

## JQUERY

- We use JQuery to -
  - 1) to simplify DOM
  - 2) for plugins
  - 3) for form validation
- It is the library of Javascript which is used to simplify the Javascript DOM manipulations.
- Latest Version is 3.5.1.

PROGRAM Example :-

→  
(body)  
<script>  
let employees = [  
];  
(same as previous program)  
{};

// filter function.

let activeEmps = employees.filter((employee) =>  
{  
return employee.gender === 'Male';  
});  
console.log(activeEmps);

</script>

</body>

</html>

f array.

od.

ent))

```

let filteredEmployees = [];
for(let employee of employees)
{
    if(employee.gender == 'Female') →
        console.log(employee);
    filteredEmployees.push(employee);
}
console.log(filteredEmployees);
</script>
<body>
</body>
</html>

```

### \* filter function :-

- It is a predefined function.
- It is used to store the values in form of array.
- We define this function using 'filter' method.
- Syntax :- arrayname.filter(function(argument))

≡ } // code ;

PROGRAM

body /  
script  
let

// fi

let

else

if

else

{ body

{ html }

```
<!DOCTYPE html>
<html>
<head>
<title> JS Employees </title>
</head>
<body>
<script>
let employees = [
    {
        name: 'Sunil',
        age: 23,
        gender: 'Male',
        active: true
    },
    {
        name: 'Sara',
        age: 26,
        gender: 'Female',
        active: false
    },
    {
        name: 'Soumen',
        age: 25,
        gender: 'Male',
        active: false
    },
    {
        name: 'Snigdha',
        age: 24,
        gender: 'Female',
        active: true
    }
];
```

\* In arrow function if we are taking single parameter then ~~we are~~ brackets are optional.

If we are taking more than one parameter then brackets are mandatory.

Q. callback function:- Introduced in ES6.

PROGRAM :-

```
<!DOCTYPE html>
<html>
<head>
    <title> JS Call Back Function </title>
</head>
<body>
    <script>
        let sum = (a, b) =>
            {
                return a+b;
            }
        let mul = (a, b) =>
            {
                return a*b;
            }
        let calculate = (a, b, callback) =>
            {
                let result = callback(a, b);
                return result;
            }
        let output = calculate(20, 20, sum);
        console.log(output);
    </script>
</body>
</html>
```

### For Each :-

- In Java for each is a loop.
- But in JavaScript forEach is a function.

### Syntax :-

```
array.forEach(function(another function as  
argument))  
{  
    print statement/code;  
}
```

Example :- let birds = ['sparrow', 'hen', 'crow', 'eagle'];

```
birds.forEach(function(bird))
```

```
{  
    console.log(bird);  
}
```

→ O/P will be  
same for both.

### PROGRAM :-

```
<body>  
<script>  
    //for Each  
    let birds = ['sparrow', 'parrot', 'peacock', 'hen'];  
    birds.forEach((bird) =>  
        {  
            console.log(bird);  
        }  
    );  
</script>  
</body>
```

PROGRAM :- {for-in Loop} Example :-

```
<!DOCTYPE html>
<html>
<head>
    <title> JS Array Loops </title>
</head>
<body>
<script>
    let birds = ['sparrow', 'parrot', 'Eagle', 'hen'];
    // normal for loop
    for (let i=0; i<birds.length; i++) {
        console.log(birds[i]);
    }
    // for-in loop
    for (let bird in birds) {
        console.log(birds[bird]);
    }
    // for-of loop (ES6)
    for (let bird of birds) {
        console.log(bird);
    }
</script>
</body>
</html>
```

for  
for Each : -  
→ In Java for  
→ But In Java  
→ Syntax : -

Example :

PROGRAM  
<body>  
<script>
 // for
 let
 bir
 {
 }
</script>
</body>

## Inner HTML

### innerHTML

ul Text  
is used to display output same as it code. So, it displays HTML text.

ed;"> Good </span>  
ue;"> Morning

(' #msg ').innerText  
g").innerHTML

## PROGRAM

```
<!DOCTYPE html>
<html>
<head>
<title> JS List </title>
</head>
<body>
<div id="list">
</div>
<script>
let colors = ['white', 'red', 'black', 'blue', 'green'];
let output = `<ul>\n`;
for (let i=0; i<colors.length; i++) {
    output += `<li> ${colors[i]} </li>\n`;
}
output += `</ul>`;
console.log(output);
document.querySelector('#list').innerHTML = output;
</script>
</body>
</html>
```

### #. Inner Text

#### innerText

- Normal Text
- It is used to display normal text (whatever is written inside tag exact same text)

### Inner HTML

#### innerHTML

- HTML Text
- It is used to display the output same as HTML code. So, It displays HTML text.

### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> JS Inner Properties </title>
  </head>
  <body>
    <div id="msg">
      </div>
    <script>
      let htmlSnippet = `<h1>
        <span style="color:red;"> Good </span>
        <span style="color:blue;"> Morning
        </span>
      <h1>
    `;
      // document.querySelector('#msg').innerText
      // = htmlSnippet;
      document.querySelector('#msg').innerHTML
      = htmlSnippet;
    </script>
  </body>
```

PROGRAM

<!DOCTYPE  
<html>  
<head>  
</head>

<body>  
<div>  
</div>

let  
let  
for  
if

<script>  
<body>  
</html>

## JS Arrays

Syntax :- let arrayName = [ ..... ];  
                  {  
                  };

Example of Array

<body>

```
let colors = ['white', 'red', 'black', 'blue', 'green'];
console.log(colors);
console.log(colors[2]);
console.log(colors[6]); //undefined
```

</script>

</body>

</html>

PROGRAM

<body>

<script>

```
let colors = ['white', 'red', 'black', 'blue', 'green'];
```

```
let output = ` `;
```

```
for (let i=0; i<colors.length; i++) {
```

{

```
// console.log(colors[i]);
```

```
output += `${colors[i]} `;
```

}

```
console.log(output);
```

</script>

</body>



### Set Timeout :-

If you want to call a function after some time.

```
<script>
let count = 0;
let sayWishes =>
```

```
{ count ++
console.log(`Happy Anniversary : ${count}`);
console.log(new Date().toLocaleTimeString());
```

```
{ console.log(new Date().toLocaleTimeString());
```

```
// sayWishes();
```

or

```
// setInterval (sayWishes, 1000); {** setInterval is a pre-defined function }
setTimeOut (sayWishes, 3000);
```

```
</script>
```

```
</body>.
```

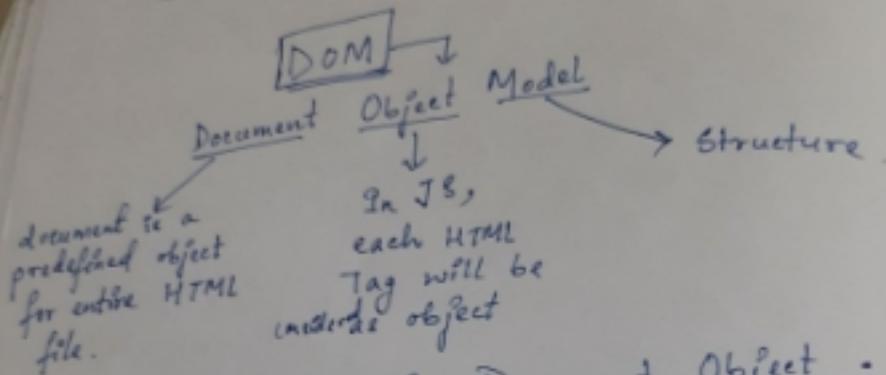
X

→ console.log(new Date().toLocaleTimeString());
interval = setInterval (sayWishes, 1000);
setTimeout ( () =>

```
{ clearInterval (interval);
```

```
}, 5000);
```

CRUD - Create, Read, Update, Delete



### B7. Predefined Methods in Document Object :-

- ① document.getElementById();
- ② document.getElementsByTagName();
- ③ document.getElementsByClassName();

### Query Selector Method :-

Syntax :- document.querySelector(' '');

→ This can be used for any kind of Selector.

20/02/21

if  
body  
script  
for  
let  
{  
}

OR

</s  
</body  
→

```
<h2 id="display"> Input </h2>  
</div>  
<script>  
let inputTag = document.querySelector('#inputtag');  
inputTag.addEventListener('keyup', function() {  
    let textEntered = inputTag.value;  
    document.querySelector('#display').innerText =  
        textEntered;  
});  
</script>  
<body>  
</body>
```

used to attach  
handler.

element.

### Anonymous Function :-

- If we are creating a function without a function  
logic name, those functions are known as Anonymous  
functions.

### Call-back Function :-

- If we are passing a function as an argument for  
another function, those are known as call-back  
functions.

\* jQuery → used for → simplifying  
↓ forms.

### PROGRAM :-

```
<body>
  <div>
    <input type="button" value="Submit" />
  </div>
</body>
<html>
  <head>
    <title> Form </title>
    <style>
      div {
        background-color:
```

```
        {
          <style>
            <head>
              <body>
```

J.S Event Listeners:- It is a method to attach an event handler.

- Get the HTML Element
- Attach an JS Event for selected HTML element.
- Write a function for event handling.

PROGRAM :-

```
<body>
  <div>
    <h2 id="h1tag">Message </h2>
    <button id="gm">Good Morning </button>
    <button id="ga">Good Afternoon </button>
    <button id="ge">Good Evening </button>
  </div>
<script>
  // JS Event Listeners
  let GmBtn = document.querySelector('#gm');
  GmBtn.addEventListener('click', function() {
    displayMessage('Good Morning', 'red');
  });
  let GaBtn = document.querySelector('#ga');
  GaBtn.addEventListener('click', function() {
    displayMessage('Good Afternoon', 'blue');
  });
  let GeBtn = document.querySelector('#ge');
  GeBtn.addEventListener('click', function() {
    displayMessage('Good Evening', 'yellow');
  });

```

\* Anonymous Function

→ If we are creating logic names, functions. ↳

\* Call-back Function

→ If we are creating another function.

\* jQuery →

PROGRAM :-

```
<body>
  <div>
    <!DOCTYPE html>
    <html>
      <head>
        <title> J
        <style>
          div {
            b
            {
              l
              style
            }
            l
            head
            body
          }
        </style>
      </head>
      <body>
```

```
<script>
let goodMorning = () =>
{
    let headingTag = document.querySelector('#h1tag');
    headingTag.innerText = 'Good Morning';
    headingTag.style.color = 'red';
}
let goodAfternoon = () =>
{
    let headingTag = document.querySelector('#h1tag');
    headingTag.innerText = 'Good Afternoon';
    headingTag.style.color = 'blue';
}
let goodEvening = () =>
{
    let headingTag = document.querySelector('#h1tag');
    headingTag.innerText = 'Good Evening';
    headingTag.style.color = 'yellow';
}
```

```
</script>
```

```
<body>
```

```
<h1>.
```

Note :-  
# click → JavaScript event → means any action which we are performing on browser.

ping

now()

**JS Event** :- Any action which we are performing on browser.

PROGRAM :-

```
<!DOCTYPE html>
<html>
<head>
<title> JS DEM </title>
<style>
body
{
    text-align: center;
}
div
{
    background-color: limegreen;
    padding: 10px;
}
div button
{
    border: none;
    padding: 10px;
    border-radius: 10px;
}
</style>
</head>
<body>
<div>
<h2 id="h1tag"> Message </h2>
<button onclick = "goodMorning();"> Good Morning
</button>
<button onclick = "goodAfternoon();"> Good Afternoon
</button>
<button onclick = "goodEvening();"> Good Evening
</button>
</div>
```

**script**  
let go  
{ let  
bread  
bread  
}  
let go  
{ let  
bread  
bread  
}  
let go  
{ let  
bread  
bread  
}  
let go  
{ let  
bread  
bread  
}

</script>  
</body>  
</html>

\* Note :-  
\* onclick

## Unary Operator :-

Example :-

let m = 1;

// m = m + 1;

// m += 1; → (Assignment Operator)

m++; (post increment)

console.log(m); → op:- 2

X

{  
if (condition)  
}

{  
=

{  
if (condition)  
}

{  
=

condition should  
true

{  
else  
}

{  
=

{  
}

{  
if (condition)  
}

{  
=

{  
else if (condition)  
}

{  
=

{  
else  
}

{  
=

→ Logical Operators :-

Example :-

① let isRelation = true;  
let parentAgreed = false;  
if (isRelation || parentAgreed) // any one condition true  
{  
 console.log ('Get Marry Soon');  
}  
else  
{  
 console.log ('Wait');  
}

Op:- Get Marry Soon

② let isRelation = true;  
let parentAgreed = false;  
if (isRelation & & parentAgreed) // both condition should  
{  
 be true  
 console.log ('Get Marry');  
}  
else  
{  
 console.log ('Wait');  
}

Op:- Wait

### Example of Relation Operator :

$==$  → double equals-to operator checks only the values and it won't check the datatypes.

$== =$  → triple equals operator checks both value as well as datatypes.

Ex:-

```
① let x = 35;  
let y = '35';  
if (x == y) // true  
{  
    console.log ('Fail: Prepare Well');  
}  
else  
{  
    console.log ('Pass');  
}
```

O/P:- Fail: Prepare Well

```
② let x = 35;  
let y = '35';  
if (x === y) // false  
{  
    console.log ('Fail');  
}  
else  
{  
    console.log ('Pass');  
}
```

O/P:- Pass

→ null datatype :-

\* null is a dummy value.

null ≠ 0

null ≠ " ";

null ≠ null.

let mName = null;

Operator

→ typeof Operator :-

Typeof is a special operators which is used to check what type of value we are assigning to a variable.

### OPERATORS

1). Arithmetic Operators → +, -, \*, /, %

2). Relational Operator → >, <, >=, <=, !=, ==, ===

3). Logical Operator → &, ||

4). Unary Operator → ++, -- (Increment & decrement)

5). Assignment Operator → syntax :-

6). Assignment Operator → condition ? : -

→ Arithmetic :-      ↓      ↓  
concatenation Example: -      if True      if false.

Let a = 10;

Let b = 20;

Let result = a + b;

console.log(`Addition of A & B is : \${result}`);

console.log(`Addition of a & b is: \${result}`);

console.log(`A value is : \${a}`);

concatenation

Example of Re

" " → double  
It won't

" " → triple  
data

Ex:-

① let x =

let y =

if (x ==

y

else

y

else

② let x =

let y =

if (x ==

y

else

y

else

y

of value,  
e.  
data).

### String datatype :-

→ String datatype is used to store characters or a group of characters.

\* In JS, string can be enclosed within single quotes or double quotes & also back tick (`-`);

#### Example:-

```
let name_1 = 'a', OR 'a' { - } → back tick  
console.log(typeof name_1);  
let name_2 = "abc".  
console.log(typeof name_2);
```

### Boolean datatype :-

\* It is used to store true or false values.

\* 'true' & 'false' values should be written in lower case.

#### Example:-

```
① let isManager = TRUE; X  
    console.log(typeof isManager); * // Error
```

```
② let isManager = true; ✓  
    console.log(typeof isManager); // op:- boolean.
```

### Undefined datatype :-

\* If we declare a variable without assigning the value then the op will be undefined.

#### Example:-

```
let isManager;  
console.log(typeof isManager); // undefined.
```

13/02/21

### Note :-

- For some variables, we can assign any type of value, because JS is weakly typed programming language.  
(no need to mention which type of data).

### Program :-

```
<body>
  <!-- Script code -->
  <script>
    let num-1 = 'Siva';
    console.log(num-1);
  </script>
</body>
```

### {HTML}

## # Data Types

- It is of 2-types : - 1) Primitive and  
2) Non-Primitive / derived.

### 1) Primitive Datatype

- Number
- String
- boolean
- undefined
- null

### Non-Primitive Datatypes

- Array
- Object
- RegEx

- number datatype : - Number is a datatype which is used to store numeric values (both integer & decimal).

```
<body>
  <!-- Script code -->
  <script>
    let num-1 = 10;
    console.log(typeof num-1);
    let num-2 = 10.0;
  </script>  console.log(typeof num-2);
```

### String data

- \* String data of character
- \* In JS, a double q

### Example

```
let
const
let
const
```

### boolean

- \* It is used
- \* 'true' & 'false'

### Examples

① let a=10  
console.log(a)

② let  
const

→ Undefined  
\* If we  
then

### Example

ing purpose  
or

### Rules of JS Variables :-

→ JS Variables are case sensitive.

let a = 10;

console.log(a); // 10

console.log(A); // A is not defined.

→ While declaring variables, reserved words (or keywords) cannot be used.

let if = 10; X

let true = 20; X

→ While declaring the variable, always variable names should start with alphabets and not numbers.

let abc1 = 10; X // error → unexpected token.

let abc1 = 20; ✓

→ Special characters should not be used while declaring variables except dollar (\$) and underscore(\_) symbol.

let abc\_1 = 10; ✓

let \$abc = 20; ✓

let be@d = 30; X

→ While declaring the variable names, if we are taking multiple words, then follow the ~~the~~ <sup>capital</sup> case.

let employeeName = 'Manju';

son  
gation.

- case.  
alization

classmate

# Console Messages : → console is used for checking purpose whether the code is working or not.

- ↳ console.log(a+b);
- ↳ console.error(a+b);
- ↳ console.warn(a+b);
- ↳ console.info(a+b);

Example Program :-

<!-- Script Code -->

<script>

let a = 10;  
let b = 20;

// Console Messages

- console.log(a+b);
- console.error(a+b);
- console.warn(a+b);
- console.info(a+b);

</script>

</body>

---

JavaScript

---

Variables

\* Syntax :- Keyword variable-name ; → Declaration  
variable name = value ; → Initialization.

\* Keywords are → let  
→ var  
→ const.

\* All the keywords must be written in lower-case.

\* Example :- ①. let a = 10; → declaration & initialization  
in same line.  
②. let b; → declaration  
b = 20; → initialization.

Rules of J.S  
→ J.S Variable

→ While decl  
cannot be

→ While dec  
should be

→ Special  
variable

→ While  
multiple

→ Internally JS :-

In internal JS, it is advisable to write the script code before closing the body tag.

Q. PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Basic JS </title>
  </head>
  <body>
```

<!-- Script Code -->

<script>

document.write ('Welcome to JS')

</script>

object → predefined method

</body>

to print anything on  
webpage.

</html>.

Note :-

\* document → object

\* write → predefined method of document object.

→ used to print anything on webpage.

\* JS is case sensitive.

\* JS is weakly typed programming language.

12/02/21

## JAVASCRIPT

### JavaScript (J.S) :-

- JavaScript was developed by Brendon Eich in the year 1994, at Netscape Navigator company.
- The initial name of J.S. was liveScript. (To get a good fame of liveScript, they changed to javascript.)
- J.S is used to make dynamic changes on webpages and to perform client side validation.
- It is a weakly typed programming language.

### \* Client Side

- ↳ Javascript
- ↳ Angular
- ↳ React

### \* Server Side

- ↳ Node JS
- ↳ Express JS

### \*. Data Base (DB)

Mongo DB (No SQL)

↓  
JSON

### \* Ways of writing script code to HTML

#### Types of JavaScript

- 1). Inline
- 2). Internal
- 3). External

→ Internal  
In inter  
code be

### EJS. PROGRAM

```
<!DOCTYPE  
<html>
```

    <head>

        <he

        <ba

    <!

    <se

        </

    </ba

        <html>.

### Note :-

\* document

\* write

\* J.S is

\* J.S is

### PROGRAM :-

<body>

===== same as previous.

</body>

```

<button class = "btn btn-outline-white btn-sm">
    <i class = "fa fa-user" /> </i> Signup </button>
<button class = "btn btn-outline-white btn-sm">
    <i class = "fas fa-sign-in-alt" /> </i>
    Login </button>

```

</body>

### BOOTSTRAP CAROUSEL

<div class = "container int-5">

<div class = "carousel slide" data-ride = "carousel"

id = "car" data-interval = "1000" >

controls the speed of slide

in terms of ms  
1000ms = 1 sec.

<ol class = "carousel-indicators" >

<li data-target = "#car" data-slide-to = "0" > </li>

<li data-target = "#car" data-slide-to = "1" > </li>

<li data-target = "#car" data-slide-to = "2" > </li>

</ol>

```
<div>
  <ul class="navbar-nav">
    <li><class="nav-item"><a href="#" class="nav-link">
      <i class="fa fa-home"> </i> HOME </a> </li>
    <li class="nav-item"><a href="#" class="nav-link">
      <i class="fa fa-id-badge"> </i> About </a>
    </li>
    <li class="nav-item"><a href="#" class="nav-link">
      <i class="fa fa-graduation-cap"> </i>
      COURSES </a> </li>.
```

</ul>

</div>.

</nav>

</body>.

---

~~scribble~~ PROGRAM :-

```
<body>
  <div class="container">
    <a href="#" class="navbar-brand">
      <i class="fas fa-snowflake pr-2" style="font-size: 1.5em; color: #007bff; vertical-align: middle; margin-right: 0.5em;"> </i>
      SPIDER </a>.
    <button class="close navbar-toggler" data-toggle="collapse" data-target="#links" style="font-size: 1.5em; color: #007bff; vertical-align: middle; margin-right: 0.5em;"> </button>
    <span class="navbar-toggler-icon" style="font-size: 1.5em; color: #007bff; vertical-align: middle; margin-right: 0.5em;"> </span>
    <button class="collapse navbar-collapse" id="links" style="font-size: 1.5em; color: #007bff; vertical-align: middle; margin-right: 0.5em;"> </button>
    <div class="collapse navbar-collapse" style="font-size: 1.5em; color: #007bff; vertical-align: middle; margin-right: 0.5em;">
      <ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: 1.2em; color: #007bff; vertical-align: middle; margin-right: 0.5em;">
        <li> — (same as previous). </li>
      </ul>
    </div>
  </div>.
```

08 | 02 | 01

## Bootstrap NavBars

<body>

<div class = "container mt-5">

<!-- Simple Navbar -->

<nav class = "navbar bg-dark navbar-dark">

<div class = "container">

<a href = "#" class = "navbar-brand">

<i class = "fas fa-snowflake pr-2" > </i>

JSPIDER </a>

</div>

<nav>

</body> <!-- Script Code -->

# PROGRAM

<body>

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

<div class = "container">

<a href = "#" class = "navbar-brand">

<i class = "fas fa-snowflake pr-2" > </i>

JSPIDER </a>

```
<div class="fa fa-times-circle"> </i>
</button>
</div>
<div class="modal-body bg-light">
<form>
<div class="form-group">
<input type="text" class="form-control"
placeholder="Enter Username">
</div>
<div class="form-group">
<input type="password" class="form-control"
placeholder="Enter Password">
</div>
</form>
</div>
<div class="modal-footer bg-secondary">
<button class="btn btn-outline-white" type="button">Login </button>
</div>
</div>
<div>
<!-- Script Code --> {should be mentioned here}
<script src=" " type="text/javascript"> </script>
{Same as BS dropdown Buttons Program}
</div>
</body>
</html>
```

```
<!-- Append -->
<body> <div class="container" -><div class="row" -> -
    <div class="input-group">
        <input type="text" class="form-control bg-light"
            placeholder="Enter email id">
    <div class="input-group"> - append ">
        <span class="input-group-text bg-primary
            text-white">@gmail.com </span>
    </div>
    </div>
</div>
</div>
</body>
```

[Q]- Bootstrap Modal :-

## PROGRAM :-

```
<body>
  <div class = "container mt-5">
    <button class = "btn btn-secondary" data-toggle = "modal"
           data-target = "#register">Login</button>
  </div>
```

`<!-- BS Modal -->`

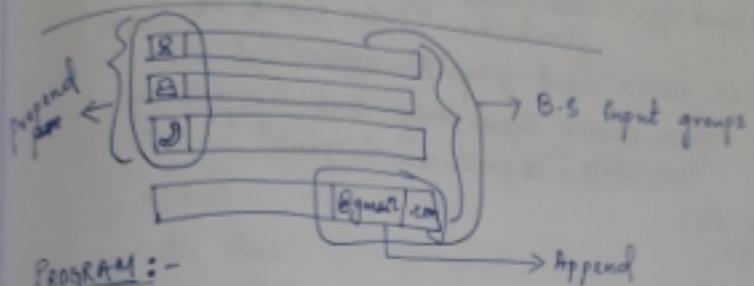
```
<div class="modal" id="register">
  <div class="modal-dialog">
    <div class="modal-content" style="background-color: #f0f0f0; border-radius: 10px; padding: 20px; text-align: center; width: fit-content; margin: auto; border: 1px solid #ccc; box-shadow: 0 0 10px #ccc; font-family: sans-serif; font-size: 16px; line-height: 1.5; color: #333; transition: all 0.3s ease-in-out; opacity: 0.9; z-index: 1000;>
      <div class="header" style="margin-bottom: 10px; border-bottom: 1px solid #ccc; padding-bottom: 5px; font-weight: bold; font-size: 1.2em; color: #333; position: relative; z-index: 1001;>
        <h3> Register Here </h3>
        <button class="close" data-dismiss="modal" style="position: absolute; top: -10px; right: -10px; background: none; border: none; font-size: 1.2em; color: #333; font-weight: bold; cursor: pointer; transition: all 0.3s ease-in-out; z-index: 1002;>
          ×
      </div>
      <div class="body" style="font-size: 0.9em; color: #666; margin-top: 10px; position: relative; z-index: 1001;>
        <form style="width: 100%; position: relative; z-index: 1001;>
          <input type="text" placeholder="First Name" style="width: 100%; height: 40px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 10px; padding: 5px; font-size: 1em; color: #333; position: relative; z-index: 1001;>
          <input type="text" placeholder="Last Name" style="width: 100%; height: 40px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 10px; padding: 5px; font-size: 1em; color: #333; position: relative; z-index: 1001;>
          <input type="text" placeholder="Email" style="width: 100%; height: 40px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 10px; padding: 5px; font-size: 1em; color: #333; position: relative; z-index: 1001;>
          <input type="password" placeholder="Password" style="width: 100%; height: 40px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 10px; padding: 5px; font-size: 1em; color: #333; position: relative; z-index: 1001;>
          <input type="password" placeholder="Confirm Password" style="width: 100%; height: 40px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 10px; padding: 5px; font-size: 1em; color: #333; position: relative; z-index: 1001;>
          <button type="button" style="width: 100%; height: 40px; border: 1px solid #007bff; border-radius: 5px; background-color: #007bff; color: white; font-size: 1em; font-weight: bold; text-decoration: none; padding: 0; margin: 0; position: relative; z-index: 1001;>
            Register
        </form>
      </div>
    </div>
  </div>
</div>
```

c = "dropdown")  
> {/button},  
42 </a>  
(/a)  
(/a)  
(a)

Q. Horizontal form :-

UserName  
Password  
Login

} 768 px min size.

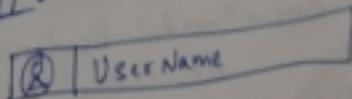


PROGRAM :-

<!-- Prepend -->

```
<body>
  <div class = "container mt-5">
    <div class = "row">
      <div class = "col-md-6">
        <div class = "input-group">
          <div class = "input-group-prepend">
            <span class = "input-group-text bg-warning">
              <i class = "fa fa-user-circle"></i> <span>
            <input type = "text" class = "form-control bg-light"
                  placeholder = "Enter Username">
          </div>
        </div>
      </div>
    </div>
  </body>
```

Output:-



e 768 px  
n size.



# <!-- Large & Small Buttons -->

```
<div>
  <button class="btn btn-secondary">Button</button>
  <button class="btn btn-secondary btn-sm">Button</button>
  <button class="btn btn-warning btn-md">Button</button>
  <button class="btn btn-warning btn-large">Button</button>
  <button class="btn btn-warning btn-block">Button</button>
</div>
```

# <!-- Button-Group -->

```
<div class="btn-group">
  <button class="btn btn-primary">One</button>
  <button class="btn btn-primary">Two</button>
  <button class="btn btn-primary">Three</button>
<div class="btn-toolbar btn-warning">
  <div class="btn-group">
    <button class="btn btn-primary">One</button>
    <button class="btn btn-primary">Two</button>
</div>
</div>
```

# <!-- /a -->

```
<div class="btn-group">
  <button class="btn btn-primary">Three</button>
  <button class="btn btn-primary">Four</button>
</div>
```

</div>

classmate



## PROGRAM :- (Image Example)

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-4">
        <div class="card">
          
          → to get default size of
          <div class="card-body bg-light"> image.
          <h2 class="card-title"> ICE CREAM </h2>
          <p class="card-text"> -- -- --
```

</p>

```
<button class="btn btn-primary"> Read
  More </button>
```

</div>

</div>

</div>

</div>

</div>

</body>

ht  
0

1

-2

-3

-4

→ For margin top and bottom, we write as:-

mg - 0  
mg - 1  
mg - 2  
mg - 3  
mg - 4  
mg - 5

PROGRAM:-

<body>  
<div class  
<div cl  
<div

→ For margin left and right both, we write as:-

mx - 0  
mx - 1  
mx - 2  
mx - 3  
mx - 4  
mx - 5

→ for default/automatically set margin, we write

m - auto

\* Padding :-

| <u>all</u> | <u>top</u> | <u>bottom</u> | <u>left</u> | <u>right</u> | <u>top</u> | <u>bottom</u> | <u>left</u> | <u>right</u> |
|------------|------------|---------------|-------------|--------------|------------|---------------|-------------|--------------|
| p - 0      | pt - 0     | pb - 0        | pl - 0      | pr - 0       | py - 0     | py - 0        | px - 0      | px - 0       |
| p - 1      | pt - 1     | pb - 1        | pl - 1      | pr - 1       | py - 1     | py - 1        | px - 1      | px - 1       |
| p - 2      | pt - 2     | pb - 2        | pl - 2      | pr - 2       | py - 2     | py - 2        | px - 2      | px - 2       |
| p - 3      | pt - 3     | pb - 3        | pl - 3      | pr - 3       | py - 3     | py - 3        | px - 3      | px - 3       |
| p - 4      | pt - 4     | pb - 4        | pl - 4      | pr - 4       | py - 4     | py - 4        | px - 4      | px - 4       |
| p - 5      | pt - 5     | pb - 5        | pl - 5      | pr - 5       | py - 5     | py - 5        | px - 5      | px - 5       |

```

<div> class = "form-group"
    <label> Password: </label>
    <input type = "password" class = "form-control" />
</div>
</form>
</div>

<div class = "card-footer bg-secondary text-white text-center">
    <button class = "btn btn-outline-white"> Submit </button>
    <button class = "btn btn-outline-white"> Reset </button>
</div>
</div>
</div>
</div>
</body>

```

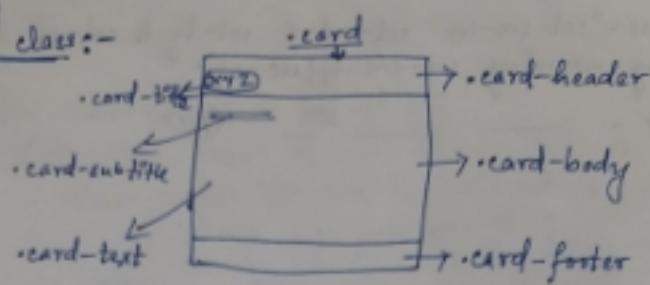
## BS → Margins & Paddings

Margins:-

| <u>All</u> | <u>top</u> | <u>bottom</u> | <u>left</u> | <u>Right</u> |
|------------|------------|---------------|-------------|--------------|
| mt-0       | mt-0       | mb-0          | ml-0        | mr-0         |
| mt-1       | mt-1       | mb-1          | ml-1        | mr-1         |
| mt-2       | mt-2       | mb-2          | ml-2        | mr-2         |
| mt-3       | mt-3       | mb-3          | ml-3        | mr-3         |
| mt-4       | mt-4       | mb-4          | ml-4        | mr-4         |
| mt-5       | mt-5       | mb-5          | ml-5        | mr-5         |

## 2]. Bootstrap Cards :-

\* card class :-



\* animation class :-

To give animation, we use this class.

class = "animated (name)"

↳ bounce  
↳ slide  
↳ fade  
etc.

PROGRAM :-

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-5">
        <div class="card animated bounce infinite">
          <div class="card-header bg-secondary text-white text-center">
            <h3 class="card-title">Login Here </h3>
          </div>
          <div class="card-body bg-light">
            <form>
              <div class="form-group">
                <label>UserName: </label>
                <input type="text" class="form-control" />
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-sm-12 col-md-6 col-lg-4 col-xl-3"
          bg-warning col-sm-offset-3>
        <p> - - - - - </p>
```

lg-4

1p>

lg-4

1p>

PROGRAM :-

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-4 bg-danger
          text-white">
```

<p> - - - - - </p>

<div>

<div class="col-8 bg-light">

<p> - - - - - </p>

<div>

<div class="row">
 <div class="col-6 bg-secondary">

<p>.

<p> - - - - -

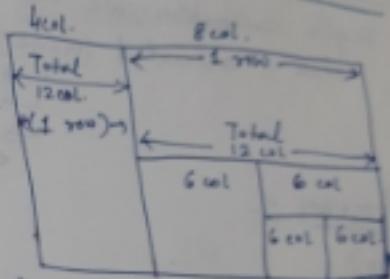
<div>
 <div class="col-6 bg-warning">

<p> - - - - -

<div class="row">
 <div class="col-6 bg-primary">

<p> - - - - -

<div><div><div><div>



bt.

i.e., no.

more.

of

we

To make column & layout Responsive

Example :-

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-sm-12 col-md-6 col-lg-4
        col-xl-3 bg-primary">
        <p> - - - - - </p>
      </div>
      <div class="col-sm-12 col-md-6 col-lg-4
        col-xl-3 bg-danger">
        <p> - - - - - </p>
      </div>
      <div> same as previous-div </div>
      <div> same as previous-div </div>
      <div> same as previous-div </div>
    </div>
  </body>
```

\* Offset class :-

To move any & column layout towards right.

Syntax:- ①. offset - (\*) → star represent column value i.e., no. of columns we want to move.  
It is given in terms of numbers.

②. col-sm-offset-\* ] → To make responsive, we use this syntax.

<body>
 <div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-primary">
 <p> - - - - - </p>
 </div>
 </div>

<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-danger">
 <p> - - - - - </p>
 </div>

PROGRAM

<body>
 <div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-primary">
 <p> - - - - - </p>
 </div>
 </div>

<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-danger">
 <p> - - - - - </p>
 </div>

<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-primary">
 <p> - - - - - </p>
 </div>
 </div>

<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-danger">
 <p> - - - - - </p>
 </div>

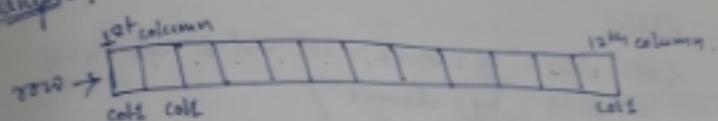
<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-primary">
 <p> - - - - - </p>
 </div>
 </div>

<div class="row">
 <div class="col-sm-12 col-md-6 col-lg-4
 col-xl-3 bg-danger">
 <p> - - - - - </p>
 </div>

## Bootstrap Grid System :-

- To give layouts we use grid system.
- By using grid system, we get layouts of a webpage.
- The class name is row.
- 1 row can have a total of 12 columns. After column division also, each row has 12 columns.

Example:-



- Each column is named as col-1.
- To merge the columns we write as col-2|col-4|col-12 and so on.

Program:-

```
<body>
  <div class="container mt-5">
    <div class="row">
      <div class="col-4 bg-primary"> </p>
      <p> -- -- -- -- </p>
      <div>
        <div class="col-8 bg-danger"> </p>
        <p> -- -- -- -- </p>
        <div>
          <div>
            </div>
        </div>
      </div>
    </div>
  </div>
```

### Example Program :-

```
<body>
  <h1>Screen size : <span id="screen"></span></h1>
  <div class = "container mt-2" style="text-align: center;">
    <h1 class = "text-sm-right text-md-center text-lg-left
      text-xl-right">. Text Alignments </h1>
  </div>
  <!-- Script code -->
  <script>
    window.addEventListener('resize', function () {
      let screenSize = window.innerWidth;
      document.querySelector('#screen').innerHTML = screenSize;
    })
  </script>
</body>
```

Example :-

```
<html>
  <body>
    <div class="container" style="text-align: center;">
      <a href="#" class="bg-secondary d-block text-white">HOME </a>
      <a href="#" class="bg-secondary d-block text-white">ABOUT </a>
      <a href="#" class="bg-secondary d-block text-white">COURSES </a>
    </div>
  </body>
</html>
```

\* Text Alignments :- To adjust the text alignments

class = "text-center"  
↳ "text-left"  
↳ "text-right"

\* Bootstrap Responsive Classes :- Also known as Bootstrap Breakpoints.

- (i). sm (small screen) → mobiles - 501px to 767px.
- (ii). md (medium screen) → tabs - 768px to 992px.
- (iii). lg (large screen) → Laptops/desktops - 992px to 1199px.
- (iv). xl (Extra-large screen) → Television, etc. - 1200px and maximum

\* text-align class : - To give proper alignment to the paragraph.

Bootstrap colors :-

1. Primary - Blue
2. Secondary - purple
3. Info - sky blue
4. Danger - red
5. Success - green
6. Warning - orange
7. Dark - black
8. Light - grey
9. White - white itself.

Syntax :-

text-color name  
btn-color name  
bg-color name  
badge-color name  
table-color name.

Example :-

text-primary.  
btn-secondary/primary.  
bg-primary  
badge-primary  
table-primary

\* Converting block-level to inline & vice-versa:

→ block class = "d-block" → convert to block  
→ class = "d-inline" → convert to inline.  
→ class = "d-inline-block"  
→ class = "d-flex" → to make display flexible  
→ class = "d-none" → to make ~~display~~ no display property.

.mt-5 → margin top value is 5

PROGRAM  
body  
<div  
<a  
<br  
</div  
</body>  
</html>

\* Text A

class = "  
" ↗  
" ↘

\* Bootstrap

(i). sm ( )  
(ii). md ( )  
(iii). lg ( )  
(iv). xl ( )

display classes :-

~~# disp~~ This is used to increase the font size compared to h1 tag font size

- display - 1
  - display - 2
  - display - 3
  - display - 4

→ highest font size  
→ lowest font size.

Example :-

```
<span class="display-1"> Example Heading </span>
```

\* Container class :-

To give padding to an element we use this class.  
In container class, it

+ container-fluid class :- compared to container class, it gives less padding (empty space) to the element.

→ gives only left & right padding. (We won't get top  
and bottom)

→ Aim :- To increase paragraph's font size.  
instead

- \* lead class :- To increase paragraph form
- \* text-muted :- Text color becomes grey instead of black.

Example :-

role :-   
 <p class="lead text-muted"> --> LIP.

08/02/21

## BOOTSTRAP TYPOGRAPHY

Bootstrap Typography :-

↳ content specific styles.

PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <link
      </link>
    <link
      </link>
    <title> Bootstrap Typography </title>
  </head>
  <body>
    <!-- Heading classes -->
    <h1> Example Heading </h1>
    <h2> " " </h2>
    <h3> " " </h3>
    <h4> " " </h4>
    <h5> " " </h5>
    <h6> " " </h6>
    <span class = "h1" > Example Heading </span>
    <span class = "h2" > Example Heading </span>
  </body>
</html>
```

They initially need  
project  
blueprint.  
Select - 5.0 (only JS)

class name.  
single zip file.

will get home  
SS file and link

### Local Method :-

Go to Google → Type MDB and click on first link → Select  
MDB with Jquery → click on download zip → extract the  
file and css.

### Sample Example :-

#### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title> Bootstrap </title>
    <link rel="stylesheet" href="mdb/css/bootstrap.css">
    <link rel="stylesheet" href="mdb/css/mdb.css">
  </head>
  <body>
    <button class="btn btn-primary"> Click Here </button>
  </body>
</html>
```

02/02/21

## BOOTSTRAP

\* Bootstrap :-

→ Invented by Mark Otto in 2011.  
↳ worked in Twitter.

# Before they used Bootstrap, they initially used this framework on twitter project which is called as twitter blueprint.

→ Version :- 4.5.1 (with JQuery) & Latest - 5.0 (only JS)

→ Bootstrap is an open-source-framework.

→ It is used for fast development purpose.

→ It is the file with HTML & CSS.

→ We can use Bootstrap in 2 ways —

i). CDN (Content Delivery Network)

ii). Local Method

\* Steps :- Google

↓  
getbootstrap.com

\* In Bootstrap, we only use class name.

\* MDB Bootstrap comes in a single zip file.

i). CDN :-

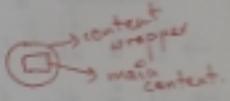
Go to Google → getbootstrap.com → we will get home page → scroll down → Copy the CSS file and link with the HTML file.

ii). Local Method :-  
Go to Google -  
MDB with JQuery file and etc.

Sample Example

PROGRAM :-

```
{!DOCTYPE html>
<html>
  <head>
    <meta
      <title>
        <link
          <link
            </head>
            <body>
              <but
                </body>
                </html>
```



```
@media (max-width: 550px)
{
    #main-nav
    {
        flex-direction: column;
    }
    #nav-items
    {
        margin-top: 5px;
    }
}
#main-content
{
    background-image: url('grp.jpg');
    height: 150vh;
}
#main-content #content-wrapper
{
    background-color: rgba(0, 0, 0, 0.3);
    height: 150px;
    display: flex;
    color: white;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
#main-content #content-wrapper h2
{
    margin: 0px;
}
#main-content #content-wrapper button
{
    background-color: orange;
    padding: 5px;
    outline: none;
    border: none;
    border-radius: 10px;
    box-shadow: 0 0 10px white;
    cursor: pointer;
}
</style>
```

```
<head>
<body>
<nav id=>
<h2><
<insti
<div>
</div>
</body>
</html>
```

## CSS Landing Page :-

### PROGRAM:-

```
<span> </span>
<h1>
<!DOCTYPE html>
<html>
<head>
<title> CSS Landing Page </title>
<style>
body
{
    margin: 0px;
}
#main-nav
{
    background-color: black;
    color: white;
    display: flex;
    padding: 10px;
    align-items: center;
    justify-content: space-between;
}
#main-nav a
{
    color: white;
    text-decoration: none;
    margin: 10px;
    padding: 4px;
}
#main-nav a:hover
{
    background-color: orange;
    color: black;
    border-radius: 5px;
    box-shadow: 0 0 10px white;
}
#main-nav h2
{
    margin: 0px;
}
```

`<html>`

`<head>`

`<body>`

`<h1> ScreenSize : <span id="screen"> </span> </h1>`

`<div>`

`<h2> Small Screen </h2>`

`<h2> Medium Screen </h2>`

`<h2> Large Screen </h2>`

`<h2> X-Large Screen </h2>`

`</div>`

`<!-- Script Code --> [ * case sensitive ]`

`<script>`

```
window.addEventListener('resize', function () {
    let screenSize = window.innerWidth;
    document.querySelector('#screen').innerHTML =
        ` ${screenSize} px`;
});
```

`</script>`

`</body>`

`</html>`

01/02/21

PROGRAM

<!DOCTYPE  
<html>  
<he

@media (min-width: 501px) and (max-width: 600px)

{ body

{ background-color: yellow;

{ div h2:nth-child(2)

{ display: block;

{ }

}

@media (min-width: 601px) and (max-width: 650px)

{

{ body

{ background-color: blue;

{ div h2

{ display: block;

{

}

@media (min-width: 651px)

{

{ body

{ background-color: cyan;

{ div h2

{ display: block;

{ }

## CSS Media Queries :-

Program Ex:-

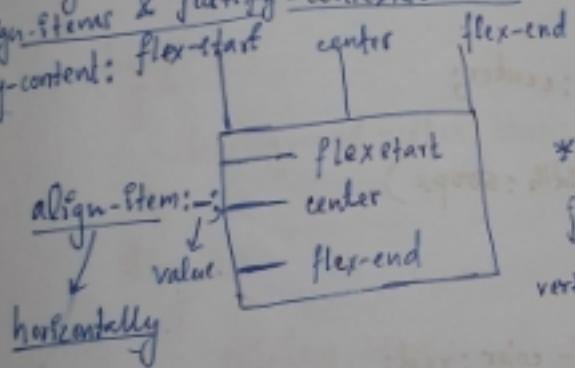
```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS Media Queries </title>
    <style>
      h1 {
        text-align: center;
      }
      h2 {
        display: none;
      }
      div h2 {
        text-align: center;
      }
      @media (max-width: 500px) {
        body {
          background-color: red;
        }
        div h2: nth-child(1) {
          display: block;
        }
      }
    </style>
  </head>
  <body>
    <h1> CSS Media Queries </h1>
    <h2> CSS Media Queries </h2>
    <div>
      <h2> CSS Media Queries </h2>
    </div>
  </body>
</html>
```

```

</style>
</head>
<body>
<div>
  <h1>Item-1 </h1>
  <h1>Item-2 </h1>
  <h1>Item-3 </h1>
</div>
</body>
</html>

```

\* To align text horizontally we use flex-start, center, flex-end  
align-items & justify-content.



\* also we can give  
justify-content: —;  
 vertical space-between;  
 space-around;  
 space-evenly;

```

</head>
<title>
<style>
  div {

```

\* flex-wrap: wrap;  
 no-wrap; → default

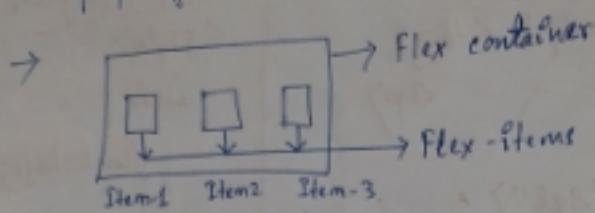
same as previous prgm.

flex-direction: row;  
 align-items: flex-start;  
 justify-content: space-around; +/  
 height: 300px;  
 flex-wrap: wrap;

29/01/21

## CSS FLEX BOX

→ To make any box flexible we have to use the property : → `display: flex;`



→ `flex-direction :`  
Values  
↓  
row  
column  
row-reverse  
column  
column-reverse

### PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS Flexbox </title>
  <style>
    div
    {
      background-color: green;
      padding: 10px;
      display: flex;
      flex-direction: row;
    }
    div h1
    {
      color: red;
      background-color: yellow;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div>
    <h1>Hello</h1>
  </div>
</body>

```

\* To align  
align-items  
justify-content

align  
↓  
horizontal

vertical  
↓  
vertical

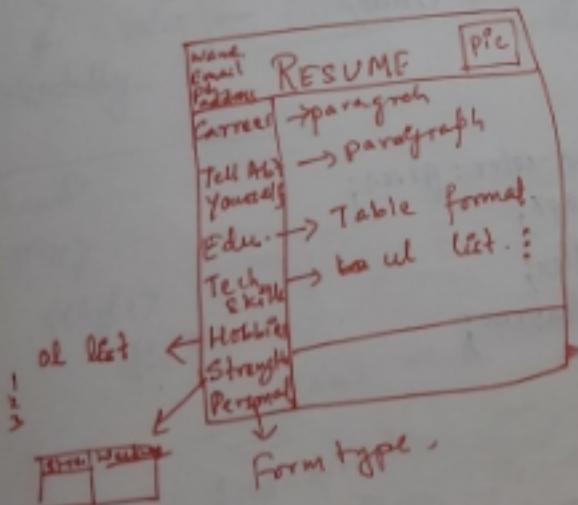
```
<article>
  <h2> JAVA </h2>
  <p> --- </p>
</article>
<article id="script">
  <h2> Script </h2>
  <p> --- </p>
</article>
<article id="sql">
  <h2> SQL </h2>
  <p> --- </p>
</article>
<section>
</body>
</html>.
```

text style appearance

```
<style>
  aside {
    float: left;
  }

```

## ASSIGNMENT



28/01/23

### Continuation of HTML-5 tags Topic

#### SIDE-BAR

→ Aaside Tag:-  
+ → <aside>

#### BODY/Main Section

→ SECTION tag)  
→ article tag

\* footer tag.

→ # PROGRAM :-

continue with previous program

~~<!Sidebar>~~

<!-- Sidebar -->

<div>

<aside>

<a href="#html">HTML </a>

<a href="#css">CSS </a>

<a href="#java">JAVA </a>

<a href="#script">Script </a>

<a href="#sql">SQL </a>

</aside>

<!-- Main Section -->

<section>

<article> id="html" >

<h2>HTML </h2>

<p> — - - - </p>

<article> <h2>CSS </h2>

<p> — — — </p>

<article>

Steps to get Icons Offline :-

1). Go to Google



2). Font-Awesome



3). click on first link



4). click on left side menu icon



5). Click on Start



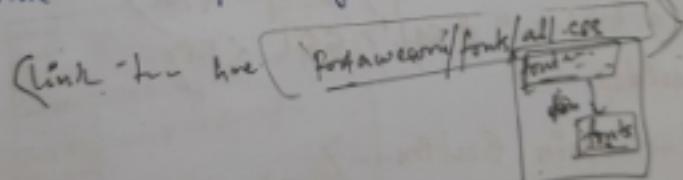
6). Scroll down & Click on Download.



7). Zip file → extract to normal folder



8). Link address file to your HTML.



## # Steps to get Icons ~~OFFLINE~~ ONLINE :-

1). Go to Google



2). Font Awesome



3). click on first link



4). Click on left side menu icon =



5). Click on start



Give your gmail id & verify the  
email.



6). Fill your details (set password, etc. . .)



7). Connected to font-awesome page.



8). Copy the Script Tag & paste it to  
your HTML file.

<nav>

<a href="#"><i class="fa fa-home"></i> HOME

<a href="#">

<i class="fa fa-id-badge"></i> ABOUT

<a href="#">

<i class="fa fa-graduation-cap"></i> COURSES

<a href="#">

<i class="fa fa-cogs"></i> SERVICES

<a href="#">

<i class="fa fa-phone"></i> CONTACT US

</nav>

<body>

</html>

{ Note:-

<i class="fa fa-... "></i>

This above tag is used to give icons from font-awesome from Google

\* Font-Awesome → Google :- To give icons.

<i class=" " ></i> :→ ~~an~~ tag used to give icons.

Ques

```
nav {  
    background: linear-gradient(45deg, indigo,  
                                green, blue);  
    padding: 10px;  
}
```

```
nav a:hover {  
    background: linear-gradient(45deg, orange,  
                                white, green);  
    color: blue;  
    border-radius: 3px;  
    box-shadow: 0 0 10px gray;  
}
```

```
{  
    a {  
        padding-right: 3px;  
    }  
}
```

```
<style>  
<script src="https://kit.fontawesome.com/get the  
path from fontawesome after  
verifying gmail"/>
```

```
<script>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1> <span> style="color: gold;"> JSPIDERS </span> </h1>
```

```
 <span style="color: orange;"> INSTITUTE </span>
```

```
</h1>
```

```
</header>
```

```
<nav>
```

```
    <a href="#">
```

```
        <span>
```

```
            href
```

```
        </span>
```

```
    </a>
```

```
    <a href="#">
```

```
        <span>
```

```
            href
```

```
        </span>
```

```
    </a>
```

```
    <a href="#">
```

```
        <span>
```

```
            href
```

```
        </span>
```

```
</nav>
```

```
<body>
```

```
<html>
```

```
{  
    N  
    L  
    This  
    —
```

```
*
```

```
Font-Awe
```

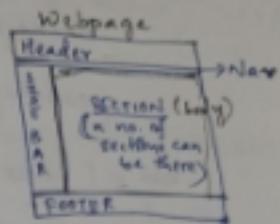
```
{& clas
```

```
zea
```

## HTML-5 TAGS

(SATURDAY WORK)

- div tag:- to create divisions in a webpage
- header tag:- to give heading
- aside tag:- side bar
- section tag:- body part
- footer tag
- article tag.
- nav tag → Helps to set navigation keys like Home, About, etc.



### # PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> HTML-5 TAGS - HEADER & NAV </title>
    <style>
      body {
        margin: 0px;
        text-align: center;
      }
      header {
        background: linear-gradient(45deg, red, green, indigo);
        color: white;
        padding: 16px;
      }
      header h1 {
        margin: 0px;
      }
    </style>
  </head>
  <body>
    <header>
      <h1> Welcome to my website! </h1>
    </header>
    <section>
      <p> This is a sample section. </p>
    </section>
    <nav>
      <a href="#">Home</a>
      <a href="#">About</a>
      <a href="#">Contact</a>
    </nav>
    <footer>
      <p> Copyright © 2023. All rights reserved. </p>
    </footer>
  </body>
</html>
```

should  
be.

50%.

{  
transform: translate Y (-100px);

100%.

{  
transform: translate Y (0px);

}

</style>

</head>

<body>

<div>

<ul>

<li> </li>

<li> </li>

<li> </li>

<li> </li>

<li> </li>

</ul>

</div>

</body>

</html>

27/01/21

→ div tag  
→ header  
→ aside  
→ section  
→ footer  
→ article  
→ nav tag

## # PROGRAM

<!DOCTYPE

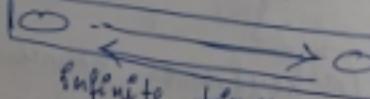
<html>

<head>

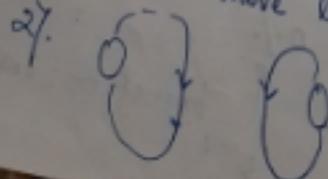
<

## Assignment

1). 100% width block



Infinite times content should move left to right.



```
ul li:nth-child(2)
{
    background-color: yellow;
    animation-delay: 0.2s;
}

ul li:nth-child(3)
{
    background-color: green;
    animation-delay: 0.3s;
}

ul li:nth-child(4)
{
    background-color: skyblue;
    animation-delay: 0.4s;
}

ul li:nth-child(5)
{
    background-color: pink;
    animation-delay: 0.5s;
}

ul
{
    border-bottom: 3px solid white;
    width: 260px;
}

@keyframes bounce
{
    0% {
        transform: translateY(0px);
    }
}
```

# PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Bounce Animation </title>
    <style>
      body {
        background-color: black;
      }
      div {
        position: relative;
        top: 120px;
        left: 180px;
      }
      div ul li {
        display: inline-block;
        background-color: black;
        width: 20px;
        height: 20px;
        border-radius: 50%;
        margin: 10px;
        box-shadow: 0 -7px 20px white;
        animation: bounce 1s infinite;
      }
      ul li:nth-child(1) {
        background-color: red;
        animation-delay: 0.1s;
      }
    </style>
  </head>
  <body>
    <div>
      <ul>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
      </ul>
    </div>
  </body>
</html>
```

20%.

{  
    top: 100px;  
    left: 60px;  
}

40%.

{  
    top: 0px;  
    left: 120px;  
}

60%.

{  
    top: 100px;  
    left: 180px;  
}

80%.

{  
    top: 0px;  
    left: 240px;  
}

100%.

{  
    top: 100px;  
    left: 300px;  
}

}

</style>

</head>

<body>

<div>

</div>

</body>

</html>

# PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Animation </title>
    <style>
      div {
        background: linear-gradient(45deg, red, blue, gray);
        width: 50px;
        height: 50px;
        border-radius: 50%;
        box-shadow: 0 0 10px black;
        border-top: 2px solid black;
        border-bottom: 2px solid red;
        border-right: 2px solid violet;
        border-left: 2px solid green;
        position: relative;
      }
    </style>
  </head>
```

```
  <body>
    <div>
      /* Animation Properties */
      animation: football 2s linear 0s infinite
                  alternate;
    </div>
  </body>
```

@ keyframes football

```
{ 0% {
  top: 0px;
  left: 0px;
}
```

</body>  
</html>

```

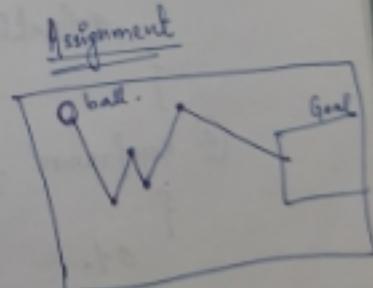
50% {
    top: 100px;
    left: 100px;
    border-radius: 0px 40px 40px 0px;
}

75% {
    top: 100px;
    left: 0px;
    border-radius: 0px 0px 40px 0px;
}

100% {
    top: 0px;
    left: 0px;
    border-radius: 40px;
}

</style>
<head>
<body>
    <div>
        </div>
    </body>
</html>

```



Point / Line  
Line / Area  
Area / Volume

## css Animation Examples

PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS animation </title>
  <style>
    div {
      background: linear-gradient(45deg, yellow, blue, red);
      width: 50px;
      height: 50px;
      position: relative;
      margin-bottom: 20px;
    }
    /* Animation Properties */
    div::after {
      content: '';
      width: 100px;
      height: 100px;
      border-radius: 50%;
      background: linear-gradient(45deg, yellow, blue, red);
      position: absolute;
      left: -50px;
      top: -50px;
      animation: square 2s linear 0s infinite;
    }
    @keyframes square {
      0% {
        transform: rotate(0deg);
        border-radius: 0px;
      }
      25% {
        border-radius: 50px;
      }
      50% {
        border-radius: 0px;
        transform: rotate(45deg);
      }
      75% {
        border-radius: 50px;
      }
      100% {
        border-radius: 0px;
        transform: rotate(90deg);
      }
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

```
@keyframes abc
{
    0% {
        transform: perspective(200px) rotateX(0deg);
        background-color: yellow;
    }
    50% {
        background-color: red;
    }
    100% {
        background-color: green;
        transform: perspective(200px)
                    rotateX(360deg);
    }
}

</style>
</head>
<body>
    <div>
        <p>ANIMATIONS</p>
    </div>
</body>
</html>
```

5). animation-iteration-count :  
indicating the number of times  
animation to be played (1, 2, ..., infinity)

@keyframes

{

0%

{

6). animation-direction :  
Directions for the animation.  
(normal, reverse, alternate, alternate-reverse).

# PROGRAM :-

<!DOCTYPE html>

<html>

<head>

<title> CSS Animation Properties </title>

<style>

div {

background-color: yellow;

color: red;

border: 3px solid blue;

width: 100px;

position: relative;

top: 10px;

left: 300px;

padding: 20px;

/\* Animation Properties \*/

animation-name: abc;

animation-duration: 2s;

animation-timing-function: linear;

animation-delay: 0s;

animation-iteration-count: infinite;

animation-direction: alternate;

}

@keyframes

{

0%

{

</style>

</head>

<body>

<div>

</div>

</body>

</html>

- 20px;

to Y(0 deg);

de Y(0 deg);

### # CSS ANIMATIONS:-

→ We go for animations for continuous transformation.

→ To animate HTML elements we must use @ Keyframes block.

Syntax :- @keyframe @keyframes.name (animation-name)

{ keyframe-block; }

{ keyframe-rules; }

{ keyframe-block; }

{ keyframe-rules; }

}

### @keyframe-block:

→ It is used to default set of keyframe blocks

→ Each keyframe block must be defined in percentage (%)

→ Each keyframe rule must be a CSS property.

### # Animation Properties:-

i). animation-name:  
one or more @keyframe names.

ii). animation-duration:  
time in sec/ms take to complete animation.

iii). animation-timing-function:  
one or more behaviour of animations  
(linear, ease-in, ease-out, ease-in-out)

iv). animation-delay:  
time delay in sec/ms to begin the  
animation.

```

padding: 6px;
border-radius: 20px 0px 0px 20px;
left: 0px;
transform: perspective(200px) rotateY(0deg);
transition: all .2s;
transform-origin: left;

} hover, content
{
    transform: perspective(200px) rotateY(0deg);
}

div .content .text
{
    color: black;
}

<style>
</style>
</head>
<body>
<div>
    
    <div class="content">
        <p class="text"> abc...xyz {</p>
    </div>
</div>
</body>
</html>

```

# CSS ANIMATION

- We go for animation
- To animate HTML

Syntax :- @keyframe  
 keyframe  
 {  
 keyframe  
 {  
 keyframe  
 {  
 }

@keyframe-block

- It is used to define keyframes
- Each keyframe
- Each keyframe

# Animation I

1). animation -

2). animation -

3). animation -

4). animation

25/01/21

### CSS Transform Example :-

# PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS Transform </title>
    <style>
      div {
        width: 200px;
        height: 200px;
        margin: 60px auto;
        box-shadow: 0 0 20px black;
        border-radius: 20px;
        overflow: hidden;
        position: relative;
      }
      div img {
        height: inherit;
        width: inherit;
        border-radius: 20px;
        transition: all 2s;
      }
      div:hover img {
        transform: scale(1.2, 1.2);
      }
      div::content {
        background-color: rgba(255, 255, 255, 0)
        position: absolute;
        width: 45%;
        top: -60px;
      }
    </style>
  </head>
  <body>
    <div>
      <img alt="A small image inside a rounded square container.">
    </div>
  </body>
</html>
```

```
transition: all 2s;
filter: grayscale(100%);  
}  
div:hover img {  
    transform: scale(1.3, 1.3);  
    filter: grayscale(0%);  
}  
div #wrapper {  
    background-color: rgba(0, 0, 0, 0.8);  
    position: absolute;  
    top: -50px;  
    left: 0px;  
    height: 100%;  
    transition: all 3s;  
}  
div:hover #wrapper {  
    transform: translateY(-100%);  
}  
</style>  
</head>  
<body>  
<div>  
    <img alt="path/address">  
    <div id="wrapper">  
        </div>  
</div>  
</body>  
</html>
```

25/01/21

# PROG

<!DOCTYPE

<html>

<

### # Transform-origin :-

→ It is used for fixing & giving some imaginary settings at a point.

Example :-

`div {`

`transform-origin: right;`

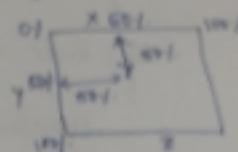
`}`

`div: hover {`

`transform: perspective(200px)`

`rotateY(-90deg);`

`}`



### # PROGRAM :-

`<title> CSS Transform </title>`

`<style>`

`div {`

`box-shadow: 0 0 10px black;`

`border: 3px solid black;`

`width: 300px;`

`height: 200px;`

`margin: 50px auto;`

`border-radius: 20px;`

`overflow: hidden;`

`position: relative;`

`}`

`div img {`

`width: inherit;`

`height: inherit;`

`border-radius: inherit;`

### PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS Transform </title>
  <style>
    div {
      background-color: yellow;
      color: red;
      padding: 10px;
      width: 100px;
      border: 3px solid blue;
    }
    /* Transition Properties */
    transition-property: transform;
    transition-duration: 2s;
    transition-timing-function: linear;
    transition-delay: 3s;
  }
  div:hover {
    transform: translateX(100px);
    background-color: cyan;
  }
  </style>
</head>
<body>
  <div> <p> Transforms </p> </div>
</body>
</html>
```

# Transform  
→ It is  
a  
Example:-

# P

## CSS Transformation Properties :-

### 1-D Transform functions

Note:- x, y values  
can be in px or  
px, but z value  
must be in px.  
→ translate X(x)  
→ translate Y(y)  
→ translate Z(z)

### 2-D Transform functions

→ translate(x,y)

Note:- x,y,z values  
must be in degree.  
→ rotate X(x deg)  
→ rotate Y(y deg)  
→ rotate Z(z deg)

### 3-D Transform functions

→ translate 3d(x,y,z)

Note:- x,y,z must  
be in numbers.  
→ scale X(x)  
→ scale Y(y)  
→ scale Z(z)

Here, z values for  
3D transformation.

→ scale(x,y)

→ scale 3d (x,y,z)

Note:- x, y must be  
in degree.

→ skewX(x)  
→ skewY(y)

→ skew(x,deg,y,deg)

No 3-D.

22/11/24

### Transition Properties:-

→ Transition properties are used to make smooth transitions.  
These properties are:-

- (1) transition-property;
- (2) transition-delay;
- (3) transition-timing-function: linear;
- (4) transition-duration: 6s;

\* For 3-D values, we have to use perspective.

Example:- div:hover {

    transform: perspective(200px);

    translateZ(-100px);

}

\* For 2-D values for x and y :- Example:-

Example:- div:hover {

    transform: translate(100px, 100px);

}

\* For rotate :-

Example:- div {

    transform: perspective(200px)

    rotate(40deg);

}

\* For rotate 3-D, we use:-

Example:- {

    rotate3d(x, y, z, any deg.);

### 2.3 Transformation

#### 2-D Transform Functions

Note:- x, y values  
can be in px or  
%, but z value  
must be in px.

→ translate X(x)

→ translate Y(y)

→ translate Z(z)

Note:- x, y, z values  
must be in degree.

→ rotate X(x deg)

→ rotate Y(y deg)

→ rotate Z(z deg)

Note:- x, y, z must  
be in numbers.

→ scale X(x)

→ scale Y(y)

→ scale Z(z)

Here, z values for  
3-D transformation

Note:- x, y must be  
in degree.

→ skewX(x)

→ skewY(y)

YouTube

Go to channel / site.  
Select the video  
Click on share button  
Click on Embed  
(Scroll down & Enable privacy enhanced mode if any.)  
Copy iframe tag and paste it to HTML file.

360° Images  
Go to pixevid.com

Select Create or Embed a link.  
Create a link.  
Copy the link  
Go to HTML file & copy the link in iframe tag src attribute.

Google Maps  
Go to Google Maps on Browser  
Select the location  
Click on share button  
Click on Embed Map  
Copy the iframe tag and paste it to HTML file.

My Image Convert to 360°

Go to Google (search)  
Memento 360°  
Click on 1st Link.  
First sign-up & verify mail.  
Login & click on camera icon (upload option).  
Select the image & upload.  
Click on Share button.

If this text is going to display in browser, then it will  
not support frameset.

```
<!DOCTYPE>
<html>
  <head>
    <title> framesets </title>
  </head>
  <frameset>
    <frame>
      <body>
```

This browser won't support frameset.

```
    </body>
  </frame>
</frameset>
</html>
```

→ **I Frame Tag** :-

Program :-

```
<body>
```

```
  <h1> Iframe Tag </h1>
```

```
  <iframe src = "frame3.html" width = "350px"
          height = "250px" frameborder = "0px">
  </iframe>
```

```
  <h1> Youtube Video </h1>
```

```
  <iframe width = "350px" height = "250px"
          src = " " >
```

copied from youtube.

```
  <h1> Google Map </h1>
```

```
  /> iframe tag from Google Map.
```

```
</body>
```

YouTube

Go to channel /

Select the vid



click on share



click on Em

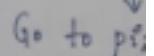
{ Scroll down &  
privacy enhanc  
if any.



Copy iframe  
and paste in  
html file



360° ↗



Go to pic

Select G  
Embed ↴

Create ↴

copy the ↴

Go to HTML  
the link in attrib

### frame 1.html file

```
<body background = "red">  
    frame One  
    <a href = "frame4.html" target = "fr3"> click here </a>  
</body>
```

### Nested Framelets :-

PROGRAM :-

```
<!DOCTYPE>  
<html>  
    <head>  
        <title> Nested framelets </title>  
    </head>  
    <frameset rows = "10%, *, 10%">  
        <frame src = "frame1.html" noresize> </frame>  
        <frameset cols = "20%, * " >  
            <frame src = "frame2.html" > </frame>  
            <frame src = "frame3.html" noresize> </frame>  
        </frameset>  
        <frame src = "frame4.html" noresize> </frame>  
    </frameset>  
</html>
```

### PROGRAM.

```

<!DOCTYPE>
<html>
  <head>
    <title>frames</title>
  </head>
  <frameset cols="200px, *, 300px">
    <frame src="frame1.html" />
    <frame src="frame2.html" />
    <frame src="frame3.html" />
  </frameset>
</html>

```

It represent all the remaining space when we set the value in terms of pixel or %.

### using Target attribute

#### PROGRAM:-

```

<frameset>cols = "20%, *, 30%">
  <frame src="frame1.html" name="fr1">
  <frame src="frame2.html" name="fr2">
  <frame src="frame3.html" name="fr3">
</frameset>

```

frame 1.html

<body>

b

fra

a

<body>

Nested

### PROGRAM :-

```
<!DOCTYPE>
```

```
<html>
```

```
<head>
```

```
<tit
```

```
</head>
```

```
<fram
```

```
<
```

```
<
```

```
<
```

```
<fram
```

```
<
```

```
</fram
```

```
</html>
```

## HTML FRAMES :-

→ We go for HTML frames to run multiple webpages at a time in a browser.

→ We can create multiple frames in one framset by using frame tag.

Example:-

PROGRAM :-

```
<!DOCTYPE>
<html>
  <head>
    <title> HTML first frames </title>
  </head>
```

```
  <frameset cols = "50%, 50%">
    <frame src = "frame1.html" /> <frame>
    <frame src = "frame2.html" /> <frame>
```

```
</frameset>
```

→ [\* <sup>note</sup> When using frameset tag, we don't use body tag.]

```
</html>
```

frame1.html

```
<!DOCTYPE>
```

```
<html>
```

```
  <head>
```

```
    <title> Frame 1 </title>
```

```
  </head>
```

```
  <body bgcolor = "red">
```

Frame One

```
  </body>
```

```
</html>
```

frame2.html

← (Same) ..

```
<body bgcolor = "blue">
```

`</tr>`

`<td align="center" colspan="2">`

`<input type="submit"/>`

`</td>`

`</tr>`

`</table>`

`<form>`

`</div>`

`</body>`

`</html>`

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title>`

`<body>`

`<table border="1" align="center" style="width: 100%; border-collapse: collapse;">`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; background-color: #f2f2f2; border-bottom: 1px solid black; font-weight: bold;">Enter your details`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Submit`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Name`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Address`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Gender`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Age`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Email`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Phone`

`</td>`

`</tr>`

`</table>`

`</body>`

`</html>`

HTML

→ We go for HTML on a browser

→ We can create using form

Example:-

PROGRAM:

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title>`

`</head>`

`<body>`

`<form>`

`<table border="1" align="center" style="width: 100%; border-collapse: collapse;">`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; background-color: #f2f2f2; border-bottom: 1px solid black; font-weight: bold;">Enter your details`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Submit`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Name`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Address`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Gender`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Age`

`</td>`

`</tr>`

`<tr>`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Email`

`<td align="center" colspan="2" style="padding: 5px; border: none; border-bottom: 1px solid black; font-weight: bold;">Phone`

`</td>`

`</tr>`

`</table>`

`</body>`

`</html>`

• `input [type="text"]`, `input [type="password"]`

```
border: none;  
outline: none;  
height: 20px;  
border-radius: 10px;  
color: blue;  
background-color: rgba(255, 100, 0, 0.3);
```

```
}
```

```
</style>  
<head>
```

```
<body>
```

```
<div>
```

```
<form>
```

```
<fieldset>
```

```
<legend> Login Here </legend>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<label> Username: </label>
```

```
</td>
```

```
<td>
```

```
<input type="text" name="un">
```

```
</td>
```

```
<tr>
```

```
<td>
```

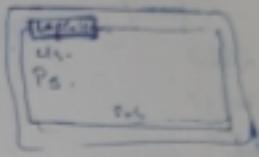
```
<label> Password: </label>
```

```
</td>
```

```
<td>
```

```
<input type="password" name="ps">
```

```
</td>
```



## fieldset & legend Tags :-

fieldset Tag :-

## fieldset & Legend Tags :-

PROGRAM :-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Fieldset & Legend </title>
```

```
<style>
```

```
div {
```

```
background-color: lightgreen;
```

```
width: 300px;
```

```
margin: 100px auto;
```

```
border-radius: 0px 10px 0px 10px;
```

```
padding: 10px;
```

```
{ box-shadow: 0px 0px 10px blue;
```

```
div form { fieldset {
```

```
border: 3px solid blue;
```

```
{ border-radius: 0px 10px 0px 10px;
```

```
fieldset legend {
```

```
border: inherit;
```

```
border-radius: inherit;
```

```
}
```

```
input { type: }
```

```
button {}
```

```
input {}
```

PROGRAM (using select tag): → To give dropdown options.  
→ option tag: → To give options.

<hr>

<div>

<form>

<label> Select a City </label>  
<select>  
<option> ... Select ... </option>  
<option> Bangalore </option>  
<option> Hyderabad </option>  
<option> Pune </option>

</form><select>

</div>.

(SSMATE)

PROGRAM (using datalist tag): -

<div>

<form>

<label> Select a course </label>  
<input list="list">  
<datalist id="list">  
<option> JAVA </option>  
<option> SQL </option>  
<option> HTML </option>  
<option> CSS </option>

</datalist>

</form>

</div>.

→ form method attribute :-

- Method is a form attribute.
- Default value of method is get.
- get method is responsible for getting the ~~method~~ data from server.
- Always, method attribute's value must be post.
- post method is responsible to push the data to the server.

→ action attribute :-

- It is used for server side URL.
- It is also a form attribute.
- The path of the next url page is given in the action attribute when the username & password is submitted.

PROGRAM (using textarea tag) :-

```
<html>
  <body>
    <div>
      <form>
        <label> Enter Feedback : </label>
        <textarea rows="4" cols="20">
        </textarea>
      </form>
    </div>
  </body>
```

Not input  
from.

```
<body>
  <div>
    <form>
      <table>
        <tr>
          <td>
            <label> Username: </label>
          <td>
            <input type="text" />
          </td>
        </tr>
        <tr>
          <td>
            <label> Password: </label>
          <td>
            <input type="password" />
          </td>
        </tr>
        <tr>
          <td colspan="2" align="center">
            <input type="submit" name=" " />
          </td>
        </tr>
      </table>
    </form>
  </div>
</body>
</html>
```

19/01/21

(Bookmark website creation)

20/01/21

## HTML FORMS:-

→ HTML Form Tags:

- \* input
- \* textarea → It defines multiple line text input control. Tag used in forms.
- \* select
- \* option
- \* datalist
- \* label

Form with Alignments :-

PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> HTML FORM TAGS. </title>
    <style>
      div
      {
        background-color: lightblue;
        padding: 20px;
        width: 400px;
        border-radius: 20px;
        box-shadow: 0 0 10px black;
      }
    </style>
  </head>
```

css Paragraph Styles :-

(Example PROGRAM:-

<!DOCTYPE html>

<html>

<head>

<title> CSS Paragraph Styles </title>

<style>

<p>

color: blue;

text-align: justify;

OR  
justify-all;

text-indent: 50px;

font-size: 20px;

word-spacing: 10px;

line-height: 35px;

letter-spacing: 20px;

</style>

</head>

<body>

<p> Abe -

- - - - -

XYZ.

</p>

</body>

</html>

PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS font Properties </title>
    <style>
      h1 {
        color: red;
        text-decoration: underline;
        text-align: center;
        font-size: 50px;
        font-weight: bold;
        font-style: italic;
        font-family: 'comic sans ms', sans-serif;
      }
    </style>
  </head>
  <body>
    <h1> Happy Bhogi </h1>
  </body>
</html>
```

CSS Paragraph  
Example PROGRAM  
<!DOCTYPE Rtm  
<html>  
<head>  
<title>  
<style>

<body>

<h3>HAPPY BHOGI </h3>

<a href="#"> happy bhogi </a>

</body>

</html>

4]. font-size :-

→ It is used to increase or decrease the font of the text.  
→ The value of font-size property is written in terms of pixels. (px).

• Ex :- font-size: 50px;

5]. font-weight: bold;

→ To make the text in bold letters.

6]. font-style: normal;

italic;

→ To make the text italic or keep it normal.

7]. font-family : 'comic sans ms', sans-serif; ] → new

→ To apply different fonts.

sans; ] → old  
monospace; ]  
cursive; → new

### \* Google Fonts

1]. Go to Google and search for Google fonts.

2]. Click on first link & Select the font style.

3]. Click on (+) icon in the right.

4]. Then copy link to HTML file.

5]. Scroll & Take the font family. (Copy & paste)

## \* CSS Text-Formatting Styles:

text-decoration: underline;  
overline;  
line-through;  
none;

27. text-transform: uppercase;  
lowercase;  
capitalize;

37. text-align: center;  
left;  
right;

Program :-

```
<!DOCTYPE html>
```

Chitral 7

*(head)*

## <title> CSS Text-Formatting Styles </title>

<style>

$A_4 \in \{$

color: blue;

text-decoration: underline;

`text-align: center;`

a f

color: red;

`text-decoration: none;`

`text-transform: capitalize;`

</style>

$\langle [head] \rangle$

Lbodj

Libe

47. font  
→ It is  
→ The  
size

57. Fento  
→ 73

### 67. font

→ To

三

10

47. Go

25

三

2

1

display: inline-block;

By using inline-block value for an HTML element, the element will have both the properties (inline as well as block-level).

PROGRAM :-

```
<!DOCTYPE html>
<html>
<head>
<title> Inline-Block Property </title>
<style>
a {
    position: relative;
    top: 20px;
    color: red;
    background-color: yellow;
    display: inline-block;
    width: 150px;
    text-align: center;
    margin-right: 15px;
    padding: 5px;
    border-radius: 10px;
    box-shadow: 0 0 10px blue;
}
</style>
</head>
<body>
<a href="#"> HOME </a>
    <!-- --> CONTACT US </a>
    <!-- --> ABOUT </a>
</body>
</html>
```

27. display: block;

→ By using this value, we can convert inline elements to block-level elements.

PROGRAM:-

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title> Display Block Property </title>
```

```
    <style>
```

```
      a {
```

```
        color: red;
```

```
        background-color: yellow;
```

```
        display: block;
```

```
        margin-top: 10px;
```

```
}
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <a href="#"> HOME </a>
```

```
    <a href="#"> CONTACT US </a>
```

```
    <a href="#"> ABOUT </a>
```

```
  </body>
```

```
</html>
```

37. display: inline-block;

→ By using this value, we can convert block-level elements to inline-block elements.

PROGRAM

```
<!DOCTYPE
```

```
<html>
```

```
  <head>
```

elements

title</title>

before

;

lement

;

(head)  
(body)

15/01/21

### CSS Display Properties :-

\* Display property is used to convert HTML elements to inline elements or block-level elements.

i) display: inline;

→ By using this value, we can convert HTML elements to inline elements.

#### PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Display Inline Properties </title>
    <style>
      h1 {
        color: red;
        background-color: yellow;
        display: inline;
        width: 100px; // width cannot be applied for
        inline elements. w/
      }
    </style>
  </head>
  <body>
    <h1> Happy Bhogi <h1>
    <h1> Happy Bhogi </h1>
    <h1> Happy Bhogi </h1>
  </body>
</html>
```

47. Position: fixed; :-

→ By using fixed position, we can make HTML elements to be fixed at given values.

### Program :-

```
<body>
  <div style="position: fixed; top: 0; left: 0; width: 100%; height: 100%; background-color: black; opacity: 0.8; z-index: 1;>
    <p>Fixed</p>
  </div>
  <p>A b c . . .<br/> . . .<br/>xyz.</p>
</body>
```

```
<head>
  <title>Fixed Position (Title)
  <style>
    div {
      position: fixed;
      top: 100px;
      left: 400px;
    }
  </style>
</head>
```

### 57. Position: střeky ; :-

→ By using sticky position, we can make HTML element to be stuck/stick at given border values.

```
<style>
    div {
        width: 100%; }
        position: sticky; }
        top: 10px; }

</style>
```

L18  
L19

L1  
L1.html

```
</style>
</head>
<body>
<div id="relative">
    <p> RELATIVE </p>
</div>
</body>
```

</html>

37. position: absolute; :-

→ We go for absolute position whenever we have to change child elements places.

PROGRAM :-

```
<!DOCTYPE html>
<html>
    <head>
        <title> Absolute Position </title>
        <style>
            div {
                position: relative;
                top: 40px;
                left: 200px;
                height: 150px;
                width: 200px;
            }
            #absparent {
                position: absolute;
                top: 10px;
                left: 30px;
            }
        </style>
    <body>
        <div id="absparent">
            <div id="absolute">
                <p> ABSOLUTE </p>
            </div>
        </div>
    </body>
</html>
```

```

    #static {
        position: static;
        top: 20px;
        left: 20px;
    }

</style>
</head>
<body>
    <div id="static">
        <p> STATIC </p>
    </div>
</body>
</html>

```

X

---

2). Position: relative; :-

→ Relative position means we can change the HTML elements from its own place.

PROGRAM :-

```

<!DOCTYPE html>
<html>
    <head>
        <title>Relative Position </title>
        <style>
            div {
                position: relative;
                top: 20px;
                left: 20px;
            }
        </style>
    </head>
    <body>
        <div>
            Same as previous program properties
        </div>
    </body>
</html>

```

## CSS Position Properties :-

- \* By using CSS position properties, we can change the place of an HTML element.
- \* To change the position of an HTML element, we have to use top, bottom, left, right CSS properties along with position property.

### 1) Position: static; :-

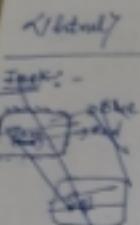
→ static position means we cannot change place of an HTML elements.

#### Ex:- PROGRAM:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Static Position </title>
    <style>
      div {
        background: linear-gradient(45deg, red, blue, green);
        color: white;
        border: 1px solid white;
        width: 150px;
        text-align: center;
        border: 1px solid white;
        padding: 10px;
        border-radius: 20px;
        box-shadow: 0 0 10px blue;
      }
    </style>
  </head>
  <body>
    <div>
      <h1> Welcome to GeeksforGeeks! </h1>
    </div>
  </body>
</html>
```

<body>  
<div>  
    fixed colors

</div>  
</body>



\* Background: radial-gradient  
    give ellipse<sup>or</sup> circle shape of colors.

Syntax:- background: radial-gradient (circle/ellipse, col<sub>1</sub>, col<sub>2</sub>, ..., col<sub>n</sub>).  
    ↳ \* Default is ellipse.  
    ↳ \* we can give either circle or ellipse.

Absolute path :- Image path given using drive information/location.

Relative path :- Image path given by just using image file name with extension.

↳ This is possible only if the image file or the image folder is present in the same folder as of the HTML program.

(\* Complete the Assignment)


12/01/21  
CSS Position Properties

- \* By using CSS position an HTML element.
- \* To change the position use top, bottom, left position property.

Q. Position: static :-  
→ static position means elements.

Ex:- PROGRAM:

<!DOCTYPE html>  
<html>

<head>

<title> Static

<style>

div {

background-

color-

border-

text-

border-

padding-

border-

border-

border-

border-

border-

border-

border-

border-

border-

}

- background-color → To set background color
- background-size → To adjust size of the background color/image.
- background-image → To set background image.
- background-repeat → To stop repeating small images in a webpage.
- background-attachment → To scroll or fix the image in a webpage.
- background-position → To set the position of the background in the webpage.
- Background : linear-gradient → It gives mixed color gradient.

### Linear Gradient

|   |   |   |   |   |
|---|---|---|---|---|
| y | x | R | G | B |
|   |   | ↓ |   |   |

Background : linear-gradient (Direction, col1, col2, ..., coln);

Degree  
↓

to top

to left

to right

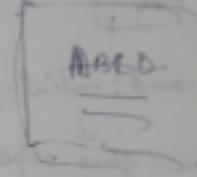
to bottom

to top right

to bottom right

to bottom left

to top left



### PROGRAM:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Mixed Colours <title>

```
<style>
```

```
div {
```

background: linear-gradient (to right, orange,  
blue, green, yellow);

color: red;  
padding: 50px;

```
</style>
```

```
</head>
```

## CSS Background Properties :-

### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> CSS Background Property </title>
    <style type="text/css">
      body {
        color: red;
        background-color: yellow;
        margin: 0px;
        background-image: url('copy url
                            from page');
        background-repeat: no-repeat;
        background-size: cover;
        background-attachment: fixed;
        background-position: 100px 100px;
      }
    </style>
  </head>
  <body>
    <h1> XYZ ABC </h1>
  </body>
</html>
```

{body}

<!--<ht> Powerful people makes places powerful </ht>-->

{/body}.

→ Hexadecimal :-

color : # 0C642B;

$$\frac{0}{8} \quad \frac{6}{8} \quad \frac{2}{8}$$

$$\begin{aligned} & 0 * 16^2 + 6 * 16^1 \\ & = 0 + 96 \\ & = 96 \end{aligned}$$

$$\begin{aligned} & 6 * 16^1 + 4 * 16^0 \\ & = 96 + 4 \\ & = 100 \end{aligned}$$

$$\begin{aligned} & 2 * 16^1 + 8 * 16^0 \\ & = 32 + 8 \\ & = 40 \end{aligned}$$

0  
1  
2  
3  
4  
5  
6  
7  
A → 10  
B → 11  
C → 12  
D → 13

→ HSL :- H - Hue → Degree on color wheel.

S - Saturation → indicates thickness of the color.

L - Lightness → lightness of the color.

Note: \* 'S' and 'L' are represented in terms of percentage.

Example:- {

color : hsl (0, 100%, 50%);

}

09/01/21

### # CSS COLORS :-

(i) Direct Names → red, blue, orange, etc.

(ii) RGB (Red, Green, Blue) → Range - (Range: 0-255)

Ex:- Red: 0-255, Green: 0-255, Blue: 0-255 → Hexadecimal :-  
color: # 000000

(iii) RGBA (Red Green Blue Alpha) → Range - (0.1 - 1.0)  
↳ used for transparency with Alpha value.

00 16 2B  
8 9 6

(iv) Hexadecimal (Ex:- # 00ffff)

(v) HSL (Hue, Saturation, Lightness)

ctrl + shift + /

### Example Program :-

<head>

<style>

Int i;

/\* color: blue; \*/

/\* color: rgb (255, 0, 0); \*/

color: rgba (255, 0, 0, 0.5);

}

</style>

</head>

→ HSL :- H-  
S-  
L-

Note: \* S.

Example:- {

color:

{}

### # OUTLINE :-

- outline will be present surrounding the border.
- It is also used for highlighting the border.

Syntax :- outline: width style color;

style type = "text/ccc">

div {

padding: 20px 50px;

border: 1px solid black;

width: 100px;

height: 150px;

outline: 2px solid orange;

}

Margin Left;

10px

### # Margin :-

- It is a gap between the outline and adjacent element or the browser edge.

→ Margin - top

Margin - left

Margin - right

Margin - bottom

- short-hand form syntax :- margin : top right bottom left  
(Same as padding)

Example :- div {

border: 1px solid black

width: 150px

height: 150px

margin: 20px 50px 35px 15px;

## # Padding :-

- It is the gap between the element and the border.
- We can even choose particular sides of padding such as :-  
padding-top: 20px;  
padding-left: 10px;  
padding-right: 50px;  
padding-bottom: 30px;
- value of padding can be written in terms of px as well as percentage.

→ We can also apply padding by writing in single line called short-hand form.

→ Short-hand form :- Syntax:- padding: top right bottom left;

(electricity)

Ex:- padding: 20px 50px 30px 10px

\* padding: 20px 50px 30px  
top ↓      left & Right      bottom.  
                ↓                    ↓

\* padding: 20px 50px;  
top ↓ & bottom      left & Right.  
                ↓                    ↓

\* padding: 20px;  
                ↓  
                (all sides).

## # Outline :-

- outline will
- It is also used

Syntax:-

list

## # Margin

→ It is a gap or the border.

→ Margin -  
Margin -  
Margin -  
Margin -  
Margin -

→ short-hand

Example

```
border-left-style: double;  
" " " color: blue;  
" " " width: 2px;
```

```
border-right-style: dotted;  
" " " width: 2px;  
" " " color: green;
```

{

\* Border properties can also be written in one single line called as short-hand writing.

Syntax :- border: width style color;  
border-left: width style color;

Example Program :-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Single Line border example </title>
```

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

```
h1 {
```

```
border: 1px solid blue;
```

{

```
h2 {
```

```
border-top: 2px dashed brown;
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> Good Afternoon </h1>
```

```
<h2> Hello </h2>
```

```
</body>
```

```
</html>
```

h3 {

border-style: dashed;  
border-width: 2px;  
border-color: brown;

}

h4 {

border-style: double;  
border-width: 3px;  
border-color: purple;

}

h5 {

border-style: inset;  
border-width: 5px;  
border-color: green;

}

h6 {

border-style: outset;  
border-width: 5px;  
border-color: grey;

}

#xyz {

border-top-style: 3px;  
border-top-width: solid;  
border-top-color: red;

border-bottom-style:

" " width: 3px;  
" " color: brown;

K Borders  
Line ca  
Synt

Example

<!DOCTYPE  
<html>  
<h1>

border

### QUESTION:-

→ Border-width :- It is used to give the thickness of the border.

→ Border-style :- It is used to choose the style of the border.

- style:
  - solid
  - dotted
  - dashed
  - double
  - inset
  - outset

→ Border-color :- It is used to give the colour of the border.

→ Border-radius :- It is used to give curved edges.

→ Border-collapse :- It is used to merge the border with an adjacent border.

Note:-  
We can even choose particular sides of the border as:-  
border-top, border-left, border-right, border-bottom.

Style - Property :- `<style type = "text/css">`

`h1 {`

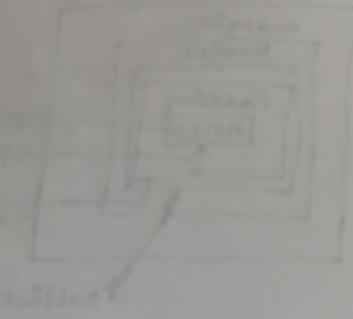
`border-style : solid ;  
border-width : 2px ;  
border-color : red ;  
}`

`h2 {`

`border-style : dotted ;  
border-width : 2px ;  
border-color : blue ;  
}`

border-style: solid;  
border-color: blue;

```
5      </style>     5  
5      </head>  
5      <body>  
5        <div> </div>  
5      </body>  
< </html>  
<
```



QUESTION:-

Border-width :  
Border-style

Q. Write a code to create a circle in the webpage  
by using CSS.

PROGRAM :-

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title> Circle using CSS </title>  
    <style type="text/css">  
      div {  
        width: 100px;  
        height: 100px;  
        background-color: red;  
        border-width: 2px;  
        border-style: solid;  
        border-color: blue;  
        border-radius: 50px;  
      }  
    </style>  
  </head>  
  <body>  
    <div> </div>  
  </body>  
</html> </body>
```

→ Border-color

→ Border-radius

→ Border-style

Note :-  
We can

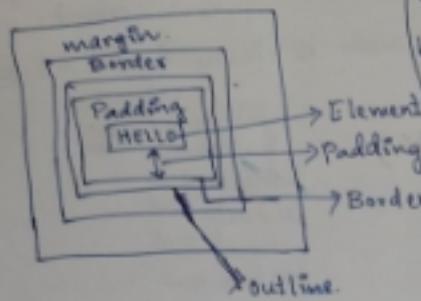
border-

style - Pr

→ OR 50% {  
 border-radius:  
}

08/03/21 (FWD)

## CSS BOX MODEL :-



### Box-Shadow Property

#### Syntax :-

box-shadow: x-axis y-axis  
blurness color;

Ex:-

box-shadow: 0px 0px 10px  
blue;

### # Border Property :-

- border-width
- border-style
- border-color
- border-radius
- border-collapse

- border-top
- border-bottom
- border-right
- border-left

Q. Write the code to create a box in the webpage by using css.

### PROGRAM :-

Laravel

```
<head>
  <title> Square using CSS </title>
<style type="text/css">
  div {
    width: 100px;
    height: 100px;
    background-color: red;
    border-width: 2px;
```

PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Pseudo Selector using Link </title>
    <style type="text/css">
      a:active {
        color: black;
      }
      a:visited {
        color: sky blue;
      }
    </style>
  </head>
  <body>
    <a href="link"> ----- </a>
    <a href = "link" > ----- </a>
  </body>
</html>
```

### 117. Pseudo Selector :-

- Syntax :- selector : event {  
    css property  
}

- pseudo selector is used to apply some css properties when an event occurs.

Example :- `<style type="text/css">`  
`input [type = "text"] : focus {`  
    background-color: yellow;  
}

~~<input>~~  
`input [type = "submit"] : hover {`  
    width: 100px;  
    height: 50px;

`</style>`

Ex :- `tr:nth-child(even) {`  
    background-color: yellow;  
}  
`tr:nth-child(odd) {`  
    background-color: grey;  
}.

### PROGRAM:-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Attribute Selector </title>
    <style> type = "text/css" >
      #form {
        border: 1px solid black;
        margin-top: 150px;
        margin-left: 150px;
        width: 300px;
        height: 200px;
      }
      input [type = "text"] {
        width: 200px;
      }
      input [type = "submit"] {
        width: 70px;
      }
    </style>
  </head>
  <body>
    <div>
      <form>
        <br>
        <br>
        <br>
      </form>
    </div>
  </body>
</html>
```

i) Pseud  
- Syntax

pseud  
when

Ex:

### 97 General Sibling Selector :-

- It is used to select all the siblings of an element.
- It is represented by " ~ " equivalent symbol.

Program :-

```
<!DOCTYPE html>  
<html>
```

```
<head>
```

```
<title> General Sibling </title>
```

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

```
p ~ h1 {
```

```
color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

[same as previous program]

```
</body>
```

```
</html>
```

### 10. Attribute Selector :-

- It is mostly used while creating forms.

Syntax :- input [type = "text"] {

css property

Selector [attribute of selector] {  
                  css property  
                  }

### 8). Adjacent Sibling Selector :-

- Sibling → Elements having common parent are known as siblings.
- Those siblings which are present side by side are known as adjacent siblings. We can apply CSS properties by using adjacent sibling selector.
- We use "+" operator to represent the adjacent sibling.

#### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Adjacent Sibling Selector </title>
    <style type = "text/css">
      P + h1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <div>
      <p> hi please don't talk </p>
      <h1> abc </h1>
      <h1> ask doubt </h1>
    </div>
  </body>
</html>
```

### If General Selector :-

- It is used to
- It is represented by " ".

#### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      <style>
        P {
          color: red;
        }
      </style>
    </head>
    <body>
      <h1> hi </h1>
      <h1> abc </h1>
      <h1> ask doubt </h1>
    </body>
  </html>
```

### 9). Attr.

- It is

S

19/01/22  
7/ Child Selector :-

- It is used to select the direct child of an element.
- It is indicated by the angular symbol ">"

Syntax:-  $\begin{matrix} \text{selector 1} \\ \text{parent} \\ \text{selector} \end{matrix} > \begin{matrix} \text{direct} \\ \text{child} \\ \text{selector} \end{matrix} \{ \text{css property} \}$

}

Program :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Child Selector </title>
    <style type = "text/css">
      #main > #p { color: red; }
    </style>
  </head>
  <body>
    <div id = "main">
      <p> Hi please don't talk </p>
      <div>
        <p> Concentrate here </p>
      </div>
    </div>
  </body>
</html>
```

### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      <style type="text/css">
        * {
          margin = 0px;
          padding = 0px;
        }
        #heading {
          width : 100%;
          height : 70px;
          background-color : blue;
        }
      </style>
    </head>
    <body>
      <div>
        <div id="heading" > </div>
        <div></div>
        <div> </div>
      </div>
    </body>
  </html>
```

10/1/21

Ch

- It is  
It is

PROB

!Doc  
htm

PROGRAM :-

```
<!DOCTYPE html>
<html>
<head>
<title> Descendent Selector Example </title>
<style type="text/css">
#hi {
    color: purple;
}
# hello #hi{
    font-size: 25px;
}
</style>
</head>
<body>
<div><p id="hi"> xyz </p> <div>
<div id="hello"><p id="hi"> abc </p> </div>
</div>
</body>
</html>
```

#### 6. Universal Selector :-

- It represents every element of webpage
- When we want to apply common property for every element in the webpage then we use universal selector.
- It is represented by star "\*" .

### ⇒ Descendent Selector :-

- This will help to select unique element by specifying its parent selector.

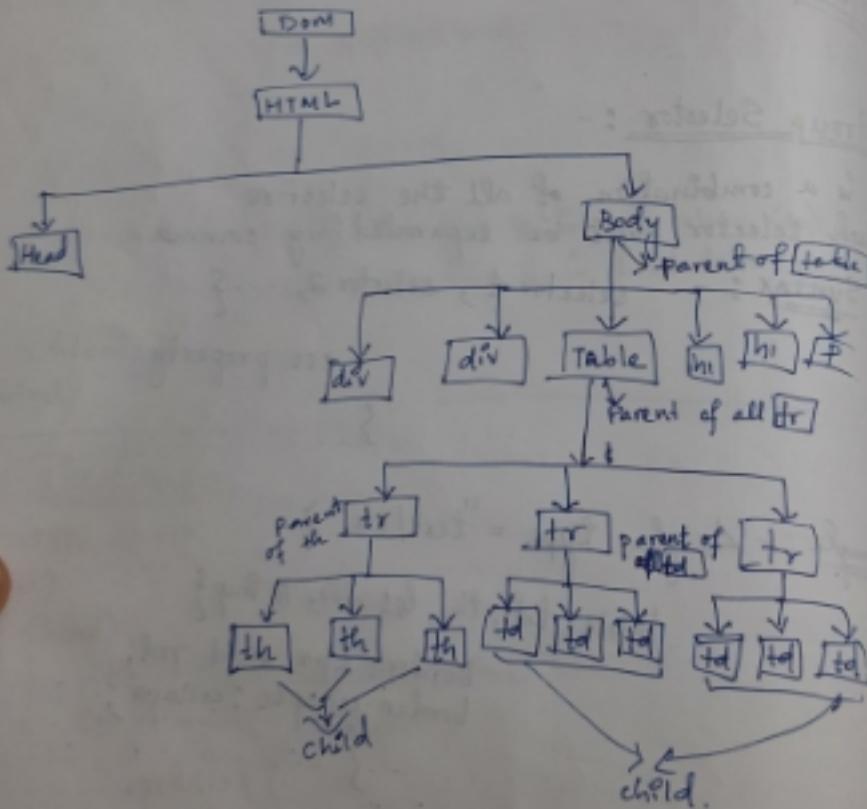
Syntax : selector 1 selector 2 selector 3

css property  
applied only for selector 3.

css property will be {

DOM tree

Document Object Module.



### PROGRAM :-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> Descendent
```

```
<style type="text
```

```
#hi {
```

```
color: #
```

```
}
```

```
# hello #,
```

```
font-
```

```
{
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div> <p id="
```

```
<div id="hello">
```

```
</body>
```

```
</html>
```

### ⇒ Universal Selector

- It represents every element in the webpage
- When we want to apply style to every element in the webpage
- It is represented by \*

```
<html>
  <head>
    <style>
  </head>
  <body>
    <p class="redtext"> This line will be in Red </p>
    <p class="boldtext"> This line is in bold </p>
    <p class="redtext boldtext"> This is in red & bold </p>
  </body>
</html>
```

06/01/21

#### a) Group Selector :-

- It is a combination of all the selectors.
- Each selector must be separated by comma.

Syntax : - selector 1, selector 2, ... {

css property

}

Example : - Let style type = "text/css" }

table, td, th, li, div, #imp {

border: 1px solid red;  
border-collapse: collapse;  
}

</style>

### PROGRAM:-

```

<!DOCTYPE html>
<html>
  <head>
    <title> Id Selector </title>
    <style type = "text/css">
      # redtext {
        color: red;
      }
      # boldtext {
        font-weight: bold;
      }
    </head>
    <body>
      <p id = "redtext"> This line is in red colour </p>
      <p id = "boldtext"> This line is in bold letter </p>
    </body>
  </html>

```

### PROGRAM:-

```

<!DOCTYPE html>
<html>
  <head>
    <title> Class Selector </title>
    <style type = "text/css">
      .redtext {
        color: red;
      }
      .boldtext {
        font-weight: bold;
      }
    </style>
  </head>

```

<head>  
 <body>  
 <p>c</p>  
 <p>b</p>  
 <p>c</p>  
 <1body>  
 <1html>

6/01/21

### i). GROUP :-

- It is a
- Each s

Synt

Example

```
<div>
  <input type="submit" value="Login">
</div>

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

### 27. Id Selector :-

- By using this, we can apply the css property for the web element with particular Id.

Syntax :- #idname {  
 css property  
}.

### 37. Class Selector :-

- A class can be created by using dot operator ". ".
- We can apply the class property by using class attribute.
- We can apply multiple class property on a single element.

```
= "ps"
</div>
```

### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Tagname Selector </title>
    <style type="text/css">
      h1 {
        color: white;
      }
      #Drawback {
        div {
          background-color: black;
        }
      }
    </style>
  </head>
  <body>
    <div>
      <div>
        <h1>Login Page </h1>
        <div>
          <form>
            <div>
              <div>Username: </div>
              <div><input type="text" name="u"/> </div>
            </div>
            <div>
              <div>Password: </div>
              <div><input type="password" name="p"/> </div>
            </div>
          </form>
        </div>
      </div>
    </div>
```

2. Id Selector  
- By using +  
web elem  
Syntax

### 3. Class Selector

- A class selector
- We can select multiple elements
- We can select multiple elements

## \* CSS SELECTORS :-

General Syntax :- selector {     property  
                      }

### Type of Selectors :-

- 1) Tag Name selector
- 2) Id selector
- 3) Class selector
- 4) Group selector
- 5) Dependent selector
- 6) child ~~selectors~~ selector
- 7) Adjacent sibling selector
- 8) general sibling selector
- 9) universal selector
- 10) attribute selector
- 11) pseudo selector

→ CSS selectors will help to select an element from the webpage and apply the CSS property on it.

### 1. Tag Name Selector :-

→ Here, we select an element by using their tagname.  
→ In Tagname selector, CSS property will be applied for every tag with the similar ~~name~~ tagname.

5/1/21

#### \* Block level Tag :-

- These tags which occupies the full width of the web page even though their actual width is small, are known as block-level tags.
- Any element which is written adjacent to block level tag will come in the next line.

Example :- `<div>`    `</div>`  
`<p>`                `</p>`  
`<h1>`              `</h1>`

#### \* Inline Tag :-

- These are the tags which occupy with their actual size/width only.
- Any element which are adjacent to these tags will be displayed in the same line.

Example :- `<img>`  
`<br>`

#### \* "Div" Tag :-

- It is used to create a division in the webpage.
- It is a block level tag.

#### CSS SELECTORS

Syntax :-

#### Types of Selectors :

- 1) Tag Name selector
- 2) Id Selector
- 3) Class Selector
- 4) Group selector
- 5) <sup>dependent</sup> Selector
- 6) Child selector
- 7) Adjacent sibling
- 8) general sibling
- 9) universal selector
- 10) attribute selector
- 11) pseudo selector

→ CSS selectors  
webpage or

#### 1. Tag Name

- Here, we select
- In tag name every tag

`<p>DOB: </p>`  
`<input type="date" name="da">`  
`<p>Ph Number: </p><input type="Number" name="num">`  
`<p>Courses: </p>`  
`<input type="checkbox" name="ja" value="Java" /> Java,`  
`<input type="checkbox" name="sq" value="Sql" /> Sql`  
`<input type="checkbox" name="py" value="Python" /> Python`  
`<input type="checkbox" name="web" value="Web" /> Web.`

`<p><input type="submit" /> </p>.`

`<form>`  
`<body>`

`</html>`

Gender:  Male  Female

M  F

DOB:  /  /

Courses:

HTML  CSS  Java

M  O  P  
 O  M  P

First Name:   
Last Name:   
Mobile:

<http://fl.com/?firstname=Urbij&lastname=Mal&age=20&gender=M>

O-Mal

One after  
0 8

firstname

fn="

"

value="

"

rel="

"

`<input type="text">`

Creates text field.  $\leftarrow$  text  $\rightarrow$  default value  
password  $\rightarrow$  text field but characters are hidden.  
Creates radio button  $\leftarrow$  radio  
checkbox  $\rightarrow$  creates checkbox.

field must be in the email  
Valid email format number  $\rightarrow$  only accepts digits.  
with @ symbol in date  $\rightarrow$  displays the calendar  
the text tel  $\rightarrow$  to set phone numbers only  
displays dropdown box  $\leftarrow$  color  
to choose the color. submit  $\rightarrow$  creates a button which submits  
selects the time  $\leftarrow$  time  $\rightarrow$  the form data on clicking it.  
reset.

$\leftarrow$  used to create a button which will remove all the form data.

### PROGRAM - 18 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Registration </title>
  </head>
  <body>
    <form>
      <p> First Name : </p> <input type="text" name="fname" />
      <p> Last Name : </p> <input type="text" name="lname" />
      <p> Password : </p> <input type="password" name="ps" />
      <p> Email : </p> <input type="Email" name="em" />
      <p> Gender : </p> <input type="radio" name="ge" value="male" /> Male <input type="radio" name="ge" value="female" /> Female
```

<p> DOB : </p>  
<input type="text" />  
<p> Ph Num : </p>  
<input type="text" />  
<p> Courses : </p>  
<input type="checkbox" />  
" " "  
" " "  
" " "  
<p> </p>  
<input type="checkbox" />  
<form>  
<body> .

Gender :  Male

Courses :  
 HTML

First Name :   
Last Name :   
Gender :  Male

```
<body>
  <table>
    <tr>
      <th>Id </th>
      <th>Name </th>
      <th>Marks </th>
    </tr>
```

```
    <tr>
      <td> 101 </td>
      <td> abc </td>
      <td> 67 </td>
```

```
    <tr>
      <td> 102 </td>
      <td> xyz </td>
      <td> 89 </td>
```

```
  </table>
```

```
</body>
```

```
</html>
```

(vimp) - HTML form Creation :-

→ We can create a form by using form tag.

```
<form>
```

```
  <input type="text" name="username">
```

```
</form>
```

→ We can create input fields by using input tag.

→ Input tag comes with a recommended attribute called type attribute.

Syntax :- type = " " [Input type = " "]

### \* Priority for applying CSS :-

1. Inline
2. Internally
3. Externally

Note :- whenever we apply same property more than once, then the recent value will be applied to the property.

`<h3 style="color:yellow;color:white;colour=pink">`

This is inline CSS `</h3>`.

→ output text will be in pink colour.

X

11/21

### PROGRAM - 17 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title>Table using CSS </title>
    <style type="text/css">
      table, td, th {
        border: 1px solid red;
        border-collapse: collapse;
      }
      table {
        width: 200px;
        height: 150px;
      }
    </style>
  </head>
```

| eno. | Name | Class |
|------|------|-------|
| 1    | John | 10A   |
| 2    | Mike | 10B   |
| 3    | Sara | 10C   |

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

<body>  
<table>

<lib>  
<html>.

vimp).

→ We can

→ We can  
→ Input  
typ

37. href → It specifies the path of external file.  
→ mandatory attribute for link tag.

### 2). Internally Way of Applying CSS:-

→ We apply the css property internally by using style tag  
in head section.

#### PROGRAM-16 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Internal css Example </title>
    <style type="text/css">
      h2 { color: blue; } → property.
      selector → mandatory attribute.
    </style>
  <head>
  <body>
    <h2> This is applied internally and in blue colour </h2>
  </body>
</html>
```

### 3). Inline Way of Applying CSS:-

→ We can apply inline css by using style attribute in  
the same line. (Written in body section)

→ Syntax:-      style = "color: yellow;"  
`<h2 style="color: yellow;"> This text is yellow colour </h2>`



```

<tr>
  1 -> <td> 102 </td>
  -> <td> 102 </td>
  3 -> <td> 71.99 </td>
        <td> 96.99 </td>
      </tr>

<table>
  <tr>
    <td> 102 </td>
    <td> 102 </td>
    <td> 71.99 </td>
    <td> 96.99 </td>
  </tr>
</table>
</body>
</html>

```

**CSS - Cascading Style Sheets**

\* CSS :-

- It stands for Cascading Style Sheets
- It was introduced in 1996.
- Current Version of CSS is CSS-3, which came up in 1999.
- CSS is used to enhance the property of HTML page.
- There are 3-ways to apply the CSS property in HTML.
- They are :-
  - 1. Externally - most recommended.
  - 2. Internally
  - 3. Inline

→ File Extension is - .css

1. External Way of Applying CSS :-

- In this, we write the CSS code in an external file and then we link that external file to HTML by using <link> tag (written under head section of HTML file).

→ Extension = .css.

### 5) rowspan & colspan attributes :-

- 'rowspan' attribute is used to merge the rows for the headings/data.
- 'colspan' attribute is used to merge the columns.

|   |     | 8 Marks |  | 8 Marks |         |
|---|-----|---------|--|---------|---------|
|   |     | 1st Sem |  | 1st Sem | 2nd Sem |
| 1 | 101 |         |  |         |         |
|   | 102 |         |  |         |         |
|   |     |         |  |         |         |

### PROGRAM-14 :-

```

<!DOCTYPE html>
<html>
  <head>
    <title> Merged Table Example </title>
  </head>
  <body>
    <table border="1" rules="all">
      <tr>
        <th rowspan="2">1<sup>st</sup><sup>nd</sup> Sem </th>
        <th rowspan="2">2<sup>nd</sup><sup>nd</sup> Sem </th>
        <th colspan="2">8 Marks </th>
      </tr>
      <tr>
        <th>1<sup>st</sup> Sem </th>
        <th>2<sup>nd</sup> Sem </th>
      </tr>
      <tr>
        <td>101</td>
        <td>abc</td>
        <td>34.99</td>
        <td>€ 56.99</td>
      </tr>
    </table>
  </body>
</html>

```

<tr>  
 1 → <td>  
 — <th>  
 3 — <th>  
 <td>  
 <tr>

↗  
 <table>  
 <tbody>

<tr>

2|2|20

\* CSS :-  
 → It stand  
 → It was  
 → Current  
 1999.  
 → CSS is  
 → There  
 They

→ File Exte

1]. Exte

→ In to  
then  
using  
file

→ Ext

<tr>  
<th> S2d </th>  
<th> SName </th>  
<th> SMark </th>

<tr>

<td> 101 </td>  
<td> abc </td>  
<td> 24.99 </td>

<tr>

<td> 102 </td>  
<td> xyz </td>  
<td> 94.99 </td>

<tr>

<table>  
</table>

</body>

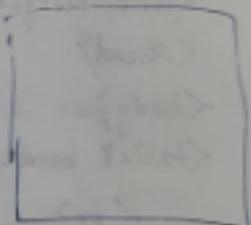
4). Align Attribute :-  
- It is used to give alignment of the table.

align = "center"

align = "left"

align = "right"

| S2d | SName | SMark |
|-----|-------|-------|
| 101 | abc   | 24.99 |
| 102 | xyz   | 94.99 |



### ATTRIBUTES OF TABLE :-

#### 27. Rules attribute :-

- It is used to give ruling for the columns and rows.
- rules = "all" → for both columns & rows
- rules = "rows" → for only rows
- rules = "cols" → for only columns.

#### 27. Border attribute :-

- It is used to give a border for table.
- Its value can be integer which represents thickness of the border.

border = "2"

#### 37. Width & height attribute :-

- It is used to set the size of table by giving the width & height in terms of pixel.

### PROGRAM-13 :-

```
DOCTYPE html>
html>
    head>
        title> Simple Table Example 
    head>
    body>
        table>
            table border = "1" rules = "all" width = "100px"
                height = "100px" align = "center" >
```

(L1) Adv. Python (L2)

(L1) Django (L2).

(L1)

(L1)

(L1)

(body)

(/html).

30/12/20

### TABLE CREATION :-

→ Table can be created by using (table) tag. It is a paired tag.

→ (table) \_\_\_\_\_ } → Root tag for Table.

(/table)

→ Table Row can be created by using (tr) tag.

(tr) \_\_\_\_\_ (/tr).

→ Table data can be given by (td) tag

(td) \_\_\_\_\_ (/td)

→ Table Q. In table, column heading can be given by

(th) tag.

(th) \_\_\_\_\_ (/th)

### \* Nested List :-

→ List inside another list is known as Nested List.

Ex:- To create a nested list like :-

#### o Full Stack Development

1. core Java
2. Adv. Java
3. Web Tech.
4. SQL

#### o Python Package

- a. Python
- b. Adv. python
- c. Django

### PROGRAM-12 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Nested List </title>
  </head>
  <body>
    <ul type="circle">
      <li> Full Stack Development
        <ol>
          <li> Core Java </li>
          <li> Adv. Java </li>
          <li> Web. Tech. </li>
          <li> SQL </li>
        </ol>
      <li>
        <ul type="a">
          <li> Python </li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

### Description List :-

- It is used to give the description for few terms.
- It can be created by `<dl>` tag.
- `<dt>` tag can be used for definition term and `<dd>` tag is used for (description / data) definition.

Syntax:- `<dl>`

`<dt> HTML </dt>`  
~~`<dd> - - - - - </dd>`~~

`<dt> CSS </dt>`

`<dd> - - - - - </dd>`

`</dl>`.

### PROGRAM-11 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title> Description List </title>
  </head>
  <body>
    <dl>
      <dt> HTML </dt>
      <dd> It is used to create a webpage </dd>
      <dt> CSS </dt>
      <dd> It is used for styling the webpage </dd>
      <dt> JS </dt>
      <dd> It is used for front end validation </dd>
    </dl>
  </body>
</html>
```

### PROGRAM-10 :

```
<!DOCTYPE html>
<html>
  <head>
    <title> Unordered List </title>
  </head>
  <body>
    <ul type = "disc">
      <li> HTML </li>
      <li> CSS </li>
      <li> JS </li>
    </ul>
    <br>
    <ul type = "square">
      same as above
    </ul>
    <br>
    <ul type = "circle">
      same as above
    </ul>
    <br>
    <ul type = "none">
      same
    </ul>
  </body>
</html>
```

`<li> JS </li>`  
`<li> Bootstrap </li>`

`<ol>`

`<br>`

`<ol type="a" start="7">`

       same as above  
      

`<ol>`

`<body>`

`</html>`

2. 8. 5. 1. Java  
4. 9. 6. 2. SQL  
3. 7. 3. Web Tech  
2. 8. 4. HTML  
1. 9. 5. CSS

⇒ Reversed Attribute :-

- By using reversed attribute, we can give indexing in Reversed order.
- Ex:- `<ol type="1" start="5" reversed="reversed">`  
`<li> same as above program`

`</li>`

`<ol>`

→ Unordered List :- In this the element will be having bullets like circle, square, etc. [●, ○, □, ♦]

- By using `<ul>` tag, we can create an unordered list.

- Syntax :- `<ul>`

        
        
      `<li>`

`<ul>`

- By default it will be disc '•'

`<br>`  
`<ol type="A">`  
≡ same as above  
`<ol>`  
`<br>`  
`<ol type="I">`  
≡ same  
`<ol>`  
`<br>`  
`<ol type="I">`  
≡ same  
`<ol>`  
`</body>`  
`</html>`

### ⇒ Start Attribute :-

- It is used to start the indexing from a particular value
- The start attribute values can only be numbers.

### PROGRAM - 9

`<!DOCTYPE html>`  
`<html>`  
`<head>`  
`<title> Ordered List with Start Attribute </title>`  
`</head>`  
`<body>`  
`<ol type="1" start="5">`  
`<li>HTML </li>`  
`<li>CSS </li>`

9/12/20  
PROGRAM

TYPE ATTRIBUTE :-

- It is used to change the bullet type in the ordered list.

- Type values can be :-

Type = "1" → number/digits → by default

Type = "A" } → alphabets

Type = "a" }

Type = "i" } → Roman numbers

Type = "I" }

PROGRAM-8 :

```
<!DOCTYPE html>
<html>
  <head>
    <title> Ordered List </title>
  </head>
  <body>
    <ol type = "5">
      <li> HTML </li>
      <li> CSS </li>
      <li> JS </li>
      <li> Bootstrap </li>
    </ol>
```

</body>

<br>

<ol type = "a">

:        (same as above)

</ol>

## \* Image as a Link:

PROGRAM-8 :

```
<!DOCTYPE html>
<html>
  <head>
    <title> Image as a Link </title>
  </head>
  <body>
    <h4> JS spiders </h4>
    <address> Old Airport Road, Murgeshpalya, Near HAL,  
        Bangalore </address>
    <a href = "tel : 9611100311" > <img src = "C :\ Vaish\ooty\>  
        Desktop\call. jpg" width = "50px" height = "50px" >
    </a>
  </body>
</html>
```

## \* List Creation :- 3-types

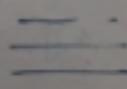
- ↳ Ordered List
- ↳ Unordered List
- ↳ Description List.

→ Ordered List :- It will be having numbers or alphabet or Roman letters as bulletts.

- By using `<ol>` tag, we can create an ordered list.

- Default is Numbers.

Syntax :- `<ol>`



`</ol>`.

12/20

PRO

TYPE

- It is  
List.

- Type

PROG

<!DOCT

<html

same page)

PROGRAM-3:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Engineering </title>
  </head>
  <body>
    <h3> Index </h3>
    <h3><a href="#ch1"> 1. chapter 1 </a> </h3>
    <h3><a href="#ch2"> 2. " 2 " </a> </h3>
    <h3><a href="#ch3"> 3. " 3 " </a> </h3>
    <h3><a href="#ch4"> 4. " 4 " </a> </h3>
```

here to open the

call us <a>

on " > click here

```
<h3 id="ch1"> Chapter 1 </h3>
```

```
<p> abc - - - xyz <br>
    abc - - - xyz </p>
```

```
<h3 id="ch2"> Chapter 2 </h3>
```

```
<p> abc - - - xyz <br>
    abc - - - xyz </p>
```

```
<h3 id="ch3"> Chapter 3 </h3>
```

```
<p> abc - - - xyz <br>
    abc - - - xyz </p>
```

```
<h3 id="ch4"> Chapter 4 </h3>
```

```
<p> abc - - - xyz <br>
    abc - - - xyz </p>
```

</body>

</html>

e window/tab

in new tab.

is in a frame.

08/01/20

Hyper Reference

href → website address  
→ local html page  
→ [Id Name] (Bookmark in same page)  
→ Phone Number.  
→ Mail Id.

PROGRAM-6 :-

```
<!DOCTYPE html>
<html>
  <head>
    <title>Anchor Example 2</title>
  </head>
  <body>
    <a href = "Image Example.html"> click here to open the
      image <a> <br>
    <a href = "tel:9611346595"> click to call us<a>
    <br>
    <a href = "mailto:jspideroar@gmail.com"> click here
      to mail us <a>
  </body>
</html>
```

\* Target Attribute :-

→ It is the attribute of Anchor Tag.

→ It can have the value as:-

(default) → target = "self" → link opens in same window/tab

target = "blank" → link opens in new tab.

target = "iframe" → link opens in a frame.

PROGRAM-7 :

```
DOCTYPE html>
<html>
  <head>
    <title> Engine
  </head>
  <body>
    <h3> Index
    <h5> <a href = "#> abc
    <h5> <a href = "#> abc
    <h5> <a href = "#> abc
    <h3 id = "abc">
      <p> abc
      abc
    </h3>
    <p> abc
      abc
    </p>
    <h3 id = "abc">
      <p> abc
      ab
    </h3>
    <p> ab
      ab
    </p>
    <h3 id = "abc">
      <p> abc
      abc
    </h3>
    </body>
  </html>
```

'alt' Attribute :-

→ It is used to display the alternative text when image is not displayed on the webpage.

- width attribute is used to set the width of the image
- height attribute " " " " " height " " "

\*. ANCHOR TAG :- (VIMP)

→ It is used to create the links in a webpage. We can visit another webpage or open a direct link.

→ Also known as Linked tag

→ It is a paired tag

Lay - ... Lay.

La href = " " 7 content L127.

→ Without href attribute, Anchor tag will not respond.  
→ Hyper Reference

## PROGRAM - 5

```
<!DOCTYPE html>
```

$\text{H}_7 = "B_{\text{S} \text{P} \text{X}}$ "

<internal>

*[head]*

<title> Anchor Example </title>

L (head)

Lbody 7

```
</head>
<body>
    <a href = "http://www.facebook.com"> click here to open
    called inline
    tag. </a>
    <a href = "http://www.youtube.com"> click to open youtube </a>
```

<(body)>

`<html>`

### \* IMAGE TAG :-

→ It is used for including image into the webpage.

Syntax:- <img src = "path" >

To minimize size of image  
tagname ↓ Attribute ↓ value.

<img src = "path" width = "50px" height = "50px" />  
or  
width = "50px" />

Notes:- px → pixel → fixed size.

→ % → percentage → depends on Browser size.

→ But height can't be in terms of percentage. It should be in pixel only.

### PROGRAM - 4

```
<!DOCTYPE html>
<html>
  <head>
    <title> Image Example </title>
  </head>
  <body>
    <img src = "C:\Users\Download\Spiderman.jpg" width = "400px"
         height = "200px" />
  </body>
</html>
```

alt' attribute :-

It is used to display if not displayed

text

width attribute

height attribute

### ANCHOR TAG

→ It is used to visit another

→ Also known as href

→ It is a pair of tags

La href

→ Without href

### PROGRAM - 5

```
<!DOCTYPE html>
<html>
  <head>
    <title> Anchors </title>
  </head>
  <body>
    <a href = "http://www.google.com" > Google </a>
    <a href = "http://www.facebook.com" > Facebook </a>
    <a href = "http://www.youtube.com" > YouTube </a>
  </body>
</html>
```

<p> <mark> This is very important </mark> </p>

<p> Please solve this  $10^{(\log 2)}$  </p>

<p> Solve this  $\log(10) \times 20$  </p>

<p> If I am <big> Big </big> </p>

<p> If I am <small> Small </small> </p>

<p> Ram says <q> Hello! Good Morning </q> </p>

<p> This is my address <address> Jspider, Old airport road, Bengaluru </address>

<p> <pre> hello good morning...

<del> had breakfast? .. I had </pre>  
<del> this is deleted text

</body>

</html>

<p> I am <del> young </del> old </p>

<p> I am <strike> young </strike> old. </p>

</body>

</html>

### \* ATTRIBUTE :-

→ Attributes are used to enhance the properties of Tag.

→ These are always written inside the opening tag.

→ Attributes are written in the form of Property name and value pair.

→ For one tag, we can use multiple attributes.

Syntax :- <tag name property name = "value"> --- </tag>

8). Underlined Text → `<u>....</u>`, or  
`<ins>....</ins>`

9). Quotations → `<q>....</q>`, or  
`<blockquote>....</blockquote>`

10). Highlighted Text → `<mark>....</mark>`

11). To display as  
Address Text → `<address>....</address>`

12). Predefined Text → `<pre>....</pre>`

26/12/20 PROGRAM - 2

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title> Text Formating Example </title>`

`</head>`

`<body>`

without any tag

`<p> <b> I am in bold </b> letter </p>`

`<p> <strong> I am in bold letter with strong tag </strong> </p>`

`<h3> <i> This element is in italic </i> </h3>`

`<h3> <em> This " " " " with em</em> </h3>`

`<h4> <u> This one is underlined </u> </h4>` Syntax:- `<tag name>`

`<h4> <ins> " " " " with ins tag </ins> </h4>`

`<p> <mark> This`  
`<p> Please solve`  
`<p> Solve this`  
`<p> Hi I am`  
`<p> Hi I am`  
`<p> Ram says`  
`<p> This is my`  
`Bengaluru`  
`<p> <pre> hello`

`<del> This is`  
`<body>`  
`</html>`  
`<p> I am <de`  
`<p> I am <st`  
`</body>`  
`</html>`

## \* ATTRIBUTE

→ Attributes are  
→ There are also  
→ Attributes are  
and value for

For one tag,

Syntax:- `<tag name>`

## PROGRAM - 2

```
<!DOCTYPE html>
<html>
  <head>
    <title> First example </title>
  </head>
  <body>
    <p> Hi! How are you? </p>
    <p> Merry Christmas </p>
    <p> I am fine. </p>
    <p> Have a nice day Thank you <br>
      Bye Bye Keep smiling <br>
      enjoy your day take rest Be calm
      abcde </p>
  </body>
</html>
```

## \* Text Formating Tags :-

- 1). Bold → <b>....</b>, or <strong>....</strong>.
- 2). Italic → <i>....</i>, or <em>....</em>
- 3). Strike out  
or Deleted text → <s>....</s>, or <del>....</del>,  
or <strike>....</strike>
- 4). Subscripted text → <sub>....</sub>.
- 5). superscript text → <sup>....</sup>.  
()
- 6). Lower case text → <small>....</small>.
- 7). Upper case text → <big>....</big>

### PROGRAM :-

```
<!DOCTYPE html>
<html>
  <head>
    <title>First example </title>
  </head>
  <body>
    <h1>Hi Good evening </h1>
    <h2> " " " </h2>
    <h3> " " " </h3>
    <h4> " " " </h4>
    <h5> " " " </h5>
    <h6> " " " </h6>
  </body>
</html>
```

### Breaking Tag :-

→ It is an unpaired tag used to break the line and give new line for the next content.

```
<br>
```

### PROGRAM

```
<!DOCTYPE
```

```
<html>
```

```
<h
```

```
</h
```

```
<b
```

```
</html>
```

### Text F

1). Bold

2). Itali

3). Strike  
or Del

4). Superscri

5). superscri

6). Lower ca

7). Upper ca

### Body Section :-

- The content of the Body Section will be displayed on the webpage.
- Body Section will be created by Body tag.

<Body>

-----  
</Body>

### Title Tag :-

- <title> ----- </title>
- It is used to give title for the webpage.
- It is written inside the Head Section

### Heading Tag :-

Font size  
h1 > h2 > h3 > h4 > h5 > h6

<h1> ----- </h1> Highest font size  
<h2> ----- </h2>  
<h3> ----- </h3>  
<h4> ----- </h4>  
<h5> ----- </h5>  
<h6> ----- </h6> Lowest font size

### Paragraph Tag :-

It is used for creating paragraph.

<p> ----- </p>

- It is a paired tag and also a block-level tag.
- In paragraph, maximum of one space is allowed.

Browser → Head.

Body

→ Displays in the Webpage.

Root Tag :- { <!DOCTYPE html>

{ <HTML>

- All HTML code will be having Root Tag.  
i.e.,

Head section

Body section.

</HTML>

Body

→ The c

the

→ Body

- The HTML code is said to be as HTML file if it contains <!DOCTYPE html> in the first line.

Head Section :-

- It contains all the information to the Browser about Body Section.

- We can create the Head section By using Head Tag

<HTML>

<Head>

=====

</Head>

Body section.

</HTML>

Heading

Font size

Paragraph

→ It is used

<P>

→ It is a par

In paragraph

- Head Section Content will be never be displayed on the Webpage

Unpaired Tags → Those tags which don't have closing tags are known as unpaired tags.

Ex:- <tagname> content.

Example :- Input Tags

Image Tags.

Break tags

Editors :-

↳ Notepad

↳ Word Pad

↳ Notepad ++

↳ Edit +

IDE

↳ Sublime

↳ Webstone

↳ Eclipse

↳ Intel j

Tags

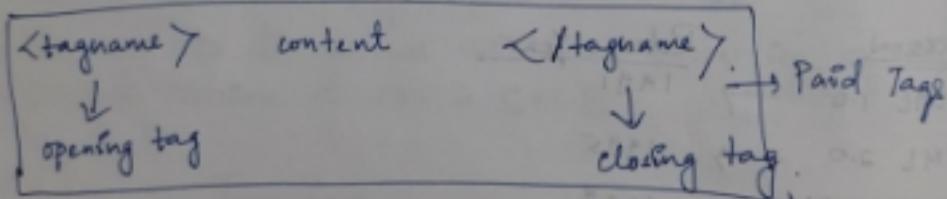
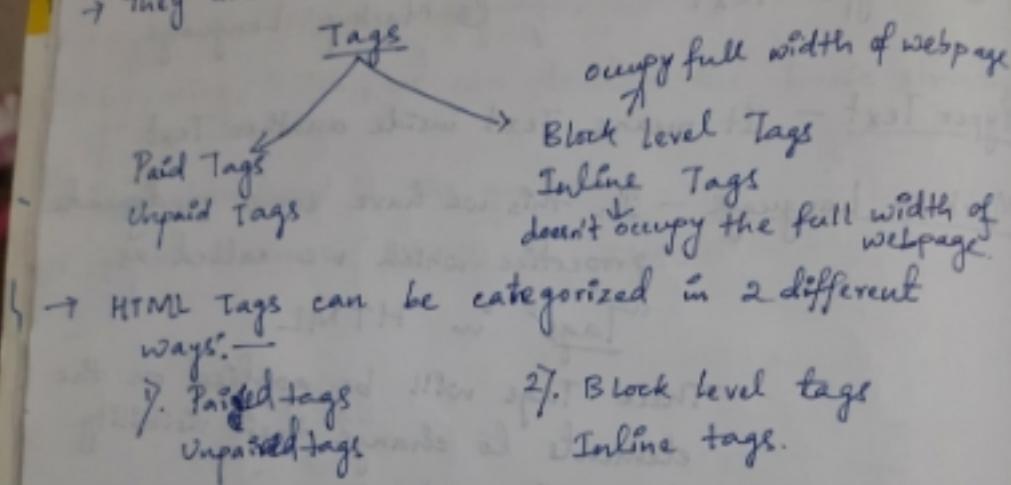
Note :-

- All HTML code is written inside a simple editor like Notepad.
- We can write the HTML code inside a simple editor like Notepad.
- All HTML must have the extension of '.html'.
- HTML code has 2-parts :-
  1. Head
  2. Body.

25/12/20

## Tags :-

- These are the predefined properties known as HTML elements and applied on some content.
- They are written inside the Angular Braces. " < > "



### Paired Tags :-

- Those tags which are having both opening & closing are known as paired tags.

Example:-

↳ Heading tags.

↳ Paragraph tags

↳ Title Tags.

### Unpaired Tags

### Example :

Editors:  
↳  
↳  
↳  
↳

IDE  
Sub  
We  
Ed  
2.

### Note:-

- All HTML
- We can use software
- All HTML
- HTML co

## HTML

Hyper Text Markup Language

7206171708

In HTML term, there are 2 parts.

- ① Hyper Text
- ② Markup Language.

① Hyper Text - It means Text inside another Text.

② Markup Language - In this we have some readymade properties which we called as "Tags" in HTML.

These Tags will be applied on the elements to change their visibility.

"< >" (Angular Braces)

| <u>Version</u> | <u>Release Year</u> | <u>Features</u> |
|----------------|---------------------|-----------------|
| HTML 1.0       | → 1991              |                 |
| HTML 2.0       | → 1995              |                 |
| HTML 3.0       | → 1999              |                 |
| HTML 4.01      | → 2000              |                 |
| XHTML          | → 2001              |                 |
| HTML 5         | → 2014              |                 |

24/02/19

and JavaScript  
→ HTML / CSS & are for Front-End Development.

## HTML

→ Hyper Text Markup Language.

→ By using HTML, we can develop the basic structure of the Webpage.

→ It was introduced in the year 1991.

→ Current version of HTML is HTML-5.

Hyper

In HTML

①

②. Hyper

③. Marku

## CSS

→ Stands for Cascading Style Sheets.

→ We use CSS to enhance the HTML properties by adding styling & colour.

→ Current Version of CSS is CSS-3.

VERSION

HTML 1

HTML 2

HTML

HTML

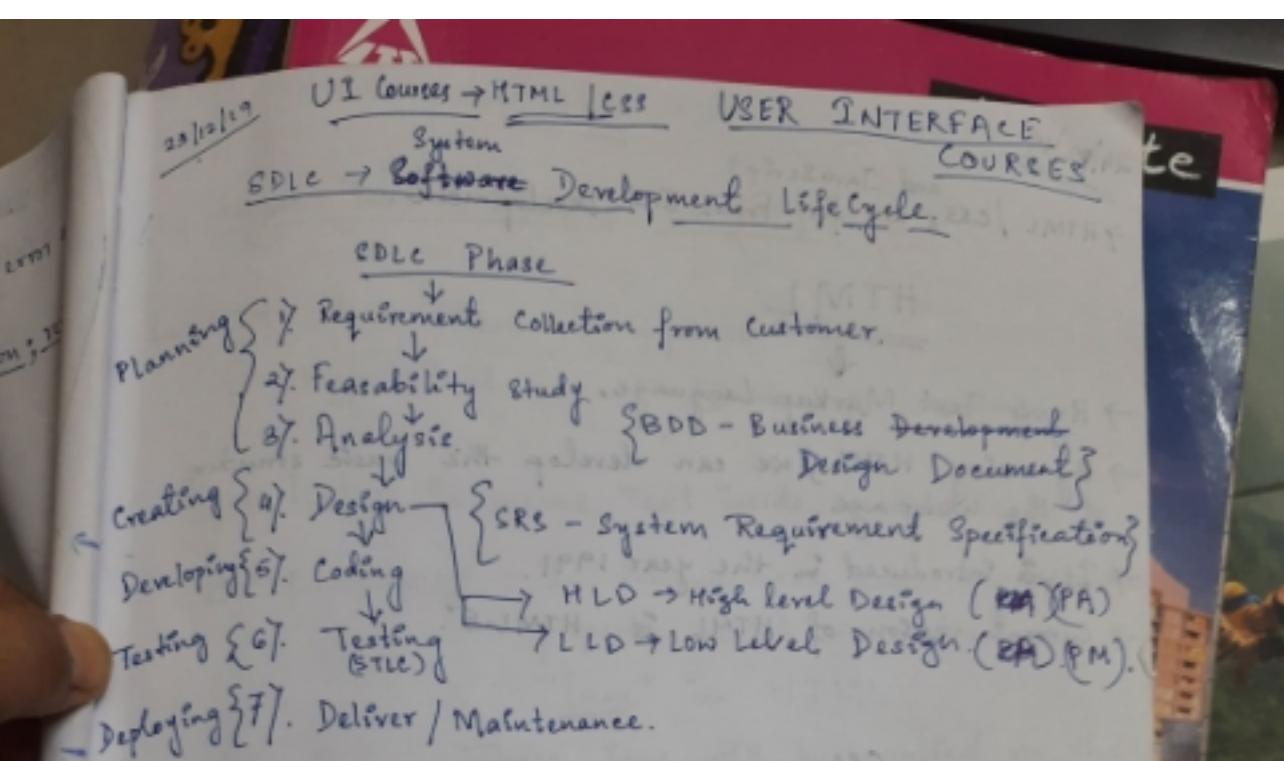
X HTML

HTML

## JavaScript

→ By using JavaScript, we can enhance the HTML and we can perform front-end validation.

→ Used to make Dynamic Changes.



HTML :- Hyper Text Markup Language.

HTML gives the structure of a Website/Webpage.

- CSS :- Cascading Style Sheets  
 CSS adds styling and colours to make the website look attractive.

JavaScript

BS :- Bootstrap  
 It is the framework of CSS.

JavaScript :- It is used to make the website dynamic.