HTML:

HTML stands for Hyper Text Markup Language <b>batch-31</b>

HTML is a language for creating static web pages <html></html>🡪paired tags <br>🡪unpaired tags/empty tag

It is a markup language.

< > tag <p> jjj oooo ssss ffff eee uuuu</p>

HTML tags normally come in pairs like <p> and </p>

<html> 🡪 This is first tag in a pair is the open or start tag.

This is a html program

</html> 🡪 this is second and end tag are also called the closing tag.

To create static **web pages.**

<html> open and close

</html> paired tags

<br> unpaired tags

Head tag: placed at the top of document immediately after the <html> tag information about the document eg: author , style etc.,

Contains the required document <title> </title> tag.

Text editor:notepad,notepad++, editplus,atom ,sublime,vs code.....

Save :filename.html s1.html

Filename:start with alphabet,it can be end with number

**Notes:**

Html is a language for describing web pages.HTML stands for Hyper text markup language .Html is not a programming language.it is a markup language.

**Structure of HTML**

<html>

<head>

Head section

</head>

<body bgcolor=”” text=””>

Body section

</body>

</html>

Most html elements can have attributes: Attribute values should always be enclosed in quotes.

<p align=”center”> the </p>

.comments can be inserted into the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed.

Comments are written like this:<!-- this is a comment -->

Formatting tags:

Logical tags and physical tags

<h1>physical html</h1>

<h2> heading</h2>

<b> hhhh</b>

<i>lllkkkkk</i>

<tt>teletype</tt><br>

<big> larger size</big><br>

<small> smaller size</small><br>

s<sub> 2</sub><br>

g<sup>5</sup><br>

<u> underline</u><br>

<strike>strike through text</strike>

4-6-876

LB nagar

hyderabad

\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

<pre> 4-6-876

LB nagar

hyderabad

\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

</pre>

<h2>logical html</h2>

<address> \*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

</address>

<cite> citation or reference to another source</cite><br>

<code> code text</code><br>

<DFN> defining instance of the enclosed term</DFN><br>

<em> emphasized text</em><br>

<kbd> text the user should user</kbd><br>

<samp> sample output</samp><br>

<strong> strongly emphasized text</strong><br>

<var> variable</var><br>

<abbr> abbreviation</abbr><br>

<acronym> Acronym</acronym><br>

Meta tag:

It is used to specify keywords that describe a document’s contents as well as a short description.

Two necessary attributes: name and content

<meta name=”keywords” content=”baseball ,”/>

<meta name=”description” content=”author or sports information page”/>

Body tag attributes: marginwidth,marginheight,leftmargin,rightmargin,topmargin.

<body marginwidth=””>

</body>

Given width of the left right top bottom margin are fix.

Bgcolor=”” it is display background color.

Text=”red” its display text color on the web page.

Link=”” alink=”” vlink=””

Image tag:

The <img> tag is used to insert an image into an html document. The image itself isn’t inserted directly into the document , the browser inserts an html image from the source specified in the <img> tag.

<img src=”path” width= “ “ height= “ “ alt=” “ hspace=” “ border=” “ vspace=” “>

Src: (source )The src attribute shows the image source. It is required as it defines the path to the image.

Alt: the alt attribute defines an alternate name for the image.

Apng:Animated probtable network graphics or png

Bmp:Bitmap file

Gif:graphic interchange format

Jpg or jpeg joint photographic expert group image

Svg: scalable vector graphic

Tiff: tagged image file format

webP webpicture format

Marquee:display scrolling text on the web page.

<marquee> online classes</marquee>

Align=”top” or “middle” or “bottom”

Bgcolor=”red”

Behaviour=”slide” or “alternate” or “scroll” or “bounce”

Direction =”left” or “right”

Height=

Hspace

Vspace

Width=

List: There are three of list,

List is collection of items.

1)Ordered list 2)Unordered list 3)Definition list

Ordered List:

Ordered lists use a numbering or lettering system to indicate that the items are sequenced way.

1.Ajay a.apple I.xxxx i.gggg A.nnn

3.srikanth c.orange II.yyyy ii.dddd B.ffff

Syntax:

<ol type=”i” start=”4”>

<li>fruits</li> d.fruits

<li>vegetables</li>e.vegetables

<li> papaya</li> f.papaya

</ol>

It is similar to ordered list but symbol are use to indicate the items.

Unordered List:

Syntax:

<ul type=””> square or circle or disc

<li> a</li>

<li> b </li>

</ul>

Definition list or description list:

In definition list , we can description the items with the help of <dd> tag.

Syntax:

<dl>(definition list)

<dt>(definition item)

<dd></dd>(data definition or data description)

</dt>

<dt><dd></dd></dt>

</dl>

Hyperlink or An Anchor tag:A link or hyperlink is a connection from one (source file)web page to another (source file)web page

To create a link we use hyperlink

<a href=”path“> text click </a>

<a name=”#top”> this is name as top</a>

<a href=”bottom”>it moves to bottom</a>

<a name=”#bottom”>this is named as bottome</a>

<a href=”top”>this is move to top</a>

To open default mail program create new message with the to field “

<a href=”<mailto:priya@yahoo.com>,nirmaanorganzation@gmail.com”> click me</a>

To add subject use a question mark(?) and write subject .

<a href=<mailto:priya@yahoo.com?subject=memo>>sri kanth</a>

<a href=<mailto:priya@yahoo.com?cc=memo>marks@gmail.com>sri kanth</a>

<a href=mailto:priya@yahoo.com?cc=someodther@yahoo.com&subject=list&body=haihowru">sri kanth</a>

Table:A table is an arrangement of data in rows and columns.HTML to create table table tag (<table> )is used . In the table tag we use <tr> tag to create row in row <td> is used to enter data or create a column.

Collection of rows cols .

|  |  |  |  |
| --- | --- | --- | --- |
| Srikanth | Web application | Ifirst | hyd |
| Sai | B.tech |  | rr |
|  |  |  |  |

Syntax:

<table border=”3”>

<caption> student details </caption>

<tr >

<td >a</td>

<td>d</td>

<td></td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

</tr>

</table>

|  |  |  |
| --- | --- | --- |
|  | D |  |
|  | | |
|  | | |
|  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time/days | 9:10 |  |  | Break |  |  |
| Monday |  |  |  |  |  |
| Tuesday |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Div tag:Formatting block text

The<Div> tag provides you with an easy way to refer to a block of text. This element is very important when applying styles because you can mark specific blocks of text to apply styles.

Span tag: This <span> tag is use to apply styles inline.

HR : tag To create the Horizontal line <hr> tag is used.

<hr> tag is use to create a horizontal line to separate or group elements vertically.

Align left, right , center

Noshade bar without shading effect.

Size: height of the horizontal rule.(pixels)

Width pixels .width of the horizontal rule.

Frames:

HTML frames are used to divide our browser window into multiple sections where each section can load a separate html documents.

A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized into rows and colums.

Html frames allow authors to present documents in multiple views which may be independent windows or subwindows.

An html document that describes frame layout (called a frameset document) has a different make up than an html document without frames.

Attributes: rows or cols this is define the number of horizontal or vertical subspaces in a frameset

Syntax:

<frameset rows=” ” or cols=” “>

<frame src=””>

<frame src=””>

</frameset>

Attributes:

Frame name

When you give a frame a name, you can use it as a target to load new pages into it.

Frameset :border bordercolor ,frameborder framespacing,cols ,rows

Iframe:

An inline frame with html tag <iframe>. This tag is not some how < frameset> tag, it can appear anywhere in your document. The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current html document.

The src attribute is used to specify the URL of the document that occupies the inline frame.

Syntax:

<iframe>

</iframe>

**FORM:**

An html form is used to collect user input .The user input is mostly sent to a server for processing.

The html <form> element is used to create an html form for user input.

The <form> element is a container for different types of input elements, such as text fields checkboxes, radio buttons , submit and reset etc,.

|  |
| --- |
| Gg |

Name:

Gender:

* female
* Male
* Other

Hobbies:

* dance
* music

chess

tv

**syntax:**

<form name=”form1” method=” get or post” action=” ”>

</form>

Action:The URL that processes the form submission.

Method: The HTTP method to submit the form with.

Post: the POST method form data sent as the request body.

Get:The GET method from data appended to the action URL with a ? separator.

Input:

|  |
| --- |
| Priya |

Name:

<input type=”text” name=”n1” id=”nn” size=”25”>

F name:

Label: Html label acts as a caption for a specified element.

<label> Name of the student</label><input type=”text”>

<fieldset> groups elements in an html form.

Audio: The html <audio> element is used to play an audio file on the web page.

The controls attribute adds audio controls,like play,pause,and volume.

The source element allows us to specify alternative audio files which the browser may choose from.

<audio controls src=””>

<source src=”” type=”audio/mpeg”>

</audio>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
| Aaaa | Msc | Female/male | C | C++ |

C++

CSS:

Cascading style sheet:cascading style sheets are used to format the layout of web pages. They can be used to define text styles, table sizes and other styles.Cascading style sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. It sets the background color,font-size,font-family, color,...etc property of elements on a web page.

There are three ways to implement CSS :

1) Internal

2) External and

3) Inline styles.

1) External CSS

With an external style sheet, we can change the look of an entire website by changing just one file.

Each Html page must include a reference to the external style sheet file inside the <link> element, inside the head section.

<!doctype html>

<html>

<head>

<link rel=”stylesheet” href=””>

</head>

<body>

<h1> this is a h1 heading</h1>

<p> paragraph </p>

</body>

</html>

Body{

Background-color:lightblue; property:value;

}

H1{

Color:navy;

Margin-left:20px;

}

Internal CSS or embedded style sheet

An internal style sheet may be used if one single Html page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

Internal styles are defined within the <style> element, inside the <head> section of an html page:

<!doctype html>

<html>

<head>

<style>

Body{

Background-color:linen;

}

H1{

Color:maroon;

Margin-left:40px;

}

</style>

</head>

<body>

<h1>

The heading h1</h1>

<p>this is a paragraph</p>

</body>

</html>

**Inline:**

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any css property.

Inline styles are defined within the “style” attribute of the relevant element.

<!doctype html>

<html>

<body>

<h1 style=”color:blue;text-align:center;”> this is a heading</h1>

</body>

</html>

Selectors: CSS selectors are used to find or select the html elements you want to apply the style.

The ID selector selects the id attribute of an html element to select a specific element. An Id is always unique within the page so it is chosen to select a single ,unique element. It is written with the hash character(#),followed by the id of the element.

#b{

Color:red;

}

The class selector selects html elements with a specific class attribute. It is used with a period character.(.full stop symbol) followed by the class name.

.sss{

Font-size:24px;

}

**Class selector for specific element.**:

If you want to specify that only one specific html element should be affected then you should use the element name with class selector.

p.cen1{

Color:yellow;

}

**Universal selector**:

The Universal selector is used as a wildcard character . it selects all the elements on the pages.

\*{

color:red;

font-size:30px;

}

**Group selector:**

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code commas are used to separate each selector in grouping.

H1{

Color:red;

}

H1,h2,p{

Color:red;

Text-align:center;

Font-size:30px;

}

Font:

Font-family

Font-size

Font-weight

Font-style

Font-stretch

Font-variant

Text:

Color

Text-align

Text-decoration

Text-indent

Text-shadow

Text-transform

Text-spacing

Word-spacing

Letter-spacing

White-spacing

Direction

Vertical align

**Box:** The layout of a web document. There are two way to create box . 1)Block level box and 2) inline box.

Block level box allows <div> , paragraph or heading to utilize the horizontal space available at the beginning of the line.

Inline box: An inline box allows <span> or image to utilize the space available before and after inline elements.

Box model: margin, border, padding, content

Margin: it is

Border:

|  |
| --- |
|  |

Border-style:solid;

Border-width:thin;

Border-color:red;

Padding:

Padding-top:100px;

Padding-right:30px;

Padding-left:40px;

Padding-bottom:45px;

|  |
| --- |
| Gggggggg  Jjjjjjjjjj  Mmmmm |

**Display property:**

This property is used to specify the display behaviour of an element.(ie block,inline, none)

List style position property:

The list style position property specifies the bullet points that have multiple lines of text by aligning the second line according to the beginning of the first line

1. inside : The second line is aligned underneath the start of the marker
2. outside:The second line is aligned with the start of the first line.

Ul{list-style-position:inside/outside}

Table:

Location of the content .

Positioning : position property (static , relative , absolute and fixed)

Normal positioning

Float positioning ----(relative positioning)

Absolute positioning.

Events can trigger actions in the browser,

Events: can be classified into 4 categories .

1)Window Events 2) Mouse Events 3)Key board Events 4)Form Events

1)Window Events:The window itself has its own events, which trigger when a new page is starting up(onLoad)shutting down(onUnload) being resized (onResize)moved (onMove) cancelled(onAbort) or when an error occurs(onError).

There is also an event triggered when the window moves to foreground (onFocus) or changes to background (onBlur).

2)Mouse Events: The mouse has a few events associated with it when a button is pressed (onMousedown)on top of an element and when it is released (onMouseup);

When the mouse moves and the pointer is already over an element (onMousemove)and when it moves away from the element(onMouseout).

Events are triggered also when the pointer is over an element (onMouseover) and when it is clicked once(onClick) or twice(onDblclick).

3) Key board Event:Events can be triggered when the key is pressed down(onKeydown) and when it is released(onKeyup).

The complete key sequence down press and up release, trigger another event( onKeypress).

4) Events can be triggered when the reset button on the form is clicked (onReset) or when the submit button is clicked (onSubmit).

Even when a content is selected on a page (onSelect) event is generated.

Display :This property is used to specify the display behavior (block, inline ,none) of an element.

Block: the element generates a block element box generating line breaks both before and after the element when in the normal flow.

Inline:The element generates one or more inline element boxes that do not generate line breaks before or after themselves. In normal flow,the next element will be on the same line if there is space.

Navigation Menu:Navbar in css refers to a group of links that lead to different pages of a web site. Navbars are either vertical or horizontal .

A css navigation bar is a collection of links.

You use the addEventListener method to attach an event handler to an HTML element, which can be a div, link, or anything you want. Using addEventListener lets you do something useful when an event is triggered. The following is general syntax for

addEventListener: object.addEventListener(event, eventListenerFunction);

**JavaScript:**

Javascript is a language that is largely used in the world wide web to add client side interactivity to webpages. Client side javascript is simply javascript that runs on the client.

Javascript is a client side scripting language. A scripting language is programming code .

There are three common ways of inserting javascript code in a webpage

1)Inside an html tag script

2)In an external file

3)As a value of some html attributes.

1)<script type=”text/javascript”>

Alert(‘welcome to javascript’)

</script>

2)<script type=”text/javascript” src=” ff2.js” ></script>

3)<a href=”javascript: Alert(‘welcome to javascript’).”>link</a>

**Javascript is a lightweight** ,crossplatform and interpreted scripting language.

Javascript contain a standard library of objects.like Array,Date and math string so on .

DOM:Document object Model is an interface using which given programs/scripts can dynamically alter the contents of a given web document.

Document

Html

Head body

Title p h1

Text id class

Text text

Data types: primitive and Non Primitive

1)numeric 🡪 integer ,floating point

Integer🡪byte,short,int,long

Floating point🡪 double, float

2)Non numeric 🡪char, Boolean

Data type: number ,Boolean undefined(undefined value),string ,null(null value)

(A=2 b=4 a is type of integer memory n=”ppp” 🡪string)

2)Non Primitive 🡪 String ,arrays and ,object user defined .

parseFloat() Parses a string and returns a floating point number

parseInt() Parses a string and returns an integer.

Variable: a=5 aaa=4 name=”priya”

A1=9 name2=”pppp”

First\_name=”jyothi”

\_name

Name must start with a letter(a to z) (A to Z) \_

After first or in between letter we can use digits(0 to 9)

Javascript variables are case sensitive. A a

Operators:An operator performs mathematical or logical operations. They are used to manipulated data and variables called operands.

Sum=a+b sum,a and b ara the operands.

= and + are the operators.

The console.log() is a function in javascript which is used to print any kind of variables defined before in it or to just print any message that needs to be displayed to the user.

Console provides you with the ability to write manage and monitor javascript on demand.

Arithematic operators

Addition(+),subtraction(-) ,multiplication(\*),division(/), Modulus(%),increment(++) and Decrement(--).

Increment (++) operator :post increment and pre increment

Decrement( -- ) operator :post decrement and pre decrement.

Assignment operators

+= (a+=b; mean :a=a+b),-= , \*= ,/= ,%=,&= , |= ,^=, <<=, >>=

Relational operators: or comparison operators

>, < , = , != or <> ,<=, >= , ==

Logical operators:

The operators which returns the Boolean values true or false are called logical operators or Boolean operations .

&& ,|| and !

Function are two types: one predefine functions and user define functions

Pow(2,3)=8

Sqrt(36)=6

Function:A function is a self contained block of code that performs specific and defined tasks. It simplifies complex task by dividing the program into modules .

Syntax:

Function Function\_name ()

{

Statements;

}

Function Hello()

{ defining function

document.write(“this is hello function”);

}

Hello(); 🡪 calling function

Function with return value;

Function without return value;

Function with return and parameters;

Function without return and parameters;

Function area(len ,wid)

{

Statements;

//Return values;

}

Area(4,5);

Recursion: is a process of calling self. Or a function is calling itself

Function recrse(){

Statements;

Recrse()

}

Recrse()

5\*4\*3\*2\*1=120

Function fact(n){

Return F=f\*fact(n-1)

}

5\*fact(5-1)

5\*4\*fact(n-1)

5\*4\*3\*fact(n-1)

5\*4\*3\*2\*fact(n-1)

5\*4\*3\*2\*1=120

Fact(5)

**Conditional statements:** can be defined as the statements get executed based on a specific condition. if the condition is true, block of statements are executed. Otherwise , control flow comes out of the block.

Syntax:

If(condition or expression)

{

Statements;

}

If --- else

If – else is an extension to if control statement. The only difference between them is the else statement contain certain block of statements which gets executed only if its superior if condition becomes false.

If(condition)

{

Statements1;

}

Else

{

Statements2;

}

Nested if:

If(condition)

{

If(condition)

{

Statement;

}else{statements

}else

Switch condition: The switch statement executes a block of code depending on different cases.The switch statement is a part of conditional statements, which are used to perform different action.

The break keyword breaks out of the switch block. This will stop the execution of more execution of code or case testing inside the block.

Switch(expression or choice)

{

Case constant: statements;

Break;

Case constant: statements;

Break;

Case constant: statements;

Break;

Case constant: statements;

Break;

Default : default statements;

Break;

}

Control statements or loops:The control structures within javascript allow the program flow to change within unit of code or function.

While

Do – while

For

1. While : This loop is used to execute a set of statements until the condition remains true.

Syntax:

Initialization i=0;

While(condition) while(i<10)

{ {

Executable statements; document.write(i);

Increment/dec i++;

} }

1. Do --- while: loop is same as that of while loop but the condition for it is specified at the end of the loop.

Syntax:

Initi

Do{

Statements;

Incre/dec

}while(condition);

The body of the loop gets executed at least once even if the specified condition is false.

1. For loop: A set of instructions can be executed repeatly for a specific number of times by using for loop

Syntax:

For(initial;condition; incre/dec;)

{

Statements;

}

The body of the loop will be executed if and only if the condition is true.

Events: Responses provided by the browser as a response to user’s interactions are referred as events.

Event Handler: Once the event is generated there is often requirement of code to process these events. Such code is known as event Handler.

There are two event handler are: Interactive event handler and Non interactive event handler.

a)Interactive event handler: The event handler which simply rely on the user’s activity to be invoked are interactive event handler.eg. onClick() onBlur()...

b)Non interactive event handler: The event handler which do not rely on the user’s to be invoked are non interactive event handler. OnPageLoad()..

DOM: (Document object model)The html DOM is an object Model for html .

Html elements as objects , Properties for all html elements,Methods for all html elements ,events for all html elements.

When a web page is loaded. The browser creates a Document object model of the page.The Html Dom model is constructed as a tree of objects.

Document

Html (root element)

<head> <body>

<title> <a><h1> <p><div>

Title name text <p>

text:”javascript”

Text: “Welcome” <p id=”we”>paragraph</p>

Attributes:class,id color...

Get , set method

innerHTML

Text

getAtribute

When you want to access html elements with javascript :

Finding html elements by ID,tag Name , class Name ,by css ,html object collections

document.getElementById();

document.getElementByTagName(“p”);

document.getElementByClassName(“cc”);

document.querySelectorAll(“p.cccc”);

document.forms

document.image

example: document.getElementById(“btn”).addEventListener(“click”,disp);

<input type=”button” name=”btn”>

The addEventListener() method attaches an event handler to the specified element.

You can add event listener to any DOM object not only html elements. i.e.,click events.

Object:An object is an abstract data type with the addition of polymorphism and inheritance.An objects has state(data) and behaviour(code).

Javascript is designed on a simple object based paradigm. An object is a collection of properties and a property is an association between a name and a value.

Built in objects in javascript:

String, date , math ,Array, and window objects.

String is built in objects:

Name=”Nirmaan”;or name= new String(“Nirmaan online”)

Class friend{ class student{

Int sno; int roll;

String name; string name;

Sum(){ void sum(){ } };

Printf(“this is sum”); student s1,s2; s1.roll=7;s2.roll=4

}

Friend f;

f.sno;f.name;

f1.sno;f1.name,f1.sum();

one method object :var d=new Date();

d.dddd:”suj”;

d.ffff:”sss:

array:var a=[“ram”,”kumar”,”sai”,”india”]-🡪 this is array

another method object🡪var a={

(properties)🡪fname:”ram”,

lname:”kumar”,

age:21,

country:”india”

};

String:

Roll=”34”;

Name=”priya”;

Array: An Array contains a set of (similar) items or data.but in javascript an array contains different items. Arrays are similar to structure variables and they can be accessed randomly by making use of index and the array name. Thus, arrays in javascript are also called as Associative arrays.

* By creating instance of Array directly ( using new keyword)
* By array literal
* By using an Array constructor(using new keyword)

Syntax: arr =new Array();

Syntax : arrayname=[value,value,....]

Syntax : array1=new Array(“srirama”,”sita”,”laxmana”,45,);

FILO

Arr=[4,5,2]; 🡪Arr[0]=4;arr[1]=5 7 a[0] l

Int roll; a 4 a[1]

2 a[2] a n a

A=3

A=[4,5,6,3,4,7,8,...]

A[0]=4,a[1]=5,a[2]=6....

Arr=new Array()

A 1 0

2 4

Prompt dialog boxes:

Form validation:

Form validation is the process verifying or validating the data that is entered by the user.

[jjjjjj@kkkkkk.com](mailto:jjjjjj@kkkkkk.com) 00/00/0000 or 0000/00/00

client side and server side validation78900995345:

ssc id :

a= 3 5 [ 4 5 6 7 ]

2 7

4 5

I, j

A[i][j][k]=a[0][0]=3

A[0][1]=5;a[0][2]=9

A[1][0]=2

A[1][1]=7

A[2][0]=4

A[2][1]=5

Palindrome:

456

Find remainder of number (6)

Use reverse=reverse \*10+remainer ;(6) ii) 6\*10+5=

Find no=no/10(45)

repeats

BOOTSTRAP: is a free front end framework for faster and easier web development.(html n css)

Bootstrap is famous for being developed with components that have the ability to follow the property of responsive designs.

Responsive design is about using css and html to resize, hide,shrink enlarge or over the content to make it look good on any screen.

Responsive design allow your page works for computer ,tablets and mobile phones.

It creates Responsive web pages, it creates platform independent web pages.

Or bootstrap is a collection of css classes and javascript functions the you get

Position :

Bootstrap comprises of mainly three components:

Css fonts javascript

Bootstrap🡪twitter developed company

CSS based front end framework

Faster and easier to web development

Easily create Responsive web layouts

Bootstrap : A framework that makes web development much more efficient.

Framework includes: HTML & CSS based design templates for typography, forms, buttons, tables, navigation,modals, and image carousels.

Also includes optional JS plugins.

open source framework

Most popular and widely used

Css & html development can often be time consuming.

Creating responsive webpages from scratch is challenging .

Streaming the development process.

We can use built – in components.

Responsiveness ,styling and cross-browser functions already tested.

Key Benefits:

Ease of use: Anybody with a basic understanding of HTML and CSS can use bootstrap.

Responsic Features: Bootstrap layouts adjust for optimal viewing on all devices, including phones, tables and desktops.

Cross browser compatibility: Bootstrap is compatible with all modern browsers including (chrome,firefox, internet Explorer,safari and opera).

Layout

Typography

Navigation menu

Form layout

Buttons

Pagination

Icon fonts

Responsive ness

Web site name home gallery contant us office

Advantages:

Save lots of time

Responsive ness

Consistent design

Easy to use

Compatible with browsers

Open source or free code

Bootstrap contain container row and col class

.container

.1140px

<div></div>

One more class called container-fluid(100%)

Screen size: acc. Bootstrap container width size(screen size)

Extra large🡪 >=1200px 1140px-- xl

Large 🡪 >=992px 960px ---lg

Medium🡪 >=768px 720px--md

Small🡪 >=576px 540px --sm

Extra small🡪 <576px atuo-------

<div class=”container”>

<div class=”row”>

</div>

<div class=”row”>

</div>

</div>

If we want to make any break between columns use a tag class= w-100

Grid col class:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Col1 | Col2 | Col3 | Col4 | Col5 | Col6 | Col7 | Col8 | Col9 | Col10 | Col11 | Col12 |

|  |  |  |  |
| --- | --- | --- | --- |
| Col1col2col3 | Col4col5col6 | Col7col8col9 | Col10col11col12 |

|  |  |
| --- | --- |
|  |  |

Off set class:offset classes move(push) columns to the right .

.offset-md-\*

.offset-lg-4

Reordering:

|  |  |  |  |
| --- | --- | --- | --- |
| .col1 | .col2 | .col3 | .col4 |
|  |  |  |  |

.orderlast order-first

Extralarge order-xl-first order-xl-last

Large order-lg-first order-lg-last

Medium order-md-first order-md-last

Small order-sm-first order-sm-last

Extra small order------

|  |  |  |  |
| --- | --- | --- | --- |
| .col1 | .col2 | .col3 | .col4 |

0 1 2 3

1 2 3 4

Nesting row or col

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 100%(col 12)   |  |  |  | | --- | --- | --- | | 4 | 4 | 4 | | 3   |  |  | | --- | --- | | 6 | 6 | | 6   |  |  |  | | --- | --- | --- | | 4 | 4 | 4 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 students details (100%)   |  |  |  |  | | --- | --- | --- | --- | | 3 | 3 | 3 | 3 | | 6 |

Margin Classes : M (m-auto)

Extra large m-xl-auto m-xl-2

Large m-lg-auto

Medium m-md-auto

Small m-sm-auto

Extra small ----

Margin .m-\*

Margin-right .mr-\* mr-md-\* mr-xl-\* mr-lg-\* mr-sm-\*

Margin-left .ml-\*

Margin-top .mt-\*

Margin-bottom .mb-\*

**Margin .m-n\***

Margin-right .mr-n\* mr-md-n\* mr-xl-\* mr-lg-\* mr-sm-\*

Margin-left .ml-\*

Margin-top .mt-\*

Margin-bottom .mb-\*

Sizes:

1. 0px
2. 0.25rem
3. 0.50rem
4. 1rem
5. 1.5rem
6. 3rem

Padding : space between the text and box or col or row.

In built in classes.

P pr-\* pl-\* pt-\* pb-\*

Pr-xl-\* pr-lg-\* pr-md-\* pr-sm-\*

Pl-xl-\*

Pt-xl-\*

Pb-xl-\*

Py-\* px-\*

My-2 mx-\*

|  |  |
| --- | --- |
| Cola | colb |
|  |  |
|  |  |

Flex direction classes:

Col1 col 2 col3 col4

|  |  |
| --- | --- |
| Col | |
| Col2 | |
| Col3 | Col4 |

Flex-row-reverse – reverse inorder all the cols

Display :none 🡪d-none 🡪d-xl-none 🡪d-md-none

:inline 🡪d-inline

:inline-block 🡪d-inline-block

:block🡪d-block

:table🡪d-table

:table🡪d-table-cell

:table-row 🡪d-table-row

:flex🡪d-flex

:inline-flex🡪d-inline-flex

Button

classes: The .btn classes are designed to be used with the button element .when using button classes on <a> element that are used to trigger in page functionality ,(<input> <button>)

btn-primary,btn-secondary,

<a href=”#” class=”btn btn-primary”>button</a>

outline buttons: outline of the button visible.

Sizes: larger , small and medium . but medium is default

Block button: create block level buttons that span the full width of a parent by adding btn-block.

Active state:Buttons will appear pressed when active. “active”

Disabled state : make buttons look inactive by adding the disabled Boolean attribute to any button.

Group button:

## Toolbar:

<button type=”button”> click me </button>

<input type=”button” value=”BUTTON”>

Class=”btn”

Gray :btn-secondary,

Blue:btn-primary

Green:btn-success

Red:btn-danger

Orange:btn-waring

White:btn-light

Black :btn-dark

Link:btn-link

<a href=”” class=”btn btn-primary”>name</a>

<input class=”btn btn-success” type=”button” value=”COLOR”>

<button type=”button”></button>

Value=”submit” reset

Outline button classes:

Btn-outline-primary...

Button size: large btn-lg

Small: btn-sm

Medium:

Class: block

Btn-block

Group button:

Toolbar class to create icons

Icon on the group buttons

To icons on the button use toolbar class.

Dropdown-toggle🡪dropdown menu dropdown-toggle🡪data-toggle🡪dropdown

<div class=”dropdown-menu”>

<a href=”#” class=”dropdown-item”> ffff</a>

<a href=”#” class=”dropdown-item”> ffff</a>

Dropdown-item🡪 items of dropdown menu

Jqery.js,popper.js,and bootstrap.js

List :collection of items.

List-group🡪creating the list

List-group-items🡪 collection of items

List-group-flush🡪used to remove the outline of the list box .

List-group-item-primary🡪 is used to set the background color.

We can use <ul> tag and <li> tag to create list and list items. And also use <div> tag to create list and list items.

List-group-item-action 🡪 to remove underline for the list items when we use <a> tag to create items of the list.

Tables: .table ,.table-striped🡪alternate color light and dark display in the rows

.table-dark 🡪 complete table display in dark color.

,table-hover: 🡪 highlighted when we move on the text and when we move the cursor,

To add color in the table : bg-primary,bg-success.bg-waring....

All table styles are interited in bootstrap

Pagination:

Previous 1 2 3 next

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| previous | 1 | 2 | 3 | 4 | next |

We use a large block of connected links for our pagination, making links hard to easily scalable all while providing large hit areas. Pagination is built with list HTML elements on screen readers can announce the number of available links.

<nav >

<ul class=”pagination”>

<li class=”page-item”><a href=”#”> previous</a></li>

<li class=”page-item active”><a href=””> 1</a></li>

<li class=”page-item”><a href=””> 2</a></li>

<li class=”page-item”><a href=””> 3</a></li>

<li class=”page-item”><a href=””> next</a></li>

</nav>

Bread crumbs classes:

|  |
| --- |
| Home/library/data/gallery/ |

<nav>

<ol class=”breadcrumb”>

<li class=”breadcrumb-item”><a href=”” >home</a></li>

<li class=”breadcrumb-item”><a href=”” >library</a></li>

</ol>

</nav>

Jumbotron:

A lightweight ,flexible component that can optionally extend the entire viewport to showcase key marketing messages on your web site.

<div class=”jumbotron”>

<h1></h1>

<p> </p>

<a class=”btn btn-primary btn-lg”> click me </a>

Jumbotron-fluid: modified jumbotron that occupies the entire horizontal space of its parent. To make the jumbotron full width, without rounded corners,

</div>

<div class=”jumbotron jumbotron-fluid”>

</div>

Forms: Bootstrap forms support the following form controls:input ,textarea,button,checkbox, radio and select.

Various types of form layouts such as vertical form, horizontal form and inline form .

.form .form-control

<form>

<div class=”form-group form-control”>

Id=”” placeholder=” readonly”

<input readonly> .form-control-plaintext class to remove the default form field styling and preserve the right margin and padding.

domain@mail.com

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| submit |

To upload file we need to take class form-control-file .

To show the range add form-control-range.

Input group classes: .input-group <div class=”input-group-prepend>

|  |  |
| --- | --- |
| $Rupee |  |

|  |  |
| --- | --- |
| Kl;dskf;lsdkf | @domain.com |

<div class=”input-group-append”>🡪rightside

<span class=”input-group-text”>$</span>

Border and shadow classes:

.border

.border-top

.border-left

.border-right

.border-bottom

.border-none

.border-top-0

.border-left-0

.border-right-0

.border-bottom-0

.border-rounded

.border-rounded-top

Shadow class:

Shadow-none—to remove the shadow

Shadow-sm ,shadow-lg, shadow.

Badge classes: can be used as part of link or buttons to provide a counter.

Class=”badge badge-success”>

Notification 2

Hhhhhhh nnnn mmmm

Progress bar classes:custom progress bars are used to support for stacked bars animated background and text labels.

Progress

50%

.progress

Spinner:Bootstrap “spinners” can be used to show the loading state in your project.

Grow, border

Carousel classes:The carousel is a slideshow for cycling through a series of content.

carousel slide carousel-fade

<div class=”Carousel slide”>

<div class=”carousel-inner”>

<div class=”carousel-item ”>

<img class=”d-block w-100” scr=”” alt=”f slide> slide 1 </div>

</div>

</div>

<div class=”carousel-item active”>

<img class=”d-block w-100” scr=”” alt=”f slide> slide 1 </div>

</div>

Card classes:

.card-img-top

.card-img-bottom

.card

.card-title

.card-body

Bootstrap tooltips:

Jquery.js

Popper.js

Bootstrap.js

bundles

Data-placement=”top” or “bottom”,”left”,”right”

Data-toggle=”tooltip”

tooltip

<script>

$(function(){

$(‘[data-toggle=”tooltip”]’).tooltip()

}

);

</script>

Nav:

<nav>

<a href=””>fff</a>

<a href=””>fff</a>

<a href=””>fff</a>

</nav>

Navbar

MYSQL : Database management System🡪 A Database is a collection of data stored in a format that can easily be accessed.

Eg: name:

Age:

Gender

Submit

|  |  |  |
| --- | --- | --- |
| Name | Age | Gender |
| Sunita | 32 | Female |
| Anil | 34 | Male |
| Kiran | 23 | Male |
|  |  |  |
|  |  |  |

System🡪DBMS🡪 Database

Oracle,Mysql,mysql server,postgresql,mongodb

Two type of database:

Relational, NoSql

1)Relational :it is stored in table forms.(rdbms) relational database.

To create this relational between tables use a language SQL (structured query language).

Bank:employee, customer, account

2)Nosql:is not a table based database.

Nosql databases are document based.

MongoDb,Redis, Cassandra etc,.

Advantage of mysql:

Cross platform

Used with multiple language (php,noodejs,python,c#)

Mysql software is open source

Mysql is Rdbms

The mysql database server is fast, reliable, scalable(small or large site) and easy to use.

Using mysql site:

Facebook ,twitter, google, Wikipedia ,youtube, flickr, pinterest...

Create database and tables in db

Add data in db,update data in db, read data form db, delete data form db.

Go to command prompt:

Type cd xampp

Xampp>cd mysql

Xampp>mysql>cd bin

Xampp>mysql>bin> mysql –u root –p –h 127.0.0.1

mariaDB[<none>]> show databases;

>create database databasename;

mariaDB[<none>]> use databasename

mariaDB[<databasename>]>show tables;

to create table:syntax:

create table tablename(colname1 datatype(size),colname2 datatype(size).....);

create table stud(sroll int(5), sname varchar(10),quail varchar(10),dept int(3));

to see the structure of the table:

describe tablename;

to see the rows in the table;

select \* from tablename;

to insert data in the table syntax:

insert into tablename values(data);

insert into faculty values(34,’dddd’,’ddd’,3434);

**distinct clause**: is used to remove duplicate rows from the result set.

Syntax: select distinct col from tablename;

Using distinct clause with multiple columns

Filtering data(where) : where clause and where with logical operator (and , or)

Where clause allows to filter rows based on certain condition or expression

Syntax:

Select col1,col2,..... from table1 ,table2 where condition;

Operation describe

= equal

< > or != not equal to

< less than

>

<=

>=

**IN and NOT IN operator with some query:**

IN operator is used to filter data and allows to determine if the value matches any one of the list or result of sub query.

Select col1,col2 ... from table1 table2.. where (expre|col1) IN (value1,value2..);

MariaDB [colle]> select \* from faculty;

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

7 rows in set (0.002 sec)

MariaDB [colle]> select \* from faculty where deptid=15 or deptid=102;

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 501 | jyothi | M.tech | 15 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

4 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where deptid in(15,102);

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 501 | jyothi | M.tech | 15 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

4 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where deptid in(15,102,101);

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

6 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where qualif in('m.tech','b.tech');

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

+------+-----------+--------+--------+

5 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where qualif in('m.tech','b.tech','b.sc');

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

6 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where qualif not in('m.tech','b.tech','b.sc');

+------+-------+--------+--------+

| fid | fname | qualif | deptid |

+------+-------+--------+--------+

| 34 | ravi | b.a | 102 |

+------+-------+--------+--------+

1 row in set (0.000 sec)

MariaDB [colle]> select \* from faculty where qualif not in('m.tech','b.tech');

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

2 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty where deptid not in(15,102,101);

+------+---------+--------+--------+

| fid | fname | qualif | deptid |

+------+---------+--------+--------+

| 37 | sandeep | M.tech | 42 |

+------+---------+--------+--------+

1 row in set (0.000 sec)

MariaDB [colle]>

**Between and not between operator:**

Between operator is used to filter data and allows to specified range of values of test.

Select col1,col2,... from table1, table2.. where (expre | col1) between begin(>=) and end(<=);

MariaDB [colle]> select \* from student where marks between 350 and 450;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 13 | sunitha | ece | 357 |

+------+---------+--------+-------+

1 row in set (0.000 sec)

MariaDB [colle]> select \* from student;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

+------+---------+--------+-------+

7 rows in set (0.000 sec)

MariaDB [colle]> select \* from student where marks between 350 and 467;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | 357 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

+------+---------+--------+-------+

5 rows in set (0.000 sec)

MariaDB [colle]> select \* from student where marks between 350 and 467;

**Limit and is null operator:**

Limit is used to constrain the number of rows returned by the select statement.

Syntax:

Select col1,col2 from table1 table2.. limit offset ;

Select col1 col2 from table1 table limit N;

count;

MariaDB [colle]> select \* from student limit 4;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

+------+---------+--------+-------+

4 rows in set (0.001 sec)

MariaDB [colle]> select \* from student limit 2,4;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

+------+---------+--------+-------+

4 rows in set (0.001 sec)

MariaDB [colle]>

MariaDB [colle]> select \* from student limit 0,4;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

+------+---------+--------+-------+

]> select \* from student ord

er by marks desc limit 0,4;

+------+--------+--------+-------+

| sid | sname | branch | marks |

+------+--------+--------+-------+

| 14 | anitha | it | 567 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

| 15 | ramesh | eee | 467 |

+------+--------+--------+-------+

4 rows in set (0.045 sec)

MariaDB [colle]> select \* from student order by sname desc limit 0,4;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 15 | vijay | eee | 467 |

| 13 | sunitha- | ece | NULL |

| 13 | sunitha | ece | 357 |

| 16 | sunil | cse | 467 |

+------+---------+--------+-------+

4 rows in set (0.000 sec)

MariaDB [colle]> select \* from student where marks is null;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 13 | sunitha | ece | NULL |

+------+---------+--------+-------+

1 row in set (0.001 sec)

MariaDB [colle]> select \* from student where marks is not null;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

+------+---------+--------+-------+

6 rows in set (0.044 sec)

**Like operator:** like operator is used to select rows/data based on patterns following two wild card characters are used with like operator.

1)percentage (%) it matches any string of zero or more characters.

2)underscore(\_): it matches exactly one single character.

Select col... from table... where condition like ‘pattern’;

MariaDB [colle]> select \* from faculty where fname like 's%';

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 37 | sandeep | M.tech | 42 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

+------+-----------+--------+--------+

3 rows in set (0.001 sec)

MariaDB [colle]> select \* from faculty where fname like '%i';

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 501 | jyothi | M.tech | 15 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

3 rows in set (0.001 sec)

MariaDB [colle]> select \* from faculty where fname like '%h%';

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

+------+-----------+--------+--------+

3 rows in set (0.001 sec)

MariaDB [colle]> select \* from faculty where fname like '\_y%';

+------+--------+--------+--------+

| fid | fname | qualif | deptid |

+------+--------+--------+--------+

| 501 | jyothi | M.tech | 15 |

+------+--------+--------+--------+

1 row in set (0.001 sec)

MariaDB [colle]> select \* from faculty where fname like 'r\_\_\_';

+------+-------+--------+--------+

| fid | fname | qualif | deptid |

+------+-------+--------+--------+

| 34 | ravi | b.a | 102 |

+------+-------+--------+--------+

1 row in set (0.000 sec)

MariaDB [colle]> select \* from faculty where fname like 'r\_\_\_\_';

Empty set (0.000 sec)

MariaDB [colle]> select \* from faculty wheroe fname like 'r\_';

Empty set (0.000 sec)

MariaDB [colle]> select \* from faculty where fname like 'r\_%';

+------+-------+--------+--------+

| fid | fname | qualif | deptid |

+------+-------+--------+--------+

| 34 | ravi | b.a | 102 |

+------+-------+--------+--------+

1 row in set (0.000 se

MariaDB [colle]> select \* from student where sname not like '\_a%';

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

+------+---------+--------+-------+

5 rows in set (0.001 sec)

MariaDB [colle]> select \* from student where sname not like 's\_\_\_\_';

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

+------+---------+--------+-------+

6 rows in set (0.000 sec)

MariaDB [colle]>

Order by clause:( sorting data):order by clause is used to sort the result of the result statement

Syntax:select column from table order by col1 [asc/desc];

MariaDB [colle]> select \* from student order by sname asc;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 12 | sairam | B.tec | 456 |

| 16 | sunil | cse | 467 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 15 | vijay | eee | 467 |

+------+---------+--------+-------+

7 rows in set (0.045 sec)

MariaDB [colle]> select \* from student order by sname desc;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 15 | vijay | eee | 467 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 16 | sunil | cse | 467 |

| 12 | sairam | B.tec | 456 |

| 15 | ramesh | eee | 467 |

| 14 | anitha | it | 567 |

+------+---------+--------+-------+

7 rows in set (0.000 sec)

Multiple columns using order by clause:

Syntax: select \* from tablename order by colname [asc/desc] ,col2 [asc/desc];

Otherwise:

Select \* from tablename order by field( colname, ‘value’,’value’,.,,,,);

MariaDB [colle]> select \* from student;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

| 16 | sunil | eee | 478 |

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

+------+---------+--------+-------+

10 rows in set (0.000 sec)

MariaDB [colle]> select \* from student order by sname asc ,marks desc;

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 12 | sairam | B.tec | 456 |

| 16 | sunil | eee | 478 |

| 16 | sunil | cse | 467 |

| 13 | sunitha | ece | 357 |

| 13 | sunitha | ece | NULL |

| 18 | varma | it | 543 |

| 17 | varma | cse | 476 |

| 15 | vijay | eee | 467 |

+------+---------+--------+-------+

10 rows in set (0.000 sec)

MariaDB [colle]> select \* from student order by field(branch,'eee','cse');

+------+---------+--------+-------+

| sid | sname | branch | marks |

+------+---------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 18 | varma | it | 543 |

| 14 | anitha | it | 567 |

| 13 | sunitha | ece | 357 |

| 13 | sunitha | ece | NULL |

| 15 | ramesh | eee | 467 |

| 16 | sunil | eee | 478 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

| 17 | varma | cse | 476 |

+------+---------+--------+-------+

10 rows in set (0.000 sec)

Joins: Join is a method of linking one table with another table .

Types of joins in mysql are following:

Cross join, inner join , left join and right join.

Cross join: it returns the Cartesian product of rows from the joined tables.

Syntax: select \* from t1 cross join t2 where joincondition(option);

MariaDB [colle]> select \* from faculty cross join department where department.deptid=15;

+------+-----------+--------+--------+--------+----------+

| fid | fname | qualif | deptid | deptid | deptname |

+------+-----------+--------+--------+--------+----------+

| 32 | priya | M.tech | 15 | 15 | M.tech |

| 37 | sandeep | M.tech | 42 | 15 | M.tech |

| 501 | jyothi | M.tech | 15 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 34 | ravi | b.a | 102 | 15 | M.tech |

| 35 | kumar sai | b.sc | 102 | 15 | M.tech |

+------+-----------+--------+--------+--------+----------+

7 rows in set (0.003 sec)

MariaDB [colle]> select \* from faculty cross join department where department.deptid=faculty.deptid;

+------+-----------+--------+--------+--------+----------+

| fid | fname | qualif | deptid | deptid | deptname |

+------+-----------+--------+--------+--------+----------+

| 32 | priya | M.tech | 15 | 15 | M.tech |

| 501 | jyothi | M.tech | 15 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 101 | b.tech |

| 502 | sangeetha | b.tech | 101 | 101 | b.tech |

| 34 | ravi | b.a | 102 | 102 | b.sc |

| 35 | kumar sai | b.sc | 102 | 102 | b.sc |

+------+-----------+--------+--------+--------+----------+

6 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty;

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

7 rows in set (0.000 sec)

MariaDB [colle]> select \* from department;

+--------+----------+

| deptid | deptname |

+--------+----------+

| 15 | M.tech |

| 101 | b.tech |

| 102 | b.sc |

| 103 | b.com |

+--------+----------+

4 rows in set (0.000 sec)

MariaDB [colle]>

MariaDB [colle]> select \* from faculty cross join department where department.deptid=15;

+------+-----------+--------+--------+--------+----------+

| fid | fname | qualif | deptid | deptid | deptname |

+------+-----------+--------+--------+--------+----------+

| 32 | priya | M.tech | 15 | 15 | M.tech |

| 37 | sandeep | M.tech | 42 | 15 | M.tech |

| 501 | jyothi | M.tech | 15 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 34 | ravi | b.a | 102 | 15 | M.tech |

| 35 | kumar sai | b.sc | 102 | 15 | M.tech |

+------+-----------+--------+--------+--------+----------+

7 rows in set (0.000 sec)

MariaDB [colle]> select \* from faculty cross join department;

+------+-----------+--------+--------+--------+----------+

| fid | fname | qualif | deptid | deptid | deptname |

+------+-----------+--------+--------+--------+----------+

| 32 | priya | M.tech | 15 | 15 | M.tech |

| 32 | priya | M.tech | 15 | 101 | b.tech |

| 32 | priya | M.tech | 15 | 102 | b.sc |

| 32 | priya | M.tech | 15 | 103 | b.com |

| 37 | sandeep | M.tech | 42 | 15 | M.tech |

| 37 | sandeep | M.tech | 42 | 101 | b.tech |

| 37 | sandeep | M.tech | 42 | 102 | b.sc |

| 37 | sandeep | M.tech | 42 | 103 | b.com |

| 501 | jyothi | M.tech | 15 | 15 | M.tech |

| 501 | jyothi | M.tech | 15 | 101 | b.tech |

| 501 | jyothi | M.tech | 15 | 102 | b.sc |

| 501 | jyothi | M.tech | 15 | 103 | b.com |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 101 | b.tech |

| 502 | sangeetha | b.tech | 101 | 102 | b.sc |

| 502 | sangeetha | b.tech | 101 | 103 | b.com |

| 502 | sangeetha | b.tech | 101 | 15 | M.tech |

| 502 | sangeetha | b.tech | 101 | 101 | b.tech |

| 502 | sangeetha | b.tech | 101 | 102 | b.sc |

| 502 | sangeetha | b.tech | 101 | 103 | b.com |

| 34 | ravi | b.a | 102 | 15 | M.tech |

| 34 | ravi | b.a | 102 | 101 | b.tech |

| 34 | ravi | b.a | 102 | 102 | b.sc |

| 34 | ravi | b.a | 102 | 103 | b.com |

| 35 | kumar sai | b.sc | 102 | 15 | M.tech |

| 35 | kumar sai | b.sc | 102 | 101 | b.tech |

| 35 | kumar sai | b.sc | 102 | 102 | b.sc |

| 35 | kumar sai | b.sc | 102 | 103 | b.com |

+------+-----------+--------+--------+--------+----------+

28 rows in set (0.000 sec)

**Inner join**:It matches rows of one table with another table.

Select collist from tablename t1 inner join tablename t2 on joincondition;

Select f.fname,f.deptid from faculty f inner join department d on f.deptid= d.deptid;

select f.fname,d.deptname from faculty f inner join department d on f.deptid=d.deptid;

+-----------+----------+

| fname | deptname |

+-----------+----------+

| priya | M.tech |

| jyothi | M.tech |

| sangeetha | b.tech |

| sangeetha | b.tech |

| ravi | b.sc |

| kumar sai | b.sc |

+-----------+----------+

6 rows in set (0.003 sec)

MariaDB [colle]> select \* from faculty;

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

+------+-----------+--------+--------+

7 rows in set (0.001 sec)

MariaDB [colle]> select \* from department;

+--------+----------+

| deptid | deptname |

+--------+----------+

| 15 | M.tech |

| 101 | b.tech |

| 102 | b.sc |

| 103 | b.com |

| 16 | eee |

| 17 | cse |

| 18 | ece |

+--------+----------+

Left join: result in all the matching rows from two table and also the rows that are not matched from left table.

Syntax: select cols from table1 left join table2 on joincondition

Right join result in all the matching rows from two tables and also the rows that are not matched from right table.

Select columns from table1 right join table2 on join condition;

select f.fname,f.deptid,d.deptname from faculty f right join department d on f.deptid =d.deptid;

+-----------+--------+----------+

| fname | deptid | deptname |

+-----------+--------+----------+

| priya | 15 | M.tech |

| jyothi | 15 | M.tech |

| sangeetha | 101 | b.tech |

| sangeetha | 101 | b.tech |

| ravi | 102 | b.sc |

| kumar sai | 102 | b.sc |

| NULL | NULL | b.com |

+-----------+--------+----------+

7 rows in set (0.001 sec)

MariaDB [colle]> select f.fname,d.deptname from faculty f right join department d on f.deptid =d.deptid;

+-----------+----------+

| fname | deptname |

+-----------+----------+

| priya | M.tech |

| jyothi | M.tech |

| sangeetha | b.tech |

| sangeetha | b.tech |

| ravi | b.sc |

| kumar sai | b.sc |

| NULL | b.com |

+-----------+----------+

Self join is used to join a table to itself.

MariaDB [colle]> insert into emp values(25,'ravi',24);

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> select \* from emp;

+------+---------+--------+

| eid | ename | esuper |

+------+---------+--------+

| 21 | veeru | 23 |

| 22 | kavitha | 25 |

| 23 | sudhams | 22 |

| 24 | gayatri | 21 |

| 25 | ravi | 24 |

+------+---------+--------+

5 rows in set (0.000 sec)

MariaDB [colle]> select e.eid as "employee id",s.ename as "supervisor" from emp e ,emp s where e.esuper=s.eid;

+-------------+------------+

| employee id | supervisor |

+-------------+------------+

| 24 | veeru |

| 23 | kavitha |

| 21 | sudhams |

| 25 | gayatri |

| 22 | ravi |

+-------------+------------+

5 rows in set (0.002 sec)

MariaDB [colle]> select e.eid as "employee id",e.ename,s.ename as "supervisor",s.eid from emp e ,emp s where e.esuper=s.eid;

+-------------+---------+------------+------+

| employee id | ename | supervisor | eid |

+-------------+---------+------------+------+

| 24 | gayatri | veeru | 21 |

| 23 | sudhams | kavitha | 22 |

| 21 | veeru | sudhams | 23 |

| 25 | ravi | gayatri | 24 |

| 22 | kavitha | ravi | 25 |

+-------------+---------+------------+------+

1. rows in set (0.000 sec)

**Grouping data:**

Group by and having clause:

Group by clause groups a set of rows into a set of summary rows by values of columns or expressions

Generally it is used with aggregate functions like, sum ,average, max, min and count.

Syntax: select col1,col2... aggregate function(col) from tablename where condition group by col1,col2..

select sum(marks) from student;

+------------+

| sum(marks) |

+------------+

| 4278 |

+------------+

1 row in set (0.009 sec)

MariaDB [colle]> select branch count(\*) from student group by branch;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'count(\*) from student group by branch' at line 1

MariaDB [colle]> select branch ,count(\*) from student group by branch;

+--------+----------+

| branch | count(\*) |

+--------+----------+

| B.tec | 1 |

| cse | 2 |

| ece | 2 |

| eee | 3 |

| it | 2 |

+--------+----------+

5 rows in set (0.014 sec)

MariaDB [colle]> select branch ,sum(marks) from student group by branch;

+--------+------------+

| branch | sum(marks) |

+--------+------------+

| B.tec | 456 |

| cse | 943 |

| ece | 357 |

| eee | 1412 |

| it | 1110 |

+--------+------------+

5 rows in set (0.002 sec)

MariaDB [colle]> select branch ,min(marks) from student group by branch;

+--------+------------+

| branch | min(marks) |

+--------+------------+

| B.tec | 456 |

| cse | 467 |

| ece | 357 |

| eee | 467 |

| it | 543 |

+--------+------------+

5 rows in set (0.003 sec)

MariaDB [colle]> select branch ,max(marks) from student group by branch;

+--------+------------+

| branch | max(marks) |

+--------+------------+

| B.tec | 456 |

| cse | 476 |

| ece | 357 |

| eee | 478 |

| it | 567 |

+--------+------------+

5 rows in set (0.001 sec)

Group by with having clause is used to filter the groups returned.

Syntax: select col1,col2 ... aggregate function(col) from tablename where condition group by col1 ,col2 having condition;

MariaDB [colle]> select branch, avg(marks) from student group by branch ;

+--------+------------+

| branch | avg(marks) |

+--------+------------+

| B.tec | 456.0000 |

| cse | 471.5000 |

| ece | 357.0000 |

| eee | 470.6667 |

| it | 555.0000 |

+--------+------------+

5 rows in set (0.008 sec)

MariaDB [colle]> select branch, avg(marks) from student group by branch having branch in('cse','it');

+--------+------------+

| branch | avg(marks) |

+--------+------------+

| cse | 471.5000 |

| it | 555.0000 |

+--------+------------+

2 rows in set (0.000 sec)

**Sub query (nested query) and exists and not exists:**

Subquery is a nested query within another queries like select ,insert ,update or delete.

Syntax:

Select col1,col2,... from tablename where condition [=in|not in] {nested query}

(note: if we are getting single row or single value then we can use ‘=’ or other wise use other option;)

select fname ,deptid from faculty where deptid in(select deptid from department where deptname in('M.tech','b.tech'));

+-----------+--------+

| fname | deptid |

+-----------+--------+

| priya | 15 |

| jyothi | 15 |

| sangeetha | 101 |

| sangeetha | 101 |

+-----------+--------+

4 rows in set (0.001 sec)

MariaDB [colle]> select fname from faculty where deptid in(select deptid from department where deptname ='M.tech');

+--------+

| fname |

+--------+

| priya |

| jyothi |

+--------+

2 rows in set (0.000 sec)

MariaDB [colle]> select fname ,deptid from faculty where deptid in(select deptid from department where deptname ='M.tech');

+--------+--------+

| fname | deptid |

+--------+--------+

| priya | 15 |

| jyothi | 15 |

+--------+--------+

2 rows in set (0.000 sec)

MariaDB [colle]> select fname ,deptid from faculty where deptid=(select deptid from department where deptname ='M.tech');

+--------+--------+

| fname | deptid |

+--------+--------+

| priya | 15 |

| jyothi | 15 |

+--------+--------+

2 rows in set (0.000 sec)

select \* from department;

+--------+----------+

| deptid | deptname |

+--------+----------+

| 15 | M.tech |

| 101 | b.tech |

| 102 | b.sc |

| 103 | b.com |

| 16 | eee |

| 17 | cse |

| 18 | ece |

+--------+----------+

7 rows in set (0.000 sec)

MariaDB [colle]> select branch,avg(marks) from student where branch in (select deptname from department where deptid in(15,16));

+--------+------------+

| branch | avg(marks) |

+--------+------------+

| eee | 470.6667 |

+--------+------------+

1 row in set (0.000 sec)

MariaDB [colle]> select branch,avg(marks) from student where branch in (select deptname from department where deptid not in(15,16));

+--------+------------+

| branch | avg(marks) |

+--------+------------+

| ece | 433.3333 |

+--------+------------+

1 row in set (0.000 sec)

Exists and not Exists:if sub query doesnot give any rows then main query doesn’t fires.

Select \* from student where exists (select \* from department where deptid=15);

**Set operators:**

Union ,intersect and minus:

Union ,union all:

Union operator is used to combine two or more result sets into signle result set .

By default union operator removes duplicate rows (use union all to retain duplicate rows)

Rules: Number and the orders of columns must be same.

Data types of the column must be same.

Syntax: select col1 ,col2 … from table1 union|union all select col1,col2….

MariaDB [colle]> select fname from faculty union select sname from student;

+-----------+

| fname |

+-----------+

| priya |

| sandeep |

| jyothi |

| sangeetha |

| ravi |

| kumar sai |

| sairam |

| sunitha |

| anitha |

| ramesh |

| vijay |

| sunil |

| varma |

+-----------+

13 rows in set (0.012 sec)

MariaDB [colle]> select fname from faculty union all select sname from student;

+-----------+

| fname |

+-----------+

| priya |

| sandeep |

| jyothi |

| sangeetha |

| sangeetha |

| ravi |

| kumar sai |

| sairam |

| sunitha |

| sunitha |

| anitha |

| ramesh |

| vijay |

| sunil |

| sunil |

| varma |

| varma |

| sangeetha |

| anitha |

+-----------+

19 rows in set (0.001 sec)

Intersect:operator is used to fetch common rows from two or more tables. It compares the result of two queries and returns the distinct rows that are output by both left and right queries.

Note : no intersect key in mysql .

MariaDB [colle]> select distinct fname from faculty where fname in(select sname from student);

+-----------+

| fname |

+-----------+

| sangeetha |

| ramesh |

+-----------+

2 rows in set (0.000 sec)

Minus: operator is used to fetch rows which are present in left query but not in right query eg:a={2,3,4} b={4,5,6} a-b={2,3}

Select col1,col2 from table1 minus select col1,col2… from table2;

MariaDB [batch31]> select did from faculty left join dept using(did) where dept.did is null;

+------+

| did |

+------+

| 45 |

+------+

1 row in set (0.010 sec)

MariaDB [batch31]> select did,fname from faculty left join dept using(did) where dept.did is null;

+------+-------+

| did | fname |

+------+-------+

| 45 | thanu |

+------+-------+

1 row in set (0.000 sec)

MariaDB [batch31]> select \* from faculty;

+------+---------+------+--------+

| fid | fname | did | qualif |

+------+---------+------+--------+

| 201 | naveen | 22 | b.tech |

| 202 | praveen | 24 | m.tech |

| 203 | ajay | 26 | b.tech |

| 204 | chinnu | 23 | m.tech |

| 205 | thyish | 25 | m.tech |

| 201 | naveen | 22 | b.tech |

| 201 | naveen | 22 | b.tech |

| 206 | karunya | 21 | m.tech |

| 456 | thanu | 45 | bsc |

+------+---------+------+--------+

9 rows in set (0.000 sec)

MariaDB [batch31]> select \* from dept;

+------+-------+

| did | dname |

+------+-------+

| 21 | cse |

| 22 | eee |

| 23 | ece |

| 24 | mech |

| 25 | civil |

| 26 | it |

| 27 | b.com |

| 21 | cse |

| 61 | MA |

| 62 | B.ed |

+------+-------+

10 rows in set (0.000 sec)

Sql commands:DDL(data definition language): create, alter ,drop truncate, comment,rename

DML(Data manipulation language): insert ,update,delete

DCL(data control language):grant, revoke

TCL(transaction control language):commit, rollback,savepoint ,set transaction

DDL:data definition language which deals with database schemasand descriptions,of how the data should reside in the database.

Create : to create a database and its objects like table,index ,views, store procedure, function and triggers)

Alter: alters the structure of the existing database.

Drop: delete objects from the database.

Trumcate: remove all records from a table, including all spaces allocatedfor the records are removed.

Rename: rename an object.

DML(data manipulation language)which deals with data manipulation and includes most common mysql statements such insert,update delete.... and it is used to store , modify, retrieve,delete and update data in a database.

Insert : statement is used to insert data into the db tables.

Insert into tablename values(......);

Insert into tablename(colname,....) values(.....);

Insert with select clause: it is used to insert data into table from another table.

Insert into tablename select col1,col2.... from tablename;

MariaDB [colle]> select \* from faculty;

+------+-----------+--------+--------+

| fid | fname | qualif | deptid |

+------+-----------+--------+--------+

| 32 | priya | M.tech | 15 |

| 37 | sandeep | M.tech | 42 |

| 501 | jyothi | M.tech | 15 |

| 502 | sangeetha | b.tech | 101 |

| 502 | sangeetha | b.tech | 101 |

| 34 | ravi | b.a | 102 |

| 35 | kumar sai | b.sc | 102 |

| 15 | ramesh | eee | 564 |

+------+-----------+--------+--------+

8 rows in set (0.075 sec)

MariaDB [colle]> insert into fac1 select fid,fname deptid from faculty;

ERROR 1146 (42S02): Table 'colle.fac1' doesn't exist

MariaDB [colle]> create table fac1(fid int(5),fname varchar(15));

Query OK, 0 rows affected (0.050 sec)

MariaDB [colle]> insert into fac1 select fid,fname deptid from faculty;

Query OK, 8 rows affected (0.048 sec)

Records: 8 Duplicates: 0 Warnings: 0

MariaDB [colle]> select \* from fac1;

+------+-----------+

| fid | fname |

+------+-----------+

| 32 | priya |

| 37 | sandeep |

| 501 | jyothi |

| 502 | sangeetha |

| 502 | sangeetha |

| 34 | ravi |

| 35 | kumar sai |

| 15 | ramesh |

**To enter multiple rows at a time ,**

Insert into tablename values(value1,value2,...),

(value1,value2,....),

(value1,value2...);

MariaDB [colle]> insert into fac1 values(36,'rithivka'),

-> (56,'vrusha'),

-> (45,'raj');

Query OK, 3 rows affected (0.002 sec)

Records: 3 Duplicates: 0 Warnings: 0

MariaDB [colle]> select \* from fac1;

+------+-----------+

| fid | fname |

+------+-----------+

| 32 | priya |

| 37 | sandeep |

| 501 | jyothi |

| 502 | sangeetha |

| 502 | sangeetha |

| 34 | ravi |

| 35 | kumar sai |

| 15 | ramesh |

| 36 | rithivka |

| 56 | vrusha |

| 45 | raj |

+------+-----------+

11 rows in set (0.000 sec)

MariaDB [colle]>

Update statement: is used to update existing data in a table.

Update tablename set colname=expr1 ,col2=expr2 where condition;

update student set sid=19 ,sname='kirankumar' where marks=478;

Query OK, 1 row affected (0.012 sec)

Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [colle]> select \* from student;

+------+------------+--------+-------+

| sid | sname | branch | marks |

+------+------------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

| 19 | kirankumar | eee | 478 |

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

| 34 | sangeetha | b.tec | 579 |

| 35 | anitha | it | 657 |

+------+------------+--------+-------+

12 rows in set (0.001 sec)

Delete: statement is used to delete data from table .

Syntax: delete from tablename where condition[option].

Delete from table limit number;

MariaDB [colle]> delete from stud limit 4;

Query OK, 4 rows affected (0.001 sec)

MariaDB [colle]> select \* from stud;

+------+-----------+--------+-------+

| sid | sname | branch | marks |

+------+-----------+--------+-------+

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

| 34 | sangeetha | b.tec | 579 |

| 35 | anitha | it | 657 |

+------+-----------+--------+-------+

4 rows in set (0.001 sec)

Constraints: is used to define rules to allow or restrict what values can be stoted in columns.

Column level constraints:These constraints are applied only to the single column that limits the type of particular column data.

Table level constraints: These constraints are applied to the entire table that limits the type of data for the whole table.

Create table tablename(col1 datatype constraint,col2......);

Create table tablename ( col1 datatype ........ constraints);

Various constraints: not null or null, auto\_increment, default, unique ,check,

Primary key ,foreign key.....

Not null constraint: This constraint forces the values of a column non-null values only.

alter table stud change sid sid int(15)not null;

Query OK, 0 rows affected (0.068 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [colle]> describe stud;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| sid | int(15) | NO | | NULL | |

| name | varchar(10) | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

2 rows in set (0.038 sec)

**Unique constraint**: This constraint forces the column to have unique value only .

Syntax:create table tname(col1 datatype(size) constraint,......);

**create table e2(eid int(2),ename varchar(10),sal int, constraint u1 unique(eid,ename));**

Query OK, 0 rows affected (0.042 sec)

MariaDB [colle]> describe e2;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| eid | int(2) | YES | MUL | NULL | |

| ename | varchar(10) | YES | | NULL | |

| sal | int(11) | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

3 rows in set (0.020 sec)

MariaDB [colle]> insert into e1 values(2,'rrrrr');

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> insert into e1 values(2,'rrrrr');

ERROR 1062 (23000): Duplicate entry '2' for key 'eid'

MariaDB [colle]> insert into e1 values(3,'rrrrr');

Query OK, 1 row affected (0.001 sec)

MariaDB [colle]> select \* from e1;

+------+-------+

| eid | ename |

+------+-------+

| 2 | rrrrr |

| 3 | rrrrr |

+------+-------+

2 rows in set (0.000 sec)

MariaDB [colle]> insert into e2 values(1,'eeee',45500);

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> insert into e2 values(1,'eeee',45500);

ERROR 1062 (23000): Duplicate entry '1-eeee' for key 'u1'

MariaDB [colle]> insert into e2 values(2,'eeee',45500);

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> insert into e2 values(2,'pppp',45500);

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> insert into e2 values(2,'pppp',45500);

ERROR 1062 (23000): Duplicate entry '2-pppp' for key 'u1'

MariaDB [colle]> select \*from e2;

+------+-------+-------+

| eid | ename | sal |

+------+-------+-------+

| 1 | eeee | 45500 |

| 2 | eeee | 45500 |

| 2 | pppp | 45500 |

+------+-------+-------+

3 rows in set (0.000 sec)

**Create table stud2(sid int(4) unique ,sname varchar(10));**

Query OK, 0 rows affected (0.039 sec)

MariaDB [colle]> describe stud2;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| sid | int(4) | YES | UNI | NULL | |

| sname | varchar(10) | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

2 rows in set (0.035 sec)

MariaDB [colle]> insert into stud2 values(11,'dddd');

Query OK, 1 row affected (0.002 sec)

MariaDB [colle]> insert into stud2 values(11,'dddd');

ERROR 1062 (23000): Duplicate entry '11' for key 'sid'

MariaDB [colle]> insert into stud2 values(12,'dddd');

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> select \* from stud2;

+------+-------+

| sid | sname |

+------+-------+

| 11 | dddd |

| 12 | dddd |

+------+-------+

**Default constraint**:The constraint is used to insert a default value into a column.

Create table tname (col1 datatype ,col2 datatype defaultvalue);

MariaDB [colle]> create table s1(eid int ,ename varchar(10),sal int default 2000);

Query OK, 0 rows affected (0.034 sec)

MariaDB [colle]> describe s1;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| eid | int(11) | YES | | NULL | |

| ename | varchar(10) | YES | | NULL | |

| sal | int(11) | YES | | 2000 | |

+-------+-------------+------+-----+---------+-------+

Syntax: alter table tname salary default 3000;

insert into s1(eid,ename) values(101,'pppp');

Query OK, 1 row affected (0.002 sec)

MariaDB [colle]> select \* from s1;

+------+-------+-------+

| eid | ename | sal |

+------+-------+-------+

| 101 | pppp | 60000 |

| 101 | pppp | 2000 |

+------+-------+-------+

2 rows in set (0.000 sec)

alter table stud2 alter sname set default 'nirmaan';

Query OK, 0 rows affected (0.020 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [colle]> describe stud2;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| sid | int(4) | YES | UNI | NULL | |

| sname | varchar(10) | YES | | nirmaan | |

+-------+-------------+------+-----+---------+-------+

2 rows in set (0.023 sec)

MariaDB [colle]> insert into stud2(sid) values(23);

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> select \* from stud2;

+------+---------+

| sid | sname |

+------+---------+

| 11 | dddd |

| 12 | dddd |

| 23 | nirmaan |

+------+---------+

3 rows in set (0.000 se

**Auto increment constraint**: This constraint allows a unique number to be generated when a new record is inserted into a table.

By default the starting value for auto increment is 1. And it will increment by 1 for each new record.

It is used to create sequence in mysql.

Create table s3(col1 datatype ,...auto-increment primary key,....);

create table s2(sid int auto\_increment primary key, sname varchar(10));

Query OK, 0 rows affected (0.043 sec)

MariaDB [colle]> describe s2;

+-------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+----------------+

| sid | int(11) | NO | PRI | NULL | auto\_increment |

| sname | varchar(10) | YES | | NULL | |

+-------+-------------+------+-----+---------+----------------+

2 rows in set (0.018 sec)

MariaDB [colle]> insert into s2(sname) value('rrr');

Query OK, 1 row affected (0.002 sec)

MariaDB [colle]> select \* from s2;

+-----+-------+

| sid | sname |

+-----+-------+

| 1 | rrr |

+-----+-------+

1 row in set (0.001 sec)

MariaDB [colle]> insert into s2(sname) value('pppp');

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> select \* from s2;

+-----+-------+

| sid | sname |

+-----+-------+

| 1 | rrr |

| 2 | pppp |

+-----+-------+

2 rows in set (0.000 sec)

Check constraint: It constraint that a certain column must satisfy the condition (Boolean expression)

Syntax: colname datatype check(condition)

Mysql doesn’t support checkkeyword .

There are two ways to satisfy this , using tiggers or using view with check option .

**Primary key and foreign key** :

Primary key: It is column or combination of column that uniquely identifies a row in table

Rules:primary key must contain unique values.

Primary key cant be null.

There is only one primary key in the table.

Rollno🡪pk; mobile no🡪(candidate key)or (can be said alternat key)

create table s3(sid int primary key,sname varchar(10));

Query OK, 0 rows affected (0.024 sec)

Emp (eid(pk),ename,sal);

Mgr(exp,loc,eid,did)

Dept(did(pk),dname,eid(fk))

MariaDB [colle]> describe s3;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| sid | int(11) | NO | PRI | NULL | |

| sname | varchar(10) | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

2 rows in set (0.017 sec)

MariaDB [colle]> insert into s3 values(45,'yyy');

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> insert into s3 values(45,'yyy');

ERROR 1062 (23000): Duplicate entry '45' for key 'PRIMARY'

MariaDB [colle]> insert into s3 values(46,'yyy');

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> select \* from s3;

+-----+-------+

| sid | sname |

+-----+-------+

| 45 | yyy |

| 46 | yyy |

+-----+-------+

2 rows in set (0.001 sec

Foreign key: It is a field in a table that matches another field of another table . it is used to maintain referential integrity .

A table may have more than one foreign key and each foreign key in the child table may refer to a different parent table.

Eid(pk) ename esal did(fk) did(pk) dname

12 203 201

13 504 203

14 201 504

15

Student:( sid,sname,smarks,branch,course,bid(fk))

Course(cname,branches,bid)🡪btech 🡪(ece,civil,mech,it,cse,eee,ae,agr)1,2,3,4,5,6,7,8

or mtech🡪

1 sss,45,2,btech,

Promotion:

create table e1(eid int primary key) engine=InnoDB;

Query OK, 0 rows affected (0.040 sec)parent table.

create table d1(did int,foreign key(did)references e1(eid)) engine=InnoDB;

Query OK, 0 rows affected (0.035 sec)(child table)

insert into e1 values(12);

Query OK, 1 row affected (0.006 sec)

MariaDB [colle]> insert into e1 values(13);

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> insert into e1 values(14);

Query OK, 1 row affected (0.008 sec)

MariaDB [colle]> insert into d1 values(12);

Query OK, 1 row affected (0.010 sec)

MariaDB [colle]> insert into d1 values(16);

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`colle`.`d1`, CONSTRAINT `d1\_ibfk\_1` FOREIGN KEY (`did`) REFERENCES `e1` (`eid`))

Other table for foreign key :

create table e2(eid int primary key,ename varchar(10)) engine=InnoDB;

Query OK, 0 rows affected (0.049 sec)

MariaDB [colle]> create table d2(did int ,dname varchar(10), eid int,foreign key(eid) references e2(eid))engine=InnoDB;

Query OK, 0 rows affected (0.033 sec)

MariaDB [colle]> insert into e2 values(1,'priya');

Query OK, 1 row affected (0.003 sec)

MariaDB [colle]> insert into e2 values(2,'sam');

Query OK, 1 row affected (0.008 sec)

MariaDB [colle]> insert into e2 values(3,'jyothi');

Query OK, 1 row affected (0.002 sec)

MariaDB [colle]> insert into e2 values(4,'ramesh');

Query OK, 1 row affected (0.003 sec)

MariaDB [colle]> insert into d2 values(10,'cs',3);

Query OK, 1 row affected (0.012 sec)

MariaDB [colle]> insert into d2 values(11,'eee',6);

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`colle`.`d2`, CONSTRAINT `d2\_ibfk\_1` FOREIGN KEY (`eid`) REFERENCES `e2` (`eid`))

Eid exp

Alter table: It is used to change existing table structure such as adding or removing column changeing column attributes...

View:A view is a database object that has no values.its contents are based on the base table. It contains rows and columns similar to the real table. The view is a virtual table created by a query by joining one or more tables.

It is operated similarly to the base table but does not contain any data of its own.

Syntax:create view viewname as select columns from tables where conditions;

MariaDB [colle]> create view stud\_view as select sid,branch, marks from student where branch='eee';

Query OK, 0 rows affected (0.025 sec)

MariaDB [colle]> select \* from stud\_view;

+------+--------+-------+

| sid | branch | marks |

+------+--------+-------+

| 15 | eee | 467 |

| 15 | eee | 467 |

| 19 | eee | 478 |

+------+--------+-------+

3 rows in set (0.003 sec)

Update view:

The alter view statement is used to modify or update the already created view without dropping it.

MariaDB [colle]> insert into student values(32,'priya','eee',498);

Query OK, 1 row affected (0.009 sec)

MariaDB [colle]> select \* from student;

+------+------------+--------+-------+

| sid | sname | branch | marks |

+------+------------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 16 | sunil | cse | 467 |

| 19 | kirankumar | eee | 478 |

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

| 34 | sangeetha | b.tec | 579 |

| 35 | anitha | it | 657 |

| 13 | NULL | cse | 589 |

| 13 | NULL | eee | 589 |

| 32 | priya | eee | 498 |

+------+------------+--------+-------+

13 rows in set (0.000 sec)

MariaDB [colle]> select \* from stud\_view;

+------+--------+-------+

| sid | branch | marks |

+------+--------+-------+

| 19 | eee | 478 |

| 13 | eee | 589 |

| 32 | eee | 498 |

+------+--------+-------+

3 rows in set (0.001 sec)

MariaDB [colle]> update stud\_view set marks=480 where sid=19;

Query OK, 1 row affected (0.010 sec)

Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [colle]> select \* from stud\_view;

+------+--------+-------+

| sid | branch | marks |

+------+--------+-------+

| 19 | eee | 480 |

| 13 | eee | 589 |

| 32 | eee | 498 |

+------+--------+-------+

3 rows in set (0.000 sec)

MariaDB [colle]> select \* from student;

+------+------------+--------+-------+

| sid | sname | branch | marks |

+------+------------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 16 | sunil | cse | 467 |

| 19 | kirankumar | eee | 480 |

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

| 34 | sangeetha | b.tec | 579 |

| 35 | anitha | it | 657 |

| 13 | NULL | cse | 589 |

| 13 | NULL | eee | 589 |

| 32 | priya | eee | 498 |

+------+------------+--------+-------+

13 rows in set (0.001 sec)

Syntax: alter view\_name as select col from tablename where conditions;

Insert in the view:

insert into stud\_view values(13,'cse',589);

Query OK, 1 row affected (0.016 sec)

MariaDB [colle]> select \* from stud\_view;

+------+--------+-------+

| sid | branch | marks |

+------+--------+-------+

| 15 | eee | 467 |

| 15 | eee | 467 |

| 19 | eee | 478 |

+------+--------+-------+

3 rows in set (0.001 sec)

MariaDB [colle]> select \* from student;

+------+------------+--------+-------+

| sid | sname | branch | marks |

+------+------------+--------+-------+

| 12 | sairam | B.tec | 456 |

| 13 | sunitha | ece | NULL |

| 13 | sunitha | ece | 357 |

| 14 | anitha | it | 567 |

| 15 | ramesh | eee | 467 |

| 15 | vijay | eee | 467 |

| 16 | sunil | cse | 467 |

| 19 | kirankumar | eee | 478 |

| 17 | varma | cse | 476 |

| 18 | varma | it | 543 |

| 34 | sangeetha | b.tec | 579 |

| 35 | anitha | it | 657 |

| 13 | NULL | cse | 589 |

+------+------------+--------+-------+

13 rows in set (0.001 sec)

MariaDB [colle]> insert into stud\_view values(13,'eee',589);

Query OK, 1 row affected (0.004 sec)

MariaDB [colle]> select \* from stud\_view;

+------+--------+-------+

| sid | branch | marks |

+------+--------+-------+

| 15 | eee | 467 |

| 15 | eee | 467 |

| 19 | eee | 478 |

| 13 | eee | 589 |

+------+--------+-------+

4 rows in set (0.000 sec)

Index: A database index is a data structure that improves the speed of operations in a table .indexes can created using one or more columns, providing the basis for both rapid random lookups and efficient ordering of access to records.

A index helps in fast retrieval of data from tables. It is just like a index of book where you first search index to look for content.

🡪All primary key columns are in the primary index of the table automatically.

Syntax:

Create [unique|fulltext|spatial ] index index\_name using [btree|hash|rtree] on table\_name(colname [(length)][asc/desc]...)

Or create index index\_name on tablename (colname);

System🡪request🡪server

js 🡨 response php

Client 🡪 server

PHP(personal home page)

PHP:php is an acronym for “php” HYPERTEXT PREPROCESSOR”. Php is a widely used open source scripting language. Php scripts are executed on the server. Php is free to download and use.

It is powerful enough to be the biggest blogging system on the web(word press). It is deep enough to run the largest social network (facebook). It is also enough to be a beginners first server

🡪easy to learn

🡪supports differet types of db

🡪cross platform

🡪supports multiple server types

Php files can contain text,html,css, javascript and php code.

Php code is executed on the server and the result is returned to the browser as plain html.

Php files have extension .php

Php can generate dynamic page content

Php can collect form data

Php can send and receive cookies

Php can create open, read, write delete,and close files on the server.

Php can add, delete ,modify data in your database.

Php can be used to control user access.

Variable is a name given to a memory location which acts as container for storing data

Eg: $a=34

Constants are fixed values that do not change during execution time .

Define (‘pi’,3.14159)

Primitive datat types:Integers, strings ,Boolean ,float, undefined (?).

Reference data types:Arrays :An Array is a data structure that contains a list of elements. These elements are all of the same datatype such as integer or string.

$chocolates= array(‘eclairs’,’kitkat’,’mars’);

Variables: it should start with $ symbol ,eg:$num=908;

$name=”php online”;

It is case sensitive , $age 🡪 not $AGE;

$first\_name

$first name🡪 not correct

Comments : single line comments //

Multiple line comments /\* ffffff \*/

And single line comments #

Constant variables:

Define(name,value,caseinsensitive) default(false) but two value (true and false).

Cant use $sign with constant variable name.

Constant variable are global variables.

Arithmetic operators: +,-,\*,/,%,\*\*,++,-- $a\*\*$b a=3 b=5 35=243

Assignment operators:=,+=,-=,\*=,/=,%=,\*\*=,

Comparison operators: ==,===, != or <>, !==,>,<,>=,<=,

Conditional statements and control statements(loops):php supports a number of different control statements(structure).

If,else, elseif,switch,while,do\_while, for and foreach ...

Conditional statements :

If conditional statements :if the condition is true ,then executes true block statements, otherwise comes out the loop.

Syntax:if(condition or expression)

{

Statements

}

If else : if else statement :if the condition is true it executes true block else executes false block .

Syntax:

If(condition or expression)

{

Statements( true block)

}

Else{

Statements(false block)

}

If --- else --- if condition: if the condition is true, it executes true block statements then if condition should check again for another condition then it is called nested if condition.

/\*If()

{}

Elseif(){

} elseif(){}\*/

If()

{

If() // true block;

{

True block for second if

}

Else()

{

False block for second else part

}

}else {

Statements( false block for the first if loop)

}

**Conditional ternary:**

Condition?true statements :false statements

($a>$b) ? “a is >”:”a is <”;

Switch :The switch statement is similar to the else if statement where we have executes one block if the condition is true, in the case of elseif statement we have a set of different conditions ,it is similar to elseif .action will be executed based on a condition .

In the switch condition is checks than it goes to that case and executes that block then comes out of the switch block.

Switch(condition)

{

Case 1:

Statements;

Break;

Case 2:statements;

Break;

Case 3: statements;

Break;

..

Default:statements

Break;

Control loops:Loops in php are useful to execute a piece of code repeatedly until a true.(block of code a specified number of times).

Syntax: inti

While(condition)

{

Statements;

Incre/decr;

}

Do—while:The do while loop is similar to while loop with the only difference that is its condition is checked at the end of the first iteration.so the loop code is executed at least once , irrespective of the condition(true or false).

Syntax:

Inti;

Do{

Statements;

Incre/decr;

}while(condition);

For loop:loops through a block of code a specified number of times.

For(initialization;condition;incre/decr)

{

Statements

}

Foreach loop: The foreach statement is used to loop through arrays. For each pass the value of the current array element is assigned to $value and the array pointer is moved by one and in the next pass next element will be processed.

Syntax: foreach(array as value)

{

Statements;

}

Break statement:The php break keyword is used to terminate the execution of a loop prematurely.

Continue statement: The php continue keyword is used to halt the current iteration of a loop but it does not terminate the loop. For the pass encountering continue statement , rest of the loop code is skipped and next pass starts.

Goto statement:The goto statement is used to send flow of the program to a certain location in the code. The locationis specified by a user defined label.

Syntax:

Statements

If(condition)

Goto label1;

Statements

Goto label2;

Label1: statements

Statements;

Label2: statements;

Functions : Php functions are similar to other programming languages. A function is a piece of code which takes input in the form of parameter and does some processing and returns a value.

Creating or defining a php function

Calling a php function

Function functionname()

{

Executable statements;

}

Functionname();🡪calling function

Function functionname(parameters)

{

}

Functionname(parameters values);🡪 calling function

Array: An array is a data structure that stores one or more similar type of values in a single value(variable).

There are three different kind of arrays and each array value is accessed using an id which is called array index.

1. Numeric array: An array with a numeric index . values are stored and accessed in linear fashion.

$a=array(10,20,30);

Foreach($a as $value)

{

echo $value;

}

$x[0]=56;

$x[1]=78;

2)Associative array:The associative arrays are very similar to numeric arrays in terms of functionality but they are different in terms of their index .Associative array will have their index as string so that you can establish a strong association between key and values.

$name=array(“studname”=>’aaa’,”rollno”=>34,”fee”=>4500);

Echo $name[‘studname’];

3)Multi-dimensional arrays:A multi-dimensional array each element in the main array can also be an array. And each element in the sub array can be an array,...values in the multideimensional array are accessed using multiple index.

$stud= array(“priya”=>array(“maths”=>45,“Science”=>67,”social”=>86),

“sam”=>array(“maths”=>45,“Science”=>67,”social”=>86),

“jyothi”=>array(“maths”=>45,“Science”=>67,”social”=>86)

);

Echo $stud[priya][maths];

Echo $stud[priya][science];

Print\_r() function prints the information about a variable in a more readable way.

Print\_r(variable, return)(return option)

List:The list() function is used to assign values to a list of variables in one operation.

List(var1,var2,........)

Array methods:

Count: counts all the value of an array.size of array.

Sizeof : it is similar to count() function.

Syntax: echo count($arrayname);

Echo sizeof($arrayname);

Array\_count\_values($arrayname): it compares the no. Of values in the array.

Eg: $food= array('Orange','banana','apple','banana','grapes','Orange');

// This function doesn't return any value it return one Array so use print\_r

//echo array\_count\_values($food);

echo "<pre>";

print\_r(array\_count\_values($food));

echo "</pre>";

Array

(

[Orange] => 2

[banana] => 2

[apple] => 1

[grapes] => 1

)

In\_array(‘value’,$arrayname):

if the values is present in the array it gives out put 1 other wise 0.

Array :specifies the array

Mode : specifies the mode possible values:

0:default ,doesnot count all elements of multidimensional array.

1: ( ture) : it checks case sensitive.

Local and global variables:The scope of a variable is the context within which it is defined.

$x=1;global variable

Function text()

{

Echo $x;

}

Text();

$x variable will be available with the included

$x=7;

Function text()

{

Global $x;

Echo $x;

}

Text();

PHP Super global variables:some predefined variable in PHP are “superglobals” which means that they are always accessible, regardless of scope and You can access them from any function, class or file without having to do anything special.

$\_SERVER

$\_REQUEST

$\_POST

$\_GET

$\_SESSION

$\_FILES

$\_COOKIES

File1 🡪 file2

$a=90; echo$a;

$\_POST:is a php super global variable which is used to collect form data after submitting an html form with the method “post: $\_POST is used to pass variables.

$\_GET:is a php super global variable which is used to collect form data after submitting an html form with method=”get”

$\_SERVER: is a PHP super global variable which holds information about headers ,paths, and script locations.

$\_SERVER[‘SERVER\_NAME];

$\_SERVER[‘SERVER\_HOST];

Php\_self,server\_addr,server\_name,server\_software,server\_protocol,gateway\_interface,server\_method,request\_time,query\_string,http\_accept,http\_accept\_charset,http\_host,http\_referer,https,remote\_addr,script\_filename,server\_admin,remote\_host,server\_port,script\_url...etc...,

Name:priya -🡪database🡪server

Age:30 print

submit

$\_SESSION[]:

Request🡪server🡪sessionstart()🡪message ur session expired u have login again.

Object🡪service🡪destroy

Three steps to set and get session value:

Step 1: session start();

Step 2: $\_session[‘name’]=value;

Step 3: session\_destroy();

Detele session:

Session\_unset();

Remove all session variables

Session\_destroy();

After create the session destroy the session but first unset all then destroy.

Cookies:A cookies is often used to identify a user. A cookies is a small file that the server embeds on the user’s computer. Each time the same computer requests a page with a browser, it will send the cookie too. With php ,you can both create and retrieve cookie values.

1)To create cookie:

setCookie(name,value,expire,path,domain,secure,httponly);

Name: it is used to set the name of the cookie.

Value: it is used to set the value of the cookie.

Expire: it is used to set the expiry timestamp of the cookie after which the cookie can’t be accessed.

Path : It is used to specify the path on the server for which the cookie will be available,

Domain : it is used to specify the domain for which the cookie is available.

Security: It is used to indicate that the cookie should be sent only if a secure https connection exists

2)view cookie value

$\_COOKIE[name];

<input type=”file” name=”aaa” >

$\_FILE:

“Require\_once “ffff.php”

Include jjkljkljk.php

System

(system)

Internet

server

Database

CRUD Operations:CRUD create, Read,Update,and Delete.

Crud operations are basic data manipulation for database.

Three steps:

1.connection

2.run sql query

3.close connection

1)Mysqli \_connect(serverName,username,password,databasename)

2)run sql query;

3)close connection

Mysqli \_close(connectionname);

OOPs :object oriented programming

* **Class** − This is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.
* **Object** − An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.
* **Member Variable** − These are the variables defined inside a class. This data will be invisible to the outside of the class and can be accessed via member functions. These variables are called attribute of the object once an object is created.
* **Member function** − These are the function defined inside a class and are used to access object data.
* **Inheritance** − When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.
* **Parent class** − A class that is inherited from by another class. This is also called a base class or super class.
* **Child Class** − A class that inherits from another class. This is also called a subclass or derived class.
* **Polymorphism** − This is an object oriented concept where same function can be used for different purposes. For example function name will remain same but it take different number of arguments and can do different task.
* **Overloading** − a type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly functions can also be overloaded with different implementation.
* **Data Abstraction** − Any representation of data in which the implementation details are hidden (abstracted).
* **Encapsulation** − refers to a concept where we encapsulate all the data and member functions together to form an object.
* **Constructor** − refers to a special type of function which will be called automatically whenever there is an object formation from a class.
* **Destructor** − refers to a special type of function which will be called automatically whenever an object is deleted or goes out of scope.

Class🡪methods and properties.

Class employee{

Public $eid;

Function sum(){

Echo $this->eid;

}

}

$obj=new employee();

$obj->sum();

Car: color student:name,roll,sec,student(){total=s1+s2;}avg(){avg=total/2}

Engine student dd,a;

Seats dd.name;dd.roll, student() ;a.name,a.roll

Ac

price

method1(){

miles=j\*r

}

blueprint

<?php

class person{

public $name,$age;

function \_\_construct($name,$age){

$this->name =$name;

$this->age=$age;

}

function show(){

echo "ur name:".$this->name ."-".$this->age;

}

}

$p1 = new person("priya online",23);

$p1->show();

?>

<?php

class person{

public $name ="no name";

public $age;

function show(){

echo " your Name :".$this->name;

echo " your age: ".$this->age;

}

}

$p1= new person();

//$p1->name="priya online";

$p1->age=34;

$p1->show();

?>

Constructor

<?php

class person{

public $name,$age;

function \_\_construct($name,$age){

$this->name =$name;

$this->age=$age;

}

function show(){

echo "ur name:".$this->name ."-".$this->age;

}

}

$p1 = new person("priya online",23);

$p1->show();

?>:

INTERFACE

<?php  
interface InterfaceName {  
  public function someMethod1();  
  public function someMethod2($name, $color);  
  public function someMethod3() : string;  
}  
?>

+

Angular: Angular is a platform that makes its easy to develop application with the web.

Angular is a platform and framework for building single page client applications using html and typescript.

Angular is a single page application

Angular is written in typescript. It implements core and optional functionality as a set of typescript libraries that you import into your apps.

Angular components that are organized into NgModules. NgModules collect related code into functional sets.

Components define views ,which are sets of screen elements that angular can choose among and modify according to your program logic and data.

Components use services, which provide specific functionality not directly related to views. Service provider can be injected into components as dependenices making your code modular, reusable , and efficient.

Modules, components and services are classes that use decorators. These decorators mark their type and provides metadata that tells angular how to use them.

Nodejs:Nodejs is an open source ,crossplatform from javascript runtime environment.

Npm is a node package manager for javascript programming language it is automatically installed with nodejs.

Typescript need to install it using command

Npm install -g typescript

Npm install –g @angular/cli (command line interface) is a tool to that allows us to create a project build and run it on development server directly using command line.

Module:default module : app.module

This is compulsory

@NgModule;@-->decorator

App.component.ts

App.component.html

App.component.css

App.component.spec.ts

Decorators: are a feature of typescript and implemented as functions.The name of the decorator starts with @ symbol following by bracket and arguments .

Decorators are simply function that returns function these function supply metadata to angular about a particular class, property, value or method,,

Decorators are invoked at runtime.

Decorators allow you to execute functions eg:@component execute the component function imported from angular.

Some common decorators:

@NgModule()🡪 to define modules.

@component()🡪to define components

@injectable() to define services.

@input() & @output() to define properties that send & receive data from the dom.

Types of decorators:

Class decorator:

@component and @NgModule

Property decorator : for properties inside classes @input() & @output

Method decorators : for methods inside classes @HostListener

Parameter decorators: for parameters inside class ,constructors @inject

Each decorator has a unique role.

Angular Router: is an official angular router library, written and maintained by the angular core .

First of all m angular router takes care of the duites of a javascript router.

It activates all required angular components to compose a page when a user navigates to certain url.

Users navigate from onepage to another without page reload.

Component: Every Angular application has atleast one component , the root component that connects, a component hierarchy with page document object model(DOM) . Each component defines a class that contains application data and logic and is associated with an html template that defines a view to be displayed in a target environment.

Binding:

{{expression}}🡪 interploation

Property binding: [target]=”expression”;

Event binding: (target)=”statement”;

Two way binding:[(target)]=”expression”;

Property binding: to set a property of a view element.

Rendering value from data source to template

Property binding as one-way data binding

Data souce (component)🡪 template( one way binding)

[target]=’expression’;

Event binding: Event binding app respond to user input in the target environment by updating your application data.

(target)=’statement’;

Two way Binding:[(target)]=’expression’;

Two way binding gives components in your application away to share data .

Use two way binding to listen for events and update values .

Angular two way binding syntax is a combination of square brackets and parenthesis.[()].i.e(property binding and event binding binding).

Data binding🡪1)one way binding 2) two way binding

1. One way binding 🡪 1)interpolation binding, style binding, class binding property binding 2) event binding

Interpolation is a technique that allows the user to bind a valueto a UI element .

{{value}}

Directives:Directives are elements which change the appearance or behaviour of the DOM element. These are 3 types of directives

1. Components 1) directives with own template
2. Structural :change DOM by adding removing elements
3. Attribute: change appearance or behaviour of the DOM.
4. Structural Directives: are responsible for the html layout. The shape or reshape the html view by simply adding or removing the elements in the DOM .These directives are the way to handle how the component or the element renders in a template.

1)ngIf(\*ngif)

2)ngFor(\*ngFor)

3)ngSwitch(\*ngSwitch)

1) ngIf :conditionally add or remove an element from the DOM

2) ngFor: Repeat a template for each item in a list.

3)ngSwitch: A set of directives that switch among alternative views

Attribut directives:is away to modify the appearance of the DOM element or component .

There are 2 built in attribute directives in angular.

ngStyle: angular provides a built inn ngstyle attribute to modify the element appearance and hebavior.

Ngclass: this attribute is used to change the class attribute of the element in DOM or the component to which it has been attached.

Pipes:

Angular pipes can be used to transform data to desired output

Pipes takes in a data input and transforms data to a different output

Using the pipe operator (|) we can apply the pipes features to any of the property in our angular project.

Pipes ( | ) in angular are used to transform the data before displaying it in a browser Angular provides a lot of built in pipes to translate the data before displaying it into the browser and as we know, angular lets us extend its feature , we can even create custom pipes in angular.

Input date 🡪 filter 🡪 final output

Types of pipes:

Pipes🡪 1)built in pipes 2) custom pipes

1. Built in pipes🡪 parameterized , chaining

Custom pipes :

To create a pipe in angular ,you have to apply the @pipe decorator to class, which we can import form the core angular library.

The @pipe decorator allows you to define the pipe name that you’ll use with in template expressions

Syntax:

Import { pipe, pipeTransform } form ‘@angular/core’;

App.component.html

<h3> Custome pipes </h3>

<table border="1">

  <tr \*ngFor="let emp of emp ">

    <td>{{emp.code}}</td>

    <td>{{emp.name|mypipe1:emp.gender}}</td>

    <td>{{emp.gender}}</td>

    <td>{{emp.salary}}</td>

  </tr>

</table>

<app-test></app-test>

<router-outlet></router-outlet>

Mypipe.pipe.ts

transform(value: string,gender: string): string {

    if(gender.toLowerCase()=="male")

    return "Mr."+value;

    else

    return "Mrs."+value;

  }

App.component.ts

emp :any=[

    {code:'01',name:'priya',gender:'fenmmale',salary:50000},

    {code:'02',name:'kranthi',gender:'male',salary:50000},

    {code:'03',name:'eshwar',gender:'male',salary:50000}

  ];

**Service**: Service are a piece of code that are used to perform a specific task, a service can contain a value or function or combinations of both. Services are injected into application using dependency injection mechanism. Services prevent us from writing the same code at multiple sections of our app’n . serices provide store, and interact with data and a communication b/w classes. Service is a mechanism used to share the functionality b/w the components.

You can register a service

🡪Root level (any where)

🡪Module level ( any component)

🡪Component level ( with in the component [same component])

Step:1 create the service file

>ng generate service servicefilename

@Injectable()

export class NewserviceService {

news = [

  {id: 1,title:'news title',description: 'This is news description'},

  {id: 1,title:'news title',description: 'This is news description'},

  {id: 1,title:'news title',description: 'This is news description'},

  {id: 1,title:'news title',description: 'This is news description'}

];

  constructor() { }

  getNews() {

    return this.news;

  }

}

import { Component, OnInit } from '@angular/core';

import {  NewserviceService } from '../newservice.service';

@Component({

  selector: 'app-test1',

  templateUrl: './test1.component.html',

  styleUrls: ['./test1.component.css'],

  providers: [NewserviceService]

})

export class Test1Component implements OnInit {

  news:{};

  constructor(newservice: NewserviceService) {

    this.news = newservice.getNews();

   }

  ngOnInit(): void {

  }

}

<p>test1 works!</p>

{{news | json}}

import { Component, OnInit } from '@angular/core';

/\*import {  NewserviceService } from '../newservice.service';\*/

@Component({

  selector: 'app-test2',

  templateUrl: './test2.component.html',

  styleUrls: ['./test2.component.css']

})

export class Test2Component implements OnInit {

 /\* news:{};

  constructor(newservice: NewserviceService) {

    this.news = newservice.getNews();

   }

\*/

  ngOnInit(): void {

  }

}

{{news | json}}

@Input property:pass data from parent to child

Parent component🡪child component(@input property)

Child component🡪 parent component(@output property)

Create Reactive form:

There are more scalable,resusable and testable.

A form control instance that tracks the valueand validation status of an individual form control.

Form group: A form group instance that tracks the same values and status for a collection of form controls.

Nested form groups: when building complex forms, managing the different areas of information is easier in smaller sections.

Routing : Routing in angular helps us navigate from one view to another as users perform tasks in web apps.

The Router Link directive helps you customize the anchor element. In this case, the route, or url contains one fixed segment,/products. The final segment is variable ,inserting the id property of the current product.

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { DepartmentComponent } from './department/department.component';

import { EmployeeComponent } from './employee/employee.component';

import { PagenotfoundComponent } from './pagenotfound/pagenotfound.component';

const routes: Routes = [

  {path: '', redirectTo:'/departments',pathMatch:'full'},

  {path: 'departments', component: DepartmentComponent},

  {path: 'employees', component: EmployeeComponent},

  {path: "\*\*",component: PagenotfoundComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

export const routingcomponents = [DepartmentComponent,EmployeeComponent]

<div style="text-align: center;"> <h3>Routing Navigation</h3></div>

<nav>

  <a routerLink="/departments" routerLinkActive="active">Departments</a>

  <a routerLink="/employees" routerLinkActive="active">Employees</a>

</nav>

<router-outlet></router-outlet>

GIT INTRODCUTION:

Git is an open source ,version control tool created

Untracked unmodified modified staged

Add the file------------------------------------------------------🡪(1)

Edit the file(3)

-------------------🡪

----------------🡪stage(4)

Commit(2)

🡨-----------------------------------------