
HTML

HTML :- Hyper Text Markup Langauge

Tags :

Heading tags (<h1> to <h6>) :

```
<body>
  <h1>This is my first Web page.</h1>
  <h2>This is my second Web page.</h2>
  <h3>This is my third Web page.</h3>
  <h4>This is my fourth Web page.</h4>
  <h5>This is my fifth Web page.</h5>
  <h6>This is my sixth Web page.</h6>
</body>
```

This is my first Web page.

This is my second Web page.

This is my third Web page.

This is my fourth Web page.

This is my fifth Web page.

This is my sixth Web page.

List Tags : There are three lists [1)unordered list, 2)ordered list, 3)describtion list]

```
<body>
    <ol type="1">
        <h3>Boys List :</h3>
        <li>Ram</li>
        <li>Shyam</li>
        <li>Mohan</li>
    </ol>
    <ul style type ="square">
        <h3>My ridding Instruments :</h3>
        <li>Car</li>
        <li>Bike</li>
        <li>Cycle</li>
    </ul>
    <dl>
        <h3>receppie</h3>
        <li>Maggi</li>
        <li>Burger</li>
        <li>Chai</li>
    </dl>
</body>
```

Boys List :

- 1. Ram
- 2. Shyam
- 3. Mohan

My ridding Instruments :

- Car
- Bike
- Cycle

receppie

- Maggi
- Burger
- Chai

NOTE DOWN : No closing tag in img tag.

```

<body>

<h4>Madan Mohan Malaviya University of Technology</h4>

</body>

```



Madan Mohan Malaviya University of Technology



Paragraph Tag : (<p> </p>) : It is use to creat paragraph.

In this tag we'll use some formatting tags like <i>, , <en>, <s>, , <sup>, <sub> etc.

```

<body bgcolor="pink">
<p><b>Lorem ipsum dolor sit amet</b><br>
<i>consectetur adipisicing elit.</i><br>
<s>Quasi voluptates deleniti</s><br> ad
<u>praesentium quia consequuntur</u><br>
soluta nisi assumenda alias nam beatae<br>,
repudiandae necessitatibus asperiores<br>
sed laboriosam et harum excepturi eveniet!</p>
<p>2<sup>5</sup></p>
<p>0<sub>2</sub></p>
</body>

```

```
  Lorem ipsum dolor sit amet
  consectetur adipisicing elit.
  Quasi voluptates deleniti
  ad praesentium quia consequuntur
  soluta nisi assumenda alias nam beatae
  , repudiandae necessitatibus asperiores
  sed laboriosam et harum excepturi eveniet!
```

2⁵

O₂

Pre Tag (<pre> tag) is used to same printing.

```
<body>
<pre>Ram goes to school.
He is good boy.
His brother name is Shyam.
</pre>
</body>
```

```
Ram goes to school.
He is good boy.
His brother name is Shyam.
```

div(<div>) Tag : It is a container which devide some part of web.(Kisi part me kuchh bhi karna ho tab isko use karte hai)

```
</head>
<body >
<div style="background-color: #burlywood;">This is my First Div.<br>
Div is a just like container which<br>
| contain some part of our Web.</div>
</body>
</html>
```

```
This is my First Div.
Div is a just like container which
contain some part of our Web.
```

Ancor Tag (ABCD) : It is used to creat a link .

NOTE DOWN : target="_blank" attribute lagane se new tab me jayega.

```
<body>
  <a href="https://www.google.com" target="_blank">Google</a>
  <br>
  <a href="mailto: anugrahit20@gmail.com">Contact with Email</a>
  <br>
  <a href="tel: 7068464328">Call us</a>
```

[Google](https://www.google.com)
[Contact with Email](mailto: anugrahit20@gmail.com)
[Call us](tel: 7068464328)

Table pe charcha :

```

<body bgcolor="pink">
    <table border="15">
        <thead>
            <th>Student ID</th>
            <th>Student Name</th>
            <th>Gender</th>
            <th>Age</th>
        </thead>
        <tbody>
            <tr>
                <td>01</td>
                <td>Ram</td>
                <td>M</td>
                <td>18</td>
            </tr>
            <tr>
                <td>02</td>
                <td>Shyam</td>
                <td>M</td>
                <td>19</td>
            </tr>
            <tr>
                <td>03</td>
                <td>Aashtha</td>
                <td>F</td>
                <td>17</td>
            </tr>
        </tbody>
    </table>
</body>

```

Student ID	Student Name	Gender	Age
01	Ram	M	18
02	Shyam	M	19
03	Aashtha	F	17

Q. Write a program for this type of page ?

AAAAAA	BBBBBB	CCCCCC
DDDDDD	EEEEEE	FFFFFF
GGGGGG HHHHHH		
IIIII		

```

<body bgcolor="yellow">
    <table border="2">
        <tr>
            <td>AAAAAA</td>
            <td>BBBBBB</td>
            <td>CCCCCC</td>
        </tr>
        <tr>
            <td rowspan="2">DDDDDD</td>
            <td>EEEEEE</td>
            <td>FFFFFF</td>
        </tr>
        <tr>
            <td>GGGGGG</td>
            <td>HHHHHH</td>
        </tr>
        <tr>
            <td colspan="3">IIIII</td>
        </tr>
    </table>
</body>

```

HTML Form : It is used to creat a Form. (use of form tag)

#NOTE DOWN :

- 1) Input ka "id" aur Label ka "for" same hona chahiye

2)"placeholder" se andar likhta hai .

```
<body bgcolor="pink">
  <form>
    <label for="mail">Enter EmailId :</label>
    <input type="email" id="mail" placeholder="enter email here">
    <br>
    <br>
    <label for="password">Enter the password here :</label>
    <input type="password" id="password" placeholder="*****">
    <br>
    <br>
    <input type="submit" value="Submit Karo">
  </form>
</body>
```

Enter EmailId :

Enter the password here :

Q. Write a program to execute this form ?

Amazon Application

Name :

Email :

Phone Number:

Resume : Choose File No file chosen

Experience(year):

Skill

HTML : CSS : JS :

Higher Education : ▾

Availability

Full Time Part-Time

Additional Comment :

```
<body bgcolor="pink">
    <h2>Amazon Application</h2>
    <label for="name">Name :</label>
    <input type="text" id="name">
    <br>
    <br>
    <label for="mail">Email :</label>
    <input type="email" id="mail">
    <br>
    <br>
    <label for="pin">Phone Number:</label>
    <input type="number" id="pin">
    <br>
    <br>
    <label for="file">Resume :</label>
    <input type="file" id="file" name="Upload your resume">
    <br>
    <br>
    <label for="ex">Experience(year):</label>
    <input type="number" id="ex" min="0" max="30">
    <br>
    <br>
```

```

<fieldset>
    <legend>Skill</legend>
    <label for="1">HTML :</label>
    <input type="checkbox" id="1">
    <label for="2">CSS :</label>
    <input type="checkbox" id="2">
    <label for="3">JS :</label>
    <input type="checkbox" id="3">
</fieldset>
<br>
<br>
<label for="">Higher Education :</label>
<select name="Education" id="abcd">
    <option value="a">10th</option>
    <option value="b">12th</option>
    <option value="c">Graduation</option>
</select>
<br>
<br>
<fieldset>
    <legend>Availability :</legend>
    <label for="ABCD">Full Time</label>
    <input type="radio" name="abcd" >
    <label for="ABCD">Part-Time</label>
    <input type="radio" name="abcd">
</fieldset>
<br>
<br>

```

```

    <label for="text">Additional Comment :</label>
    <textarea name="text" id="text" cols="30" rows="10"></textarea>
    <br>
    <br>
    <input type="submit" value="Submit">
    <input type="reset" value="Reset">
</body>

```

Cascading Style Sheet(CSS)

Q. What is CSS ?

Ans :- It is a language that is used to describe the style of a document.

Types of CSS on the basis of use :

- **Inline**

```
<h1 style="color: red"> Apna College </h1>
```



- **<style> tag**

```
<style>
  h1 {
    color : red;
  }
</style>
```

- **External Stylesheet**

Writing CSS in a separate document & linking it with HTML file

How to link a CSS file in HTML file ?

```
<link rel="stylesheet" href="ABDC.css">
```

Kuchh CSS ke sath pyar ho jayeee 😊

```
<body>
    <h1>Anugrah Pratap Singh</h1>
    <h2>My university name is MMMUT.</h2>
    <p>Lorem ipsum dolor sit amet consectetur <br>
        adipisicing elit. Veniam maiores ab architecto <br>
        debitis maxime eum veritatis, omnis quos deleniti! <br>
        Repellendus iusto pariatur maxime perspiciatis praesentium <br>
        cupiditate odit blanditiis animi nam.</p>
    <button>Submit</button>
    <button>Reset</button>
</body>
```

```
h1 {
    color: green;
}
h2 {
    color: #rgb(172, 115, 124);
}
p {
    color: orange;
}
button {
    color: white;
    background-color: #rgb(112, 149, 228);
}
body {
    background-color: #rgb(188, 241, 241);
}
```

Anugrah Pratap Singh

My university name is MMMUT.

 Lorem ipsum dolor sit amet consectetur
 adipisicing elit. Veniam maiores ab architecto
 debitis maxime eum veritatis, omnis quos deleniti!
 Repellendus iusto pariatur maxime perspiciatis praesentium
 cupiditate odit blanditiis animi nam.

Selectors

- Universal Selector

```
* { }
```

- Class Selector

```
.myClass { }
```



- Element Selector

```
h1 { }
```

- Id Selector

```
#myId { }
```

1) Universal Selector :- Sab kuch select kar leta hai 😊

```
<h1>Anugrah Pratap Singh</h1>
<h2>My university name is MMMUT.</h2>
| <button>Submit</button>
| <button>Reset</button>
</body>
```

```
* {
|   color: blue;
}
```

Anugrah Pratap Singh

My university name is MMMUT.

Submit **Reset**

2) Element Selector :- Simple kisi element ko select karta hai like h1, p etc.

```
<h1>Anugrah Pratap Singh</h1>
<h2>My university name is MMMUT.</h2>
| <button>Submit</button>
| <button>Reset</button>
```

```
h1 {  
    color: #rgb(196, 94, 30);  
}
```

Anugrah Pratap Singh

My university name is MMMUT.

3) Id Selector :- Kisi bhi element ko id diya jayega usake bad '#' use karke use karenge ham log

NOTE DOWN : Bhaiya Id unique hoga but class more than one ho sakta hai

```
<div id="anugrah">  
    <h1>Anugrah Pratap Singh</h1>  
    <h2>My university name is MMMUT.</h2>  
    <button>Submit</button>  
    <button>Reset</button>  
</div>
```

```
#anugrah {  
    background-color: #rgb(23, 111, 145);  
}
```

Anugrah Pratap Singh

My university name is MMMUT.

4) Class Selector :- Class selector ke bhi element ko class name dete hain usake bad '.' use karke use karte hai

```
<div>  
    <h1 class="ABCD">Anugrah Pratap Singh</h1>  
    <h2 class="EFGH">My university name is MMMUT.</h2>  
    <button class="ABCD">Submit</button>  
    <button class="EFGH">Reset</button>  
</div>
```

```
*{
    background-color: #rgb(151, 199, 199);
}
.ABCD{
    color: red;
}
.EFGH{
    color: blue;
}
```

Anugrah Pratap Singh

My university name is MMMUT.

Text Properties

1) Text - align :- Text ko align karo ---->(life is jhingalala) !

text-align : left / right / center

```
<h1>Anugrah Pratap Singh</h1>
<h2>My university name is MMMUT.</h2>
<p>Lorem ipsum dolor sit amet consectetur <br> adipisicing elit. <br>
    Explicabo commodi
<br> aliquam iste! Quia reiciendis excepturi sequi eos <br>
assumenda quo suscipit fugiat animi, eius debitis <br>
quis cumque iste. Explicabo, laudantium omnis!</p>
<button>Submit</button>
<button>Reset</button>
```

```
*{  
    background-color: #rgb(99, 188, 213);  
}  
p {  
    text-align: center;  
}  
h1 {  
    text-align: left;  
}
```

Anugrah Pratap Singh

My university name is MMMUT.

Explicabo commodi
aliquam iste! Quia reiciendis excepturi sequi eos
assumenda quo suscipit fugiat animi, eius debitis
quis cumque iste. Explicabo, laudantium omnis!

2) text-decoration :- Text ko decorate karo 😎

text-decoration : underline / overline / line-through etc

```
<h1>Anugrah Pratap Singh</h1>
<h2>My university name is MMMUT.</h2>
<p>Lorem, ipsum dolor sit amet consectetur adipisi</p>
<button>Submit</button>
<button>Reset</button>
```

```
*{  
    background-color: #rgb(99, 188, 213);  
}  
p {  
    text-decoration: overline;  
}  
h1{  
    text-decoration: underline;  
}  
h2 {  
    text-decoration: line-through;  
}
```

Anugrah Pratap Singh

~~My university name is MMMUT.~~

— Lorem, ipsum dolor sit amet consectetur adipisicing

3) font-weight :- Font ko mota patla karo

font-weight : normal / bold / bolder / lighter

font-weight : 100-900



```
*{  
    background-color: #rgb(99, 188, 213);  
}  
p {  
    font-weight: bold;  
}  
h1{  
    font-weight: 700;  
}  
h2 {  
    font-weight: bolder;  
}
```

Anugrah Pratap Singh

My university name is MMMUT.

Lorem, ipsum dolor sit amet consectetur adipisci

4) font-family :- isme to font ke families hi aa jayenge.



```
<a id="link" href="https://www.mmmut.ac.in" target="_blank">Go to MMMUT</a>
<p id="para">Lorem ipsum dolor sit amet consectetur <br>
adipisicing elit. Molestias, eaque debitis. Qui <br>
a explicabo, sed iusto in mollitia ducimus quas. <br>
Sapiente repellat corrupti optio quia facilis. <br>
Et consectetur eum porro deserunt!</p>
```

```
*{
    background-color: #rgb(99, 188, 213);
}
#para {
    font-family: 'Courier New', Courier, monospace;
}
#link {
    font-family: 'Courier New', Courier, monospace;
}
```

[Go to MMMUT](https://www.mmmut.ac.in)

Lorem ipsum dolor sit amet consectetur
adipisicing elit. Molestias, eaque debitis. Qui
a explicabo, sed iusto in mollitia ducimus quas.
Sapiente repellat corrupti optio quia facilis.
Et consectetur eum porro deserunt!

5) Line height :- Line ke liye utna hi jagah chhad deta hai jitna ham dete hai.

```
*{  
    background-color: #rgb(99, 188, 213);  
}  
#para {  
    line-height: 40px;  
}
```

[Go to MMMUT](#)

Lorem ipsum dolor sit amet consectetur

adipisicing elit. Molestias, eaque debitis. Qui

a explicabo, sed iusto in mollitia ducimus quas.

Sapiente repellat corrupti optio quia facilis.

Et consectetur eum porro deserunt!

6) text-transform :-

text-transform : uppercase / lowercase / capitalize / none

```
*{  
    background-color: #rgb(99, 188, 213);  
}  
#para {  
    text-transform: capitalize;  
}  
#link {  
    text-transform: lowercase;  
}
```

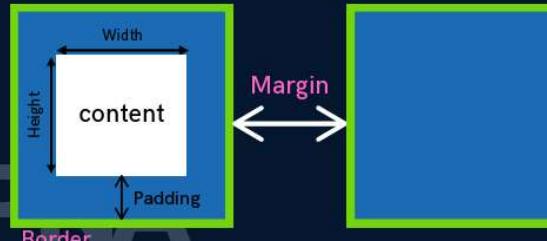
[go to mmmut](#)

Lorem Ipsum Dolor Sit Amet Consectetur
Adipisicing Elit. Molestias, Eaque Debitis. Qui
A Explicabo, Sed Iusto In Mollitia Ducimus Quas.
Sapiente Repellat Corrupti Optio Quia Facilis.
Et Consectetur Eum Porro Deserunt!

Box Model in CSS

Box Model in CSS

- Height
- Width
- Border
- Padding
- Margin



Height, Width and Border :-

```
<body>
  <div>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing
       elit. Atque, quod aspernatur veritatis nam excepturi
       natus pariatur provident accusantium sunt dolorem fugit
       maiores amet quae aperiam exercitationem sint id omnis cumque?</p>
  </div>
</body>
```

```
div {
  height: 200px;
  width: 300px;
  background-color: #pink;
  border-width: 5px;
  border-style: dashed;
  border-color: #blue;
  border-radius: 15px;
}
```

Quae aperiam exercitationem sint id omnis cumque?

Padding, Margin :-

1) Padding :- Padding me border aur content ke bich ka distance hota hai 😊

Padding

- padding-left
- padding-right
- padding-top
- padding-bottom

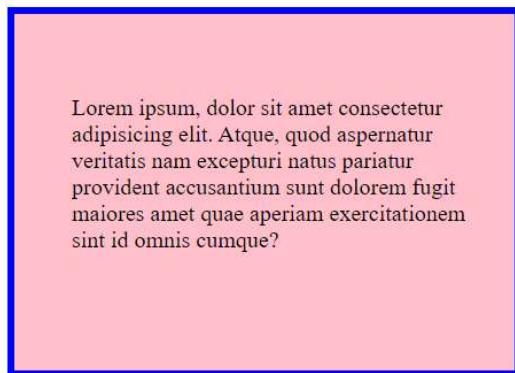
APNA
COLLEGE



The image shows the APNA COLLEGE logo in large, bold, sans-serif letters. Overlaid on the logo are two examples of CSS padding declarations. The first example, 'padding: 50px;', shows the entire logo shifted inward from its original position. The second example, 'padding: 1px 2px 3px 4px;', shows the logo shifted inward with varying widths on each side: top (1px), right (2px), bottom (3px), and left (4px). The text is color-coded: 'padding:' is in blue, '50px;' is in red, '1px 2px 3px 4px;' is in blue, and the side-specific padding values (1, 2, 3, 4) are in red.

```
<div>
<p>Lorem ipsum, dolor sit amet consectetur adipisicing
    elit. Atque, quod aspernatur veritatis nam excepturi
    natus pariatur provident accusantium sunt dolorem fugit
    maiores amet quae aperiam exercitationem sint id omnis cumque?</p>
</div>
```

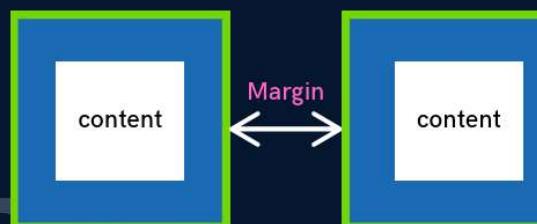
```
div {
    height: 200px;
    width: 300px;
    background-color: #pink;
    border-width: 5px;
    border-style: solid;
    border-color: #blue;
    padding: 40px 9px 8px 40px;
}
```



2) Margin :-

Margin

- margin-right
- margin-left
- margin-top
- margin-bottom



Margin

Shorthand

margin: 50px;



margin: 1px 2px 3px 4px;

top | right | bottom | left -> clockwise

Display Property

display: inline / block / inline-block / none

- **inline** - Takes only the space required by the element. (no margin/ padding)
- **block** - Takes full space available in width.
- **inline-block** - Similar to inline but we can set margin & padding.
- **none** - To remove element from document flow.

Visibility

visibility: hidden;



Note : When visibility is set to none, space for the element is reserved.

But for display set to none, no space is reserved or blocked for the element.

NOTE DOWN :- display ke none me gya to pura khatam hoke usaka jagah bhi khatam ho gya but visibility ke hidden me jayega to jagah khatam nahi hogta.

```

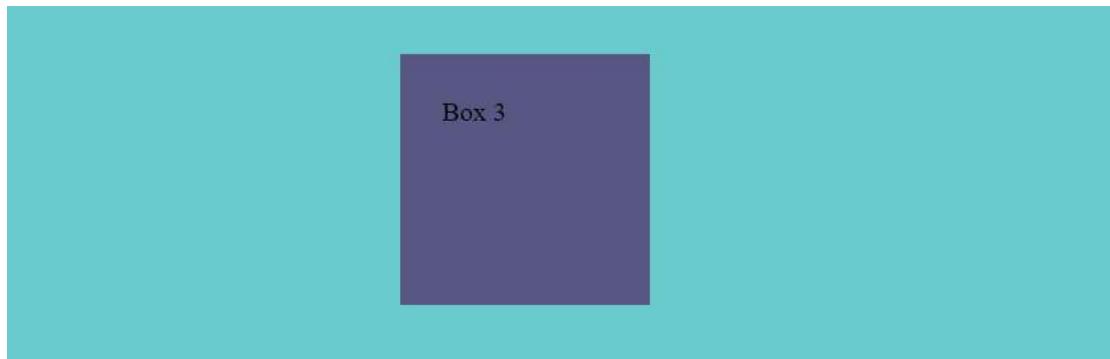
<body>
  <div id="box1" style="background-color: green;">Box 1</div>
  <div id="box2" style="background-color: red; visibility: hidden;">Box 2</div>
  <div id="box3" style="background-color: blue; display: none;">Box 3</div>
</body>

```

```

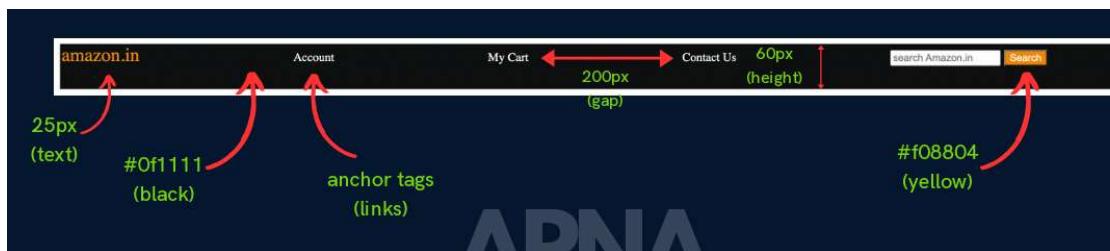
*{
  background-color: #rgb(105, 204, 204);
}
div {
  height: 100px;
  width: 100px;
  margin: 25px;
  padding: 25px;
  display: inline-block;
}
#box2 {
  visibility: hidden;
}
#box1{
  display: none;
}

```



Upper wale me 2 wale jagah khatam ho gya hai but 1 ka nahi hua hai.

Q. Create the following nav bar



Sol :-

```
<body>
  <div id="navbar">
    <a id="logo" href="">amazon.in</a>
    <a href="">Account</a>
    <a href="">My Cart</a>
    <a href="">Contact Us</a>
    <div id="barsearch">
      <input placeholder="Search Amazon.in" type="text" name="" id=""
        <button>Search</button>
    </div>
  </div>
</body>
```

```
padding: 0;
margin: 0;
color: white;

}
#navbar {
  height: 60px;
  background-color: #0f1111;
}
button {
  background-color: #f08804;
}
#logo {
  color: #f08804;
  font-size: 25px;
}
a {
  margin-right: 200px;
  margin-top: 18px;
  text-decoration: none;
}
div {
  display: flex;
}
#barsearch{
  height: 25px;
  margin-top: 18px;
  border-width: 2px;
  border-color: #f08804;
}
```

Position

The position CSS property sets **how an element is positioned** in a document.

position : static / relative / absolute / fixed



Position

- **static** - default position (The top, right, bottom, left, and z-index properties have no effect)

- **relative** - element is relative to itself. (The top, right, bottom, left, and z-index will work)

- **absolute** - positioned relative to its closest positioned ancestor. (removed from the flow)

- **fixed** - positioned relative to browser. (removed from flow)

- **sticky** - positioned based on user's scroll position

```
#box2{  
    position: relative;  
    top: 10px ;  
    left: 20px ;  
}
```

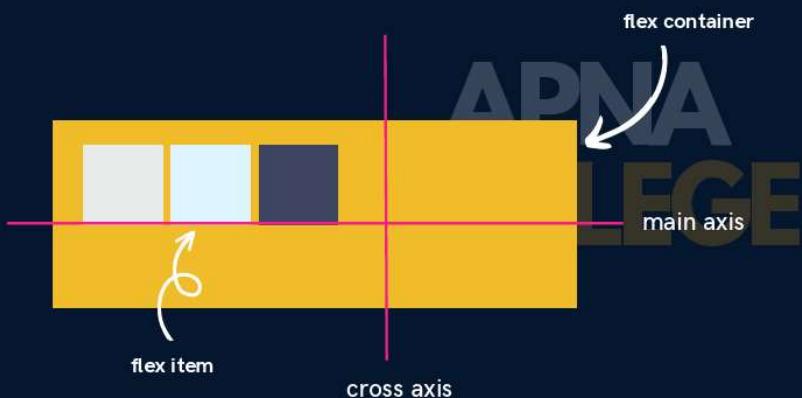
Flexbox

Flexible Box Layout

It is a one-dimensional layout method for arranging items in rows or columns.



The Flex Model



Flexbox Direction

It sets how flex items are placed in the flex container, along which axis and direction.

- `flex-direction : row;` (default)
- `flex-direction : row-reverse;`
- `flex-direction : column;`
- `flex-direction : column-reverse;`

```
<body>
  <div id="container">
    <div id="box1">Box 1</div>
    <div id="box2">Box 2</div>
    <div id="box3">Box 3</div>
    <div id="box4">Box 4</div>
    <div id="box5">Box 5</div>
  </div>
</body>
```

```

body{
    text-align: center;
    background-color: #rgb(114, 176, 155);
}

div{
    height: 100px;
    width: 100px;
    display: inline-block;
    border: 2px solid black;
}

#container {
    width: 800px;
    height: 200px;
    display: flex;
    flex-direction: column;
}

```



Flex Properties

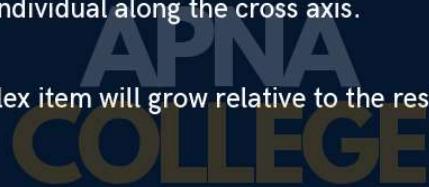
for Flex Container

- **justify-content** : alignment along the main axis.
flex-start / flex-end / centre / space-around / space-between
- **flex-wrap** : nowrap / wrap / wrap-reverse
- **align-items** : alignment along the cross axis.
- **align-content** : alignment of space between & around the content along cross-axis

Flex Properties

for Flex Item

- **align-self** : alignment of individual along the cross axis.
- **flex-grow** : how much a flex item will grow relative to the rest of the flex items if space is available
- **flex-shrink** : how much a flex item will shrink relative to the rest of the flex items if space is available



Media Queries

Help create a **responsive** website

```
@media (width : 600px) {  
    div {  
        background-color : red;  
    }  
}
```



```
@media (min-width : 600px) {  
    div {  
        background-color : red;  
    }  
}
```

Transitions

Transitions enable you to define the transition between two states of an element.

- `transition-property` : property you want to transition (font-size, width etc.)
- `transition-duration` : 2s / 4ms ..
- `transition-timing-function` : ease-in / ease-out / linear / steps ..
- `transition-delay` : 2s / 4ms ..

Transition Shorthand

property name | duration | timing-function | delay

transition: font-size 2s ease-in-out 0.2s;

CSS Transform

Used to apply 2D & 3D transformations to an element

- `rotate`

transform: rotate(45deg);

CSS Transform

- scale

```
transform: scale(2);  
transform: scale(0.5);  
transform: scale(1, 2);  
  
transform: scaleX(0.5);  
transform: scaleY(0.5);
```



CSS Transform

- translate

```
transform: translate(20px);  
transform: translate(20px, 50px);  
  
transform: translateX(20px);  
transform: translateY(20px);
```



Animation

To animate CSS elements

```
@keyframe myName {  
  from { font-size : 20px; }  
  to { font-size : 40px; }  
}
```



Animation Properties

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

APNA
COLLEGE

```
.loder{  
    height: 100px;  
    width: 100px;  
    border-radius: 50%;  
    border: 20px solid #FFE66D;  
    border-top: 20px solid #4ECDC4;  
    border-bottom: 20px solid #4ECDC4;  
    animation: spinAnimate 3s infinite normal;  
}  
  
@keyframes spinAnimate {  
    from {  
        transform: rotate(0deg);  
    }  
    to {  
        transform: rotate(360deg);  
    }  
}
```

Javascript

console.log("") ka power : Ye "" ke andar wala sab kuchh print karta hai .

Variables in JS : Variables are containers for data.

Types of Variables :

1) var,

2) let,

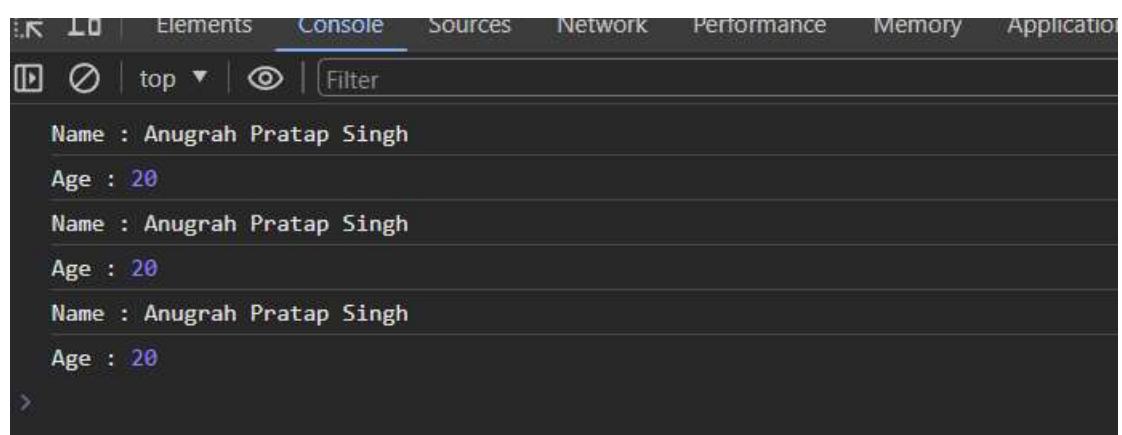
3) const

var : Variable can be re-declared & updated. A global scope variable.

let : Variable cannot be re-declared but can be updated. A block scope variable.

const : Variable cannot be re-declared or updated. A block scope variable.

```
1 var name="Anugrah Pratap Singh";
2 console.log("Name :",name);
3 var age = 20;
4 console.log("Age :",age);
5 let name2="Anugrah Pratap Singh";
6 console.log("Name :",name2);
7 let age2 = 20;
8 console.log("Age :",age2);
9 const name3="Anugrah Pratap Singh";
10 console.log("Name :",name3);
11 const age3 = 20;
12 console.log("Age :",age3);
```



Data Types in JS :

Primitive Types : Number, String, Boolean, Undefined, Null, BigInt, Symbol

Object Wala Baaat :

```
1 const Student = {  
2     fullName : "Vivek Singh",  
3     Age : 20,  
4     CGPA : 8.73,  
5     PassorNot : true,  
6  
7 };  
8 console.log(Student);
```

```
▼ {fullName: 'Vivek Singh', Age: 20, CGPA: 8.73, PassorNot: true} i  
  Age: 20  
  CGPA: 8.73  
  PassorNot: true  
  fullName: "Vivek Singh"  
  ▶ [[Prototype]]: Object
```

Operators in JS : Used to perform some operation on data.

Arithmetic Operators

+, -, *, /

- Modulus
- Exponentiation
- Increment
- Decrement

NOTE DOWN :- For Exponentiation, we use '**'

```

let a=6;
let b=7;
console.log("a + b =", a+b);
console.log("b - a =", b-a);
console.log("a * b =", b*a);
console.log("b / a =", b/a);
console.log("b % a =", b%a);
console.log("7(b) to the power 6(a) =", b**a);

```

```

a + b = 13
b - a = 1
a * b = 42
b / a = 1.1666666666666667
b % a = 1
7(b) to the power 6(a) = 117649

```

Assignment Operators

= += -= *= %= **=

Comparison Operators

Equal to ==

Equal to & type ===

Not equal to !=

Not equal to & type !==

Logical Operators

Logical AND **&&**

Logical OR **||**

Logical NOT **!**

Conditional Statements : if, else if, else

Qs2. Write a code which can give grades to students according to their scores:

- 80-100, A
- 70-89, B
- 60-69, C
- 50-59, D
- 0-49, F

Sol :-

```
let score = 78;
if(score >=80 && score<=100 ){
    console.log("Grade A");
}
else if (score >=70 && score <=79){
    console.log("Grade B");
}
else if (score >=60 && score <=69){
    console.log("Grade C");
}
else if (score >=50 && score <=59){
    console.log("Grade D");
}
else {
    console.log("Grade F");
}
```

Output : Grade B.

Ternary Operators

condition ? true output : false output

```
age > 18 ? "adult" : "not adult";
```

```
let age = 20;
let voterange = age >= 18 ? "Can give vote" : "Can not give vote";
console.log(voterange);
```

Output : Can give vote.

Qs1. Get user to input a number using prompt("Enter a number:"). Check if the number is a multiple of 5 or not.

Sol :-

```
let num = prompt("Enter the Number");
if(num%5==0){
    console.log("Number is divisible by 5");
}
else {
    console.log("Number is NOT divisible by 5");
}
```

Loops in JS :- It is use to print anything again and again.

Strings in JS :- String is a sequence of characters used to represent text.

Create String

```
let str = "Apna College";
```

String Length

```
str.length
```

String Indices

```
str[0], str[1], str[2]
```

String Methods in JS

These are built-in functions to manipulate a string

- `str.toUpperCase()`
- `str.toLowerCase()`
- `str.trim()` *// removes whitespaces*

- **str.slice(start, end?)** *// returns part of string*

- **str1.concat(str2)** *// joins str2 with str1*

- **str.replace(searchVal, newVal)**

- **str.charAt(idx)**

```
let str = "AbCdE fGhIjK lMnOpQ rStUvW xYz";
let newStr = str.toUpperCase();
let newStr2 = str.slice(6, 12);
console.log(str);
console.log(newStr);
console.log(newStr2);
```

AbCdE fGhIjK lMnOpQ rStUvW xYz
ABCDE FGHIJK LMNOPQ RSTUVW XYZ
fGhIjK

Ques . Prompt the user to enter their full name. Generate a username for them based on the input.Start username with @, followed by their full name and ending with the fullname length.

eg: user name = “shradhakhapra” , username should be “@shradhakhapra13”

Sol :-

```
let user = prompt("Enter Your Name please !");
let leng = user.length;
let userName ="@" + user + leng ;
console.log(userName);
```

Output : @anugrahpratapsingh17

Functions in JS :Block of code that performs a specific task, can be invoked whenever needed

Functions in JS

Function Definition

Function Call

```
function functionName( ) {
    //do some work
}
```

```
function functionName( param1, param2 ...) {
    //do some work
}
```

```
function sum_function(x, y) {
    sum = x + y;
    return sum;
}
let val = sum_function(5, 3);
console.log("Sum is : ", val);
```

Output : Sum is 8

Arrow Functions

Compact way of writing a function

```
const functionName = ( param1, param2 ... ) => {  
    //do some work  
}
```

```
const sum = ( a, b ) => {  
    return a + b;  
}
```

```
const mul_function = ( a, b ) => {  
    console.log("Multiplication is : ", a*b);  
}  
mul_function(9, 3);
```

Output : Multiplication is : 27

forEach Loop in Arrays

arr.forEach(callBackFunction)

CallbackFunction : Here, it is a function to execute for each element in the array

*A callback is a function passed as an argument to another function.

```
arr.forEach( ( val ) => {  
    console.log(val);  
})
```

Qs. For a given array of numbers, print the square of each value using the forEach loop.

Solution :

```
let numbers = [2, 3, 4, 5, 6];

numbers.forEach((num)=>{
  console.log(num*num);
})
```

Output : 4

9
16
25
36

Some More Array Methods :

1)

Map

Creates a new array with the results of some operation. The value its callback returns are used to form new array

arr.map(callbackFnx(value, index, array))

```
let newArr = arr.map( ( val ) => {
  return val * 2;
})
```

```
let arr = [2, 3, 4, 5, 6];

arr.map((newarr)=>{
  console.log(newarr*5);
})
```

```
10  
15  
20  
25  
30
```

2)

Filter

Creates a new array of elements that give true for a condition/filter.

Eg: all even elements

```
let newArr = arr.filter( ( val )=>{  
    return val % 2 === 0;  
})
```

```
let arr = [2, 3, 4, 5, 6];  
  
let evnum=arr.filter((number)=>{  
    return number%2==0;  
})  
  
console.log(evnum);
```

```
▼ (3) [2, 4, 6] i  
  0: 2  
  1: 4  
  2: 6  
  length: 3  
▶ [[Prototype]]: Array(0)
```

3)

Reduce

Performs some operations & reduces the array to a single value. It returns that single value.

JavaScript Demo: Array.reduce()

```
1 const array1 = [1, 2, 3, 4];
2
3 // 0 + 1 + 2 + 3 + 4
4 const initialValue = 0;
5 const sumWithInitial = array1.reduce(
6   (accumulator, currentValue) => accumulator + currentValue,
7   initialValue,
8 );
9
10 console.log(sumWithInitial);
11 // Expected output: 10
```

```
let arr = [2, 3, 4, 5, 6];

let outputnum = arr.reduce((num1, num2) =>{
  return num1 + num2;
})
console.log(outputnum);
```

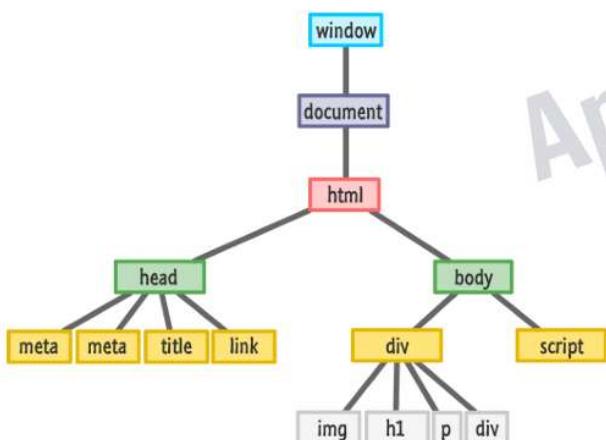
Output : 20 😊(2+3+4+5+6=20)

Window Object

The window object represents an open window in a browser. It is browser's object (not JavaScript) & is automatically created by browser.

It is a **global** object with lots of properties & methods.

When a web page is loaded, the browser creates a **Document Object Model (DOM)** of the page.



DOM Manipulation

Selecting with id

```
document.getElementById("myId")
```

Selecting with class

```
document.getElementsByClassName("myClass")
```

Selecting with tag

```
document.getElementsByTagName("p")
```

Query Selector

```
document.querySelector("#myId / .myClass / tag")
```

//returns first element

```
document.querySelectorAll("#myId / .myClass / tag")
```

//returns a NodeList

Properties

- **tagName** : returns tag for element nodes
- **innerText** : returns the text content of the element and all its children
- **innerHTML** : returns the plain text or HTML contents in the element
- **textContent** : returns textual content even for hidden elements

```
let para1 = document.getElementsByClassName("Myclass");
console.dir(para1);
let para2 = document.getElementById("ppppp");
console.dir(para2);
let para3 = document.getElementsByTagName("div");
console.dir(para3);
```

```
▶ HTMLCollection(3)
▶ p#ppppp
▶ HTMLCollection(1)
```

Events in JS : The change in the state of an object is known as an Event. Events are fired to notify code of "interesting changes" that may affect code execution.

- **Mouse events (click, double click etc.)**
- **Keyboard events (keypress, keyup, keydown)**
- **Form events (submit etc.)**
- **Print event & many more**

```
<h1>Events in Javascript</h1>
<br>
<br>
<div onmouseover="console.log('You have hovered on div')">This is div.</div>
<button onclick="console.log('You have clicked on button.')">Click me !</button>
<script src="first.js"></script>
```

Classes & Objects : A javaScript object is an entity having state and behavior (properties and method). JS objects have a special property called prototype.

NOTE DOWN : We can set prototype using '_ _ proto _ _'

An object :)

```
const student = {
    fullName : "Anugrah Pratap Singh",
    marks : 95,
    rollNum : 2023021112
}
console.log("Name : ",student.fullName);
console.log("Marks : ",student.marks);
console.log("Roll Num : ",student.rollNum);
```

```
Name : Anugrah Pratap Singh
Marks : 95
Roll Num : 2023021112
```

Classes in JS

Class is a program-code template for creating objects.

Those objects will have some state (variables) & some behaviour (functions) inside it.

```
class MyClass {  
    constructor() { ... }  
    myMethod() { ... }  
}  
  
let myObj = new MyClass();
```

Constructor() method is :

- automatically invoked by new
- initializes object

```
class MyClass {  
    constructor() { ... }  
    myMethod() { ... }  
}
```

Form Validation in Javascript : 😊

```

<form action="abcd.html" onsubmit=" return data()">
    <label for="Name">Name : </label>
    <input type="text" id="Name">
    <br><br>
    <label for="Contact">Contact : </label>
    <input type="text" id="contact">
    <br><br>
    <label for="password">Password</label>
    <input type="password" id="password">
    <br><br>
    <label for="spassword">Confirm Password : </label>
    <input type="password" id="spassword">
    <br><br>
    <button id="submit">Submit</button>
</form>

```

```

function data(){
    var a = document.getElementById("Name").value;
    var b = document.getElementById("contact").value;
    var c = document.getElementById("password").value;
    var d = document.getElementById("spassword").value;
    var e = document.getElementById("submit").value;

    if (a === "" || b === "" || c === "" || d === "" || e === "") {
        alert("All fields are mandatory, Please fill all fields !");
        return false;
    }
    else if (b.length < 10 || b.length > 10) {
        alert("Number should be 10 digits !");
        return false;
    }
    else if (isNaN(b)) {
        alert("Only number is allowed .");
        return false;
    }
    else if (c != d) {
        alert("Password should be same !");
        return false;
    }
    else {
        true;
    }
}

```

Name :

Contact : 76757

Password

Confirm Password :

All fields are mandatory. Please fill all fields !

Angular JS

What is Angular Javascript ?

Ans : AngularJS is a JavaScript framework written in JavaScript. AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag.

AngularJS Extends HTML

AngularJS extends HTML with **ng-directives**.

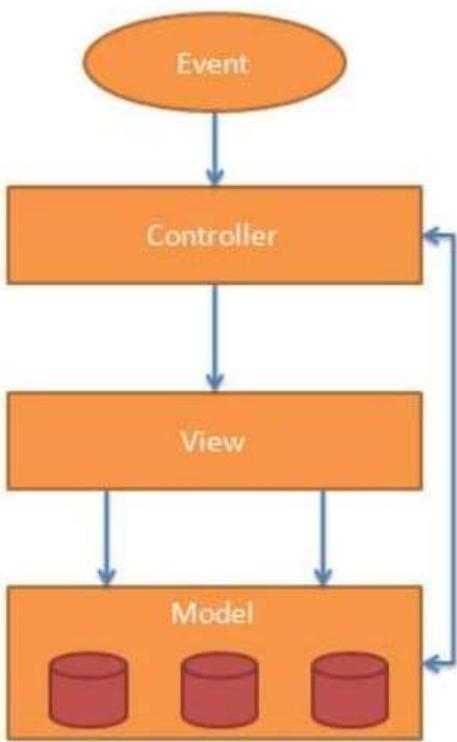
The **ng-app** directive defines an AngularJS application.

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.

The **ng-bind** directive binds application data to the HTML view.

MVC Architecture : - Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts –

- **Model** – It is the lowest level of the pattern responsible for maintaining data.
- **View** – It is responsible for displaying all or a portion of the data to the user.
- **Controller** – It is a software Code that controls the interactions between the Model and View.



AngularJS Expressions : AngularJS binds data to HTML using Expressions.

AngularJS expressions can be written inside double braces: `{{ expression }}` .

AngularJS expressions can also be written inside a directive: `ng-bind="expression"` .

AngularJS will resolve the expression, and return the result exactly where the expression is written.

AngularJS expressions are much like **JavaScript expressions**: They can contain literals, operators, and variables.

Example `{{ 5 + 5 }}` or `{{ firstName + " " + lastName }}`

AngularJs Module :- Module in AngularJS refers to a place where you can group the components, directives, pipes, and services, which are related to the application.

How to create a Module in AngularJS : A module is created by using the AngularJS function `angular.module`

```
<div ng-app="myApp">...</div>

<script>

var app = angular.module("myApp", []);

</script>
```

Directives :- Directives are classes that add additional behavior to elements in your Angular applications.

AngularJS directives are extended HTML attributes with the prefix `ng-`.

The `ng-app` directive initializes an AngularJS application.

The `ng-init` directive initializes application data.

The `ng-model` directive binds the value of HTML controls (input, select, textarea) to application data.

Data Biding :- The HTML container where the AngularJS application is displayed, is called the view & Model represents the data structure and business logic of an application. Data binding in AngularJS is the synchronization between the model and the view. When data in the model changes, the view reflects the change, and when data in the view changes, the model is updated as well.

AngularJS Controllers :- AngularJS controllers control the data of AngularJS applications. AngularJS controllers are regular JavaScript Objects.

```

<div ng-app="myApp" ng-controller="myCtrl">

First Name: <input type="text" ng-model="firstName"><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
```

</div>

```

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
</script>

```

Filters :- AngularJS provides filters to transform data.

- `currency` Format a number to a currency format.
- `date` Format a date to a specified format.
- `filter` Select a subset of items from an array.
- `json` Format an object to a JSON string.
- `limitTo` Limits an array/string, into a specified number of elements/characters.
- `lowercase` Format a string to lower case.
- `number` Format a number to a string.
- `orderBy` Orders an array by an expression.
- `uppercase` Format a string to upper case.

```

<div ng-app="myApp" ng-controller="personCtrl">

<p>The name is {{ lastName | uppercase }}</p>

</div>

```

```
<div ng-app="myApp" ng-controller="costCtrl">  
  
  <h1>Price: {{ price | currency }}</h1>  
  
</div>
```

AngularJS Forms :- Forms in AngularJS provides data-binding and validation of input controls.

Input controls are the HTML input elements:

- **input** elements
- **select** elements
- **button** elements
- **textarea** elements

-: Completed Web-D-1 :-

Created/Made By:

Anugrah Pratap Singh

(CSE Student @ MMMUT)