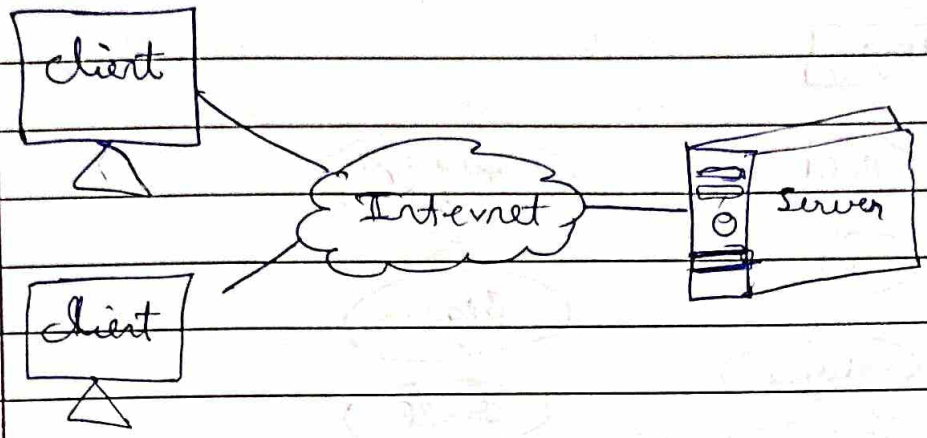
HTML

(Tag Based Language)




```
<html>
        <head>
        <title>
        </title>
        ~~<head>~~ </head>

<body>
  <h1>        </h1>
</body>
</html>
```

```
<center>
_____
_____
_____
_____
_____
_____
_____
</center>
```

```
<hr/>
```

Heading	→ line
FirstPage	

```
<p>
}  →  ..........
</p>
```

```
<pre>
}
</pre>
```

Attribute

JAY PRAKASH Date_____ Page_____

<body bgcolor = "tan">

<h1 align = "right" > This is a heading </h1>

<h1 align = "left" ></h1>
<h2 align = "center" > </h2>
<h3 align = "right > </h3>

Formatting Tag

- Bold
- Italic
- Underline
- Subscript (H_2O)
- Superscript (10^2)
- Big
- Small
- Monospeed

<u> First Text </u>

LIFO → Last In first out

<strike> This is strike </strike>

Subscript H ₂ O (H_2O)

Superscript 10 ³ (10^3)

Small : This is <small> Small Text </small>

Big : This is <big> Big Text </big>

Phrase Tag

- Emphasis
- Address
- Market Text
- Abbreviation
- Strong
- Quotation

<P> welcome to <attribute = "Tutorial Point">TP
</abbr></P>

Meta Tag

Meta name ⟶ keywords

<meta name = "author" Content = "Vijay Sharma"/>

Comments

<!-- ---->

Images

<img src = " " alt = " "
border = "0" align = "left"/>

Table

```
<table>
    <tr>
        <td> Row1, Column1 </td>
        <td> Row2, Column1 </td>
    </tr>
    <tr>
        <td> Row2, Column1 </td>
        <td> Row2, Column2 </td>
    </tr>
</table>
<table>
    <tr>
        <th> Name </th>
        <th> Salary </th>
    </tr>
</table>
```

head	Name	Salary
Body	60	20
	320	40

List

PRAKASH

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

1) Unordered list

```
<html>
  <head>
    <title> ~~~~ </title>
  </head>
  <body>
    <h3> Web development </h3>
    <hr align="left" width=50% "/>
    <ul>   <ul type="circle">
      <li> ASP.NET </li>
      <li> EJB </li>
      <li> PHP </li>
      <li> MVC </li>
    </ul>
  </body>
</html>
```

- ASP.NET
- EJB
- PHP
- MVC

2) Ordered list

```
<ol>   <ol type="i">
  <li> ASP.NET </li>
  <li> EJB </li>
  <li> PHP </li>
  <li> MVC </li>
</ol>
```

i. ASP.NET
ii. EJB
iii. PHP
iv. MVC

```
<ol type="a" start="d">
```

d.
e.
f.
g.

HTML

Hyperlink

JAY PRAKASH

Date

Page

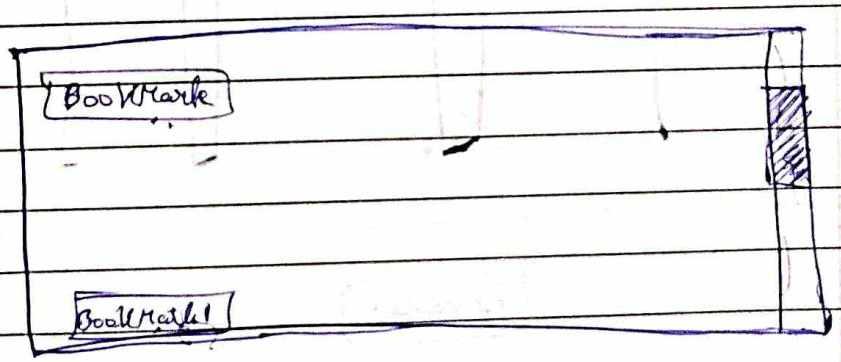
```
<a href = "                    "> Tutorial Point </a>
```

```
< a href = "bookmark" > </a>
```

```
<br>
<br>
<br>
<br>
```

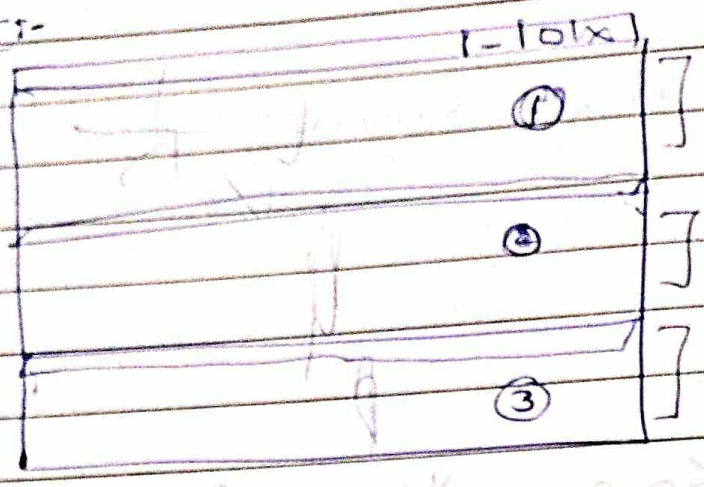
```
<a name = "bookmark" > BOOKMARK 1 </a>
```

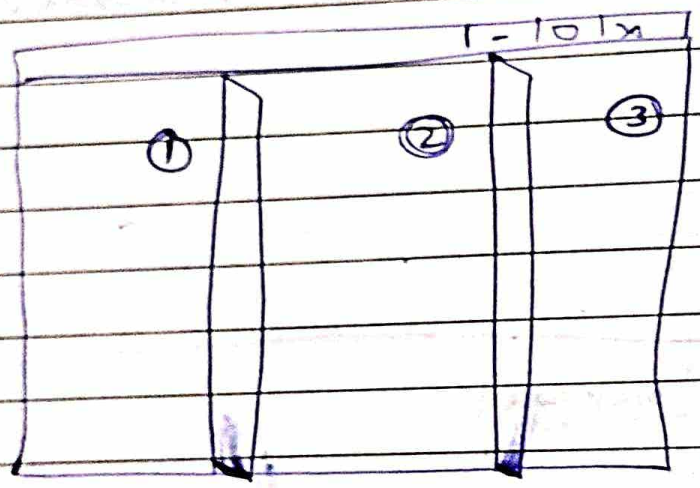
```
<br>
<br>
<br>
<br>
```




```
<html>
    <head>
        <title> HTML Frames </title>
    </head>
<body> <frameset rows = 25%, 50%, 25%" >
        <frame name = "top" src = "https://                "/>
        <frame name = "main" src = "https://               "/>
        <frame name = "bottom" src = "https://             "/>
    <noframes>
        Your browser does not support frames
</body> </noframes>
</frameset>
```

OUTPUT:-



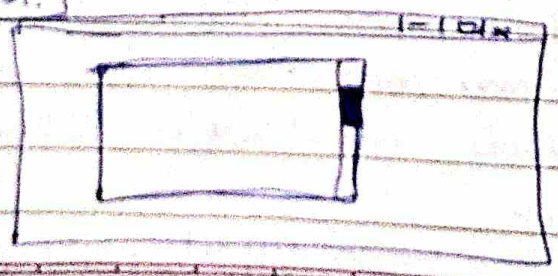


"left"
"center"
"right"

iframes

```
<body>
    <iframe src="http://                "></iframe>
        Browser Support inline frames
    <iframe>
</body>
```

OUTPUT:



HTML

Blocks

```
<ul>
    <li> Point1 </li>
    <li> Point2 </li>
    <li  Point3  </li>
</ul>
```

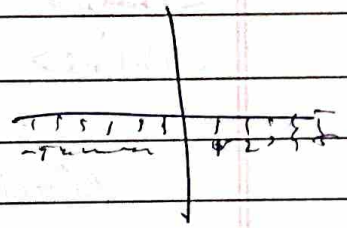
- Point1
- Point2
- Point3

```
<ol>
    <li> Point1 </li>
    <li> Point2 </li>
    <li> Point3 </li>
</ol>
```

(i) Point1
(ii) Point2
(iii) Point3

HTML

fonts.



```
<font size="1"> font size="1" </font><br>
```

+1 → increase font size
-1 → decrease font size

Action

Method

Target

Controls

TextBox	RadioButton
Password Box	Buttons
CheckBox	DROP DOWN

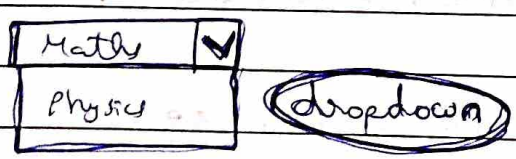
HTML

Form

form → Get
→ Post

```
<body>
<form action = " " method = "POST">

  First name: <input type= "text" id = "first_name"/>
  <br>
  Lastname: <input type = "text" name = "last_name"/>
</form>
</body>
</html>
```

Placeholder




```
<input type = "file" name = "fileupload" accept = "image/*"/>
```

HTML

Marquee

```
<Marquee> MovingText </Marquee>

<Marquee direction = "right"> MovingText </Marquee>
```

whatever you put in Marquee it starts moving

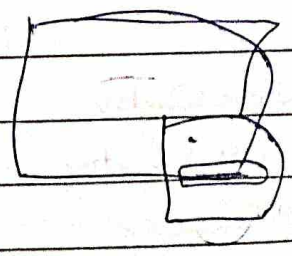
"left"
"right"
"Top"
"down"
"slide"
"alternate"

HTML
Embedding Media

Audiofile
Videofile

<embed src="video.mp4" height=" ">
<noembed> </noembed>
</embed>

Password.



CSS

JAY PRAKASH

Basic Syntax

```
<html>
    <head>
        <title> CSS Tutorial </title>
        <style type = "text/Css" >
        h1
        {
          color : blue;
          font-family : comic sans ms;
        }
        h2 {
        color:green;
        }
        </style>
    </head>
    <body>
        <h1> welcome to tutorial point </h1>
        <h2> Simply Easy Learning </h2>
    </body>
</html>
```

CSS

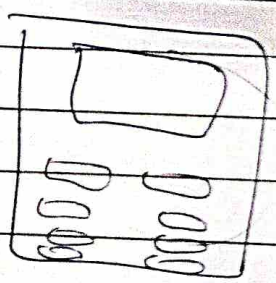
Selectors

Name Selector

Class Selector

id Selector

Universal Selector



```
<html>
    <head>
        <title> CSS Example </title>
        <style type = "text/css" >
          h1
          {
            color: blue;
          }
          h2 {
              Color: green;
              text-decoration : underline;
          }
          b, i {
              text-decoration : underline;
          }
        </style>
    </head>
    <body>
        <h1> Welcome To Tutorial Point </h1>
        <h2> Simply Easy learning </h2>
        <b> This is bold </b>
        <i> This is ITALIC </i>
    </body>
</html>
```

Background color for paragraph.

```
<p> {
    background-color: limegreen;
} </p>

// <h1 color : "green" > welcome </h1>
```

JAY PRAKASH

```
<html>
    <head>
        <title> CSS Example </title>
        <style>
            .style1 {
                color : green;
                font-family : Times new roman;
            }
            .style2 {
                color:red;
                font-family : monotype corsiva;
            }
        </style>
    </head>
    <body>
    <div>
        <h2 class="style1"> Heading 1 </h2>
        <p class="style1"> This is a paragraph for
                heading 1 </p>

        <h2 class="style2"> Heading 2 </h2>
        <p class="style2"> This is a paragraph
                for heading 2 </p>
    </body>
</html>
```

```
#mydiv {

}
<div id="mydiv">
```

CSS

JAY PRAKASH

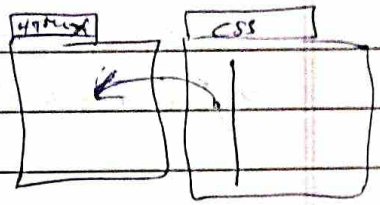
Date

Page

Inclusion in HTML

- Inline CSS
- Embedding CSS
- Linking CSS
- Importing CSS

<header>



Embedding only in <header> Section (X)

Semicolon → HTML (X)
Semicolon → CSS (✓)

Semicolon is used only in CSS

~~HTML~~ CSS

Colors

Hex code

#000000 – Blue (Black) #00FF00 – Green

Red

Green

000000

#FF0000 – Red #0000FF – Blue

#FFFFFF – RGB

Short Hex Code

#000 – Black #0F0 – Green

#F00 – Red #00F – Blue

000000 000 0000000/odd

RGB Absolute

rgb(0,0,0) – Black
rgb(255,0,0) – Red
rgb(0,255,0) – Green
rgb(0,0,255) – Blue

Keywords

Black
Red
Green
Blue

CSS

Background

```
<P style="background-color: yellow">
This is a paragraph </P>
```

```
<body style="background-image: url('images/bg.jpg')">
<P>
This is a paragraph
</P>
```

```
<body style="background-image: url('image/bg.jpg');
background-attachment: fixed;">
```

CSS

TEXT

```
<p style="Text-transform: capatalize;">
    This will be capatilized
<\p>
<p style="text-transform: uppercase;">
    This will be in uppercase
<\p>
<p style="text-transform: lowercase;">
    This will be in lowercase
<\p>
<\body>
```

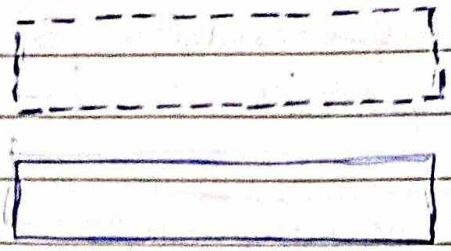
CSS

links

```
<html>
<head>
<style type="text/css">

a. {
    text-decoration: none;
  }
  a: link {color: Red}
  a: visited {color: Gray}
  a: hover { color: Pink}
  a: active { color: Green}
<\style>
<\head>
<body>
    <a href="https:// ">Click here <\a>
<\body>
<\html>
```

CSS

Tables



```
<p style="border-width:4px; border-style:dotted;">
This is a dotted border
</p>
```

CSS

Lists

```
<ul>
    <li> Tea </li>
    <li> Coffee </li>
    <li> Soft drink </li>
        <ul>
            <li> Tea </li>
            <li> Coffee </li>
            <li> Soft drink </li>
        </ul>
</ul>
```

Output:

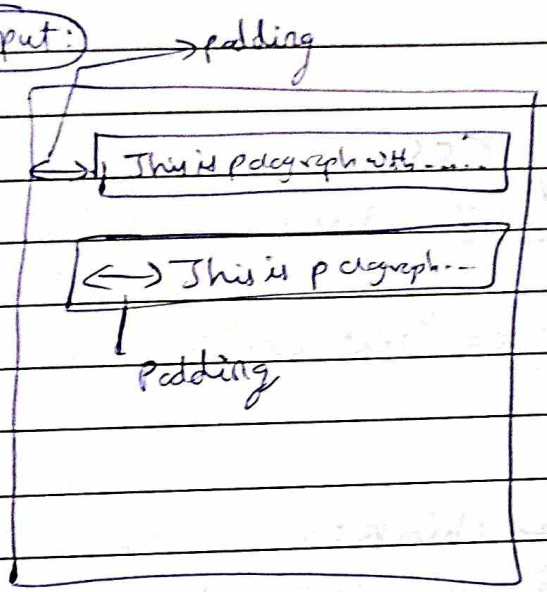
- Tea
- Coffee
- Soft drinks
    - Tea
    - Coffee
    - Soft drinks

CSS

Padding

```
<p style = "padding-left : 15px ; border : 1px solid black">
This is a paragraph with specified bottom padding
</p>
```

Output:



CSS

Cursors

```
<body>
    <p> Move the mouse over the words to see the cursor
        change </p>

    <div style = "cursor : auto" > Auto </div>
</body>
```

1) crosshair	6) ne-resize	11) s-resize
2) default	7) nw-resize	12) w-resize
3) pointer	8) n-resize	14) text
4) move	9) se-resize	15) wait
5) e-resize	10) sw-resize	16) help

CSS

Dimensions

Para Dimensions

width

max-height

padding

margin

CSS

Scroll Bars

```
<html>
<head>
</head>
    <style type="text/css">

    .scroll {
        display : block;
        border : 1px solid red;
        padding : 5px;
        margin-top : 5px;
        width : 300px
        height : 50px;
        overflow : scroll;
    }
    </style>

<body>
    <div class="scroll">
    I am going to keep lot of content.....

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

CSS

Visibility

JAY PRAKASH Date_____ Page_____

```
<p style ="visibility : hidden / ">

This paragraph should be visible in normal way
```

CSS

Positioning

1) Absolute Positioning
2) Relative positioning
3) Fixed positioning

```
<div style ="position : relative ; left : 80px; top:20px;
            background-color: yellow;">
This div has relative positioning.
</div>
```

fixed position : Does not change Even if we Scroll.

CSS

Layers

```
<body>
    <div style ="background-color: green;
width : 300px ; height : 100px; position: absolute; top:0px;
left : 120px z-index : 3;">
</div>
```

CSS 3

Introduction

JAY PRAKASH

Date
Page

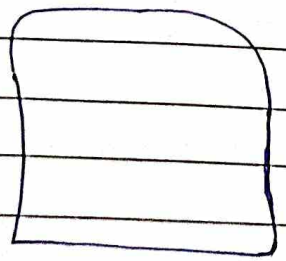
1) Rounded corners
2) Border Images
3) Multi Background
4) Gradient Colors
5) Shadowing
6) 2-D Transform
7) 3-D Transform
8) Animation

CSS 3

Roundercorners

```
<style>
    #rcorners1 {
        border-radius : 15px 50px 30px 5px;
        background : #a44170;
        padding : 20px;
        width : 100px;
        height : 100px;
    } </style>
```

OUTPUT:-



CSS 3

Border Images

CSS3

Multiple Backgrounds

JAY PRAKASH

```
<Style>
    # multibackground {
        background-image : url (images/logo.png),
                           url (images/bg.jpg);
        background-position : left, Top, left Top;
        background-repeat : no-repeat, no-repeat;
        padding : 75px;
    }
<1Style>
```

CSS

Colors

```
<html>
    <head>
        <style>                                  rgba
            # p1 {background-color : rgba (255, 0,
                                          0, 0, 3);}

            # p2 { background-color : rgba (0, 255, 0,
                                          0,3) ;}

            # p3 { background-color : right (0,0,255,0,3);
                                          }
        <1Style>
    </head>
    <body>
        <p> RGBA colors : <1P>
        <P id = "p1"> Red <1P>
        <P id = "p2"> Green <1P>
        <P id = "p3"> Blue <1P>
    <1body>
<1html>
```

CSS3

Gradient

id = " " → # Something

```
<head>
<style>
  #grad1 {
    height: 100px;
    width: 550px;
    background :-webkit -repeating -radial-
      gradient (blue, yellow 10%, green 15%);
  }
</style>
</head>
```

CSS3

Shadow

```
<html>
  <head>
    <style>
      .style1 {
        text-shadow : 2px, 2px;
      }
    </style>
  </head>
  <body>
    <h2 class="style1">Tutorialpoint.com</h2>
  </body>
</html>
```

CSS3

Text

JAY PRAKASH

```
<style>
  p.text1 {
    width : 140px;
    border : 1px solid #000000;
    word-break : keep-all;
  }
</style>
```

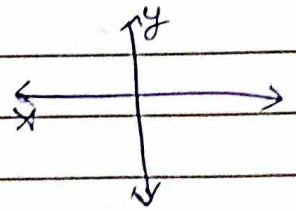
CSS3

Web Font

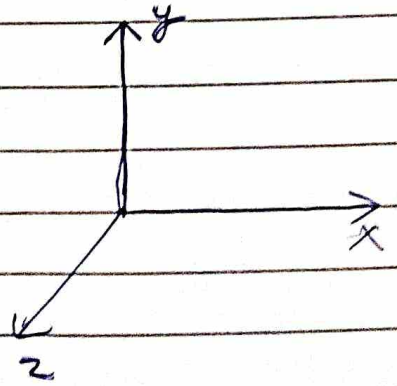
CSS font file download

CSS3

2D Transform



3D Transform



Animations

CSS3

Multiple Columns

JAY PRAKASH

```
<html>
<head>
<style>
 .multi {
    -webkit-column-count : 3;
    -moz-column-count : 3;
    column-count : 3;

    -webkit-column-gap : 40px;
    -moz-column-gap : 40px;
    column-gap : 40px;
 }
</style>
</head>
<body>
 <div class = "multi">

 </div>
```

JS

JAY PRAKASH

Date

Page

JavaScript

```
<html>
  <head>
    <title> JavaScript Demo </title>
  </head>
  <body>
    <script language = "javascript" type="text/java
                                        script">

    document.writeln ("Hi, welcome to JavaScript...");
    </script>
  </body>
</html>
```

Local variable → Declare within function
Global variable → Declare outside function

Second Program

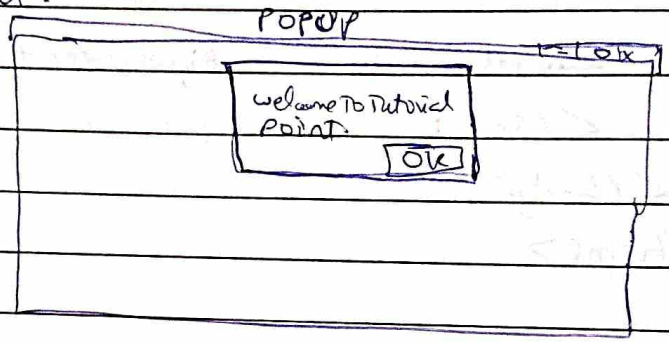
```
<html>
  <head>
    <title> JavaScript Demo </title>
  </head>
  <body>
    <script language = "javascript" type="text/javascript">

      var name = "Anadi";
      var age = 28;
      var salary = 5000.45;

      document.write ("Name: "+ name);
      document.write ("Age: "+ age);
      document.write ("Salary: "+ salary);
    </script>
  </body>
</html>
```

Javascript

Dialog Box

```
<body>
    <script language="javascript" type="text/javascript">
    alert ("welcome to Tutorialpoint.... ");
    </script>
</body>
```

Output:



Prompt → To Take User Input.

```
<body>
    <script language="javascript" type="text/javascript">
    var name = prompt ("Enter your name", "name...");
    alert ("Hello "+name);
    </script>
</body>
</html>
```

```
<body>
    <script language="javascript" type="text/javascript">
    var result = confirm ("Are you Sure to do this task",
                "confirm");
    document.write (result);
    </script>
</body>
```

JavaScript

[operators]

JavaScript has a rich collection of built in data types.

- → Assignment operator
- → Arithmetic operator
- → Comparison operator
- → Logical operator
- → Ternary Operator

Arithmetic operator

(+)
(-)
(*)
(/)
(%)

Hadson:

```
<body>
    <script language = "javaScript" type = "text/java
                                        script">

        var num1 = 40 , num2 = 10;
        document.write (num1 + num2);
        document.write ("<br/>");
        document.write (num1 - num2);
        document.write ("<br/>");
        document.write (num1 * num2);
        document.write ("<br/>");
        document.write (num1 / num2);
        document.write ("<br/>");
        document.write (num1 % num2);
        document.write ("<br/>");
    </Script>
</Html> </body>
```

post increment

```
Var num = 10;
Var result = num++;
document.write (result);
```

Output : 10

pre increment

```
Var num = 10;
Var result = ++num;
document.write (result);
```

Output : 11

Comparison operator

==	<body>
!=	<Script language = "java script" type="text/ javascript">
>	
<	Var num1 = 100;
>=	document.write (num1 == 10);
<=	document.write (num1 != 10);
	document.write (num1 >= 10);
	document.write (num1 <= 10);
	document.write (num1 > 10);
	document.write (num1 < 10);
	</Script)
	</body>

{ Logical operators }

&&

||

!

Hands on:

AND operator

```
<body>
    <Script language = "javascript" type = "text/javascript">

    var uid = prompt ("Enter User Id", "User Id");
    var pwd = prompt ("Enter Password", "Password");

    var message = (uid == "andi" && pwd == "tutorialpoint")
                    ? "welcome" : "Invalid user");
    alert (message);
</body>
```

OR operator

[No Password Input] [only user Input]

```
var message = (user == "andi" || user == "anand") ?
                "welcome" : "Invalid user";
alert (message);
```

NOT Operator

```
var message = ! (user == "andi" || user == "anand") ?
                "welcome" : "Invalid user";
alert (message);
```

Apart from "andi" and "anand", Everything is valid.

Java Script

Conditional Statement

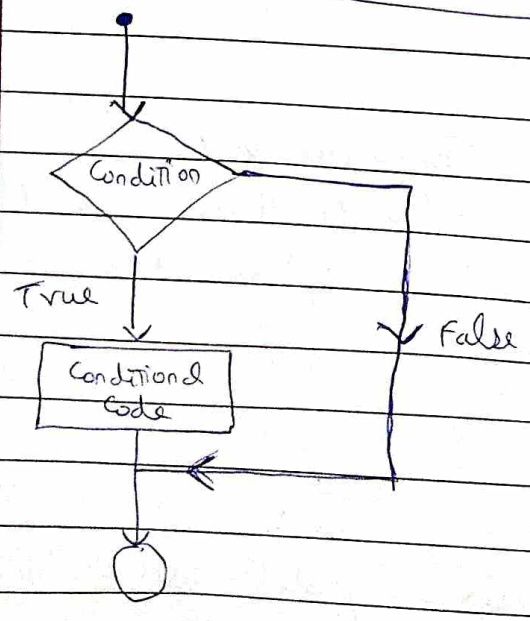
if else

Switch

```
String if (s1 == s2)
{
s.o.p ("Same");
}
else
{
s.o.p ("different");
}
```

Output: Same It works

JavaScript
if-Else Statement



```
<script laguage = "javascript" type = "text/javascript">
var marks1 = parseInt (prompt ("Enter Marks1", "Enter"));
Var marks2 = parseInt (prompt ("Enter Marks2", "Enter"));
Var marks3 = parseInt (prompt ("Enter Marks3", "Enter"));

Var total = marks1 + marks2 + marks3;
var percent = total * 100 / 150;
alert (percent);
if (percent >= 80)
{
alert ("Grade A");
}
else if (percent >=60)
{
alert ("Grade B");
}
else if (percent >=40)
{
alert ("Grade C");
}
else
{
alert ("Grade D");
}
</script>
</body>
```

parseInt → Convert to Integer

↓

marks1 + marks2 + marks3

add rather than concat

Example: 40+40+40 = 120

= 120/150 * 100

⇒ 80 %

JavaScript

Strings

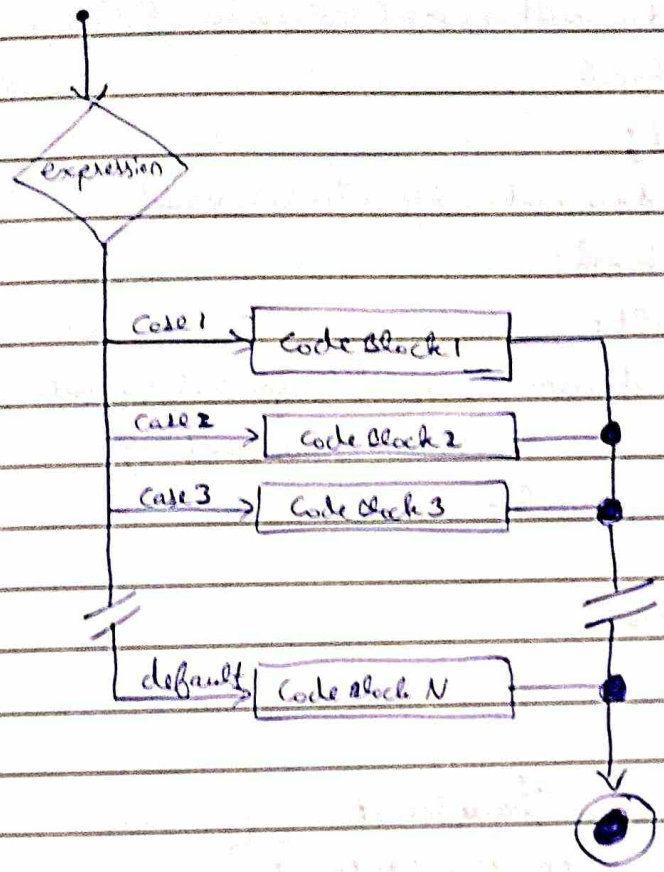
Hands on:

```
<html>
<head> </head>
<body>
<script language="javascript" type="text/javascript">
var str1 = "Tutorials Point India Pvt Ltd.";
var str2 = "Simply Easy Learning.";

document.write ("str1 :" + str1);
document.write ("str2 :" + str2);
document.write ("charAt : " + str1.charAt(3));
document.write ("charCodeAt : " + str1.charCodeAt(3));
document.write ("concat : " + str1.concat(str2));
document.write ("index of : " + str1.indexOf("p"));
document.write ("toLowerCase : " + str1.toLowerCase());
document.write ("toUpperCase :" + str1.toUpperCase());
document.write ("split :" + split.split(" ",3));
</script>
</body>
</html>
```

OUTPUT:

```
str1 : Tutorial Point India Pvt Ltd
str2 : Simply Easy Learning
charAt : o
charCodeAt : 111
concat : Tutorial Point India Pvt Ltd Simply Easy Learning
index of : 10
toLowerCase : tutorial point india pvt ltd
toUpperCase : TUTORIAL POINT INDIA PVT LTD
split : Tutorials Point India
```

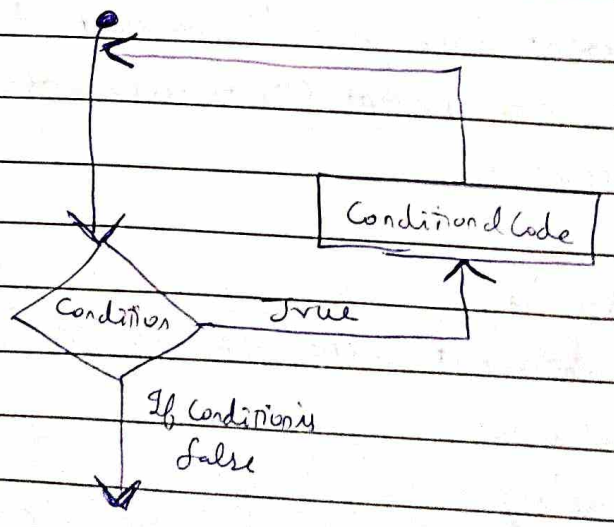
Switch Statement




```
<body>
<script language = "javascript" type = "text/javascript">
var ch = prompt ("Enter a character," "Enter ....");
switch(ch)
{
 case 'a':
    document.write ("A is vowel");
    break;
 case 'e':
    document.write("E is a vowel");
    break;
 case 'i':
    document. write ("I is a vowel");
    break;
```

```
case 'o':
        document.write("o is a vowel");
        break;
case 'u':
        document.write("u is a vowel");
        break;
default:
        document.write("Special character /
                        consonant");
        break;
}
</script>
```

JavaScript
Iteration Statements



1) for
2) While
3) Do While

JAY PRAKASH Date_____ Page_____

JavaScript

For loop

```
for (init ; condition ; increment)
{
   conditional code;
}
```

```
<body>
    <script language="javascript"
            type="text/javascript">

    var i;
    for (i=1 ; i<=10 ; i++)
    {
     document.write ("i = " +i);
     document.write ( "<br/>");
     }
     </script>
</body>
```

JavaScript

Whileloop

```
<body>
    <Script language="javascript" type="text/java
                                        script">

    var num = parseInt (prompt ("Enter a number", "Enter...."));
                                                        5432
    var count=0;
    while (num>0)
    {
     num = Math.trunc (num/10); 543.~~7~~ (Truncate)
     count++;
    }
```

```
        document.write ("Digits : "+count);
    </Script>
    </body>
</html>
```

JavaScript

do while loop

```
<html>
    <head>
        <title> JavaScript Demo </title>
    </head>
<body>
    <Script language ="javaScript" type= "text/javaScript"
    var actualpin ="3456";
    var pin;
    var count =0;
    do
    {
      pin= prompt ("Enter your pin ....", "Enter ....");
      count++;
    } while (pin != actualpin && count<3);
    if ( pin == actualpin)
    {
    document.write ("welcome user");
    }
    else
    {
    document.write ("Account locked");
    }
    </Script>
  </body>
</html>
```

JAY PRAKASH

JavaScript

Loop Control Statements

Break

Continue

```
<body>
    <script language="javascript" type="text/javascript">
     var i;
     for(i=1; i<=10; i++)
     {
      if (i==5)
      break;
      document.write("i =" +i+ "<br />");
      }
     <\script>
    <\body>
<\html>
```

JavaScript

Arrays

```
var fruits = new Array("apple", "orange", "mango");
```

↓

New fresh Space

```
<html>
    <head>
        <title> Java Script Demo </title>
    </head>
<body>
    <script language = "javascript" type = "text/javascript">
    var arr = new Array[5];
    arr[0] = 10;
    arr[1] = 20;
    arr[2] = 30;
    arr[3] = 40;
    arr[4] = 50;

    for( i=0; i<5; i++)
    {
      document.write (arr[i] + "<br/>");
    }
  </script>
 </body>
</html>
```

Assending order = Swapping

Descending order = Swapping

find a number in an array

```
<html>
    <head>
        <title> JavaScript Demo </title>
    </head>
    <body>
    <script language="javascript" type="text/javascript">
    var arr = new Array (7, 2, 8, 1, 3, 4);
    var num = parseInt (prompt ("Enter a number to Search",
                                "Enter .....")); 
    var found = false;

    for(i=0 ; i<6; i++)
    {
      if (arr[i] == num)
      {
        found = true;
        break;
      }
    }
    if (found)
    {
      document.write("Element Found .....");
    }
    else
    {
      document.write("Element Not Found ...");
    }
    </script>
  </body>
</html>
```

Java Script

OBJECTS

```
<body>
    <script language = "javascript" type = "text/javaScript">
    Var tutorial = {"topic": "JavaScript", "trainer": "Andy",
                    "mode": "online"};

    document.write ("Topic : "+tutorial.topic);
    document.write ("Trainer:" + tutorial.trainer);
    document.write ("Mode:" + tutorial.mode);
    </script>
</body>
</html>
```

JavaScript

Dates

```
<body>
<script language = "javascript" type = "text/javascript">
Var dt = new Date ("December 15, 2016 16:20:05");

document.write ("Date : "+Date());
document.write ("dt :"+dt);
document.write ("getDate : "+dt.getDate());
document.write ("getDay :"+ dt.getDay());
document.write ("getMonth :"+ dt.getMonth());
document.write ("getFullyear :"+ dt.getFullyear());
"  "  "  "  "  "  .getHours());
"  "  "  "  "  "  .getMinutes());
</script>
</body>
```

JavaScript

Functions

```
<html>
    <head>
        <script type="text/javascript">
        function SayHello()
        {
        document.write ("Hello there!");
        }
        </script>
    </head>
    <body>
        <p> Click the following button to call the function </p>

    <form>
        <input type="button" onclick="SayHello()"
        value="Say Hello">
    </form>
  </body>
</html>

<head>
    <script type="text/javascript">
    function SayHello(name, age)
    {
    document.write (name+" is"+age+" years old");
    }
    </script>
  </head>
  <body>
    <form>
        <input name="t1" placeholder="Enter Name"/> <br/>
        <input name="t2" placeholder="Enter Age"/> <br/>
        <input type="button" onclick="SayHello (t1.value, t2.value)"
                value="Say Hello">

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

JavaScript

Error Handling

```
<head>
  <script type="text/javascript">
  function myFunc()
  {
  var a=100;
  try
   {
   var s=a.toUpperCase();
   alert ("Value of variable a is: "+a);
   }
   catch(e)
   {
   alert ("Error");
   }
   finally
   {
   alert ("Finally block will always execute!");
   }
  } </script>
</head>
<body>
  <p> Click the following to see the result</p>
  <form>
     <input type="button" value="clickme" onclick="myFu
  </form>
</body>
```

JavaScript

Events

```
<html>
    <head>
    <Script type = "text/javascript">
    function MouseOver()
    {
    document.images ["img1"] -src= "logored.png"
    }
    function MouseOut()
    {
    document.image ["img1"] -src="logogreen.png"
    }
    function load()
    {
    document.images["img1"] -Src= "logogreen.png"
    }
    </Script>
  </head>
  <body onload ="load()">
      <img name = "img1" mouseover = "MouseOver()"
        onmouseout = "MouseOut()"/>
  </body>
</html>
```

Javascript

Page Redirect

```
<html>
  <head>
    <script type="text/javaScript">
      function Redirect()
      {
      window.location = " http: ....... ";
      }
    </script>
  </head>
  <body>
    <P> Click the following button you will be redirected
        to home page </P>
    <form>
      <input type="button" value="Redirect Me"
            onclick = "Redirect();" />
    </form>
  </body>
</html>
```

Javascript

Cookies

→ Session Cookie [Temporary]
→ Persistent Cookie [Permanent]

Java Script
(Validations)

```
<body>
    <form                    >
      <table                 >
        <tr>
            <td> </td>
            <td> </td>
        </tr>
        <tr>
            <td> </td>
            <td> </td>
        </tr>
        <tr>
            <td> </td>
            <td> </td>
        </tr> </table>
    </form>
</body>
```

JavaScript

Animation

```
<head>
    <title> Animation </title>

    <script type="text/javascript">
    var arr = new Array ("1.jpg", "2.jpg", "3.jpg");
    var i=1;
    var ref;
    function picLibrary()
    {
    document.image[0].src = arr[i];
    i++;
    if (i >3)
        i=0;
    }
    function start()
    {
    ref = setInterval ("picLibrary()", 1000);
    }
    function stop()
    {
    clearInterval (ref);
    }
    </script>
</head>
<body>
    <img src = "1.jpg" name = "img1" onmouseover = "stop()"
    onmouseout = "start()" />
</body>
</html>
```

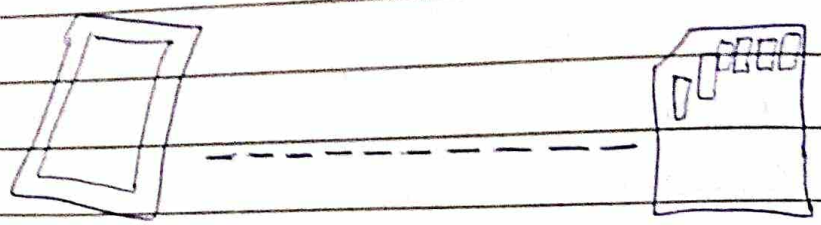
JAY PRAKASH Date_____ Page_____

JavaScript

Image Map

```
<html>
  <head>
  </head>

  <body>
    <map>
      <area>


      </area>
      </map>
  </body>
</html>
```

SQL



Database Management System — SQL — Database

1) ORACLE
2) MYSQL
3) SQL SERVER
4) POSTGRESQL

SQL → Easy Programming Language

SQL Vs NoSQL

1) Structured Data
2) Unstructured Data
3) Semastructed Data

Structured Data

1) Text
2) Number
3) Date

Unstructured Data

1) Post
2) Reels
3) Comments

3) Semistructured Data (Structured Data + Unstructured Data)

1) Email

2)

① Relational Database

SNo	Name	Age

Table ✓

① Non Relational database

SNo	Name	Age

No Table ✗

Command
- DDL – Data Definition Language
- DML – Data Manipulation Language
- TCL –
- DCL –
- DQL – Data Query Language

⇒ Data Definition Language

1) Create
2) Alter
3) Drop
4) Truncate

① Create Schema dummy
Run This ↑

Table

Name	Age	Department	Salary

use dummy

```
create table Teacher (
name varchar(100),
age int,
Salary int,
department varchar(100),
)
```

② i) alter table teacher add column degree varchar(100)

Name	Age	Department	Salary	degree

ii) alter table Teacher change department dept varchar (100)

Name	Age	Dept	Salary	degree

iii) alter table teacher change age age varchar(100)

Note: age change from int to varchar.

③ drop table teacher

④ truncate table teacher

→ The Data inside the table will get deleted.

⇒ Data Manipulation Language

Student

Name	Age	Department
John	23	CSE
Praveen	45	ECE
Suresh	34	EEE

Create table

```
create table student (
name varchar (100),
age int,
department varchar (100)
)
```

DQL Command

```
select * from student
```

	name	age	department

```
insert into students values ("John", 23, "CSE")
```

```
Select * From Student
```

	name	age	department
▶	John	23	CSE

```
insert into students values ("Praveen", 45, "ECE")
```

```
Select * from Student
```

	name	age	department
▶	John	23	CSE
	Praveen	45	ECE

insert into Student values ("Suresh", 34, "EEE")

Select * from Student

	name	age	Department
▶	John	23	CSE
	Praveen	45	ECE
	Suresh	34	EEE

update Command

update Student set age = 23

select * from Student

	name	age	Department
▶	John	23	CSE
	Praveen	~~45~~ 23	ECE
	Suresh	23	EEE

update student set department "Computer" where department "CSE"

select * from Student

	name	age	department
▶	John	23	Computer
	Praveen	23	ECE
	Suresh	23	EEE

update student set age = 30 where age = 23

select * from student

	name	age	department
▶	John	30	Computer
	Praveen	30	ECE
	Suresh	30	EEE

update student set age = 40 where name = "John"

select * from student

	name	age	department
▶	John	40	Computer
	Praveen	30	ECE
	Suresh	30	EEE

delete from student ~~where~~

→ Everything will get deleted

delete from student where department = "Computer"

→ 1st Row will get deleted

delete from student where age=30

→ 2nd and 3rd Row will get deleted.

select * from student

	name	age	department
▶	John	40	Computer

Question:

1. How do you make a table in a database?

```
Customer (
    customer_id,
    customer_name,
    customer_address,
    city,
    State,
    ZIP_CODE)
```

use dummy (Database)

```
create table customer (
customer_id int,
customer_name varchar(100),
customer_address varchar(200),
city varchar(100),
state varchar(100),
zip-code varchar(100)
)
```

Select * from Customer

2. How do you insert below records?

1. Customer ID: 1
- Name : John Doe
- Address : 392 Sunset Blvd.
- City : New York
- State : NT
- ZIP Code:

2. Customer ID: 2
- Name : Mary Smith
- Address: 6900 Main St,
- City : San Fransico
- State : CA
- ZIP Code: 94032

3. Customer ID : 3
- Name : Richard Newman
- Address : 2040 Riverside Rd.
- City : San Diego
- state : CA
- ZIP Code : 92010

4. Customer ID: 4
- Name : Cathy Cook
- Address : 4010 Speedway
- City : Tucson
- State : AZ
- ZIP Code: 85719

insert into customer values (1, "John Doe", "392 Sunset Blvd", "New York", "NT", "10059")

Select * from customer.

	customer_id	customer_name	customer_address	city	state	zipcode
▶	1	John Doe	392 Sunset Blvd	New York	NT	10059

insert into customer values (2, "Mary Smith", "6900 Main St", "San Fransico", "CA", "94032");
insert into customer values (3, "Richard Newman", "2040 Riverside", "San Diego", "CA", "92010");
insert into customer values (4, "Cathy Cook", "4010 Speedway", "Tuscon", "AZ", "85719")

Select * From Customer.

	customer_id	customer_name	customer_addre	city	state	zip_code
▶	1	John Doe	392 Sunset Blvd	Newyork	NT	10059
	2	Mary Smith	6900 Main St	San Fransico	CA	94032
	3	Richard Newman	2040 Riverside	San Dego	CA	92010
	4	Cathy Cook	4010 Speedway	Tucson	AZ	85719

(3) How can we change the customer address column to just address ?

alter table customer change column customer_address address varchar(200)

Select * from customer

	customer_id	customer_name	address	city	state	zip_code
▶	1	John Doe	392 Sunset	Newyork	NT	10059
	2	Mary Smith	6900 Main St	San Fransico	CA	94032
	3	Richard Newman	2040 Riverside	San Dego	CA	92010
	4	Cathy Cook	4010 Speedway	Tuscon	AZ	85719

4. How do you add a new column called mobile_number?

alter table customer add column mobile_number int (100)

Select * from customer

	customer_id	customer_name	address	city	state	zip_code	mobile_number
▶	1	Johnson	197 [illegible]	Newyork	NY	10001	NULL
	2	Mary Smith	6700 [illegible]	San Francisco	CA	94103	NULL
	3	Richard Moore	3040 [illegible]	San Diego	CA	92101	NULL
	4	Cally Cook	4030 [illegible]	Boston	MA	02131	NULL

5. How do you delete a column where mobile_number is null?

delete from customer where mobile_number = null

Select * from customer.

Output: Nothing will change.
mobile_number = null (will not work)

delete from customer where mobile_number is null

Select * from customer

	customer_id	Customer_name	address	city	State	Zip-Code	mobile_number

(6) How do you insert below records ?

insert into customer values (" ", " ", " ", " ");
insert into customer values (" , , , ");
insert into customer values (" , , , ");
insert into customer values (" , , , ");
insert into customer values (" , , , ");

Select * from Customer.

(7) How to show the all Records

Select customer_name from customer

	customer_name
▶	John Doe
	Mary Smith
	Richard Newman
	Cathy Cook
	Alice Johnson
	Bob Williams

Select customer_name, address from customer.

	customer_name	address
	John Doe	" "
	" "	"
	" "	"
	" "	"
	" "	"
	" "	"

(8) How can you update the phone number for 'Mary Smith' to '52206-1234'?

update customer set mobile_number = "52206-1234"
where customer_name = "Mary Smith"

select * from customer.

(9) How can you delete the record where the zip_code is "60601"?

delete from customer where zip_code = "60001"

select * from customer.

Customer_id	Customer_name	address	city	State	zip-code	M N
1						
2						
3						
4						
5						
~~6~~	~~X~~	~~X~~	~~X~~	~~X~~	~~X~~	~~X~~

→ 6th row will get deleted

(10) How can you select distinct data where the user is from a state = "CA"?

select * from customer where State = "CA"

distinct = தனித்துவமான

Customer_Id	customer_name	address	City	State	Zip-code	Mobile-num
2	"	"	"	"	"	"
3	"	"	"	"	"	"
4	"	"	"	"	"	"
5	"	"	"	"	"	"

→ 1st row will get ~~deleted~~. removed not deleted

distinct → Not delete / but remove

(11) How do you print customer IDs who are greater than 5?

Select * from customer where customer_id > 5

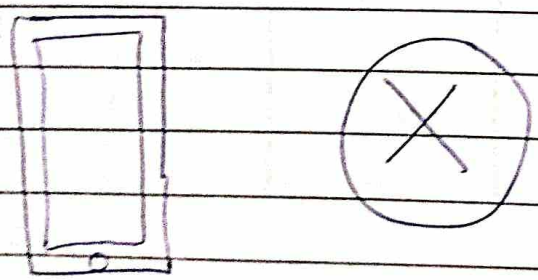
	customer_id	customer_name	address	city	state	zip-code	mobile number

Select * from customer where customer_id > 2

Customer.

	customer_id	customer_name	address	city	state	zip-code	M_N
▶	3	"	"	"	"	"	"
	4	"	"	"	"	"	"
	5	"	"	"	"	"	"

(13) How do you delete the mobile number column?



alter table customer drop column mobile_number

Select * from customer

→ Mobile number column will get deleted.

(14): How do you Truncate a table?

truncate table customer

select * from customer

	customer-id	customer-name	address	city	State	Zip-code

→ It won't delete Table but it will delete data inside the table.

Aggregate function

1) SUM()
2) MAX()
3) MIN()
4) AVG()
5) COUNT()

Entire Table function:

1) Create database
2) Create Table
3) Insert Table
4) Alter Table
5) Update Table
6) Delete Table

```
use dummy;

create table student (
student_name varchar(100),
student_mark int,
department varchar(100)
);

insert into student values ("Praveen", 1, "CSE");
insert into student values ("Manoj", 2, "MECH");
insert into student values ("Karthi", 3, "ECE");
insert into student values ("Raju", 4, "MECH");
insert into student values ("Deepak", 5, "CSE");

select * from student
```

OUTPUT:

student_name	student_mark	department
Praveen	1	CSE
Manoj	2	MECH
Karthi	3	ECE
Raju	4	MECH
Deepak	5	CSE

```
select student_mark from student.
```

→ we will get student_mark alone.

Aggregate function

① SUM()

select sum(student_mark) from student

Student_mark
1 ⊕
2 ⊕
3 ⊕
4 ⊕
5

OUTPUT

Student_mark
15

Select sum(student_mark) as total marks from student

Student_mark	→	Total marks
15		15

select * from student.

② MAX()

Select max(student_mark) from student

	max(student_mark)
▶	4

③ MIN()

select min(student_mark) from student.

	min(student_mark)
▶	1

④ AVG()

select avg(student_mark) from student

	avg(student_mark)
▶	3.0000

1, 2, 3, 4, 5 ⊕

$\frac{1+2+3+4+5}{5} = \frac{15}{5} = 3$ $\boxed{=3}$

⑤ Count()

select count(student_name) from student.

	count(student_name)
▶	5

select * from student where department = "CSE"

	studentnone	student-mark	departmnt
▶	Praveen	1	CSE
	Deepak	3	CSE

select studentnone from student where department = "CSE"

	student_none
▶	Praveen
	Deepak

select count(student-none) from student where department = "cse";

	count (student_none)
▶	2

Group by & Orderby

① Create Student Table

Select * from student where department = "CSE"

→ People those who are there in CSE will get displayed

Select * from student where department = "ECE"

→ People those who are there in ECE will be displayed.

Select student_name from student

→ All student_name will get printed.

Select student_name, mark from student

→ Both student and mark will get displayed

ORDER BY:

Select student_name, mark from student
order by mark desc

student_name	mark

high
↓
low

Descending Order

select student_name, mark from student order by mark asc

	student_name	mark	
▶			low
			↓
			high

Ascending ORDER

select sum(mark) from student
select avg(mark) from student
select MIN(mark) from student
select Max(mark) from student
select count(mark) from student

—— (x) ——

GROUP BY:

CSE 69 70 80
ECE 79 82
EEE 72 78

select avg(mark) from student group by department.

	avg(mark)	
▶	77.2500	(CSE)
	69.5000	(ECE)
	52.000	(EEE)

select avg(mark), department from student group by department

	avg(mark)	department
▶	77.2500	CSE
	69.500	ECE
	52.000	MECH

Select student_name department from student.

Student_name department

select count(student_name), department from student group by department.

	count(student_name)	department
▶	4	CSE
	2	ECE
	2	MECH

select count (student_name), department from student group by department order by count(student_name) desc

	Count (student_name)	department
▶	4	CSE
	2	ECE
	2	MECH

Create table Employee (

EmployeeID INT,
FirstName VARCHAR (50),
LastName VARCHAR(50),
Department VARCHAR(50),
Salary INT
);

INSERT INTO Employee (EmployeeID, FirstName, LastName, Department, Salary)
VALUES
(1, 'John', 'Doe', 'HR', 55000),
(2, 'Jane', 'Smith', 'IT', 60000),
(3, 'Bob', 'Johnson', 'IT', 62000),
(4, 'Alice', 'Williams', 'HR', 54000),
(5, 'Eva', 'Davis', 'Finance', 58000),
(6, 'Mike', 'Brown', 'Finance', 59000);

Select * from employee where department = "IT"

EmployeeID	Firstname	lastName	Department	Salary
2	Bob	Johnson	IT	43000
3	Jane	Smith	IT	40000

Select * from employee where department = "IT" order by Salary desc

3	Bob	Johnson	IT	48000
2	Jane	Smith	IT	43000

For no of employee use count (*)

Select salary, department from employee

Salary	department

Select avg(salary), department from employee group by department order by avg(salary) desc limit 1

avg (salary)	department
6,1000.000	IT

Having Clause

Select * from employee where department = "IT"

	EmployeeID	FirstName	LastName	Department	Salary
▶	2	Jane	Smith	IT	60000
	3	Bob	Johnson	IT	62000

select count (*) from employee where department = "IT"

	count(*)
▶	2

group by → Count.

Having Clause → If we use condition in Aggregate function.

select avg(salary), department from employee where avg (salary) <60000 group by department having avg(salary) <60000

Primary key → No duplicate Entry

Practice Question

DDL
DML
TCL
DCL
DQL
Group by
Order by
Having Clause.

Watch #10 SQL 25 Practice Question in You Tube

Unix

awk programming language

awk 'BEGIN { } → Begin Block

{ } → Action Block

END { }' → END Block

BEGIN { } and END { } → gets Executed only Once

Action Block { } → Based on ~~Eve~~ Input lines

```
awk 'BEGIN{print "Hello world"}'
{}
        END { }
```

Output:
Hello world

```
awk 'BEGIN { }
{}
   END {print "Hello world"}'
```

OUTPUT:
Hello world

```
awk 'BEGIN { }
{print "Hello world"}
END{ }'
```

OUTPUT:

Hello world
Hello world
Hello world.

```
awk 'BEGIN {print "begin"}
{print "hello world"}
END {print "end"}
```

OUTPUT:

begin
hello world
hello world
hello world
end.

```
awk 'BEGIN {FS = ","; } {print $1}'
```

$1 | $2 | $3
uday, 22, chennai
raj, 25, bombay
john, 36, kerala

```
awk 'BEGIN {FS = ","; } {print $1, $3}'
```

Print 1st and 3rd colums

```
awk 'BEGIN {FS=","; OFS="-";}{print $1, $3}'

uday - chennai
raj -
uday - chennai
raj - Bombay
john - kerala.
--
```

```
awk 'BEGIN {FS=","; OFS="/";}{print $1, $2, $3}'

uday /22/ chennai
raj / 25/ bombay
john / 36/ kerala
//
```

Unix Hands On.

(*) I want to seperate the output field by "-"

```
awk 'BEGIN {FS="/"; OFS="-"}{print $1, $2, $3}'
```

OUTPUT:-

```
uday -22- chennai
john - 22 - kerala
raj - 33 - himachal
```

(*) I want to seperate the output field by "*"

```
awk 'BEGIN{ FS="/"; OFS="*"} {print $1,$2,$3}'
```

Output:

uday * 22 * chennai
john * 22 * kerala
raj * 33 * himachal

(*) I want to seperate the output field by "/"

```
awk 'BEGIN { FS="/"; OFS="/"}{print $1, $3}'
```

OUTPUT:

uday / chennai
john / kerala
raj / himachal

(*) I want to seperate the output field by "-"

```
awk 'BEGIN { FS="/"; OFS="-"}{print $2 $3}'
```

OUTPUT:

22 - chennai
22 - kerala
33 - himachal

* If I remove OFS what will happen

```
awk 'BEGIN {FS = ",";}{print $2, $3}'
```

```
22 chennai
22 kerala
23 himachal
```

By default There will be a Space.

```
awk 'BEGIN{}{print $2, $3}'
```

```
22 chennai
22 kerala
23 himachal
```

Field Seperator by default is a Space.

BEGIN{} and END{} will get Executed only once

{} Action Block, If there are Three Inputs

1st line will go into Action Block and it will get Executed
2nd line will go into Action Block and it will get Executed
3rd line will go into Action Block and it will get Executed

print Three Times

```
awk 'BEGIN {a=0;}
{
print a;
}
END {print a}'
```

OUTPUT:-

```
0
0
0
```

Print only once

```
awk 'BEGIN {a=0;}
{}
END {print a}'
```

OUTPUT:-

```
0
```

Print Three Times

```
awk 'BEGIN {a="hello";} {print a}
END {}'
```

OUTPUT:

```
hello
hello
hello
```

Print only once

```
awk 'BEGIN{a="hello"}
{}
END{ print a}
```

Output:
hello

```
awk 'BEGIN{a="record"; OFS="-"; FS="-";}
{print a, $1, $2, $3} END{}'
```

OUTPUT:

record-uday-22-chennai
record-raj-34-kashmir
record-john-56-himachal

```
awk 'BEGIN{print "begin"}{print "action"} END
{print "end"}'
```

OUTPUT:
begin
action
action
action
end

JAY PRAKASH

(*) If a passing is greater than 35 pass, else fail

```
awk 'BEGIN {FS = "-"; OFS = " "};
{if ($3>35) {print $1, $2, "pass"}
else { print $1, $2, "fail"}
END {}'
```

OUTPUT: (Before)	output: (after)
john - ece - 70	john - ece - pass
uday - mech - 25	uday - mech - pass
kiran - ece - 21	kiran - ece - fail
raj - mech - 23	raj - mech - fail

LOGICAL AND COMPARISON OPERATOR.

1) Greater than	if (S1 > S2)
2) Lesser than	if (S1 < S2)
3) Greater than or equal to	if (S1 >= 90)
4) Lesser than or equal to	if (S2 <= 90)
5) Equal to	if (S1 == S2)
6) Not Equal to	if (S1 != S2)

Logical operator

&&	&
||	|
!=	!

→ checking if a person has 60% in college 10th and 12th we use and operator

```
if (xmark >60 && xiimark >60 && collmark>60)
{
print lligible
}
```

⊛ checking if a person has passed in any one of the subject

```
if (Sub1>= 35 || Sub2>= 35)
{ print pass}
```

⊕ I need to specify person who scored 100 as "topper"

```
awk 'BEGIN {FS="-";OFS="-";}
{
  if ($3==100 || $4==100)
{ print $1, "topper" }
}'
```

OUTPUT:

uday Topper
kishore topper
prakash topper

|uniq|sort → for printing output in assending order

Example

```
awk 'BEGIN {FS = ",";  OFS = ","}
{
        if ( $4<60)
        {
          print $1;
        }
}' END {}' |uniq |sort
```

OUTPUT:

101

102

103

Even if we don't use |uniq command will run

```
awk 'Begin{}
{
 if(      ){
   print $0;
}
END{ }" |Sort
```

OUTPUT

101

102

103

```
awk 'BEGIN {FS = "|";}
(NR>1) {
if ($4 >= 90) {
    print $2, "Gold"
} else if ($4 >=60) {
print $2, "Silver"
} else {
  print $2, "Bronze"
}
}'
```

(NR- Number Records)

Using for loop.

```
awk 'BEGIN { FS = "|" }
{
        a[$3]++;
} END { for (i in a)
    print (i, a[i]) }' |sort -t " "
```

-t changes "|" to Blank Space " "

JAY PRAKASH

Date
Page

JDBC

- Database • RDBMS • NOSQL • MongoDB

Step 1: Import the package
Step 2: Load and Register the Driver
Step 3: Connecting or establish the connection
Step 4: Create the Statement
Step 5: Execute the Query
Step 6: Process the Query
Step 7: Close

com.mysql.jdbc.driver
mysql.connector

```
import java.sqll.*;

class class_name
{
public static void main (String [] Args)
{
```

Roll No	Name
1	Naveen
2	Raj
3	Nav

```
vs.next()
                    1st column
rs.getInt(1) + "  " +
                    rs.getString(2);

while (rs.next())
{ Sout (rs.getInt(1)+"  "+
                    rs.getstring(2))
}
St.close();
con.close();
```

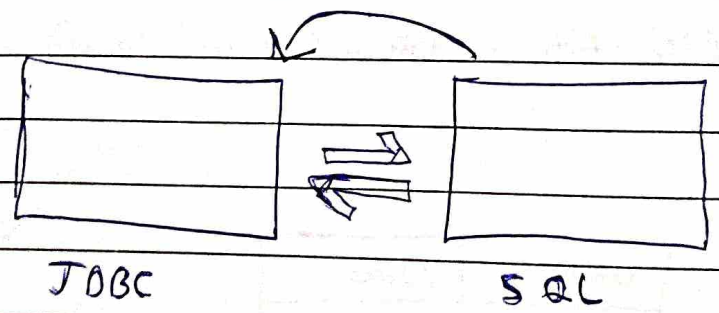
DDL → wherever you create Structure of Database or Table
DML → wherever your insert, Delete its DML
DQL → where you want to file the Query its DQL

// Java Database Connectivity Theory

1. import → java.sql
2. Load and register the driver → com.mysql.jdbc.D
3. Create connection → Connection
4. create a statement → Statement.
5. execute the query
6. process the results
7. close.

Google → mysql connector jar

mvnrepository.com



String url = used to connect JDBC and SQL

rs.next(); → Next Line

1st Column 2nd Column

```
userData = rs.getInt(1) + ":" + rs.getString(2);
System.out.println(userData);
```

executeQuery → output
executeUpdate → Input

```
class DemoClass
{
  public static void main(String[] args) throws Exception
  {
    String url = "jdbc:mysql://localhost:3307/college";
    String uname = "root";
    String pass = "";
    int userid = 5;
    String username = "Alan";
    String query = "insert into Student values ("+userid+","+
                                                 username+"");
    Class.forName("com.mysql.jdbc.Driver");
    Connection con = DriverManager.getConnection(url, unamepass);
    Statement st = con.Create Statement();
    int count = st.executeUpdate(query);
                  └Input
    System.out.println(count + "rows affected");
    st.close();
    con.close();
  }
}
```

```
class DemoClass
{
  public static void main(String[] args)
  {
    Class.forName("Pqr");
  }
}
```

→ Load the Class

```
class Pqr
{
  static
  {
    System.out.println("in static");
  }

  //instance
  {
    System.out.println("in instance");
  }
}
```

OUTPUT:

in static

```
DriverManager.registerDriver(new com.mysql.jdbc.Driver());
```

→ Load the Driver which we downloaded before.

mysql-connector-java-9.0.0

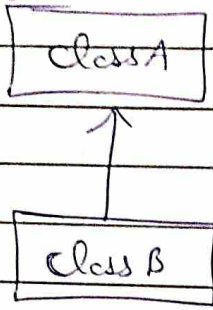
String query = " ";

→ To write a query

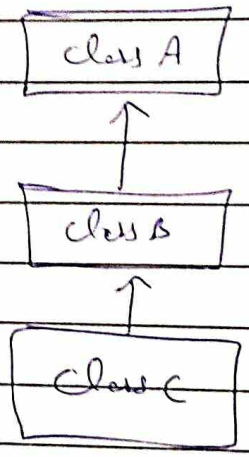
Types of Inheritance

There are (Five) Types of Inheritance

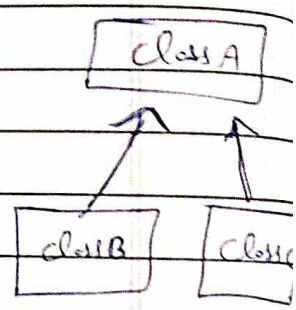
Single Inheritance



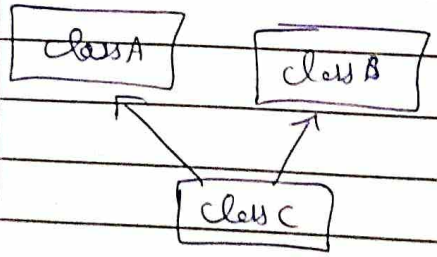
Multilevel



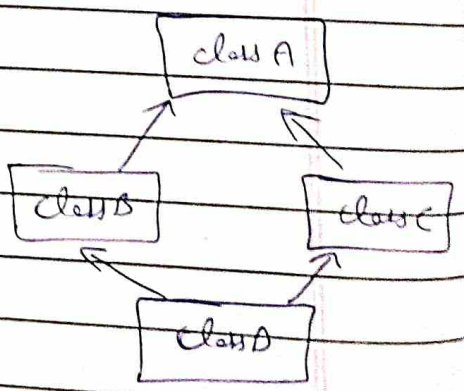
Hierarchical



Multiple



Hybrid



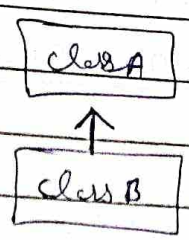
(*) Java does not support [Multiple] Inheritance

Syntax

class subclass_name extends superclass_name
{

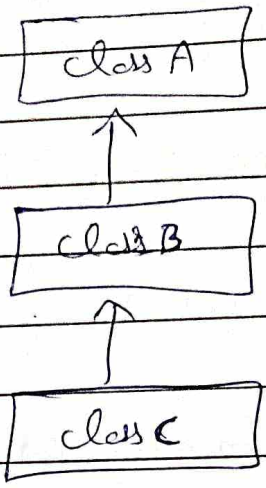
}

Single Inheritance



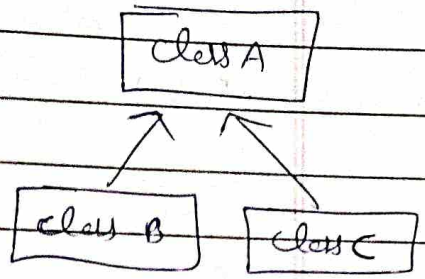

```
class Animal
{
void eat ()
{
System.out.println("eating.... ");
}
}
class Dog extends Animal
{
void bark ()
{
System.out.println("barking....");
}
}
class TestInheritance
{
public static void main (String args[])
{
Dog d = new Dog ();
d.bark();
d.eat();
}
}
```

Multilevel Inheritance



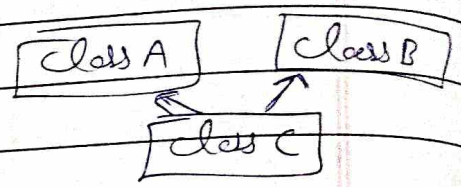

```
class Animal {

void eat()
{
System.out.println("eating ....");
}
class Dog extends Animal
{
void bark()
{
System.out.println ("barking ....");
}
class BabyDog extends Dog
{
void weep()
{
System.out.println("weeping ....");
}
}
class TestInheritance
{
public static void main(String [] args)
BabyDog d = new BabyDog();
d.eat();
}
}
```

Hierarchical Inheritance




```
class Animal
{
void eat()
{
System.out.println("eating.....");
}
class Dog extends Animal
{
void Bark()
{
System.out.println("barking.....");
}
class Cat extends Animal
{
void meow()
{
System.out.println("meowing....");
}
class TestInheritance
{
public static void main(String[] args)
{
Cat C= new Cat();
C.meow();
C.eat();
}
}
```

Multiple Inheritance



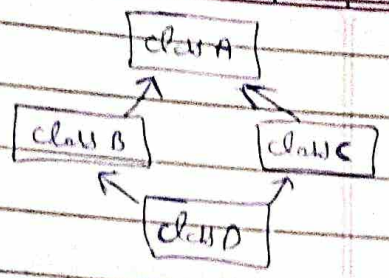

```
class A
{
void msg()
{
System.out.println("Hello");
}}
class B
{
void msg()
{
System.out.println("welcome");
}
}
class C extends A, B
{
public static void main(String[]args)
{
C obj = new C();
obj.msg();
}
}
```

Multiple Inheritance is not Supported by java

Output:

Compile Time Error

Hybrid Inheritance




```
class Animal
{
void eat()
{
System.out.println("eating....");
}
}

class Dog extends Animal
{
void bark()
{
System.out.println("barking...");
}
}

class Cat extends Animal
{
void meow()
{
System.out.println("meowing....");
}
}
interface Pet {
void play();
}

class PetDog extends Dog implements Pet
{
public void play()
{
System.out.println("playing....");
}
}
Public class TestInheritance {
public static void main (String[]args) {
Cat c = new Cat();
c.meow();
c.eat();

Dog d = new Dog();
d.bark();
d.eat();

PetDog pd=new PetDog();
pd.bark();
pd.eat();
pd.play();
}

}
```

SQL

JAY PRAKASH

Primary key

VS

Foreign key

Primary key

① Primary key is a column or set of column that uniquely identifies each row in a table

② A table can have only one Primary key which can made up of one or more fields

Primary key → always have unique data
→ No duplicate keys

Customer id	First Name	Last Name	City
24	Ajay	Rathore	Mumbai
25	Rohit	Sinha	Mumbai
26	Sakshi	Sinha	Hong Kong
27	Abhishek	Gupta	Delhi

Set Customer id as Primary key

Syntax

Alter Table Customer add Primary key (Customer-id)

Alter Table Customer drop primary key (Customer-id)

Foreign Key

* Foreign key is a column or set of column that is used to link two table together.

* The Foreign key of one Table helps to connect with the Primary key of another Table

Customer id	First Name	Last Name	City
24	Ajay	Rathore	~~Mumbai~~ 6
25	Rohit	Sinha	~~Mumbai~~ 6
26	Sakshi	Sinha	~~Banglore~~ 1
27	Abhishek	Gupta	~~Delhi~~ 3

~~City~~ ID	City
1	Banglore
2	Chennai
3	Delhi
4	Hyderabad

Syntax Reference → Primary key of another table

Alter table Customer add Foreign key (column) (City)
References table (column)
~~Customer (City)~~ city (cid)

Alter table Customer Drop Foreign key table (column)