre>	<pre>am found in \$big Bharat is my country. I am proud of it. I live in Maharashtra. India is my country. I am proud of it. I live in Maharashtra</pre>

SET A

Q: 1) Write a script to accept two integers(Use html form having 2 textboxes).

Write a PHP script to,

- Find mod of the two numbers.
- Find the power of first number raised to the second.
- Find the sum of first n numbers (considering first number as n)
- Find the factorial of second number.

(Write separate function for each of the above operations.)

Q: 2) Write a PHP script for the following: Design a form to accept a string.

- Write a function to find the length of given string without using built in functions.
- Write a function to count the total number of vowels i.e. (a,e,i,o,u)) from the string.
- Convert the given string to lowercase and then to Title case.
- Pad the given string with "*" from left and right both the sides.
- Remove the leading whitespaces from the given string.

- f. Find the reverse of given string.(use built-in functions)
(Use radio buttons and the concept of function. Use 'include' construct or 'require' stmt.)

SET B

Q. 1) :Write a PHP script for the following: Design a form to accept two strings from the user.

- a. Find whether the small string appears at the start of the large string.
- b. Find the position of the small string in the big string.
- c. Compare both the string for first n characters, also the comparison should not be case sensitive.

Q. 2) : Write a PHP script for the following: Design a form having a text box and a drop down list containing any 3 separators(e.g. #, |, %, @, ! or comma) accept a strings from the user and also a separator.

- a. Split the string into separate words using the given separator.
- b. Replace all the occurrences of separator in the given string with some other separator.
- c. Find the last word in the given string(Use **strrchr()** function).

Q. 3) : Write a PHP script having 3 textboxes. The first textbox should be for accepting name of the student, second for accepting name of college and third for accepting a proper greeting message. Write a function for accepting all the three parameters and generating a proper greeting message. If any of the parameters are not passed, generate the proper greeting message.

(Use the concept of missing parameters)

SET C

Q: 1) Write a PHP script for the following: Design a form to accept the marks of 5 different subjects of a student, having serial number, subject name & marks out of 100. Display the result in the tabular format which will have total, percentage and grade. Use only 3 text boxes.(Use array of form parameters)

Q: 2) Write a PHP script to accept 2 strings from the user, the first string should be a sentence and second can be a word.

- a. Delete a small part from first string after accepting position and number of characters to remove.
- b. Insert the given small string in the given big string at specified position without removing any characters from the big string.
- c. Replace some characters/ words from given big string with the given small string at specified position.
- d. Replace all the characters from the big string with the small string after a specified position.

(Use **substr_replace()** function)

Q: 3)Using regular expressions check for the validity of entered email-id.

- a. The @ symbol should not appear more than once.
- b. The dot (.) can appear at the most once before @ and at the most twice or at least once after @ symbol.
- c. The substring before @ should not begin with a digit or underscore or dot or @ or any other special character. (Use explode and ereg function.)

Signature of the instructor : _____ Date : _____

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2:Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1:Incomplete	<input type="checkbox"/>	3:Needs Improvement	<input type="checkbox"/>	5:Well Done	<input type="checkbox"/>

ASSIGNMENT NO. 4 : TO STUDY ARRAYS :

An array is a collection of data values. Array is organized as an ordered collection of (key,value) pairs.

In PHP there are two kinds of arrays:

Indexed array: An array with a numeric index starting with 0.

For example, Initializing an indexed array,

```
$numbers[0]=100;
```

```
$numbers[1]=200;
```

```
$numbers[2]=300;
```

Associative array: An array which have strings as keys which are used to access the values.

Initializing an Associative array,

```
$numbers[ 'one' ]=100;
```

```
$numbers[ 'two' ]=200;
```

```
$numbers[ 'three' ]=300;
```

Creating Arrays

Arrays can be created in multiple ways

1. Using simple assignment to initialize an array.

```
Eg : $addresses[0] = "spam@cyberpromo.net";
```

```
$addresses[1] = "abuse@example.com";
```

```
$addresses[2] = root@example.com
```

OR

```
$price['gasket'] = 15.29;
```

```
$price['wheel'] = 75.25;
```

```
$price['tire'] = 50.00
```

2. An easier way to initialize an array is to use the array() construct, which builds an array from its arguments. This builds an indexed array, and the index values (starting at 0) are created automatically:

```
$addresses = array("spam@cyberpromo.net", "abuse@example.com",  
"root@example.com");
```

To create an associative array with array(), use the => symbol to separate indices (keys) from values:

```
$price = array( 'gasket' => 15.29, 'wheel' => 75.25, 'tire' => 50.00 );
```

3. To construct an empty array, pass no arguments to array(): \$addresses = array();

Some Array functions:

Name	Use	Example
Array()	This construct is used to initialize an array.	<pre>\$numbers=array(100,200,300); \$cities=array('Capital of Nation'=>'Delhi', 'Capital of state'=>'Mumbai', 'My city'=>'Nashik');</pre>
List()	This function is used to copy values from array to the variables.	<pre>\$cities=array('Capital of Nation'=>'Delhi', 'Capital of state'=>'Mumbai', 'My city'=>'Nashik');</pre>
Count(), sizeof()	The count() and sizeof() functions are identical in use and effect. They return the number of elements in the array.	<pre>\$family = array("Fred", "Wilma", "Pebbles"); \$size = count(\$family); // \$size is 3</pre>
array_splice()	This function is used to remove or insert elements in array	<pre>\$student=array(11,12,13,14,15,16); \$new_student=array_splice(\$student,2,3); /* starting from index(2) and length =3 \$new_student1=array_splice(\$student,2); /* here length is not mentioned */ Output : \$new_student=(13,14,15); \$new_student1=(13,14,15,16);</pre>
array_key_exists()	This function is used to check if an element exist in the array.	<pre>\$cities=array('Capital of Nation'=>'Delhi', 'Capital of state'=>'Mumbai', 'My city'=>'Nashik'); If (array_key_exists('Capital of State',\$cities)) { Echo "key found!\n"; } Output :Key_found!</pre>
extract()	This function automatically creates local variables from the array	<pre>\$student = array('roll'=>11, 'name'=>'A', 'class'=>'TYBSC'); Extract(\$student); By this, the variables are created like this : \$roll = 11; \$name='A'; \$class='TYBSc';</pre>
array_filter()	To identify a subset of an array based on its values, use the array_filter() function	<pre>\$callback = function isOdd (\$element) { return \$element % 2; }; \$numbers = array(9, 23, 24, 27); \$odds = array_filter(\$numbers, \$callback); // \$odds is array(0 => 9, 1 => 23, 3 => 27)</pre>
shuffle()	To traverse the elements in an array in random order, use the shuffle() function	<pre>\$weekdays = array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"); shuffle(\$weekdays); print_r(\$days); Array([0] => Tuesday [1] => Thursday [2] => Monday [3] => Friday [4] => Wednesday)</pre>

Set A

1. Create your array of 30 high temperatures, approximating the weather for a spring month, then find the average high temp, the five warmest high temps and the five coolest high temps. Display the result on the browser.
Hint: a) Use array_slice b) the HTML character entity for the degree sign is & deg;.
2. Write a menu driven program to perform the following stack and queue related operations:[Hint: use Array_push(), Array_pop(), Array_shift(), Array_unshift()]
 - a) Insert an element in stack
 - b) Delete an element from stack
 - c) Display the contents of stack
 - d) Insert an element in queue
 - e) Delete an element from queue
 - f) Display the contents of queue

Set B

1. Write a PHP script that inserts a new item in an array at any position.
(hint : use array_splice())
2. Define an array. Find the elements from the array that matches the given value using appropriate search function.

Set C

1. Write a PHP script to sort the following associative array :
array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40") in
 - a) ascending order sort by value
 - b) ascending order sort by Key
 - c) descending order sorting by Value
 - d) descending order sorting by Key
2. Write a menu driven program to perform the following operations on associative arrays:
 - a) Split an array into chunks
 - b) Sort the array by values without changing the keys.
 - c) Filter the odd elements from an array.
 - d) Merge the given arrays.
 - e) Find the intersection of two arrays.
 - f) Find the union of two arrays.
 - g) Find set difference of two arrays.

Assignment Evaluation

0 : Not done		1 : Incomplete		2 : Late Complete	
3 : Needs Improvement		4 : Complete		5 : Well done	

Assignment : 5 To study Files and Database (PHP-PostgreSQL)

File : A **file** is nothing more than an ordered sequence of bytes stored on hard disk, floppy disk CD-ROM or some other storage media. Operations on file are
 Opening and closing a file.
 Reading a file and writing into file
 Deleting and renaming a file
 navigating a file

A **file handler** is nothing more than an integer value that will be used to identify the file you wish to work with until it is closed working with files

Function Name	Description														
fopen()	<p>Opening and closing a file This is used to open a file ,returning a file handle associated with opened file .It can take three arguments :fname,mode and optional use_include_path Ex:-\$fp=fopen("data.txt",r); We can also open a file on remote host List of modes used in fopen are:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Purpose</th> </tr> </thead> <tbody> <tr> <td>R</td> <td>Open for reading only; place the file pointer at the beginning of the file</td> </tr> <tr> <td>r+</td> <td>Open for reading and writing; place the file pointer at the beginning of the file.</td> </tr> <tr> <td>w</td> <td>Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td> </tr> <tr> <td>w+</td> <td>Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.</td> </tr> <tr> <td>A</td> <td>Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td> </tr> <tr> <td>a+</td> <td>Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.</td> </tr> </tbody> </table>	Mode	Purpose	R	Open for reading only; place the file pointer at the beginning of the file	r+	Open for reading and writing; place the file pointer at the beginning of the file.	w	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.	w+	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.	A	Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.	a+	Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
Mode	Purpose														
R	Open for reading only; place the file pointer at the beginning of the file														
r+	Open for reading and writing; place the file pointer at the beginning of the file.														
w	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.														
w+	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.														
A	Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.														
a+	Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.														
fclose()	<p>This is used to close file, using its associated file handle as a single argument Ex:- fclose(fp);</p>														
fread()	<p>This function is used to extract a character string from a file and takes two arguments, a file handle and a integer length Ex: fread(\$fp,10);</p>														
fwrite()	<p>This function is used to write data to a file and takes two arguments, a file handle and a string Ex: fwrite(\$fp,"HELLO");</p>														

fgetc()	Function can be used to read one character from file at a fileIt takes a single argument ,a file handle and return just one character from the file .It returns false when it reached toend of file.
fgets()	This function is used to read set of characters it takes two arguments, file pointer and length. It will stop reading for any one of three reasons:The specified number of bytes has been read A new line is encountered The end of file is reached
fputs()	This is simply an alias for fwrite() .
file()	This function will return entire contents of file.This function will automatically opens,reads,anclose the file.It has one argument :a string containing the name of the file.It can alsofetch files on remote host.
fpassthru()	This function reads and print the entire file to the web browser.This function takes one argument ,file handle.If youread a couple of lines from a file before calling fpassthru() ,then this function only print the subsequent contents of afile.
readfile()	This function prints content of file without having a call to fopen() It takes a filename as its argument ,reads a file and then writeit to standard output returning the number of bytesread(or false upon error)
fseek()	It takes file handle and integer offset , offset type as an arguments .It will move file position indicator associated withfile pointer to a position determined by offset. By default this offset is measured in bytes from the beginning of the file. The third argument is optional ,can be specified as: SEEK_SET:-Beginning of file +offset SEEK_CUR:-Current position +offset(default) SEEK_END:-End of the file +offset
ftell()	It takes file handle as an argument and returns the current offset(in bytes) of the corresponding file position indicator.
rewind()	It accepts a file handle as an argument and reset the corresponding file position indicator to the beginning of file.
file_exists()	It takes file name with detail path as an argument and returns true if file is there otherwise it returns false
file_size()	It takes file name as an argument and returns total size of file (in bytes)
fileatime()	It takes filename as an argument and returns last access time for a file in a UNIX timestamp format
filectime()	It takes filename as an argument and returns the time at which the file was last changed as a UNIX timestamp format
filemtime()	It takes filename as an argument and returns the time at which the file was last modified as a UNIX timestamp format
fileowner()	It takes filename as an argument and returns the user ID of the owner of specified file.
posix_getpwuid()	It accept user id returned by fileowner() function as an argument and returns an associative array with following references
	64
	Key Description

	<table border="1"> <tr> <td>name</td> <td>The shell account user name of the user</td> </tr> <tr> <td>passwd</td> <td>The encrypted user password</td> </tr> <tr> <td>Uid</td> <td>The ID number of the user</td> </tr> <tr> <td>Gid</td> <td>The group ID of the user</td> </tr> <tr> <td>Gecos</td> <td>A comma separated list containing user full name office phone, office number and home phone number</td> </tr> <tr> <td>Dir</td> <td>The absolute path to the home directory of the user</td> </tr> <tr> <td>Shell</td> <td>The absolute path to the users default shell</td> </tr> </table>	name	The shell account user name of the user	passwd	The encrypted user password	Uid	The ID number of the user	Gid	The group ID of the user	Gecos	A comma separated list containing user full name office phone, office number and home phone number	Dir	The absolute path to the home directory of the user	Shell	The absolute path to the users default shell
name	The shell account user name of the user														
passwd	The encrypted user password														
Uid	The ID number of the user														
Gid	The group ID of the user														
Gecos	A comma separated list containing user full name office phone, office number and home phone number														
Dir	The absolute path to the home directory of the user														
Shell	The absolute path to the users default shell														
filegroup()	It takes filename as an argument and returns the group ID of owner of the specified file														
posix_getgid()	It accept group ID returned by filegroup() function as an argument and returns an associative array on a group identified by group ID with following refernces <table border="1"> <tr> <th>Key</th> <th>Description</th> </tr> <tr> <td>Name</td> <td>The name of group</td> </tr> <tr> <td>Gid</td> <td>The ID number of group</td> </tr> <tr> <td>members</td> <td>The number of members belonging to the group</td> </tr> </table>	Key	Description	Name	The name of group	Gid	The ID number of group	members	The number of members belonging to the group						
Key	Description														
Name	The name of group														
Gid	The ID number of group														
members	The number of members belonging to the group														
filetype()	It takes filename as an argument and returns the type of specified file . the type of possible values are fifo, char, dir, block, link, file and unknown														
basename()	It takes file name as an argument and separate the filename from its directory path.														
copy()	It takes two string arguments referring to the source and destination file respectively.														
rename()	It takes two argument as old name and new name and renames the file with new name.														
unlink()	It takes a single argument referring to the name of file we want to delete.														
is_file()	It returns true if the given file name refers to a regular file.														
fstat()	The fstat() function returns information about an open file.														
	This function returns an array with the following elements: [0] or [dev] - Device number [1] or [ino] - Inode number [2] or [mode] - Inode protection mode [3] or [nlink] - Number of links [4] or [uid] - User ID of owner [5] or [gid] - Group ID of owner [6] or [rdev] - Inode device type [7] or [size] - Size in bytes [8] or [atime] - Last access (as Unix timestamp) [9] or [mtime] - Last modified (as Unix timestamp) [10] or [ctime] - Last inode change (as Unix timestamp) [11] or [blksize] - Blocksize of filesystem IO (if supported) [12] or [blocks] - Number of blocks allocated														

Examples

Use of some above mentioned functions is illustrated in the following examples: Example: 1) To read file from server use fread() function. A file pointer can be created to the file and read the content by specifying the size of data to be collected.

```
<?php
$myfile = fopen("somefile.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("somefile.txt"));
fclose($myfile);
?>
```

Example: 2) a file can be written by using fwrite() function in php. For this open file in write mode. file can be written only if it has write permission. if the file does not exist then one new file will be created. the file the permissions can be changed.

```
<?php
$filecontent="some text in file";           // store some text to enter inside the file
$file_name="test_file.txt";                // file name
$fp = fopen ($filename, "w");              // open the file in write mode, if it
does not exist then it will be created.
fwrite ($fp,$filecontent);                 // entering data to the file
fclose ($fp);                              // closing the file pointer
chmod($filename,0777);                     // changing the file permission.
?>
```

Example :

3) A small code for returning a **file-size**.

```
<?php
function dispfilesize($filesize)
{
    if(is_numeric($filesize))
    {
        $decr = 1024; $step = 0;
        $prefix =
        array('Byte','KB','MB','GB','TB','PB');
        while(($filesize / $decr) > 0.9)
        {
            $filesize = $filesize / $decr;
            $step++;
        }
        return round($filesize,2).' '.$prefix[$step];
    }
    else
    {
        return 'NaN';
    }
}
?>
```

Accessing Databases (PostgreSQL)

PostgreSQL supports a wide variety of built-in data types and it also provides an option to the users to add new data types to PostgreSQL, using the CREATE TYPE command. Table lists the data types officially supported by PostgreSQL. Most datatypes supported by PostgreSQL are directly derived from SQL standards.

The following table contains PostgreSQL supported data types for your ready reference

Category	Data type	Description
Boolean	boolean, bool	A single true or false value.
Binary types	bit(n)	An n-length bit string (exactly n) binary bits)
	bit varying(n), varbit(n)	A variable n-length bit string (upto n) binary nbits)
Character Types	character(n)	A fixed n-length character string
	char(n)	A fixed n-length character string
	character varying(n)	
	varchar (n)	
	text	A variable length character string of unlimited length
Numeric types	smallint, int2	A signed 2-byte integer
	integer, int, int4	A signed, fixed precision 4-byte number
	bigint, int8	A signed 8-byte integer, up to 18 digits in length
	real, float4	A 4-byte floating point number
	float8, float	An 8-byte floating point number
	numeric(p,s)	An exact numeric type with arbitrary precision p, and scale s.
Currency	money	A fixed precision, U.S style currency
	serial	An auto-incrementing 4-byte integer
Date and time types	date	The calendar date(day, month and year)
	time	The time of day
	time with time zone	the time of day, including time zone information
	timestamp(includes time Interval)	An arbitrarily specified length

Functions used for PostgreSQL database manipulation

S. No.	API & Description
1	<p>resource pg_connect (string \$connection_string [, int \$connect_type])</p> <p>This opens a connection to a PostgreSQL database specified by the connection_string.</p> <p>If PGSQL_CONNECT_FORCE_NEW is passed as connect_type, then a new connection is created in case of a second call to pg_connect(), even if the connection_string is identical to an existing connection.</p>
2	<p>bool pg_connection_reset (resource \$connection)</p> <p>This routine resets the connection. It is useful for error recovery. Returns TRUE on success or FALSE on failure.</p>
3	<p>int pg_connection_status (resource \$connection)</p> <p>This routine returns the status of the specified connection. Returns PGSQL_CONNECTION_OK or PGSQL_CONNECTION_BAD.</p>
4	<p>string pg_dbname ([resource \$connection])</p> <p>This routine returns the name of the database that the given PostgreSQL connection resource.</p>
5	<p>resource pg_prepare ([resource \$connection], string \$stmtname, string \$query)</p> <p>This submits a request to create a prepared statement with the given parameters and waits for completion.</p>
6	<p>resource pg_execute ([resource \$connection], string \$stmtname, array \$params)</p> <p>This routine sends a request to execute a prepared statement with given parameters and waits for the result.</p>

7	<p>resource pg_query ([resource \$connection], string \$query)</p> <p>This routine executes the query on the specified database connection.</p>
8	<p>array pg_fetch_row (resource \$result [, int \$row])</p> <p>This routine fetches one row of data from the result associated with the specified result resource.</p>
9	<p>array pg_fetch_all (resource \$result)</p> <p>This routine returns an array that contains all rows (records) in the result resource.</p>
10	<p>int pg_affected_rows (resource \$result)</p> <p>This routine returns the number of rows affected by INSERT, UPDATE, and DELETE queries.</p>
11	<p>int pg_num_rows (resource \$result)</p> <p>This routine returns the number of rows in a PostgreSQL result resource for example number of rows returned by SELECT statement.</p>
12	<p>bool pg_close ([resource \$connection])</p> <p>This routine closes the non-persistent connection to a PostgreSQL database associated with the given connection resource.</p>
13	<p>string pg_last_error ([resource \$connection])</p> <p>This routine returns the last error message for a given connection.</p>
14	<p>string pg_escape_literal ([resource \$connection], string \$data)</p> <p>This routine escapes a literal for insertion into a text field.</p>

15	<p>string pg_escape_string ([resource \$connection], string \$data)</p> <p>This routine escapes a string for querying the database.</p>
----	--

Connecting to Database

pg_connect () — is used to open a PostgreSQL connection.

Syntax :

resource pg_connect (string \$connection_string [, int \$connect_type])

pg_connect() opens a connection to a PostgreSQL database specified by the connection_string.

If a second call is made to **pg_connect()** with the same connection_string as an existing connection, the existing connection will be returned unless you pass **PGSQL_CONNECT_FORCE_NEW** as connect_typ

The following PHP code shows how to connect to an existing database on a local machine and finally a database connection object will be returned.

```
$db=pg_connect("dbname=sonal");

$db=pg_connect("host=localhost port= 5432 dbname=sonal");

$db=pg_connect("host=localhost port=5432 dbname=sonal
                user=postgres password=redhat");
```

```
<?php
    $host    = "host = 127.0.0.1";
    $port    = "port = 5432";
    $dbname  = "dbname = testdb";
    $credentials = "user = postgres password=pass123";
    $db = pg_connect( "$host $port $dbname $credentials" );
    if(!$db) {
        echo "Error : Unable to open database\n";
    } else {
        echo "Opened database successfully\n";
    }
}
```

```
?>
```

Now, let us run the above given program to open our database **testdb**: if the database is successfully opened, then it will give the following message –

```
Opened database successfully
```

Closing Connection:

`pg_close()`- function closes a PostgreSQL connection

`pg_close` — Closes a PostgreSQL connection

syntax :

```
bool pg_close ([ resource $connection ] )
```

`pg_close()` closes the non-persistent connection to a PostgreSQL database associated with the given connection resource.

Execute A Query :

`pg_query` — to Execute a query

syntax :

```
resource pg_query ([ resource $connection ], string $query )
```

`pg_query()` executes the query on the specified database connection.

Create a Table

The following PHP program will be used to create a table in a previously created database –

```
<?php
    $host      = "host = 127.0.0.1";
    $port      = "port = 5432";
    $dbname    = "dbname = testdb";
    $credentials = "user = postgres password=pass123";

    $db = pg_connect( "$host $port $dbname $credentials" );
    if(!$db) {
```

```

    echo "Error : Unable to open database\n";
} else {
    echo "Opened database successfully\n";
}

$sql = <<<<EOF
CREATE TABLE COMPANY
(ID INT PRIMARY KEY NOT NULL,
NAME TEXT NOT NULL,
AGE INT NOT NULL,
ADDRESS CHAR(50),
SALARY REAL);
EOF;
$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
} else {
    echo "Table created successfully\n";
}
pg_close($db);
?>

```

When the above given program is executed, it will create COMPANY table in your **testdb** and it will display the following messages –

```

Opened database successfully
Table created successfully

```

INSERT Operation

The following PHP program shows how we can create records in our COMPANY table created in above example –

```

<?php
$host = "host=127.0.0.1";
$port = "port=5432";
$dbname = "dbname = testdb";

```



```

$credentials = "user = postgres password=pass123";

$db = pg_connect( "$host $port $dbname $credentials" );
if(!$db) {
    echo "Error : Unable to open database\n";
} else {
    echo "Opened database successfully\n";
}

$sql = <<<<EOF
    INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
    VALUES (1, 'Paul', 32, 'California', 20000.00 );

    INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
    VALUES (2, 'Allen', 25, 'Texas', 15000.00 );

    INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
    VALUES (3, 'Teddy', 23, 'Norway', 20000.00 );

    INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
    VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00 );
EOF;

$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
} else {
    echo "Records created successfully\n";
}
pg_close($db);
?>

```

When the above given program is executed, it will create the given records in COMPANY table and will display the following two lines –

```
Opened database successfully
Records created successfully
```

pg_fetch_row

`pg_fetch_row` — Get a row as an enumerated array

Description

array **pg_fetch_row** (resource \$result [, int \$row])

pg_fetch_row() fetches one row of data from the result associated with the specified result resource.

Note: This function sets NULL fields to the PHP NULL value.

Parameters

result

PostgreSQL query result resource, returned by `pg_query()`, `pg_query_params()` or `pg_execute()` (among others).

row

Row number in result to fetch. Rows are numbered from 0 upwards. If omitted or NULL, the next row is fetched.

Return Values

An array, indexed from 0 upwards, with each value represented as a string. Database *NULL* values are returned as NULL.

FALSE is returned if row exceeds the number of rows in the set, there are no more rows, or on any other error.

SELECT Operation

The following PHP program shows how we can fetch and display records from our COMPANY table created in above example –

```
<?php
$host    = "host = 127.0.0.1";
$port    = "port = 5432";
$dbname  = "dbname = testdb";
$credentials = "user = postgres password=pass123";

$db = pg_connect( "$host $port $dbname $credentials" );
if(!$db) {
    echo "Error : Unable to open database\n";
} else {
    echo "Opened database successfully\n";
}

$sql = <<<<EOF
    SELECT * from COMPANY;
EOF;

$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
    exit;
}
while($row = pg_fetch_row($ret)) {
    echo "ID = ". $row[0] . "\n";
    echo "NAME = ". $row[1] . "\n";
    echo "ADDRESS = ". $row[2] . "\n";
    echo "SALARY = ".$row[4] . "\n\n";
}
echo "Operation done successfully\n";
pg_close($db);
```

```
?>
```

When the above given program is executed, it will produce the following result. Keep a note that fields are returned in the sequence they were used while creating table.

```
Opened database successfully
```

```
ID = 1
```

```
NAME = Paul
```

```
ADDRESS = California
```

```
SALARY = 20000
```

```
ID = 2
```

```
NAME = Allen
```

```
ADDRESS = Texas
```

```
SALARY = 15000
```

```
ID = 3
```

```
NAME = Teddy
```

```
ADDRESS = Norway
```

```
SALARY = 20000
```

```
ID = 4
```

```
NAME = Mark
```

```
ADDRESS = Rich-Mond
```

```
SALARY = 65000
```

```
Operation done successfully
```

UPDATE Operation

The following PHP code shows how we can use the UPDATE statement to update any record and then fetch and display updated records from our COMPANYY table –

```
<?php
    $host      = "host=127.0.0.1";
    $port      = "port=5432";
    $dbname    = "dbname = testdb";
    $credentials = "user = postgres password=pass123";

    $db = pg_connect( "$host $port $dbname $credentials" );
    if(!$db) {
        echo "Error : Unable to open database\n";
    } else {
        echo "Opened database successfully\n";
```

```

}
$sql =<<<<EOF
    UPDATE COMPANY set SALARY = 25000.00 where ID=1;
EOF;
$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
    exit;
} else {
    echo "Record updated successfully\n";
}

$sql =<<<<EOF
    SELECT * from COMPANY;
EOF;

$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
    exit;
}
while($row = pg_fetch_row($ret)) {
    echo "ID = ". $row[0] . "\n";
    echo "NAME = ". $row[1] . "\n";
    echo "ADDRESS = ". $row[2] . "\n";
    echo "SALARY = ". $row[4] . "\n\n";
}
echo "Operation done successfully\n";
pg_close($db);
?>

```

When the above given program is executed, it will produce the following result –

Record updated successfully

ID = 2

NAME = Allen

ADDRESS = 25

SALARY = 15000

ID = 3

NAME = Teddy

ADDRESS = 23

SALARY = 20000

ID = 4

NAME = Mark

ADDRESS = 25

SALARY = 65000

ID = 1

NAME = Paul

ADDRESS = 32

SALARY = 25000

Operation done successfully

DELETE Operation

The following PHP code shows how we can use the DELETE statement to delete any record and then fetch and display the remaining records from our COMPANY table –

```
<?php
$host    = "host = 127.0.0.1";
$port    = "port = 5432";
$dbname  = "dbname = testdb";
$credentials = "user = postgres password=pass123";

$db = pg_connect( "$host $port $dbname $credentials" );
if(!$db) {
    echo "Error : Unable to open database\n";
} else {
    echo "Opened database successfully\n";
}

$sql = <<<<EOF
DELETE from COMPANY where ID=2;
```

```

EOF;

$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
    exit;
} else {
    echo "Record deleted successfully\n";
}

$sql = <<<<EOF
    SELECT * from COMPANY;
EOF;

$ret = pg_query($db, $sql);
if(!$ret) {
    echo pg_last_error($db);
    exit;
}
while($row = pg_fetch_row($ret)) {
    echo "ID = ". $row[0] . "\n";
    echo "NAME = ". $row[1] . "\n";
    echo "ADDRESS = ". $row[2] . "\n";
    echo "SALARY = ". $row[4] . "\n\n";
}
echo "Operation done successfully\n";
pg_close($db);
?>

```

When the above given program is executed, it will produce the following result –

```

Opened database successfully
Record deleted successfully
ID = 3
NAME = Teddy
ADDRESS = 23
SALARY = 20000

```

ID = 4
NAME = Mark
ADDRESS = 25
SALARY = 65000

ID = 1
NAME = Paul
ADDRESS = 32
SALARY = 25000

Operation done successfully

SetA

Q1. Write a program to read one file and display the contents of the file with its size.

Q2. Consider the following entities and their relationships

Event (eno , title , date)

Committee (cno , name, head , from_time ,to_time , status)

Event and Committee have many to many relationship. Write a script to accept title of event and modify status committee as working.

SetB

Q1) Write a program to read a flat file “item.dat”, which contains details of 5 different items such as Item code, Item Name, unit sold, and Rate. Display the Bill in tabular format.

Q2) Considerer the following entities and their relationships

Student (Stud_id,name,class)

Competition (c_no,c_name,type)

Relationship between student and competition is many-many with attribute rank and year. Create a RDB in 3NF for the above and solve the following. Using above database write a script in PHP to accept a competition name from user and display information of student who has secured 1st rank in that competition.

SetC

Q 1. Write a menu driven program to perform various file operations. Accept filename from user.

- a) Display type of file.
- b) Display last access time of file
- c) Display the size of file
- d) Delete the file

Q 2. Property (pno, description, area)

- a. Owner (oname, address, phone)
- b. An owner can have one or more properties, but a property belongs to exactly one owner.

c. Accept owner name from the user. Write a PHP script which will display all properties which are own by that owner

Signature of the instructor: _____ **Date :** _____

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2:Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1:Incomplete	<input type="checkbox"/>	3:Needs Improvement	<input type="checkbox"/>	5: Well Done	<input type="checkbox"/>

SECTION II

CS 358

FOUNDATIONS OF DATA SCIENCE

Coordinator and Editor

Dr. Poonam Ponde

Nowrosjee Wadia College, Pune
Member, BOS (Computer Science),
Savitribai Phule Pune University.

Assignments Prepared by

Dr. Poonam Ponde
Nowrosjee Wadia College, Pune

Dr. Parag Tamhankar
Abasaheb Garware College, Pune

Dr. Harsha Patil
Ashoka Center For Business & Computer
Studies, Nashik

Prof. Amit Mogal
MVP Samaj's CMCS College, Nashik

CS 354 - FOUNDATIONS OF DATA SCIENCE

Assignment Completion Sheet

Name of Student: _____

Roll Number: _____

Division: _____

Sr. No.	Assignment Title	Marks obtained	Signature of Instructor
1	The Data Science Environment		
2	Statistical Data Analysis		
3	Data Preprocessing		
4	Data Visualization		

Total Marks : _____ / 20

Converted Marks : _____ / _____

Signature of Incharge

Date:

Internal Examiner

External Examiner

Assignment 1: The Data Science Environment

Number of Slots = 2

Objectives

- To understand the data science process.
- To get introduced to various Python programming tools like IDLE, command line, online tools like google colaboratory etc.
- To understand the purpose of packages like NumPy, SciPy, pandas, matplotlib, jupyter, etc.
- To create own dataset.
- To load an existing dataset.
- To execute simple Data Science examples on a dataset.

Reading

You should read the following topics before starting this exercise:

Typing and executing a Python script.

Introductory concepts of Data Science and Data Science Lifecycle.

Ready Reference

Installing Python

There are 2 approaches to install Python:

- You can download Python directly from its project site and install individual components and libraries (<https://www.python.org/downloads/>)
- Alternately, you can download and install a package like Anaconda, which comes with pre-installed libraries. (<https://www.anaconda.com/products/individual>)

Choosing a development environment

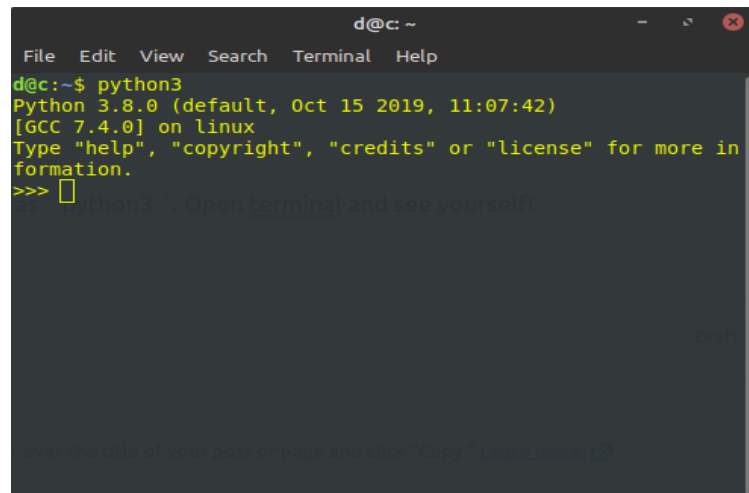
Once you have installed Python, there are various options for choosing an environment. Here are the 4 most common options:

1. Terminal / Shell based
2. IDLE
3. iPython notebook (Jupyter notebook)
4. Google Colaboratory

1. Terminal/Shell

Open the terminal by searching for it in the dashboard or pressing Ctrl + Alt + T. Right click on the desktop and click Terminal and in terminal type `python`

You should see a window like this. At the command prompt, type any Python command and press enter to execute the command.

A terminal window with a dark background and light text. The title bar reads 'd@c: ~'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main content shows the command 'python3' being executed, resulting in the Python 3.8.0 startup screen: 'Python 3.8.0 (default, Oct 15 2019, 11:07:42) [GCC 7.4.0] on linux'. It prompts the user to type 'help', 'copyright', 'credits', or 'license' for more information. The prompt '>>>' is followed by a cursor.

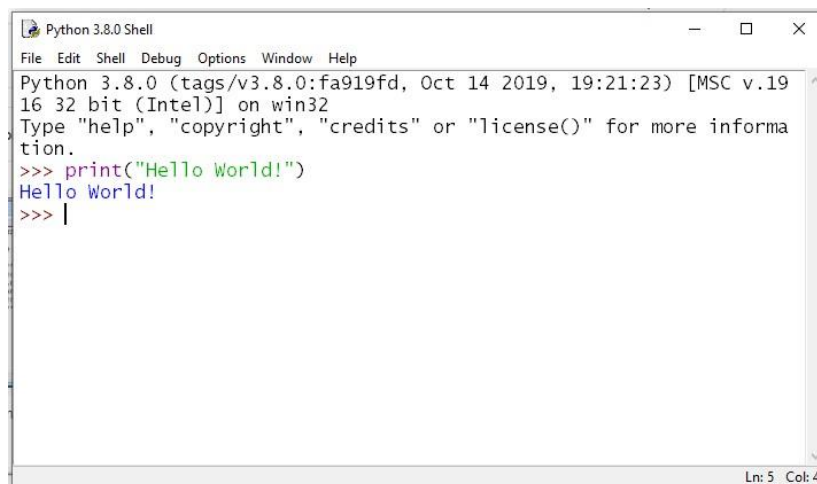
Executing a Python script:

Python programs are very similar to text files; they can be written with something as simple as a basic text editor. Type and save the script as a text file with extension .py

To run a Python script (program), Navigate the terminal to the directory where the script is located using the cd command. Type `python program_name.py` in the terminal to execute the script.

2. IDLE

IDLE stands for Integrated Development and Learning Environment. IDLE is Python's Integrated Development and Learning Environment. IDLE, is a very simple and sophisticated IDE developed primarily for beginners. It offers a variety of features like Python shell with syntax highlighting, Multi-window text editor, Code autocompletion, Intelligent indenting, Program animation and stepping for debugging, etc. (<https://docs.python.org/3/library/idle.html>)

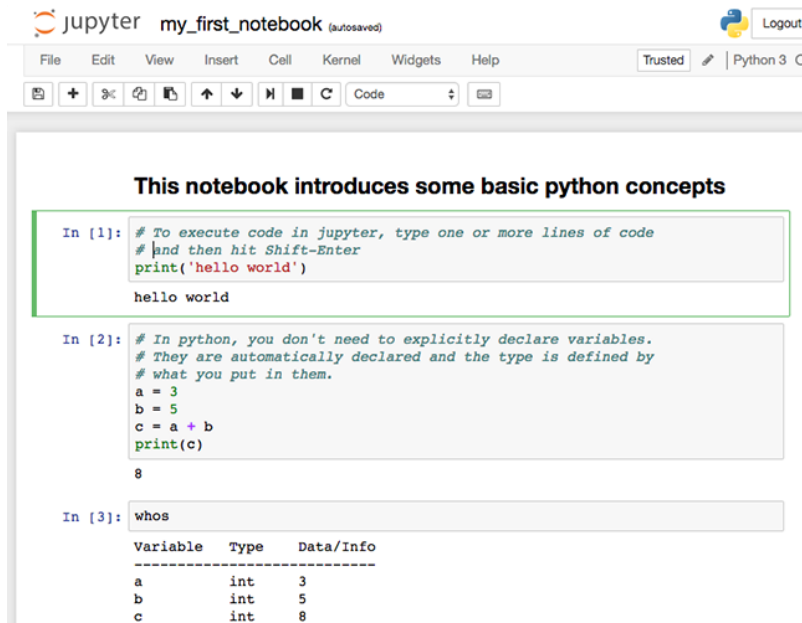
A screenshot of the Python 3.8.0 Shell window. The title bar says 'Python 3.8.0 Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main content shows the Python 3.8.0 startup screen with version and architecture information. The user has entered the command `>>> print("Hello World!")`, and the output 'Hello World!' is displayed. The prompt '>>>' is followed by a cursor. The status bar at the bottom right shows 'Ln: 5 Col: 4'.

3. Jupyter notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. It needs to be installed separately. You can use a handy tool that comes with Python called **pip** to install Jupyter Notebook like this:

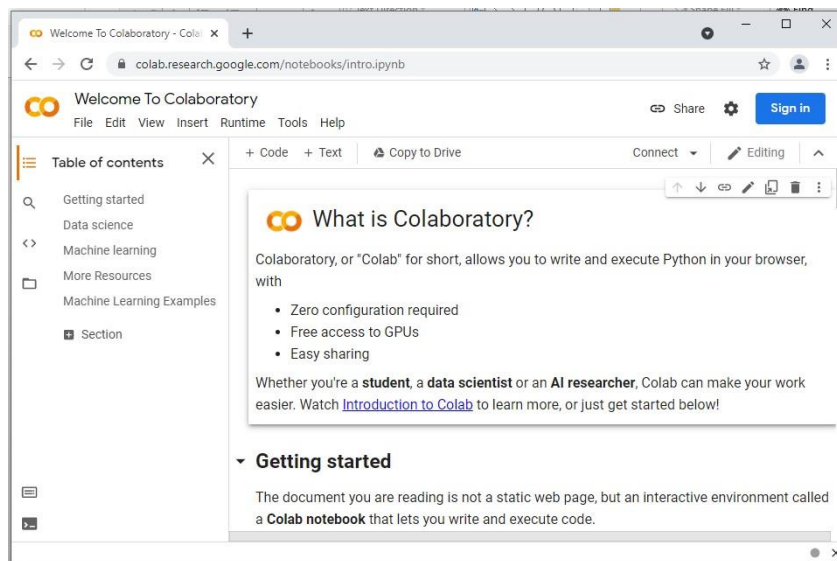
```
$ pip install jupyter
```

The other most popular distribution of Python is **Anaconda**. Anaconda has its own installer tool called **conda** that is used for installing packages. However, Anaconda comes with many scientific libraries preinstalled, including the Jupyter Notebook (<https://jupyter.org/>)



4. Google Colaboratory

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to data science, data analysis and machine learning. (<https://colab.research.google.com/notebooks/intro.ipynb>)



DATA SCIENCE

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data is the foundation of data science. Data comes from various sources, in various forms and sizes.

A **dataset** is a complete collection of all observations arranged in some order.

DATA SCIENCE TASKS

1. Data Exploration – finding out more about the data to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data and find correlations, general trends, and outliers. The basic Steps in Data Exploration are:

- a. Import the library
 - b. Load the data
 - c. What does the data look like?
 - d. Perform exploratory data analysis
 - e. Visualize the data
2. Data Munging – cleaning the data and playing with it to make it better suited for statistical modeling
 3. Predictive Modeling – running the actual data analysis and machine learning algorithms

For the above process, we need to get acquainted with some useful Python libraries.

PYTHON LIBRARIES FOR DATA SCIENCE

There are several libraries that have been developed for data science and machine learning. Following are a list of libraries, you will need for any data science tasks:



Python Libraries for Data Processing

- **NumPy** stands for Numerical Python. This library is mainly used for working with arrays. Operations include adding, slicing, multiplying, flattening, reshaping, and indexing the arrays. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities etc.
- **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
- **Pandas** is used for structured data operations and manipulations. Pandas provides various high-performance and easy-to-use data structures and operations for manipulating data in the form of numerical tables and time series. It provides easy data manipulation, data aggregation, reading, and writing the data as well as data visualization. Pandas can also take in data from different types of files such as CSV, excel etc. or a SQL database and create a Python object known as a data frame. The two main data structures that Pandas supports are:
 - **Series**: Series are one-dimensional data structures that are a collection of any data type.
 - **DataFrames**: DataFrames are two dimensional data structures which resemble a database table or say an Excel spreadsheet.

- **Statsmodels** for statistical modeling. Statsmodels is part of the Python scientific stack, oriented towards data science, data analysis and statistics. It allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.

Python Libraries for Data Visualization

- **Matplotlib** for plotting vast variety of graphs like bar charts, pie charts, histograms, scatterplots, histograms to line plots to heat plots. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers etc. It can be used to embed plots into applications. The Pyplot module also provides a MATLAB-like interface that is just as versatile and useful.
- **Seaborn** for statistical data visualization. Seaborn is a Python data visualization library that is based on Matplotlib and closely integrated with the numpy and pandas data structures. It is used for making attractive and informative statistical graphics in Python.
- **Plotly** Plotly is a free open-source graphing library that can be used to form data visualizations. Plotly (plotly.py) is built on top of the Plotly JavaScript library (plotly.js) and can be used to create web-based data visualizations that can be displayed in Jupyter notebooks or web applications using Dash or saved as individual HTML files.
- **Bokeh** for creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high-performance interactivity over very large or streaming datasets.

Python libraries for Data extraction

- **BeautifulSoup** is a parsing library in Python that enables web scraping from HTML and XML documents.
- **Scrapy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information. It gives you all the tools you need to efficiently extract data from websites, process them as you want, and store them in your preferred structure and format.

Python Libraries for Machine learning

- **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
- **TensorFlow** Developed by Google Brain Team, TensorFlow is an open-source library used for deep learning applications.
- **Keras** It is an open-source library that provides an interface for the TensorFlow library and enables fast experimentation with deep neural networks.

Installing a library

Python comes with a package manager for Python packages called **pip** which can be used to install a library. PIP is a recursive acronym for “Preferred Installer Program” or “PIP Installs Packages”

```
pip install package-name
```

Importing a library

The first step to use a package is to import them into the Programming environment. There are several ways of doing so in Python:

- i) `import package-name`
- ii) `import package-name as alias`
- iii) `from package-name import *`

In i, the specified package name is imported in to the environment. Subsequently, we have to use the whole package-name each time we have to access any function or method.

Example:

```
import numpy  
a = numpy.array([2, 3, 4,5])
```

In ii, we have defined an alias to the package-name. Instead of using the ehole package-name, a short alias can be used.

```
import numpy as np  
a = np.array([2, 3, 4,5])
```

In iii, we have imported the entire name space in the package. i.e. you can directly use all methods and operations without referring to the package-name.

```
from numpy import *  
a = array([2, 3, 4,5])
```

Importing the dataset

We can find various dataset sources which are freely available for the public to work on. Few popular websites through which we can access datasets are:

1. Kaggle : <https://www.kaggle.com/datasets>.
2. UCI Machine learning repository: <https://archive.ics.uci.edu/ml/index.php>.
3. Datasets at AWS resources: <https://registry.opendata.aws/>.
4. Google's dataset search engine <https://datasetsearch.research.google.com/>
5. Government datasets: There are different sources to get government-related data. Various countries publish government data for public use collected by them from different departments. The Open Government Data Platform, India publishes several Datasets for general use at <https://data.gov.in/>

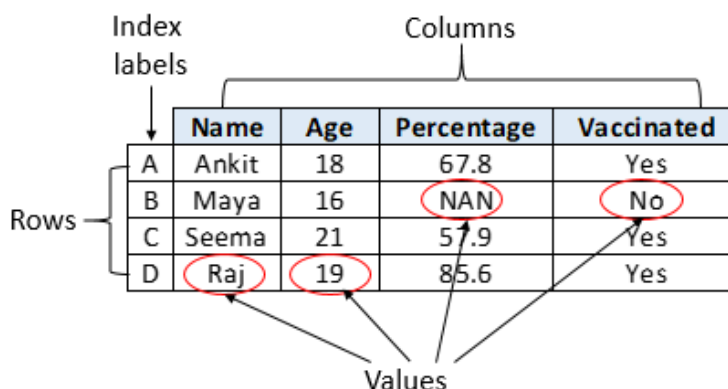
Creating own dataset

If data is not available for a particular project, you may extract or collect data on your own and create your own dataset to work with. The Self activity exercises show some examples of creating own datasets.

Data in datasets can be of the following types:

1. Numerical - quantitative data
 - a. Continuous - Any value within a range ex. Temperature
 - b. discrete - exact and distinct values ex. Number of students enrolled
2. Categorical - Qualitative data
 - a. Nominal - Naming or labeling variables without order ex. Country name
 - b. Ordinal - labels that are ordered or ranked in some particular way ex. Exam Grades

To perform operations on a dataset, it is stored as a dataframe in Python. A dataframe is a 2-dimensional labeled data structure with columns of potentially different types. The Pandas library is most useful for working with dataframes.



There are multiple ways to create own dataframes - Lists, dictionary, Series, Numpy ndarrays, using another dataframe. To carry out operations on the dataframe, the following pandas functions are commonly required:

Function	Description	Example
<code>read_csv()</code>	<code>read_csv()</code> function helps read a comma-separated values (csv) file into a Pandas DataFrame. Other functions are <code>read_json()</code> , <code>read_html()</code> , <code>read_excel()</code>	<pre>import pandas as pd df = pd.read_csv('filename.csv')</pre>
<code>head()</code> <code>tail()</code>	<code>head(n)</code> is used to return the first n rows of a dataset. By default, <code>df.head()</code> will return the first 5 rows of the DataFrame. <code>tail()</code> gives rows from the bottom of the dataframe	<code>df.head(6)</code>
<code>describe()</code>	<code>describe()</code> is used to generate descriptive statistics of the data in a Pandas DataFrame to get a quick overview of the data.	<code>df.describe()</code>
<code>info()</code>	It displays information about data such as the column names, no. of non-null values in each column(feature), type of each column, memory usage, etc.	<code>df.info()</code>
<code>dtypes</code>	Dtypes shows the data type of each column.	<code>df.dtypes</code>
<code>shape</code> <code>size</code>	<code>shape</code> returns the number of rows and columns in the dataframe. <code>size</code> returns the size of a dataframe which is the number of rows multiplied by the number of columns.	<code>df.shape</code> <code>df.size</code>
<code>value_counts()</code>	To identify the different categories in a feature as well as the count of values per category.	<code>df['column'].value_counts()</code>
<code>sample()</code>	Sample method allows you to select values randomly from a Series or DataFrame. It is useful when we want to select a random sample from a distribution.	<code>df.sample(10)</code>
<code>drop_duplicates()</code>	<code>drop_duplicates()</code> returns a Pandas DataFrame with duplicate rows removed. <code>inplace=True</code> makes sure the changes are applied to the original dataset.	<code>df.drop_duplicates(inplace=True)</code>
<code>sort_values()</code>	To sort column in a Pandas DataFrame by values in ascending or descending order. By specifying the <code>inplace=True</code> , you can make a change directly in the original DataFrame.	<code>df.sort_values(by=columnname or list of columns, inplace=True)</code>
<code>loc[:]</code>	To access a row or group of rows (a slice of the dataset). index starts from 0 in Python, and that <code>loc[:]</code> is inclusive on both values	<code>df.loc[:]</code> #All rows <code>df.loc[0]</code> #First row

		df.loc[1:4] #Rows 1,2,3 and 4 df.loc[3:] #All Rows from row 3 df.loc[1:3,['column','column']] #Rows 1,2,3 with specific columns
drop()	To drop a particular column in the dataframe. inplace=True makes changes in the dataframe.	df.drop()
query()	To filter a dataframe based on a condition or apply a mask to get certain values. Query the columns by applying a Boolean expression.	df.query('year'>2019)
insert()	Inserts a new column in a dataframe.	df.insert(pos,'columnname', values)
isnull() notnull()	Detect missing values. Return a boolean same-sized object indicating if the values are NA. notnull() detects non missing values	df.isnull()
dropna() fillna()	Pandas has a built-in method called dropna. When applied against a DataFrame, the dropna method will remove any rows that contain a NaN value. In many cases, you will want to replace missing values in a Pandas DataFrame instead of dropping it completely. The fillna method is designed for this.	df.dropna() df.fillna("No value available")

Self Activity

I. To create a dataset

	company	model	year
0	Tata	Nexon	2017
1	MG	Astor	2021
2	KIA	Seltos	2019
3	Hyundai	Creta	2015

The above dataset can be created in several ways. Type and execute the code in the following activities.

Activity 1: Create empty dataframe and add records

```
#Import the library
import pandas as pd
#Create an empty data frame with column names
df = pd.DataFrame(columns = ['company', 'model', 'year'])
#Add records
df.loc[0] = ['Tata', 'Nexon', 2017]
df.loc[1] = ['MG', 'Astor', 2021]
df.loc[2] = ['KIA', 'Seltos', 2019]
df.loc[3] = ['Hyundai', 'Creta', 2015]
#Print the dataframe
df
```

Activity 2: Create dataframe using numpy array

```
#Import the library
```

```

import numpy as np
# Pass a 2D numpy array - each row is the corresponding row in the dataframe
data = np.array([['Tata', 'Nexon', 2017],
                 ['MG', 'Astor', 2021],
                 ['KIA', 'Seltos', 2019],
                 ['Hyundai', 'Creta', 2015]])
# pass column names in the columns parameter of the constructor
df = pd.DataFrame(data, columns = ['company', 'model', 'year'])
df

```

Activity 3: Create dataframe using list

Each element of the list is one record in the dataframe. The column headers are passed separately.

```

# Import pandas library
import pandas as pd
# Create a list of lists
data = [['Tata', 'Nexon', 2017], ['MG', 'Astor', 2021], ['KIA', 'Seltos', 2019], ['Hyundai', 'Creta', 2015]]
# Create the pandas DataFrame
df = pd.DataFrame(data, columns = ['company', 'model', 'year'])
# print the dataframe.
df

```

Activity 4: Create dataframe using dictionary with lists

```

#Import the library
import pandas as pd
#Create the dictionary
data = {'company': ['Tata', 'MG', 'KIA', 'Hyundai'],
        'model': ['Nexon', 'Astor', 'Seltos', 'Creta'],
        'year': [2017, 2021, 2019, 2015]}
#Create the dataframe
df = pd.DataFrame.from_dict(data)
#Print the dataframe
df

```

Activity 5: Create dataframe using list of dictionary

```

import pandas as pd
#Each dictionary is a record in the dataframe.
#Dictionary Keys become Column names in the dataframe. Dictionary values become the values of columns
data = [{'company': 'Tata', 'model': 'Nexon', 'year': 2017},
        {'company': 'MG', 'model': 'Astor', 'year': 2021},
        {'company': 'KIA', 'model': 'Seltos', 'year': 2019},
        {'company': 'Hyundai', 'model': 'Creta', 'year': 2015}]
df = pd.DataFrame(data)
df

```

Activity 6: Create dataframe using zip() function

```

import pandas as pd
#Company list
company = ['Tata', 'MG', 'KIA', 'Hyundai']
#Model list

```

```

model = ['Nexon', 'Astor', 'Seltos', 'Creta']
#Year list
year = [2017, 2021, 2019, 2015]
# and merge them by using zip().
data = list(zip(company, model, year))
# Create pandas Dataframe using data
df = pd.DataFrame(data, columns = ['company', 'model', 'year'])
# Print dataframe
df

```

Activity 7: Understand the data

Apply functions info(), describe(), shape, size, loc, sort_values, value_counts() on the dataframe object.

Activity 8: Adding rows and columns with Invalid, duplicate or missing data

#Add rows to the dataset with empty or invalid values.

```

df.loc[4] = ['Honda', 'Jazz', None]
df.loc[5]= [None, None, None]
df.loc[6]=['Toyota', None, 2018]
df.loc[7]=['Tata', 'Nexon', 2017]

```

```

#Insert empty column
df["newcolumn"] = None
df

```

Note: a new column can be added with specific values. For example, if we want the new column to have model name + year name, the command will be:

```

df['new']=df['model']+' '+df['year'].astype(str) #Convert year to string

```

Activity 9: Check for NULL and duplicate values

```

#View null values
df.isnull() # df.notnull() shows all values that are not null
df.duplicated() #Shows the rows with duplicate entries

```

Activity 10: Replace NULL values with fillna() function

To replace all Empty values with the string “No Value Available”

```

#DataFrame.fillna() to replace Null values in dataframe
df.fillna("No Value Available", inplace = True)
df

```

Activity 11: Drop a column from the dataframe

To drop the column named “newcolumn”

```

df.drop(columns='newcolumn', axis=1, inplace=True)
df

```

II : To use a standard dataset

The **iris** flowers dataset(also known as the Fisher's Iris dataset) is the "hello world" of data science and machine learning. It is a multivariate data set which contains a set of 150 records under 5 attributes - Petal Length, Petal Width, Sepal Length, Sepal width and Class(Species). The downloadable dataset (.csv format) can be found at: <https://archive.ics.uci.edu/ml/datasets/iris> ,

<https://www.kaggle.com/uciml/iris>

Activity 12: Load the iris dataset

If you are using a local machine to run python code, download the iris dataset and type the following code. The pathname is the location of the dataset in the local drive.

```
import pandas as pd
df=pd.read_csv('\iris.csv') #Give the path as per the location in your machine
df
```

If you are using google colab, then run the following code to load the dataset.

```
from google.colab import files
iris_uploaded = files.upload()
```

This will prompt you for selecting the csv file to be uploaded. To load the file, run the following code:

```
import io
df = pd.read_csv(io.BytesIO(iris_uploaded['Iris.csv']))
df
```

Activity 13: Understand the iris dataset

Type and execute the following commands:

- i. `df.shape`
Answer the following:
 - a. How many rows and columns in the dataset?
- ii. `df.describe()`
Answer the following:
 - a. Mean values
 - b. Standard Deviation
 - c. Minimum Values
 - d. Maximum Values
- iii. `df.info()`
Answer the following:
 - a. Does any column have null entries?
 - b. How many columns have numeric data?
 - c. How many columns have categorical data?
- iv. `df.dtypes`
- v. `df.head(), df.head(20)`
- vi. `df.tail()`
- vii. `df.sample(10)`
- viii. `df.size`
- ix. `df.columns`
What are the names of the columns?
- x. For each column name, run the following command:
`df['columnname'].value_counts()`
- xi. View rows 10 to 20
`sliced_data=df[10:20]`
`print(sliced_data)`
- xii. Selecting specific columns
`specific_data=df[["Id", "Species"]]`
`print(specific_data)`

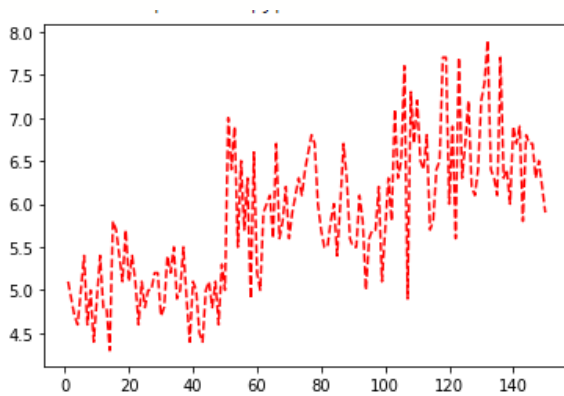
Activity 14: Visualize the dataset

Matplotlib.pyplot library is most commonly used in Python for plotting graphs both in 2d and 3d format. It provides features of legend, label, grid, graph shape, grid and many more. Seaborn is another library that provides different and attractive graphs and plots. Type and execute the following commands:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

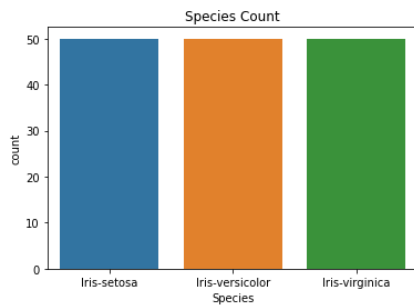
Plot Sepal length for all the entries

```
plt.plot(df.Id, df["SepalLengthCm"], "r--")
plt.show
```



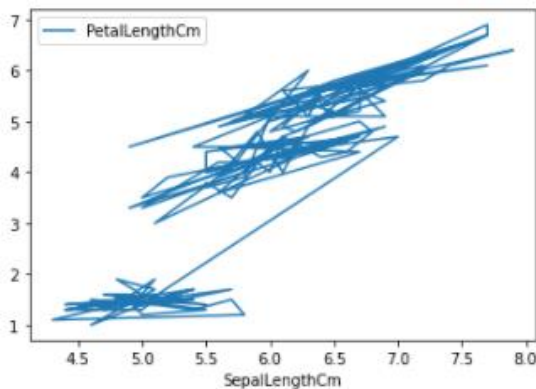
To display the species count for each species

```
plt.title('Species Count')
sns.countplot(df['Species'])
```



To create a line plot of Sepal length vs Petal Length

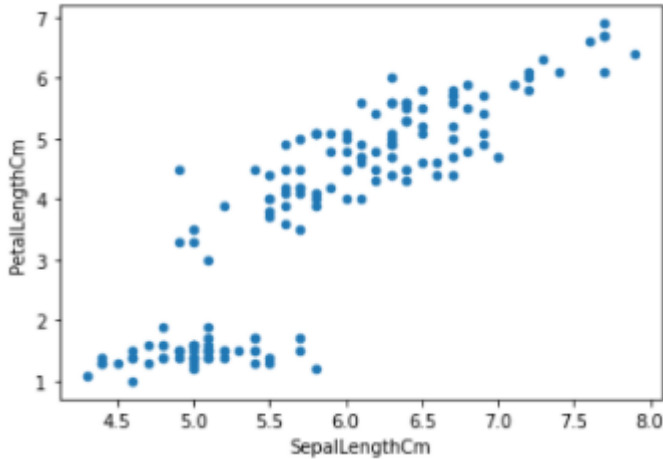
```
df.plot(x="SepalLengthCm", y="PetalLengthCm")
plt.show()
```



To create a scatterplot of Sepal length vs Petal Length using scatterplot.

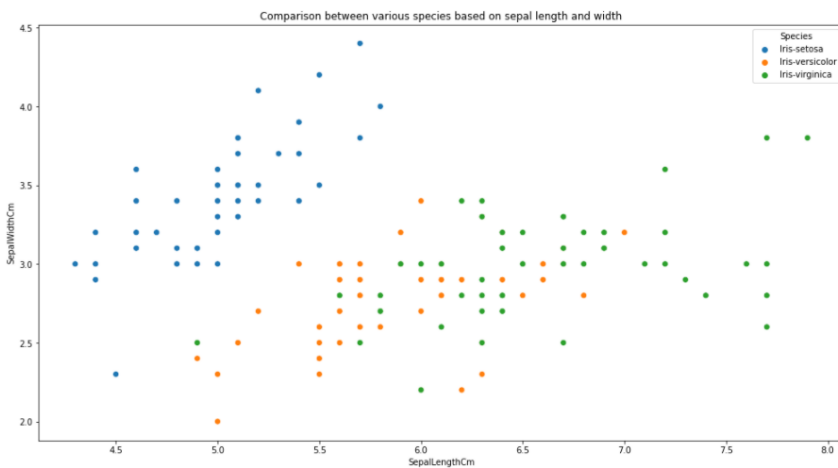
We can plot by matplotlib or seaborn. For matplotlib, we can either call the `.plot.scatter()` method in pandas, or use `plt.scatter()`. For seaborn, we use `sns.scatterplot()` function.

1. `df.plot(kind="scatter", x='SepalLengthCm', y='PetalLengthCm')`
2. `df.plot.scatter(x='SepalLengthCm', y='PetalLengthCm')`
3. `plt.scatter(df['SepalLengthCm'], df['PetalLengthCm'])`



To plot the comparative graph of sepal length vs sepal width for all species using scatterplot

```
plt.figure(figsize=(17,9))
plt.title('Comparison between various species based on sepal length and width')
sns.scatterplot(df['SepalLengthCm'], df['SepalWidthCm'], hue=df['Species'], s=50)
```



Set A

1. Write a Python program to create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe.
2. Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data
3. Write a Python program to view basic statistical details of the data.
4. Write a Python program to Add 5 rows with duplicate values and missing values. Add a column 'remarks' with empty values. Display the data
5. Write a Python program to get the number of observations, missing values and duplicate values.
6. Write a Python program to drop 'remarks' column from the dataframe. Also drop all null and empty values. Print the modified data.

7. Write a Python program to generate a line plot of name vs percentage
8. Write a Python program to generate a scatter plot of name vs percentage

Set B

1. Download the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 10 rows and random 20 rows. (<https://www.kaggle.com/burnoutminer/heights-and-weights-dataset>)
2. Write a Python program to find the shape, size, datatypes of the dataframe object.
3. Write a Python program to view basic statistical details of the data.
4. Write a Python program to get the number of observations, missing values and nan values.
5. Write a Python program to add a column to the dataframe “BMI” which is calculated as : $\text{weight}/\text{height}^2$
6. Write a Python program to find the maximum and minimum BMI.
7. Write a Python program to generate a scatter plot of height vs weight.

Signature of the instructor

Date

Assignment Evaluation

0 Not done 3 Needs Improvement	1 Incomplete 4 Complete	2 Late Complete 5 Well Done
<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>	<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>	<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>
<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>	<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>	<input style="width: 100%; height: 100%; border: 1px solid black;" type="text"/>

Assignment 2: Statistical Data Analysis

No. of sessions: 01

Objectives

- To understand an experimental foundation for data analysis using statistical methods learned in the theory.
- To learn statistical libraries in Python and use these on actual data.
- To write basic Python scripts using various functions defined in these libraries.
- To understand importance of data analysis using statistical methods in various fields.

Reading

You should read the following topics before starting this exercise:

Why to do data analysis, Difference between inferential statistics and descriptive statistics, Features of Python Programming Language, Basics of Python, Basic scripting in Python, Types of attributes,

Ready Reference

Understanding Descriptive Statistics

Descriptive statistics is about describing and summarizing data.

It uses two main approaches:

- The quantitative approach describes and summarizes data numerically.
- The visual approach illustrates data with charts, plots, histograms, and other graphs.

You can apply descriptive statistics to one or many datasets or variables. When you describe and summarize a single variable, you're performing univariate analysis. When you search for statistical relationships among a pair of variables, you're doing a bivariate analysis. Similarly, a multivariate analysis is concerned with multiple variables at once.

Types of Measures

Central tendency tells you about the centers of the data. Useful measures include the mean, median, and mode.

Variability tells you about the spread of the data. Useful measures include variance and standard deviation, range and interquartile range.

Correlation or joint variability tells you about the relation between a pair of variables in a dataset. Useful measures include covariance and the correlation coefficient.

Proximity Measures

Proximity measures are basically mathematical techniques that are used to check similarity between objects or one can say dissimilarity between objects. These measures are based on distance measures. Although it can be based on probabilities also.

Role of Distance Measures

Distance measures play vital role in case of Machine Learning (or in algorithms from data science field.) algorithms such as K-Nearest Neighbors, Learning Vector Quantization (LVQ), Self-Organizing Map (SOM), K-Means Clustering use this technique while computing.

There are four types of distances mostly getting used in the field of machine learning viz. Hamming Distance, Euclidean Distance, Manhattan Distance, Minkowski Distance.

We may have numeric, ordinal or categorical attributes and hence types of objects. In such cases, steps to be taken to get value of proximity measures are different.

e. g. Proximity measures for ordinal attributes

An ordinal attribute is an attribute whose possible values have a meaningful order or ranking among them. Here, we do not aware of gap between two values. So, first we scale up all values starting from low to high or small to large with specific range of numbers. i. e. consider movie rating (1-10), satisfaction rate at shopping mall (1-5) etc. then normalize the values using formula. Then find dissimilarity matrix for it. Such measures are helpful/required in clustering, outlier analysis, and nearest neighbour classification.

Outliers

An outlier is a data point that differs significantly from the majority of the data taken from a sample or population. There are many possible causes of outliers, but here are a few to start you off:

- Natural variation in data
 - Change in the behavior of the observed system
 - Errors in data collection
 - Data collection errors are a particularly prominent cause of outliers. For example, the limitations of measurement instruments or procedures can mean that the correct data is simply not obtainable. Other errors can be caused by miscalculations, data contamination, human error, and more.
-
- **Population and Samples**
 - In statistics, the **population** is a set of all elements or items that you're interested in. Populations are often vast, which makes them inappropriate for collecting and analyzing data. That's why statisticians usually try to make some conclusions about a population by choosing and examining a representative subset of that population.
 - This subset of a population is called a **sample**. Ideally, the sample should preserve the essential statistical features of the population to a satisfactory extent. That way, you'll be able to use the sample to glean conclusions about the population.
 - There are two major categories of sampling techniques viz. 1. Probability Sampling Techniques and 2. Non-probability Sampling Techniques.
 - Simple Random Sampling, Stratified Random Sampling and Systematic Random Sampling are probability sampling techniques whereas Convenience Sampling, Proportional Quota Sampling, Purposive Sampling, and Snowball Sampling are non-probability sampling techniques.

Self Activity

Choosing Python Statistics Libraries

Python's statistics is a built-in Python library for descriptive statistics. You can use it if your datasets are not too large or if you can't rely on importing other libraries.

NumPy is a third-party library for numerical computing, optimized for working with single- and multi-dimensional arrays. Its primary type is the array type called ndarray. This library contains many routines for statistical analysis.

SciPy is a third-party library for scientific computing based on NumPy. It offers additional functionality compared to NumPy, including scipy. stats for statistical analysis. It is an open-source library and it is useful when we want to solve problems from mathematics, scientific, engineering, and technical situations/cases. It allows us to manipulate the data and visualize results using Python command.

Pandas is a third-party library for numerical computing based on NumPy. It excels in handling labeled one-dimensional (1D) data with Series objects and two-dimensional (2D) data with DataFrame objects. Function describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. This function describe() takes argument 'include' that is used to pass necessary information regarding which columns need to be considered for summarizing. Values can be either 'all' or specific value.

df.head() is a method that displays the first 5 rows of the dataframe. Similarly, tail() is the function that displays last 5 rows of the dataframe.

Matplotlib is a third-party library for data visualization. It works well in combination with NumPy, SciPy, and Pandas.

Pandas has a describe() function that provides basic details about dataframe. But, when we are interested in more details for EDA (exploratory data analysis) pandas_profiling will be useful. It states about types, unique values, missing values, statistics basics, etc. When we use pandas profiling we can show results into formatted output such as HTML

Self-Activities

Example 1:

```
import numpy as np
demo = np.array([[30,75,70],[80,90,20],[50,95,60]])
print demo
print '\n'
print np.mean(demo)
print '\n'
print np.median(demo)
print '\n'
```

Output

```
[[30 75 70]
 [80 90 20]
 [50 95 60]]
63.33333333333333
70.0
```

Example 2:

```
import pandas as pd
import numpy as np
d = {'Name':pd.Series(['Ram','Sham','Meena','Seeta','Geeta','Rakesh','Madhav']),
     'Age':pd.Series([25,26,25,23,30,29,23]),
     'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}
df = pd.DataFrame(d)
print(df.sum())
```

Output

```
Name    RamShamMeenaSeetaGeetaRakeshMadhav
Age                                181
Rating                             25.61
dtype: object
```

Example 3:

```
import pandas as pd
import numpy as np
md= {'Name':pd.Series(['Ram','Sham','Meena','Seeta','Geeta','Rakesh','Madhav']),
     'Age':pd.Series([25,26,25,23,30,29,23]),
     'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}
df = pd.DataFrame(md)
print(df.describe())
```

Output

```
           Age  Rating
count  7.000000  7.000000
mean   25.857143  3.658571
std     2.734262  0.698628
min     23.000000  2.560000
25%    24.000000  3.220000
50%    25.000000  3.800000
75%    27.500000  4.105000
max     30.000000  4.600000
```

Example 4:

```
import numpy as np
data = np.array([13,52,44,32,30, 0, 36, 45])
print("Standard Deviation of sample is % s "% (np.std(data)))
```

Output

```
Standard Deviation of sample is 16.263455967290593
```

Example 5:

```
import pandas as pd
```

```

import scipy.stats as s
score={'Virat': [92,97,85,74,71,55,85,63,42,32,71,55], 'Rohit' :
[89,87,67,55,47,72,76,79,44,92,99,47]}
df=pd.DataFrame(score)
print(df)
print("\nArithmetic Mean Values")
print("Score 1", s.tmean(df["Virat"]).round(2))
print("Score 2", s.tmean(df["Rohit"]).round(2))

```

Output

	Virat	Rohit
0	92	89
1	97	87
2	85	67
3	74	55
4	71	47
5	55	72
6	85	76
7	63	79
8	42	44
9	32	92
10	71	99
11	55	47

Arithmetic Mean Values

Score 1 68.5

Score 2 71.17

Example 6:

```

import pandas as pd
from scipy.stats import iqr
import matplotlib.pyplot as plt
d={'Name': ['Ravi', 'Veena', 'Meena', 'Rupali'], 'subject': [92,97,85,74]}
df=pd.DataFrame(d)
df = pd.DataFrame(d)
df['Percentile_rank']=df.subject.rank(pct=True)
print("\n Values of Percentile Rank in the Distribution")
print(df)
print("\n Measures of Dispersion and Position in the Distribution")
r=max(df["subject"]) - min(df["subject"])
print("\n Value of Range in the Distribution = ", r)
s=round(df["subject"].std(),3)

```

```
print("Value of Standard Deviation in the Distribution = ", s)
v=round(df["subject"].var(),3)
print("Value of Variance in the Distribution = ", v)
```

Output

Values of Percentile Rank in the Distribution

	Name	subject	Percentile_rank
0	Ravi	92	0.75
1	Veena	97	1.00
2	Meena	85	0.50
3	Rupali	74	0.25

Measures of Dispersion and Position in the Distribution

Value of Range in the Distribution = 23

Value of Standard Deviation in the Distribution = 9.967

Value of Variance in the Distribution = 99.333

Example 7

```
mydata = np.array([24, 29, 20, 22, 24, 26, 27, 30, 20, 31, 26, 38, 44, 47])
q3, q1 = np.percentile(mydata, [75 ,25])
iqrvalue = q3 - q1
print(iqrvalue)
```

Output

6.75

Lab Assignments

SET A:

1. Write a Python program to find the maximum and minimum value of a given flattened array.

Expected Output:

Original flattened array:

```
[[0 1]
```

```
[2 3]]
```

Maximum value of the above flattened array:

3

Minimum value of the above flattened array:

0

2. Write a python program to compute Euclidian Distance between two data points in a dataset. [Hint: Use linalg.norm function from NumPy]
3. Create one dataframe of data values. Find out mean, range and IQR for this data.
4. Write a python program to compute sum of Manhattan distance between all pairs of points.

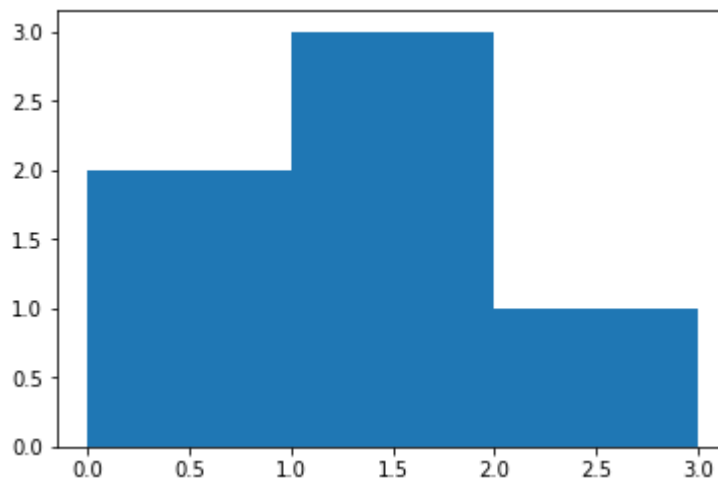
- Write a NumPy program to compute the histogram of nums against the bins.

Sample Output:

```
nums: [0.5 0.7 1. 1.2 1.3 2.1]
```

```
bins: [0 1 2 3]
```

```
Result: (array([2, 3, 1], dtype=int64), array([0, 1, 2, 3]))
```



- Create a dataframe for students' information such name, graduation percentage and age. Display average age of students, average of graduation percentage. And, also describe all basic statistics of data. (Hint: use describe()).

SET B:

- Download iris dataset file. Read this csv file using read_csv() function. Take samples from entire dataset. Display maximum and minimum values of all numeric attributes.
- Continue with above dataset, find number of records for each distinct value of class attribute. Consider entire dataset and not the samples.
- Display column-wise mean, and median for iris dataset from Q.4 (Hint: Use mean() and median() functions of pandas dataframe).

SET C:

- Write a python program to find Minkowskii Distance between two points.
- Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array.

From Wikipedia: The weighted arithmetic mean is similar to an ordinary arithmetic mean (the most common type of average), except that instead of each of the data points contributing equally to the final average, some data points contribute more than others. The notion of weighted mean plays a role in descriptive statistics and also occurs in a more general form in several other areas of mathematics.

Sample output:

Original flattened array:

```
[[0 1 2]
```

```
[3 4 5]
```

```
[6 7 8]]
```

Weighted average along the specified axis of the above flattened array:

```
[1.2 4.2 7.2]
```

- Write a NumPy program to compute cross-correlation of two given arrays.

Sample Output:

Original array1:

[0 1 3]

Original array2:

[2 4 5]

Cross-correlation of the said arrays:

[[2.33333333 2.16666667]

[2.16666667 2.33333333]]

4. Download any dataset from UCI (do not repeat it from set B). Read this csv file using read_csv() function. Describe the dataset using appropriate function. Display mean value of numeric attribute. Check any data values are missing or not.
5. Download nursery dataset from UCI. Split dataset on any one categorical attribute. Compare the means of each split. (Use groupby)
6. Create one dataframe with 5 subjects and marks of 10 students for each subject. Find arithmetic mean, geometric mean, and harmonic mean.
7. Download any csv file of your choice and display details about data using pandas profiling. Show stats in HTML form.

Signature of the instructor

Date

Assignment Evaluation

0: Not done

2: Late Complete

4: Complete

1: Incomplete

3: Needs improvement

5: Well Done

Assignment 3 : Data Preprocessing

Number of Slots = 1

Objectives

- To learn about Basic data Preprocessing
- To learn operations of Data Munging.
- Apply data processing tools of python on various datasets.

Reading

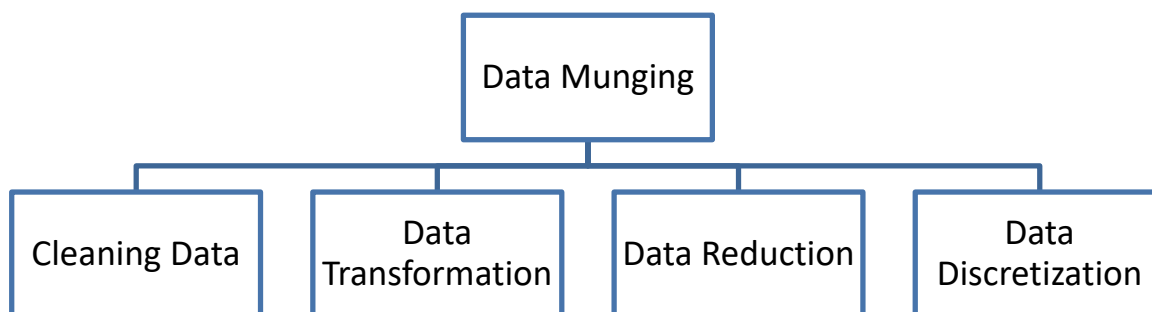
You should read the following topics before starting this exercise:

Importing libraries and Datasets, Handling Dataframe, Basic concepts of data preprocessing, Handling missing data, Data munging process includes operations such as Cleaning Data, Data Transformation, Data Reduction and Data Discretization.

Ready Reference

Introduction: Data Munging / Wrangling Operations

- Data munging or wrangling refers to preparing data for a dedicated purpose - taking the data from its raw state and transforming and mapping into another format, normally for use beyond its original intent and can be used for it more appropriate and valuable for a variety of downstream purposes such as analytics.
- Data munging process includes operations such as Cleaning Data, Data Transformation, Data Reduction and Data Discretization.



Steps for Data Preprocessing:

1. Importing the libraries
2. Importing the Dataset
3. Handling of Missing Data
4. Data Transformation techniques

Step.1 importing the libraries

1. NumPy:- it is a library that allows us to work with arrays and as most machine learning models work on arrays NumPy makes it easier
2. matplotlib:- this library helps in plotting graphs and charts, which are very useful while showing the result of your model
3. Pandas:- pandas allows us to import our dataset and also creates a matrix of features containing the dependent and independent variable.

Syntax:

- import numpy as np # used for handling numbers
- import pandas as pd # used for handling the dataset
- import SimpleImputer # used for handling missing data
from sklearn.preprocessing
- import LabelEncoder, OneHotEncoder # used for encoding categorical
data from sklearn.model_selection

Step2: Importing dataset

This step is used for providing input to programming workspace(python program)

We can use the `read_csv` function from the pandas library. The data imported using the `read_csv` function is in a Data Frame format, we'll later convert it into *NumPy arrays* to perform other operations.

	Country	Age	Income	Feedback
0	India	49	86400	Good
1	Brazil	32	57600	Excellent
2	USA	35	64800	Good
3	Brazil	43	73200	Fair
4	USA	45		Excellent
5	India	40	69600	Excellent
6	Brazil		62400	Poor
7	India	53	94800	Good
8	USA	55	99600	Fair
9	India	42	80400	Poor

Table 2.1: Data_1.csv

Syntax:

```
data=pd.read_csv('../input/Data_1.csv')
```

DataFrame

A DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

import pandas as pd

```
data = {  
    "calories": [420, 380, 390],  
    "duration": [50, 40, 45]  
}
```

#load data into a DataFrame object:

```
df = pd.DataFrame(data)
```

Locate Row using .loc[]

.loc[] is label based data selecting method which means that we have to pass the name of the row or column which we want to select. This method includes the last element of the range passed in it**1**. Selecting data according to some conditions :

For example:

```
# selecting cars with brand 'Maruti' and Mileage > 25  
display(data.loc[(data.Brand == 'Maruti') & (data.Mileage > 25)])
```

Locate Row using .iloc[]

.iloc[] is Purely integer-location based indexing for selection by position and it is primarily integer position based (from 0 to length-1 of the axis), but may also be used with a boolean array.

For eg:

```
>>>lst = [{'a': 1, 'b': 2, 'c': 3, 'd': 4},  
...      {'a': 10, 'b': 20, 'c': 30, 'd': 40},  
...      {'a': 100, 'b': 200, 'c': 300, 'd': 400 }]  
>>>d1 = pd.DataFrame(lst)  
>>>d1
```

```
   a  b  c  d  
0  1  2  3  4  
1 10 20 30 40  
2 100 200 300 400
```

.iloc[] examples with different types of inputs

- Type1(with integer inputs)

```
>>>d1.iloc[[0]]
```

```
   a  b  c  d
0  1  2  3  4
```

- Type2(with list of integer inputs)

```
>>>d1.iloc[[0, 1]]
```

```
   a  b  c  d
0  1  2  3  4
1 10 20 30 40
```

- Type3 (with a slice object)

```
>>>d1.iloc[:3]
```

```
   a  b  c  d
0  1  2  3  4
1 10 20 30 40
2 100 200 300 400
```

Describing Dataset:

`describe()` is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values.

Dataset shape:

To get the shape of DataFrame, use `DataFrame.shape`. The shape property returns a tuple representing the dimensionality of the DataFrame. The format of shape would be (rows, columns).

Step 3: Handling the missing values

Some values in the data may not be filled up for various reasons and hence are considered missing. In some cases, missing data arises because data was never gathered in the first place for some entities. Data analyst needs to take appropriate decision for handling such data.

1. Ignore the Tuple.
2. Fill in the Missing Value Manually.
3. Use a Global Constant to Fill in the Missing Value.
4. Use a Measure of Central Tendency for the Attribute (e.g., the Mean or Median) to Fill in the Missing Value.
5. Use the Attribute Mean or Median for all Samples belonging to the same Class as the given Tuple.
6. Use the most probable value to fill in the missing value.

#2.1 Program for Filling Missing Values

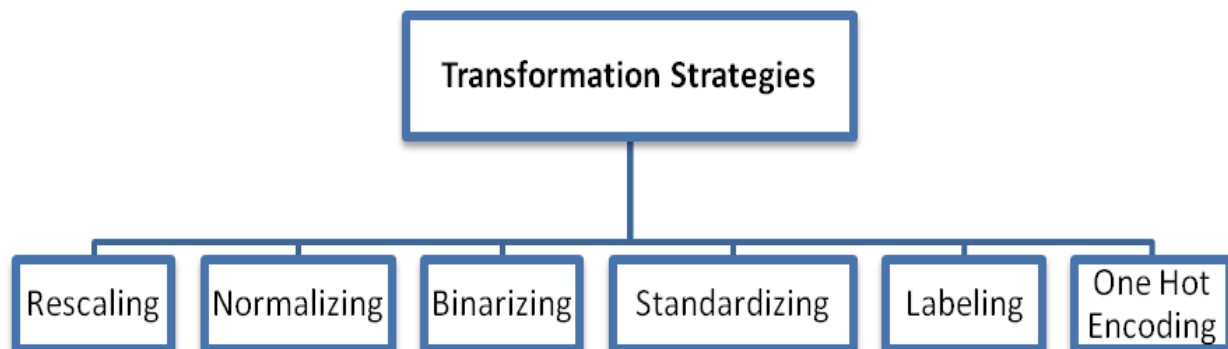
```

# import the pandas library
import pandas as pd
import numpy as np
#Creating a DataFrame with Missing Values
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'], columns=['C01', 'C02', 'C03'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print("\n Reindexed Data Values")
print("-----")
print(df)
#Method 1 - Filling Every Missing Values with 0
print("\n\n Every Missing Value Replaced with '0':")
print("-----")
print(df.fillna(0))
#Method 2 - Dropping Rows Having Missing Values
print("\n\n Dropping Rows with Missing Values:")
print("-----")
print(df.dropna())
#Method 3 - Replacing missing values with the Median
Valuedmedian = df['C01'].median()
df['C01'].fillna(median, inplace=True)
print("\n\n Missing Values for Column 1 Replaced with Median Value:")
print("-----")
print(df)

```

Step 4: Data Transformation

Data transformation is the process of converting raw data into a format or structure that would be more suitable for data analysis.



1. Rescaling:

Rescaling means transforming the data so that it fits within a specific scale, like 0-100 or 0-1. Rescaling of data allows scaling all data values to lie between a specified minimum and maximum value (say, between 0 and 1).

2. Normalizing:

The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to very different results.

3. Binarizing:

Binarizing is the process of converting data to either 0 or 1 based on a threshold value.

4. Standardizing Data:

In other words, Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation.

```
#2.2 Program for Data Transformation
import pandas as pd
import numpy as np
from sklearn import preprocessing
import scipy.stats as s
#Creating a DataFrame
d = {'C01':[1,3,7,4],'C02':[12,2,7,1],'C03':[22,34,-11,9]}
df2 = pd.DataFrame(d)
print("\n ORIGINAL DATA VALUES")
print("-----")
print(df2)
#Method 1: Rescaling Data
print("\n\n Data Scaled Between 0 to 1")
data_scaler = preprocessing.MinMaxScaler(feature_range = (0, 1))
data_scaled = data_scaler.fit_transform(df2)
print("\n Min Max Scaled Data ")
print("-----")
print(data_scaled.round(2))
#Method 2: Normalization rescales such that sum of each row is 1.
dn = preprocessing.normalize(df2, norm = 'l1')
print("\n L1 Normalized Data ")
print(" -----")
print(dn.round(2))
#Method 3: Binarize Data (Make Binary)
data_binarized = preprocessing.Binarizer(threshold=5).transform(df2)
print("\n Binarized data ")
print(" -----")
print(data_binarized)
#Method 4: Standardizing Data
print("\n Standardizing Data ")
print("-----")
X_train = np.array([[ 1., -1., 2.],[ 2., 0., 0.],[ 0., 1., -1.]])
print(" Orginal Data \n", X_train)
print("\n Initial Mean : ", s.tmean(X_train).round(2))
print(" Initial Standard Deviation : ",round(X_train.std(),2))
X_scaled = preprocessing.scale(X_train)
X_scaled.mean(axis=0)
X_scaled.std(axis=0)
print("\n Standardized Data \n", X_scaled.round(2))
print("\n Scaled Mean : ",s.tmean(X_scaled).round(2))
print(" Scaled Standard Deviation : ",round(X_scaled.std(),2))
```

Out

	C01	C02	C03
0	1	12	22
1	3	2	34
2	7	7	-11
3	4	1	9

Data Scaled Between 0 to 1
Min Max Scaled Data

```
-----
[[0.          1.    0.73 ]
 [0.33        0.09  1.    ]
 [1.          0.55  0.    ]
 [0.5         0.    0.44  ]]
L1 Normalized Data
-----
[[ 0.03  0.34  0.63]
 [ 0.08         0.05  0.87]
 [ 0.28  0.28 -0.44]
 [ 0.29         0.07  0.64]]
Binarized data
-----
[[0  1  1]
 [0  0  1]
 [1  1  0]
 [0  0  1]]
Standardizing Data
-----
Original Data
[[ 1.  -1.  2.]
 [ 2.  0.  0.]
 [ 0.  1. -1.]]
Initial Mean : 0.44
Initial Standard Deviation : 1.07
Standardized Data
[[ 0.          -1.22  1.34]
 [ 1.22         0.    -0.27]
 [-1.22        1.22 -1.07]]
Scaled Mean : 0.0
Scaled Standard Deviation : 1.0
```

5. Label Encoding:

The label encoding process is used to convert textual labels into numeric form in order to prepare it to be used in a machine-readable form. In label encoding, categorical data is converted to numerical data, and the values are assigned labels (such as 1, 2, and 3).

6. One Hot Coding:

One hot encoding refers to splitting the column which contains numerical categorical data to many columns depending on the number of categories present in that column. Each column

contains "0" or "1" corresponding to which column it has been placed. This creates a binary column for each category and returns a sparse matrix or dense array

During encoding, we transform text data into numeric data. Encoding categorical data involves changing data that fall into categories to numeric data.

Data_1.csv dataset has two categorical variables. The Country variable and the Feedback variable. These two variables are categorical variables because they contain categories. The Country contains three categories- **India**, **USA** & **Brazil** and the Feedback variable contains two categories. **Yes** and **No** that's why they're called categorical variables. The following example converts these categorical variables into numerical.

LabelEncoder encodes labels by assigning them numbers. Thus, if the feature is country with values such as ['India', 'Brazil', 'USA'], LabelEncoder may encode color label as [0, 1, 2]. We will apply LabelEncoder to Country (column 0) and Feedback column (column 3).

To encode 'Country' column, use the syntax:

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['Country'] = labelencoder.fit_transform(df['Country'])
```

The output will appear as:

	Country	Age	Income	Feedback
0	1	49	86400	Good
1	0	32	57600	Excellent
2	2	35	64800	Good
3	0	43	73200	Fair
4	2	45		Excellent

As you can see, the country labels are replaced by numbers [0,1,2] where 'Brazil' is assigned 0, 'India' is 1 and 'USA' is 2. Similarly, we will encode the Feedback column.

```
labelencoder = LabelEncoder()
df['Feedback'] = labelencoder.fit_transform(df['Feedback'])
```

	Country	Age	Income	Feedback
0	1	49	86400	2
1	0	32	57600	0
2	2	35	64800	2
3	0	43	73200	1
4	2	45		0
5	1	40	69600	0
6	0		62400	3

We could have encoded both columns together in the following way:

```
cols = ['Country', 'Feedback']
df[cols] = df[cols].apply(LabelEncoder().fit_transform)
```

As you can see, the countries are assigned values 0, 1 and 2. This introduces an order among the values although there is no relationship or order between the countries. This may cause problems in processing. Feedback values are 0,1,2,3 which is ok because there is an order between the feedback labels 'Poor', 'Fair', 'Good' and 'Excellent'.

To avoid this problem, we use One hot encoding. Here, the country labels will be encoded in 3 bit binary values (Additional columns will be introduced in the dataframe).

```
from sklearn.preprocessing import OneHotEncoder
enc = OneHotEncoder(handle_unknown='ignore')
enc_df = pd.DataFrame(enc.fit_transform(df[['Country']]).toarray())
```

	0	1	2
0	0.0	1.0	0.0
1	1.0	0.0	0.0
2	0.0	0.0	1.0
3	1.0	0.0	0.0
4	0.0	0.0	1.0
5	0.0	1.0	0.0
6	1.0	0.0	0.0

After execution of above code the Country variable is encoded in 3 bit binary variable. The left most bit represents **Brazil**, 2nd bit represents **India** and the last bit represents **USA**.

The two dataframes can be merged into a single one

```
df = df.join(enc_df)
```

	Country	Age	Income	Feedback	0	1	2
0	India	49	86400	Good	0.0	1.0	0.0
1	Brazil	32	57600	Excellent	1.0	0.0	0.0
2	USA	35	64800	Good	0.0	0.0	1.0
3	Brazil	43	73200	Fair	1.0	0.0	0.0
4	USA	45		Excellent	0.0	0.0	1.0
5	India	40	69600	Excellent	0.0	1.0	0.0
6	Brazil		62400	Poor	1.0	0.0	0.0

The last three additional columns are the encoded country values. The 'Country' column can be dropped from the dataframe for further processing.

Data Discretization

- Data discretization is characterized as a method of translating attribute values of continuous data into a finite set of intervals with minimal information loss.
- **Discretization by Binning:** The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin.

#Program for Equal Width Binning

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
#Create a Dataframe
d={'item':['Shirt','Sweater','BodyWarmer','Baby_Napkin'],
  'price':[1250,1160,2842,1661]}
#print the Dataframe
df = pd.DataFrame(d)
print("\n ORIGINAL DATASET")
print(" -----")
print(df)
#Creating bins
m1=min(df["price"])
m2=max(df["price"])
bins=np.linspace(m1,m2,4)
names=["low", "medium", "high"]
df["price_bin"]=pd.cut(df["price"],bins,labels=names,include_lowest=True)
print("\n BINNED DATASET")
print(" -----")
print(df)
```

ORIGINAL DATASET

	Item	Price
0	Shirt	1250
1	Sweater	1160
2	BodyWarmer	2842
3	Baby_Napkin	1661

BINNED DATASET

	Item	Price	price_bin
0	Shirt	1250	low
1	Sweater	1160	low
2	BodyWarmer	2842	high
3	Baby_Napkin	1661	medium

1. **Equal Frequency Binning:** bins have equal frequency.
2. **Equal Width Binning:** bins have equal width with a range of each bin are defined as $[\min + w]$, $[\min + 2w]$ $[\min + nw]$
where, $w = (\max - \min) / (\text{no of bins})$.

Equal Frequency:

Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

Output:

[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]

Equal Width:

Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

Output:

[10, 11, 13, 15, 35, 50, 55, 72]
[92]
[204]

Set A

Create own dataset and do simple preprocessing

Dataset Name: Data.CSV (save following data in Excel and save it with .CSV extension)

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40		Yes
France	35	58000	Yes
Spain		52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

*Above dataset is also available at:

<https://github.com/suneet10/DataPreprocessing/blob/main/Data.csv>

Write a program in python to perform following task

1. Import Dataset and do the followings:
 - a) Describing the dataset
 - b) Shape of the dataset
 - c) Display first 3 rows from dataset
2. Handling Missing Value: a) Replace missing value of salary,age column with mean of that column.
3. Data.csv have two categorical column (the *country* column, and the *purchased* column).
 - a. Apply OneHot coding on *Country* column.
 - b. Apply Label encoding on *purchased* column

Set B

Import standard dataset and Use Transformation Techniques

Dataset Name: winequality-red.csv

Dataset Link: <http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv>

Write a program in python to perform following task

1. Import Dataset from above link.
2. Rescaling: Normalised the dataset using MinMaxScaler class
3. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1)

4. Normalizing Data (rescale each observation to a length of 1 (a unit norm). For this, use the Normalizer class.)
5. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0)

Set C

Import dataset and perform Discretization of Continuous Data

Dataset name: **Student_bucketing.csv**

Dataset link: https://github.com/TrainingByPackt/Data-Science-with-Python/blob/master/Chapter01/Data/Student_bucketing.csv

The dataset consists of student details such as **Student_id**, **Age**, **Grade**, **Employed**, and **marks**.

Write a program in python to perform following task

- 1) Write python code to import the required libraries and load the dataset into a pandas dataframe.
- 2) Display the first five rows of the dataframe.
- 3) Discretized the **marks** column into five discrete buckets, the labels need to be populated accordingly with five values: **Poor**, **Below_average**, **Average**, **Above_average**, and **Excellent**. Perform bucketing using the **cut ()** function on the **marks** column and display the top 10 columns.

Signature of the instructor

Date

Assignment Evaluation

0: Not Done 2: Late Complete 4: Complete
1: Incomplete 3: Needs Improvement 5: Well Done

Assignment 4: Data Visualization

Number of Slots = 2

Objectives

- To learn different statistical methods for Data visualization.
- To learn about packages matplotlib and seaborn.
- To learn functionalities and usages of Seaborn.
- Apply data visualization tools on various data sets.

Reading

You should read the following topics before starting this exercise:

Introduction to Exploratory Data Analysis, Data visualization and visual encoding, Data visualization libraries, Basic data visualization tools, Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, Donut charts, Specialized data visualization tools, Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots, Advanced data visualization tools- Wordclouds, Visualization of geospatial data, Data Visualization types

Ready Reference

Python provides the Data Scientists with various packages both for data processing and visualization. Data visualization is concerned with the presentation of data in a pictorial or graphical form. It is one of the most important tasks in data analysis, since it enables us to see analytical results, detect outliers, and make decisions for model building. There are various packages available in Python for visualization purpose. Some of them are: matplotlib, seaborn, bokeh, plotly, ggplot, pygal, gleam, leather etc.

Steps Involved in our Visualization

1. Importing packages
2. Importing and Cleaning Data
3. Creating Visualizations

Step-1: Importing Packages

Not only for Data Visualization, but every process to be held in Python should also be started by importing the required packages. Our primary packages include Pandas for Data processing, Matplotlib for visuals, Seaborn for advanced visuals, and Numpy for scientific calculations.

Step-2: Importing and Cleaning Data

This is an important step as a perfect data is an essential need for a perfect visualization.

We have successfully imported and cleaned our dataset. Now we are set to do our visualizations using our cleaned dataset.

Step-3: Creating Visualizations

In this step we create different types of Visualizations right from basic charts to advanced charts.

There are many Python libraries for visualization, of which matplotlib, seaborn, bokeh, and ggplot are among the most popular. However, in this assignment, we mainly focus on the matplotlib library. Matplotlib produces publication-quality figures in a variety of formats, and interactive environments across Python platforms. Another advantage is that Pandas comes equipped with useful wrappers around several matplotlib plotting routines, allowing for quick and handy plotting of Series and DataFrame objects. The matplotlib library supports many more plot types that are useful for data visualization. However, our goal is to provide the basic knowledge that will help you to understand and use the library for visualizing data in the most common situations.

Self Activity

Installing Matplotlib

There are multiple ways to install the matplotlib library. The easiest way to install matplotlib is to download the Anaconda package. Matplotlib is default installed with Anaconda package and does not require any additional steps.

- Download anaconda package from the official site of Anaconda
- To install matplotlib, go to anaconda prompt and run the following command

```
pip install matplotlib
```

- Verify whether the matplotlib is properly installed using the following command in Jupyter notebook

```
import matplotlib
matplotlib.__version__
3.2.2
```

How to use Matplotlib

Before using matplotlib, we need to import the package. This can be done using the 'import' method in Jupyter notebook. PyPlot is the graphical module in matplotlib which is mostly used for data visualisation, importing PyPlot is sufficient to work around data visualisation.

```
import matplotlib as mpl
#import the pyplot module from matplotlib as plt (short name used for referring the object)
```

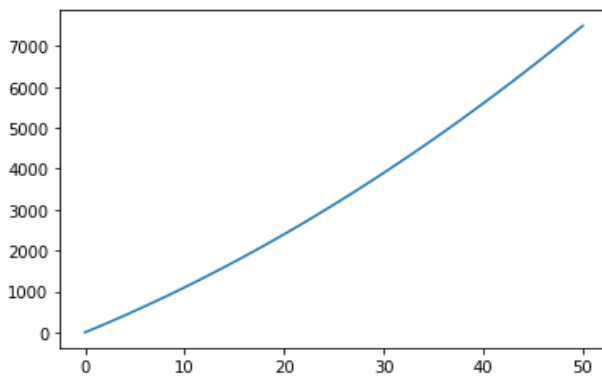


```
import matplotlib.pyplot as plt
```

Create a Simple Plot

Here we will be depicting a basic plot using some random numbers generated using NumPy. The simplest way to create a graph is using the 'plot()' method. To generate a basic plot, we need two axes (X) and (Y), and we will generate two random numbers using the 'linspace()' method from Numpy.

```
# import the NumPy package
import numpy as np
# generate random number using NumPy, generate two sets of random numbers
and store in x, y
x = np.linspace(0,50,100)
y = x * np.linspace(100,150,100)
# Create a basic plot
plt.plot(x,y)
[<matplotlib.lines.Line2D at 0x7f25569731d0>]
```



Adding Elements to Plot

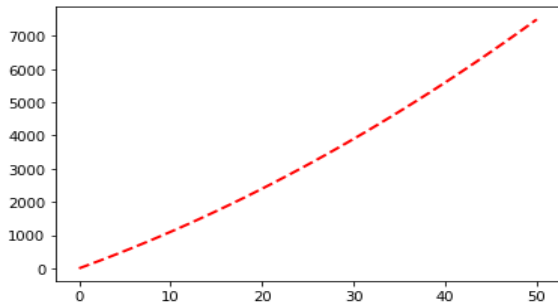
The plot generated above does not have all the elements to understand it better. Let's try to add different elements for the plot for better interpretation. The elements that could be added for the plot includes title, x-Label, y-label, x-limits, y-limits.

```
# set different elements to the plot generated above
# Add title using 'plt.title'
# Add x-label using 'plt.xlabel'
# Add y-label using 'plt.ylabel'
# set x-axis limits using 'plt.xlim'
# set y-axis limits using 'plt.ylim'
# Add legend using 'plt.legend'
```

Let's, add few more elements to the plot like colour, markers, line customisation.

```
# add color, style, width to line element
plt.plot(x, y, c = 'r', linestyle = '--', linewidth=2)
```

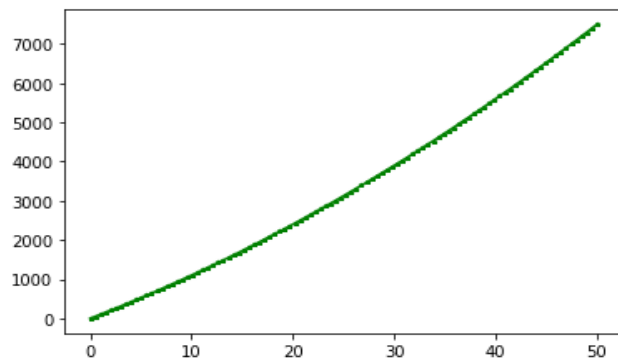
```
[<matplotlib.lines.Line2D at 0x7fdd01f12210>]
```



add markers to the plot, marker has different elements i.e., style, color, size etc.,

```
plt.plot(x, y, marker='*', c='g', markersize=3)
```

```
[<matplotlib.lines.Line2D at 0x7fdd01dacd10>]
```



add grid using grid() method

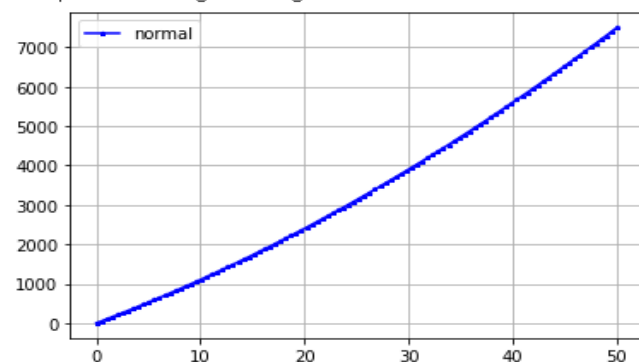
```
plt.plot(x, y, marker='*', markersize=3, c='b', label='normal')
```

```
plt.grid(True)
```

```
# add legend and label
```

```
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fdd01c345d0>
```



Plots could be customized at three levels:

- Colours

- b – blue
- c – cyan

- g – green
- k – black

- m – magenta
- r – red
- w – white
- **Line Styles**
 - ‘-‘ : solid line
 - ‘- -‘: dotted line
- **Marker Styles**
 - . – point marker
 - , – Pixel marker
 - v – Triangle down marker
 - ^ – Triangle up marker
 - < – Triangle left marker
 - > – Triangle right marker
 - 1 – Tripod down marker
- **Other configurations**
 - color or c
 - linestyle
 - linewidth
 - marker
- y – yellow
- Can use Hexadecimal, RGB formats
- ‘- .’: dash-dot line
- ‘:’ – dotted line
- 2 – Tripod up marker
- 3 – Tripod left marker
- 4 – Tripod right marker
- s – Square marker
- p – Pentagon marker
- * – Star marker
- markeredgewidth
- markeredgecolor
- markerfacecolor
- markersize

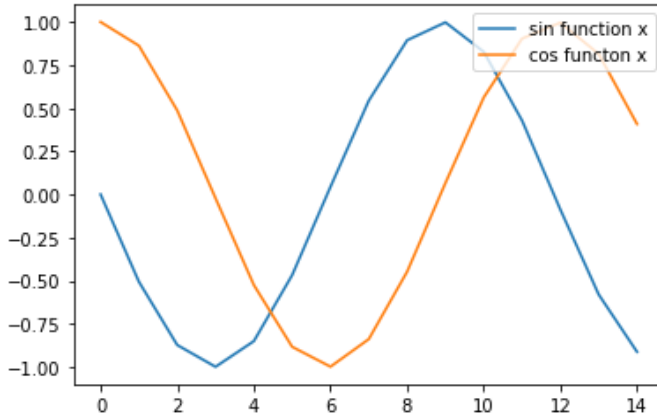
Making Multiple Plots in One Figure

There could be some situations where the user may have to show multiple plots in a single figure for comparison purpose. For example, a retailer wants to know the sales trend of two stores for the last 12 months and he would like to see the trend of the two stores in the same figure.

Let’s plot two lines $\sin(x)$ and $\cos(x)$ in a single figure and add legend to understand which line is what.

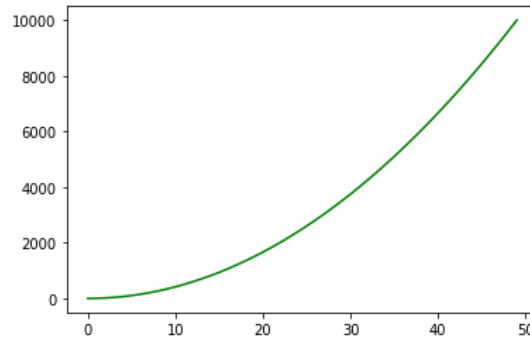
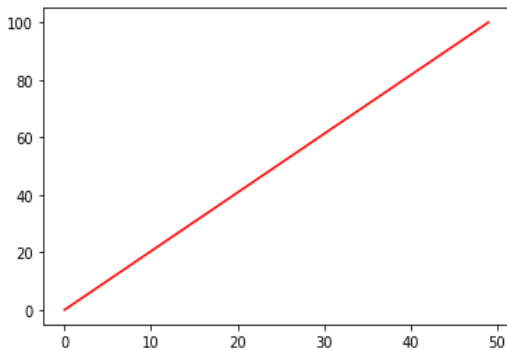
```
# lets plot two lines Sin(x) and Cos(x)
# loc is used to set the location of the legend on the plot
# label is used to represent the label for the line in the legend
# generate the random number
x= np.arange(0,1500,100)
plt.plot(np.sin(x),label='sin function x')
plt.plot(np.cos(x),label='cos functon x')
plt.legend(loc='upper right')
```

```
<matplotlib.legend.Legend at 0x7fdd01bb2f90>
```



To show the multiple plots in separate figure instead of a single figure, use `plt.show()` statement before the next plot statement as shown below

```
x= np.linspace(0,100,50)
plt.plot(x, 'r', label='simple x')
plt.show()
plt.plot(x*x, 'g', label='two times x')
plt.show()
plt.legend(loc='upper right')
```



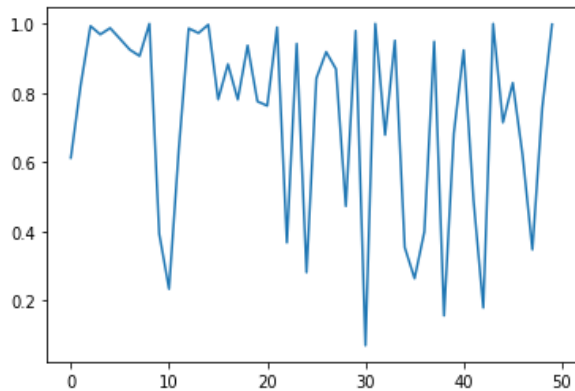
Create Subplots

There could be some situations where we should show multiple plots in a single figure to show the complete storyline while presenting to stakeholders. This can be achieved with the use of subplot in matplotlib library. For example, a retail store has 6 stores and the manager would like to see the daily sales of all the 6 stores in a single window to compare. This can be visualised using subplots by representing the charts in rows and columns.

```
# subplots are used to create multiple plots in a single figure
# let's create a single subplot first following by adding more subplots
x = np.random.rand(50)
y = np.sin(x*2)
```

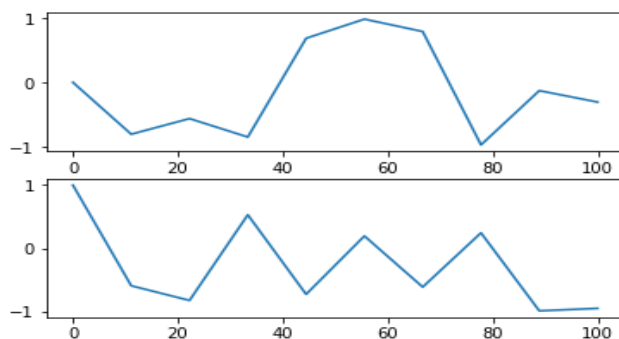
```
#need to create an empty figure with an axis as below, figure
and axis are two separate objects in matplotlib
fig, ax = plt.subplots()
#add the charts to the plot
ax.plot(y)
```

```
[<matplotlib.lines.Line2D at 0x7fdd01971dd0>]
```



```
# Let's add multiple plots using subplots() function
# Give the required number of plots as an argument in
subplots(), below function creates 2 subplots
fig, axs = plt.subplots(2)
#create data
x=np.linspace(0,100,10)
# assign the data to the plot using axs
axs[0].plot(x, np.sin(x**2))
axs[1].plot(x, np.cos(x**2))
# add a title to the subplot figure
fig.suptitle('Vertically stacked subplots')
Text(0.5, 0.98, 'Vertically stacked subplots')
```

Vertically stacked subplots

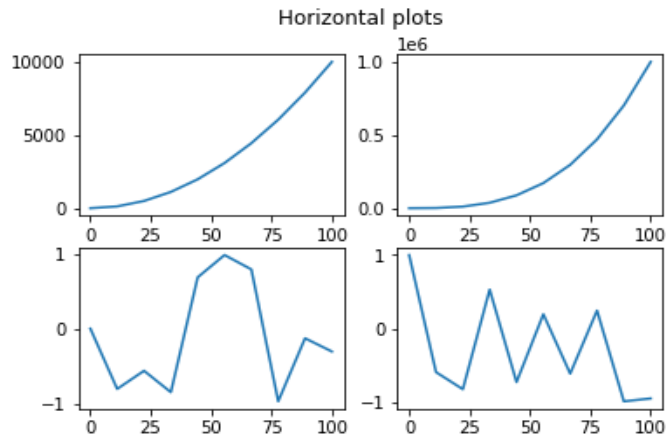


```
# Create horizontal subplots
# Give two arguments rows and columns in the subplot() function
# subplot() gives two dimensional array with 2*2 matrix
# need to provide ax also similar 2*2 matrix as below
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)
```

```

# add the data to the plots
ax1.plot(x, x**2)
ax2.plot(x, x**3)
ax3.plot(x, np.sin(x**2))
ax4.plot(x, np.cos(x**2))
# add title
fig.suptitle('Horizontal plots')
Text(0.5, 0.98, 'Horizontal plots')

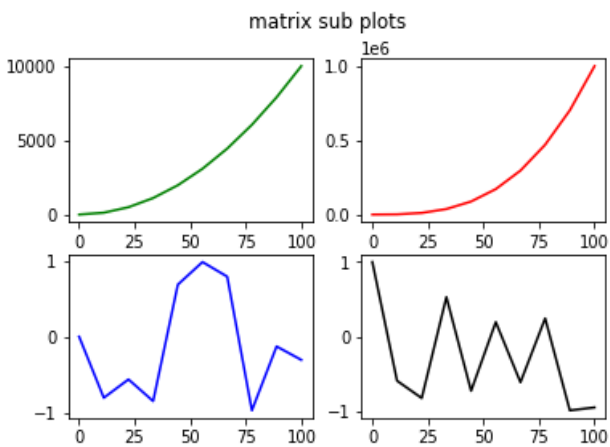
```



```

# another simple way of creating multiple subplots as below,
using axs
fig, axs = plt.subplots(2, 2)
# add the data referring to row and column
axs[0,0].plot(x, x**2, 'g')
axs[0,1].plot(x, x**3, 'r')
axs[1,0].plot(x, np.sin(x**2), 'b')
axs[1,1].plot(x, np.cos(x**2), 'k')
# add title
fig.suptitle('matrix sub plots')
Text(0.5, 0.98, 'matrix sub plots')

```



```

# let's create a figure object
# change the size of the figure is 'figsize = (a,b)' a is width
and 'b' is height in inches
# create a figure object and name it as fig
fig = plt.figure(figsize=(4,3))
# create a sample data
X = np.array([1,2,3,4,5,6,8,9,10])
Y = X**2
# plot the figure
plt.plot(X,Y)

```

[<matplotlib.lines.Line2D at 0x7fdd016ef390>]

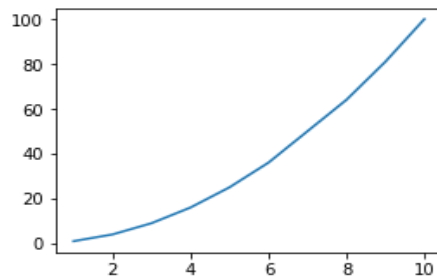


Figure Object

Matplotlib is an object-oriented library and has objects, calluses and methods. Figure is also one of the classes from the object 'figure'. The object figure is a container for showing the plots and is instantiated by calling figure() function.

'plt.figure()' is used to create the empty figure object in matplotlib. Figure has the following additional parameters.

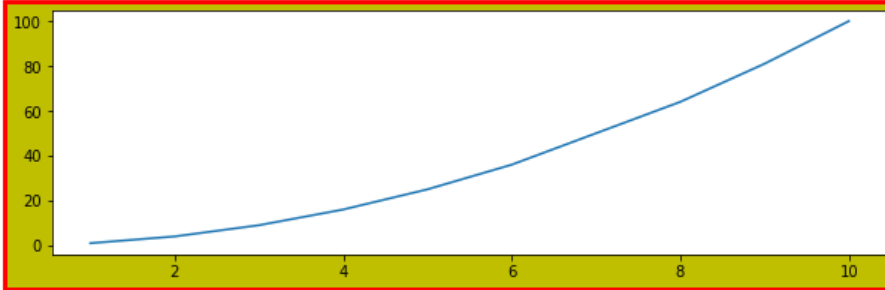
- Figsize – (width, height) in inches
- Dpi – used for dots per inch (this can be adjusted for print quality)
- facecolor
- edgecolor
- linewidth

```

# let's change the figure size and also add additional
parameters like facecolor, edgecolor, linewidth
fig =
plt.figure(figsize=(10,3),facecolor='y',edgecolor='r',linewidth
=5)
# create a sample data
X = np.array([1,2,3,4,5,6,8,9,10])
Y = X**2
# plot the figure
plt.plot(X,Y)

```

```
[<matplotlib.lines.Line2D at 0x7fdd01662390>]
```



Axes Object

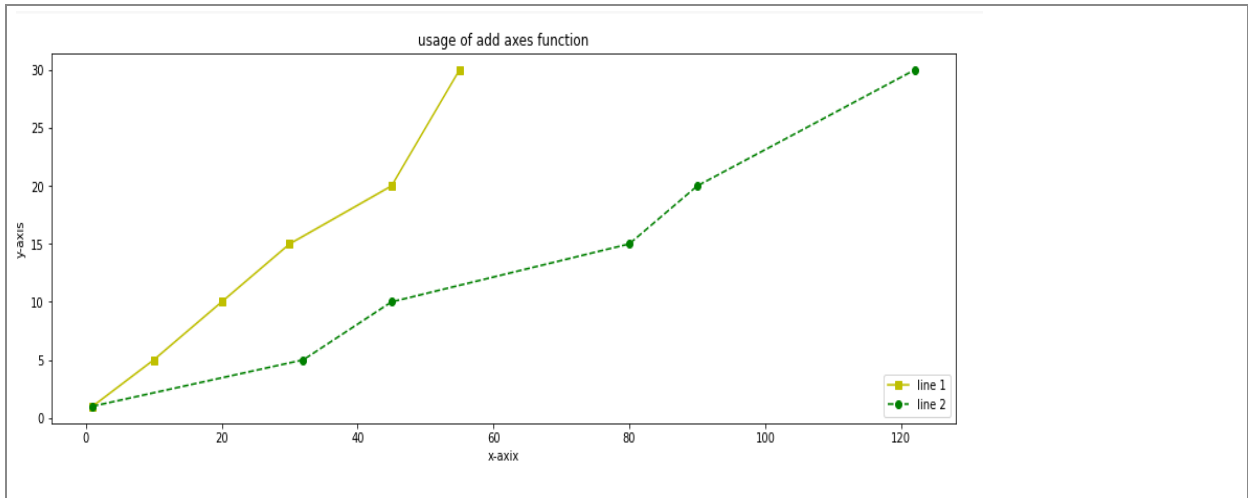
Axes is the region of the chart with data, we can add the axes to the figure using the 'add_axes()' method. This method requires the following four parameters i.e., left, bottom, width, and height

- Left – position of axes from left of figure
- bottom – position of axes from the bottom of figureList item
- width – width of the chart
- height – height of the chartList item

Other parameters that can be used for the axes object are:

- Set title using 'ax.set_title()'
- Set x-label using 'ax.set_xlabel()'
- Set y-label using 'ax.set_ylabel()'

```
# lets add axes using add_axes() method
# create a sample data
y = [1, 5, 10, 15, 20, 30]
x1 = [1, 10, 20, 30, 45, 55]
x2 = [1, 32, 45, 80, 90, 122]
# create the figure
fig = plt.figure()
# add the axes
ax = fig.add_axes([0,0,2,1])
l1 = ax.plot(x1,y,'ys-')
l2 = ax.plot(x2,y,'go--')
# add additional parameters
ax.legend(labels = ('line 1', 'line 2'), loc = 'lower right')
ax.set_title("usage of add axes function")
ax.set_xlabel('x-axis')
ax.set_ylabel('y-axis')
plt.show()
```

Different Types of Matplotlib Plots

Matplotlib has a wide variety of plot formats, few of them include bar chart, line chart, pie chart, scatter chart, bubble chart, waterfall chart, circular area chart, stacked bar chart etc.,

Bar Graph or Chart

Bar graph represents the data using bars either in Horizontal or Vertical directions. Bar graphs are used to show two or more values and typically the x-axis should be categorical data. The length of the bar is proportional to the counts of the categorical variable on x-axis.

Function:

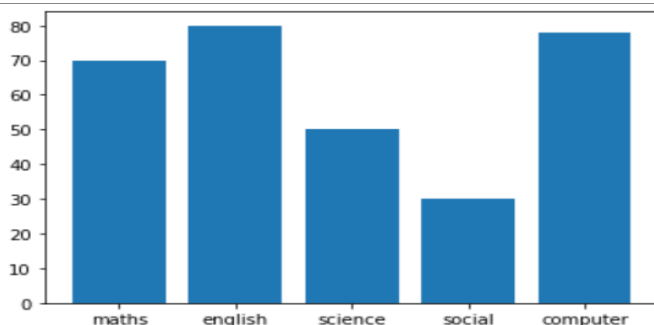
- The function used to show bar graph is 'plt.bar()'
- The bar() function expects two lists of values one on x-coordinate and another on y-coordinate

Customizations:

plt.bar() function has the following specific arguments that can be used for configuring the plot.

- Width, Color, edge colour, line width, tick_label, align, bottom,
- Error Bars – xerr, yerr

```
# lets create a simple bar chart
#x-axis is shows the subject and y -axis shows the markers in
each subject
subject = ['maths','english','science','social','computer']
marks =[70,80,50,30,78]
plt.bar(subject,marks)
plt.show()
```

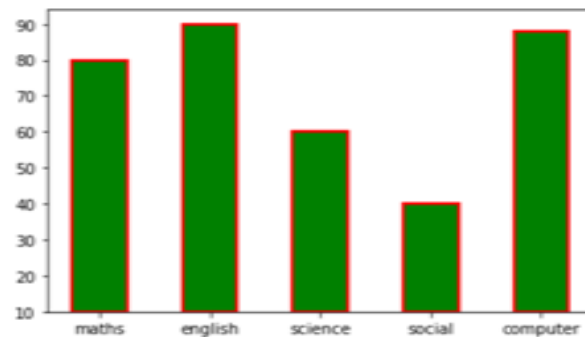


```

#let's do some customizations
#width - shows the bar width and default value is 0.8
#color - shows the bar color
#bottom - value from where the y - axis starts in the chart
i.e., the lowest value on y-axis shown
#align -move the position of x-label, has two options 'edge' or
'center'
#edgecolor - used to color the borders of the bar
#linewidth - used to adjust the width of the line around the
bar
#tick_label - to set the customized labels for the x-axis
plt.bar(subject,marks,color='g',width=0.5,bottom=10,align='cent
er',edgecolor='r',linewidth=2,
tick_label=subject)

```

<BarContainer object of 5 artists>



```

# errors bars could be added to represent the error values referring to an array value

```

```

# here in this example we used standard deviation to show as error bars

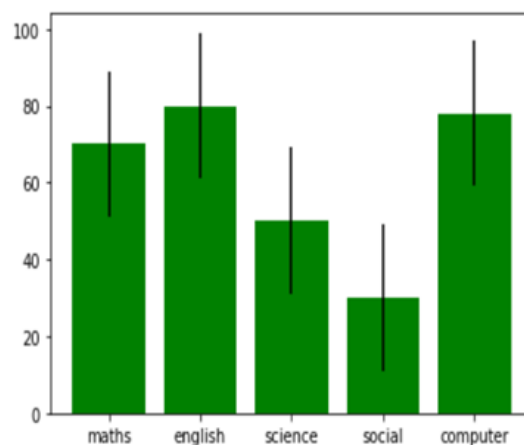
```

```

plt.bar(subject,marks,color='g',yerr=np.std(marks))

```

<BarContainer object of 5 artists>

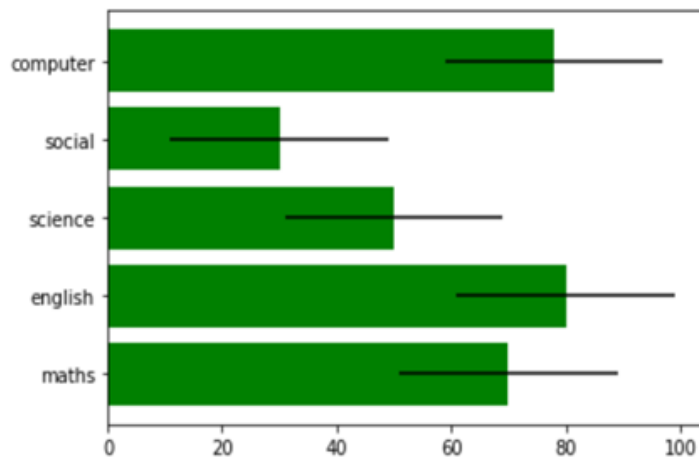


```

# to plot horizontal bar plot use plt.barh() function
plt.barh(subject,marks,color='g',xerr=np.std(marks))

```

<BarContainer object of 5 artists>



Pie Chart:

Pie charts display the proportion of each value against the total sum of values. This chart requires a single series to display. The values on the pie chart show the percentage contribution in terms of a pie called **Wedge/Widget**. The angle of the wedge/widget is calculated based on the proportion of values.

Function:

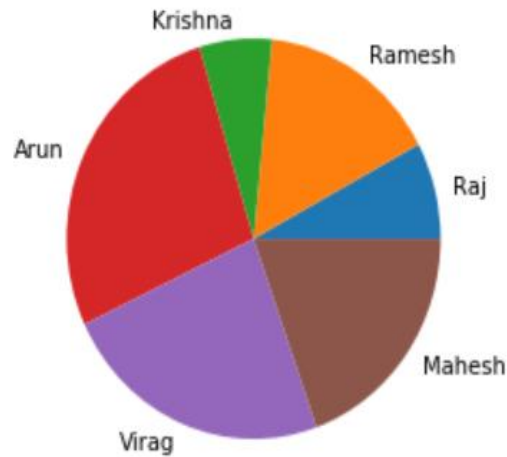
- The function used for pie chart is 'plt.pie()'
- To draw a pie chart, we need only one list of values, each wedge is calculated as proportion converted into angle.

Customizations:

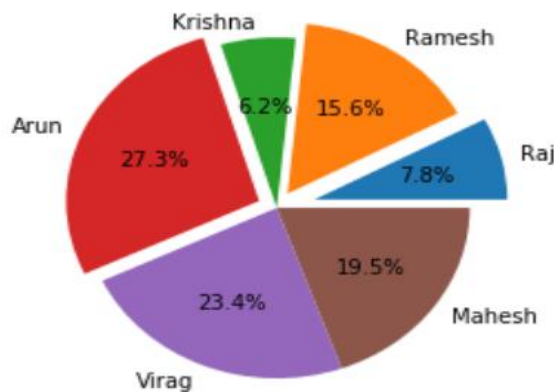
plt.pie() function has the following specific arguments that can be used for configuring the plot.

- labels – used to show the widget categories
- explode – used to pull out the widget/wedge slice
- autopct – used to show the % of contributions for the widgets
- Set_aspect – used to
- shadow – to show the shadow for a slice
- colours – to set the custom colours for the wedges
- startangle – to set the angles of the wedges

```
# Let's create a simple pie plot
Tickets_Closed = [10, 20, 8, 35, 30, 25]
Agents = ['Raj', 'Ramesh', 'Krishna', 'Arun', 'Virag', 'Mahesh']
plt.pie(Tickets_Closed, labels = Agents)
```



```
#Let's add additional parameters to pie plot
#explode - to move one of the wedges of the plot
#autopct - to add the contribution %
explode = [0.2,0.1,0,0.1,0,0]
plt.pie(Tickets_Closed,labels=Agents,          explode=explode,
autopct='%1.1f%%' )
```



Scatter Plot

Scatterplot is used to visualise the relationship between two columns/series of data. The chart needs two variables, one variable shows X-position and the second variable shows Y-position. Scatterplot helps in understanding the following information across the two columns

- Any relationship exists between the two columns
- + ve Relationship
- Or -Ve relationship

Function:

- The function used for the scatter plot is 'plt.scatter()'

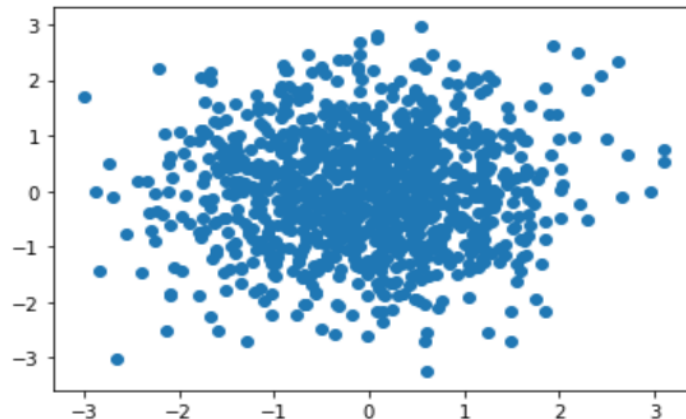
Customizations:

plt.scatter() function has the following specific arguments that can be used for configuring the plot.

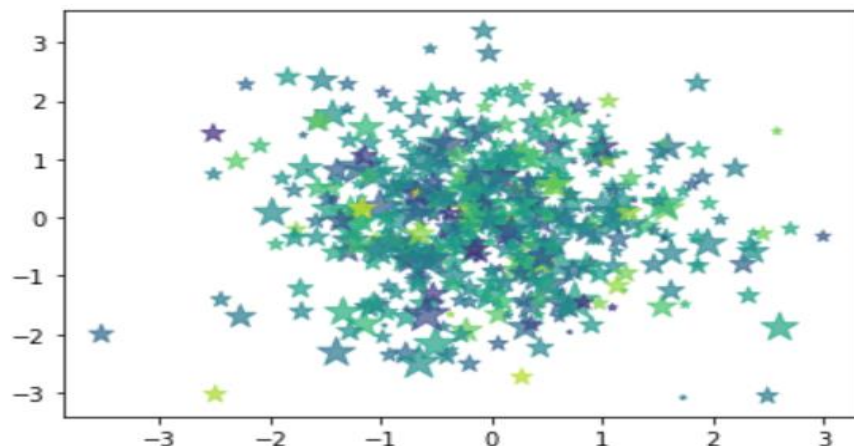
- size – to manage the size of the points
- color – to set the color of the points

- marker – type of marker
- alpha – transparency of point
- norm – to normalize the data (scaling between 0 to 1)

```
# generate the data with random numbers
x = np.random.randn(1000)
y = np.random.randn(1000)
plt.scatter(x, y)
```



```
# as you observe there is no correlation exists between x and y
# let's try to add additional parameters
# size - to manage the size of the points
#color - to set the color of the points
#marker - type of marker
#alpha - transparency of point
size = 150*np.random.randn(1000)
colors = 100*np.random.randn(1000)
plt.scatter(x, y, s=size, c = colors, marker = '*', alpha=0.7)
```



Histogram

Histogram is used to understand the distribution of the data. It is an estimate of the probability distribution of continuous data. It is similar to bar graph as discussed above but this is used to represent the distribution of a continuous variable whereas bar graph is used for discrete variable. Every distribution is characterised by four different elements including

- Center of the distribution

- Spread of the distribution
- Shape of the distribution
- Peak of the distribution

Histogram requires two elements x-axis shown using bins and y-axis shown with the frequency of the values in each of the bins form the data set. Every bin has a range with minimum and maximum values.

Function:

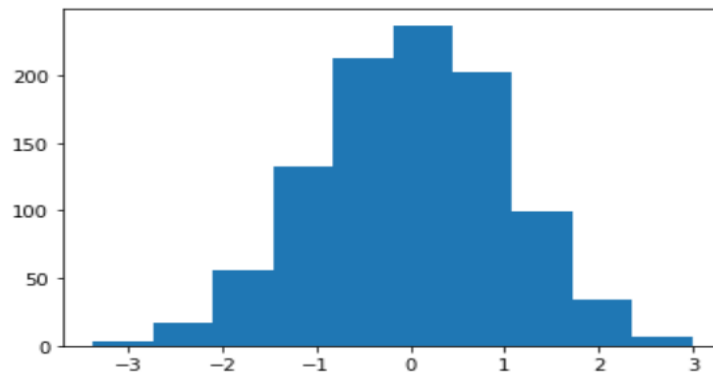
- The function used for scatter plot is 'plt.hist()'

Customizations:

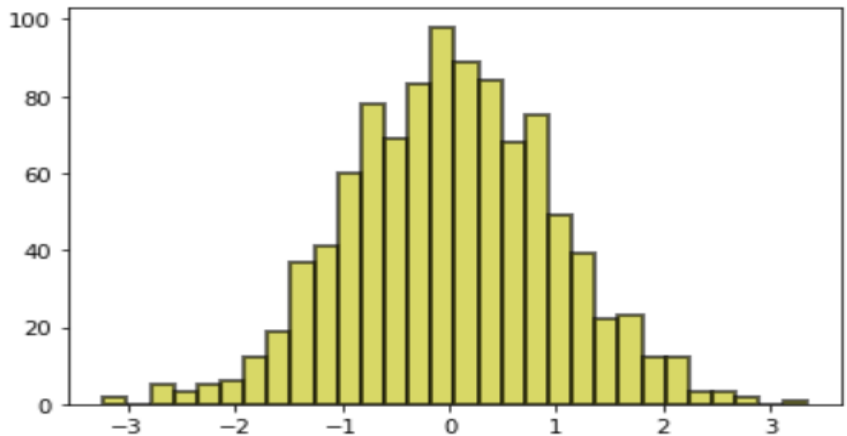
plt.hist() function has the following specific arguments that can be used for configuring the plot.

- bins – number of bins
- color
- edgecolor
- alpha – transparency of the color
- normed
- xlim – to set the x-limits
- ylim – to set the y-limits
- xticks, yticks
- facecolor, edgecolor, density

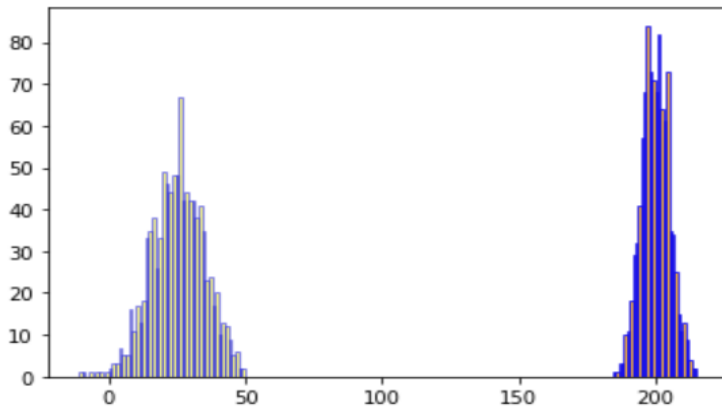
```
# let's generate random numbers and use the random numbers to
generate histogram
data = np.random.randn(1000)
plt.hist(data)
```



```
# let's add additional parameters
# facecolor
# alpha
# edgecolor
# bins
data = np.random.randn(1000)
plt.hist(data, facecolor='y', linewidth=2, edgecolor='k',
bins=30, alpha=0.6)
```



```
# lets create multiple histograms in a single plot
# Create random data
hist1 = np.random.normal(25,10,1000)
hist2 = np.random.normal(200,5,1000)
#plot the histogram
plt.hist(hist1,facecolor = 'yellow',alpha = 0.5, edgecolor
='b',bins=50)
plt.hist(hist2,facecolor = 'orange',alpha = 0.8, edgecolor
='b',bins=30)
```



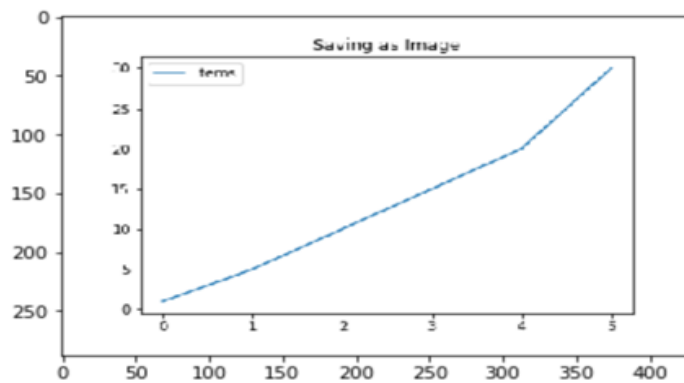
Saving Plot

Saving plot as an image using 'savefig()' function in matplotlib. The plot can be saved in multiple formats like .png, .jpeg, .pdf and many other supporting formats.

```
# let's create a figure and save it as image
items = [5,10,20,25,30,40]
x = np.arange(6)
fig = plt.figure()
ax = plt.subplot(111)
ax.plot(x, y, label='items')
plt.title('Saving as Image')
ax.legend()
fig.savefig('saveimage.png')
```

Image is saved with a filename as 'saveimage.png'.

```
#To display the image again, use the following package and
commands
import matplotlib.image as mpimg
image = mpimg.imread("saveimage.png")
plt.imshow(image)
plt.show()
```



Box Plot

It displays the distribution of data based on the five-number theory by dividing the dataset into three quartiles and then presents the values - minimum, maximum, median, first (lower) quartile and third (upper) quartile- in the plotted graph itself. It helps in identifying outliers and how much spread out data is from the center.

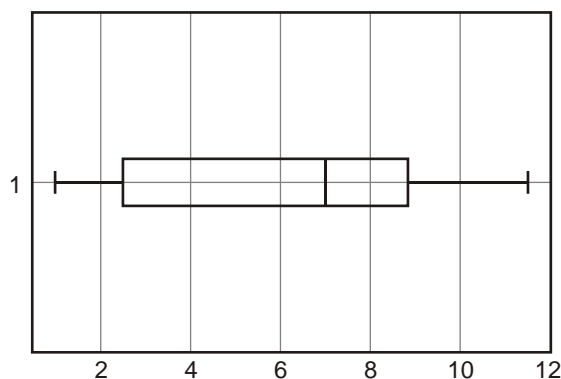
In Python, a boxplot can be created using the `boxplot()` function of the `matplotlib` library. Consider the following data elements: 1, 1, 2, 2, 4, 6, 6.8, 7.2, 8, 8.3, 9, 10, 10, 11.5

Here, min is 1, max is 11.5, the lower quartile is 2, the median is 7, and the upper quartile is 9. If we plot the box plot, the code would be:

#Let us create a Simple boxplot

```
import matplotlib.pyplot as plt
data=[1, 1, 2, 2, 4, 6, 6.8, 7.2, 8, 8.3, 9, 10, 10, 11.5]
plt.boxplot(data, vert=False)
plt.show()
```

The box plot appears as:



Lab Assignments

Student must use Iris flower data set for Lab Assignments

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper



Iris setosa



Iris versicolor



Iris virginica

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

Set A

1. Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.
2. Add two outliers to the above data and display the box plot.
3. Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart and bar chart.
4. Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data.
5. Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data.
6. Write a Python program to create a histogram of the three species of the Iris data.

Set B

1. Write a Python program to create a graph to find relationship between the petal length and petal width.
2. Write a Python program to draw scatter plots to compare two features of the iris dataset.

3. Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species.

Set C

1. Write a Python program to create a pairplot of the iris data set and check which flower species seems to be the most separable.
2. Write a Python program to generate a box plot to show the Interquartile range and outliers for the three species for each feature.
3. Write a Python program to create a joint plot using "kde" to describe individual distributions on the same plot between Sepal length and Sepal width. **Note:** The kernel density estimation (kde) procedure visualizes a bivariate distribution. In seaborn, this kind of plot is shown with a contour plot and is available as a style in joint plot().

Signature of the instructor

Date

Assignment Evaluation

0 Not done	<input type="text"/>	1 Incomplete	<input type="text"/>	2 Late Complete	<input type="text"/>
3 Needs Improvement	<input type="text"/>	4 Complete	<input type="text"/>	5 Well Done	<input type="text"/>