

UI COURSES

Open
Type Note Pad
The htm/x Pres

1. HTML
2. CSS
3. BOOTSTRAP
4. JAVASCRIPT
5. JQUERY

1. HTML :

- * HTML is a "HYPER TEXT MARKUP LANGUAGE".
- * HTML is used to Create Structure of the Web Pages (Layout, Web Page means "collection of information", Web site means "Collection of web pages").

2. CSS :

- * CSS means "Cascading Style sheets".
- * CSS is used to decorate the "html Pages".

3. Bootstrap:

- * Bootstrap is a open Source frame work of Htm/x CSS, which is used to develop a (create a) web Page with in short time..

4. Java Script :

- * JavaScript is a clint Side Programming language..
- * which is used to dynamic Pages and to Perform clint Side Validation...

*

5. Jquery :

- * Jquery is library of javaScript which is used to Simplified the javaScripte Code...

HTML

- HTML: HTML means Hyper text Markup language...
- * HTML was developed by "Tim Berners Lee" in the year of 1991.
 - * HTML is used to Create a web Page (Layouts).

What is Hyper text?

- * Hyper text is also known as "Hyper links".
- * Hyper text is a text which contains reference of another Web Page.

What is Markup language?

- * The language which we are Programming with us as "tag". Those languages are told as Markup language.

What is tag?

- * Any text which is Surrounded will be <> is known as tag. By HTML we can't Create user defined tags.

<raja>

Predefined tags

<Button>

Types of tags:

They are Two types of tags i) Paired tags.

1. Paired tags:

Paired tags mean the tags which are having opening tags as well as closing tags.

<Syntax>

or <P>.....</P>

<Tagname>.....Opening tag.....</Tagname> — Closing tag.

2. unPaired tag:

unPaired tag means the tags which are having only opening tags. ex: <tagname>.....Opening tag,
,

Tags are always known as html elements. Tag for elements.

BASIC HTML PROGRAM

```
<html>  
<head>  
<title> rajadevkar</title>  
</head>  
<body>  
    Hi WELCOME TO RAGA WORLD  
</body>  
</html>
```

- * html tag is the Parent or root tag for all html Predefined...
- * Head is used to write information about our html document.
- * Body tag is used to display the output of web pages.

HTML doctype declaration:

- * html doctype declaration teller to the browser which html version we are using.
- * <!DOCTYPE html> This represents html-5 Version.
- * doctype declaration will not be considered tag.
- * writing doctype declaration is not compulsory.
- * If we are going to write always it must be in first line of the html program..

HTML versions:

- * html 1.0 - 1991
- * html 2.0 - 1995
- * html 3.0 - 1997
- * html 4.0 - 1999
- * XHTML - 2000
- * html 5.0 - 2014

HTML heading tags:

Six heading tags is there...

1. `<h1>`
2. `<h2>`
3. `<h3>`
4. `<h4>`
5. `<h5>`
6. `<h6>`

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <title> HTML HEADINGS </title>
```

```
</head>
```

```
<body>
```

```
    <h1> example heading </h1>
```

```
    <h2> example heading </h2>
```

```
    <h3> example heading </h3>
```

```
    <h4> example heading </h4>
```

```
    <h5> example heading </h5>
```

```
    <h6> example heading </h6>
```

```
</body>
```

```
</html>
```

HTML ATTRIBUTES:

HTML attributes are used to provide additional information for HTML elements.

Syntax:

```
<Tagname attribute name = "attribute"> ---  
--- Content (Any text, img, ifany) ---  
--- </tagname>
```

ex: `<h1 align = "center"> example heading </h1>`

ex: `align = left`

`align = right & align = center.`

```
<!DOCTYPE html>
<html>
<head>
    <title> HTML headings </title>
</head>
<body>
    <h1 align="center"> example heading </h1>
</body>
</html>
```

HTML IMAGES:

Address of the image. (image location & Img name x.)

```
 image <img>
```

```
<!DOCTYPE html>
```

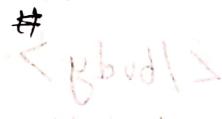
```
<html>
```

```
<head>
```

```
    <title> HTML images </title>
```

```
</head>
```

```
<body>
```

~~image location~~ = ~~#~~ 

```
    <h1> image html </h1>
```

```
    
```

~~width = "200px" height = "200px"~~

```
</body>
```

```
</html>
```

Correctly: ~~image address~~ ~~width = "200px" height = "200px"~~

```
    
```

~~src="image address" width="200px" height="200px"~~

~~src="image address" width="200px" height="200px"~~

Absolute Path & Relative Path:

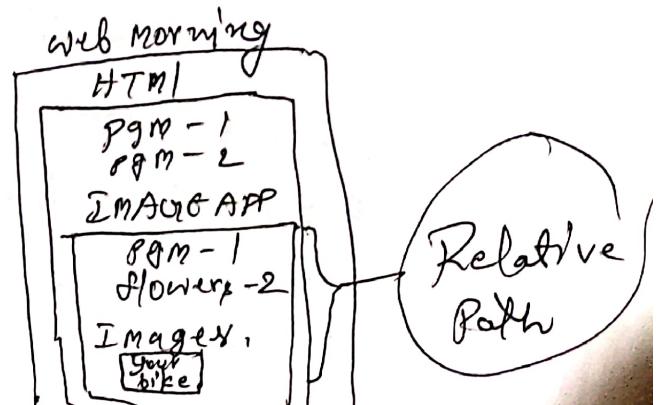
A Path is either relative (or) absolute. An absolute Path always contains the root elements & the complete directory list required to locate the file...

Relative Path:

* A relative Path needs to be combined with another Path in order to access a file. For example, `joe/foo` is a relative Path.

* A relative Path refers to a location that is relative to current directory. Relative Paths make use of two special symbols, a dot(.) & a double-dot(..) which translate in to the current directory & the parent directory.... The current directory is sometimes referred to as the root directory.

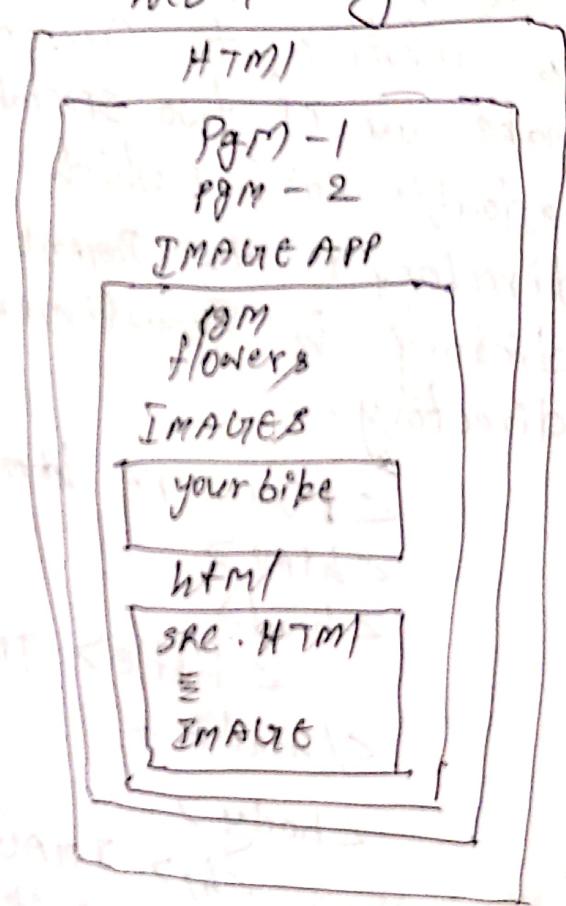
```
<!DOCTYPE html>
<html>
<head>
    <title> IMAGE APP </title>
</head>
<body>
    <h1> IMAGE ONE </h1>
    
    <h1> IMAGE TWO </h1>
    
</body>
</html>
```



```

<!DOCTYPE html>
<html>
<head>
<title> HTML IMAGES </title>
</head>
<body>
<h1> CARS WEB Page </h1>
<img src = ".. / IMAGES / CARS . jpg" width = "200px" height = "200px" >
</body>
</html>

```



CSS

- * CSS Mean "Cascading Style Sheet".
- * CSS was developed by "Hakon Wium" in the year of 1994 Oct 10.
- * CSS is used to design the HTML Pages.
- * By using CSS we can decide how the Content must be displayed on web Pages.

Types of CSS style sheets:

1. Inline style sheet.
2. Internal style sheet.
3. External style sheet.

1. Inline style sheet:

In this going to write CSS styles with in the "Tag" it self to write CSS styles we have to use on HTML attributes called or style attribute.

Syntax:-

```
<tag Name style="Property Name : Value;">....  
.....</tag Name>  
<!DOCTYPE html>  
<html>  
<head>  
  <title> Inline CSS </title>  
</head>  
<body style="background-color: yellow;">  
  <h1 style="color: red;"> Inline CSS </h1>  
</body>  
</html>
```

But here some limitations -

its 3 limitations;

i) limitation of "Inline CSS":

i) Duplicate Code:

In Case of inline CSS, we mentioned The CSS code is written by side of style attribute of each element. As a result is no need (There is no need)...

ii) Maintenance Problem:

Code will be difficult in Suddenly changing your styles (color or anything)

iii) Mixing of html & CSS:

in Single Page, you are applying the html & CSS Code so it's so difficult & confusing.

2. Internal style sheet:

* In this type of style sheet, we are going to write CSS within style tag.

* style tag must be placed within head tag.

Syntax:-

<style> → on which tag if you want apply this tag
Selector → ex: h1, tag; h2 tag; etc

{

< /* CSS style */

</style>

```
<!DOCTYPE html>
<html>
<head>
    <title>Internal CSS</title>
    <style>
        h1 {
            color: Green;
        }
    </style>
</head>
<body>
    <h1> Internal CSS </h1>
    <h2> Internal CSS </h2>
    <h3> Internal CSS </h3>
</body>
</html>
```

* Limitation of internal CSS:

By using internal CSS we can apply styles for single web pages (we can't reuse the CSS styles for another web pages).

3. external style sheet:

- * In this type of style sheet we are going to write styles with in separate file.
- * That separate file must be save with .css (dotcs)
- * The advance of CSS is we can connect one CSS files with n numbers of html files by using link tag.

```
<!DOCTYPE html>
<html>
<head>
    <title> External CSS </title>
    <link rel="stylesheet" href="external.css">
</head>
<body>
    <h1> external CSS </h1>
</body>
</html>
```

here
rel means
relationship
(relationship)

.css file or css
body, styles apply to body, h1, <body>
<h1> color: blue;
<h1> font-size: 2em;
<h1> border: 2px solid black;
<h1> margin: 10px auto;
<h1> padding: 10px;
<h1> background-color: yellow;
<h1> width: 30%;
<h1> height: 100px;
<h1> border: 1px solid black;
<h1> border-radius: 10px;
<h1> border: 1px solid black;
<h1> border: 1px solid black;

External CSS : (2) benefit of external CSS
we can use common style to multiple pages
we can change one file and reflect changes in all pages
we can add more style to one file
one file need to be linked to the page
one file need to be linked to the page

CSS Colors

Color names (hex)

- black (black)
- white (white)
- red (red)
- green (green)
- blue (blue)
- cyan (cyan)
- magenta (magenta)
- yellow (yellow)
- orange (orange)
- purple (purple)
- brown (brown)
- gray (gray)
- black (black)
- white (white)
- red (red)
- green (green)
- blue (blue)
- cyan (cyan)
- magenta (magenta)
- yellow (yellow)
- orange (orange)
- purple (purple)
- brown (brown)
- gray (gray)

Color functions

- color (color)
- linear-gradient (color, color)
- radial-gradient (color, color)
- repeating-linear-gradient (color, color)
- repeating-radial-gradient (color, color)

Color functions

- color (color)
- linear-gradient (color, color)
- radial-gradient (color, color)
- repeating-linear-gradient (color, color)
- repeating-radial-gradient (color, color)

```
<!DOCTYPE html>
<html>
<head>
    <title> CSS COLORS </title>
    <style>
        h1
        {
            /* Color : Pink; */
            /* Color : rgb(255, 105, 100); */
            /* background-color : yellow; */
            /* Color : #e99b77; */
            color : hsl(180, 60%, 50%);
        }
        /* Span
        {
            color : white;
            background-color : rgba(0, 0, 255, 0.2);
        }*/
    </style>

```

<h1> Give </h1>

eye drop extension in chrome

eye dropper </body>

① Search for eye dropper in google Chrome. ② Next click on first link & add extension to Chrome. ③ Select eye dropper & choose Pick up color from web page. ④ Now Pick your selected color & go to eye dropper the Select any color name & use...!

CSS Selectors

1. element (or) tag name Selector
2. Id Selector
3. Class Selector
4. Group Selector
5. Universal Selector
6. Attribute Selector.

CSS Combinators (or) Mixed Selector

- Descendant Selector
- Child Selector
- Adjacent Sibling Selector
- General Sibling Selector.

1. element Selector (or) tag name:

By using tag name Selector we can apply CSS styles for Selected html elements.

Syntax:

Tag name

/* CSS style */

ex: <!DOCTYPE html>
<html>
<head>

<title> CSS Selector </title>
<style>

h1

color: red;

h2

color: blue;

</style>

</head>

<body>

<h1> CSS Selector </h1>

<h2> CSS Selector </h2>

</head>

</html>

2. ID Selector :

- * By using Id Selector we can apply different CSS for some group of HTML elements.
- * To identify the HTML elements we use id element's attribute.
- * Why giving names for attribute always start with alphabet (alphabets)
- * To represent Id name we use #
- * Id name must be unique
- * We use them write CSS.

Syntax:

id name

{

/* CSS styles */

}

```
<!DOCTYPE html>
<html>
<head>
    <title> CSS SELECTORS </title>
<style>
```

abc

{

color: red;

}

xyz

{

color: blue;

}

mno

{

color: green;

}

</style>

</head>

<body>

`<h1 id="abc"> CSS Selector </h1>`

`<h1 id="xyz"> CSS Selector </h1>`

`<h1 id="mno"> CSS Selector </h1>`

`</body>`

`</html>`

3. Class Selector:

* By using class Selector We Can apply Same Css styles for Group of html elements.

* To represent class Selector We use (`. className`)

* CSS class Name Can be Duplicate

`.ClassName`

{

`/* css styles */`

}

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title> CSS class Selector </title>`

`<style>`

`.abc`

{

`color: red;`

}

`</style>`

`</head>`

`<body>`

`<h1 class="abc"> CSS Selector </h1>`

`<h1 class="abc"> CSS class Selector </h1>`

`<h1> class selector </h1>`

`</body>`

`</html>`

4. Group Selector:

* By using Group Selector we can apply some CSS styles for group of CSS Selectors.

* To represent group selector we use [, ()]
selector1, selector2, selector3, selectorN,

Syntax:

selector1, selector2, selector3

{ /* CSS styles */ }

(style block) is enclosed inside the group of
(selectors).

<!DOCTYPE html> (block id and block class)

<html>

<head>

<title> CSS Selector </title>

<style>

#abc, .mno, h3

{

color: yellow;

}

</style> </head> (block id and block class) <body>

<body>

<h1 id="abc"> CSS Selectors </h1>

<h1> CSS Selector </h1>

<h2 class="mno"> CSS Selectors </h2>

<h2 class="mno"> CSS Selectors </h2>

<h2> CSS Selector </h2>

<h3> CSS Selector </h3>

</body>

</html>

5. Universal Selector:

- * By using universal Selector We can apply same CSS styles for whole Web Page Content.
- * To represent universal Selector We Can use [* Mark] (star Mark)

```
<!DOCTYPE html>
<head>
<title> CSS Selector </title>
<style>
  * {
    color: Green;
  }
</style>
</head>
<body>
  <h1 id="abc" style="color: red;"> CSS Selector </h1>
</body>
</html>
```

6. Psudo Selector:

* Psudo Selector is a Special Selector which is having some dynamic changes.

(from Syntax) No number (number) decrease of 1
(decrements)

→ Selector : Psudo Selector

{

/* Cs styles */

}

</Doctype html>

<html>

<head>

<title> Css Selector </title>

<style>

h1 : hover

{

color : blue;

</style> "Introducing CSS" color = bi (1)

</style>

</head>

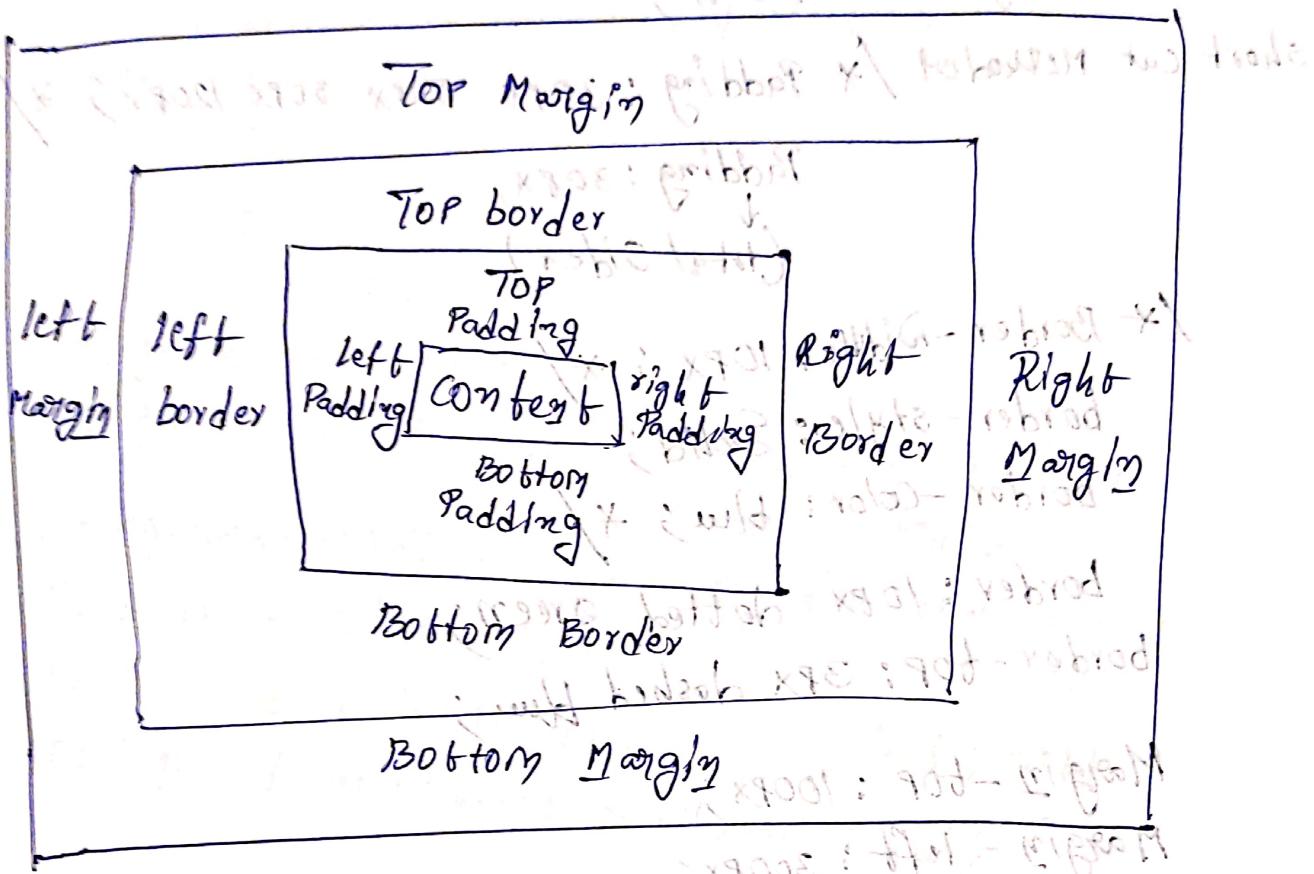
<body>

<h1> Css Selector </h1>

</body>

</html>

CSS Box Model



border: Solid, dotted, dashed, inset, outset, ridge, groove etc..!

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title> CSS Box Model </title>
```

```
<style>
```

```
div
```

```
{
```

```
    color : Red;
```

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

```
    width : 60px;
```

```
<body>
```

```
</body>
```

```
<html>
```

```
</html>
```

/* Padding - top : 30px;

Padding - right : 30px;

Padding - left : 30px;

Padding - bottom : 30px;

short cut Method /* Padding : 30px 50px 50px 120px; */

Padding : 30px;

(total sides)

/* Border - width : 10px; */

border - style : Solid;

border - color : blue; */

border : 10px dotted green;

border - top : 3px dashed blue;

Margin - top : 100px;

Margin - left : 300px;

Margin : 300px; border < body > bit ^{div means} _{body} ^{division}

}

</style>

<body>

<div> BOX ONE </div>

</body>

</html>

<html> <head> </head>

<body>

<h1>

<style>

viib

{ border : color; }

{ border : color - blue; }

{ XX : }

CSS Background Properties

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS background </title>
</head>
<body>
    color: Red;
    background-color: yellow;
    background-image: url('copy Address Path');
    background-repeat: no-repeat; (or) repeat;
    background-size: cover;
    background-position: top; bottom; left; right;
    background-attachment: scroll; (or) fixed;
</body>
</html>
```

Syntax:

Background : Color Image repeat Position attachment;

Color : (Solid, Web Safe Colors, RGB, Hex)

Image : (Local File, URL)

Repeat : (repeat, repeat-x, repeat-y, no-repeat)

Position : (top, bottom, left, right, center, center, center, center)

Attachment : (scroll, fixed)

Mixed Colors

Background: linear-gradient(Directions, color₁, color₂, color₃ ... color_n)

shape

radial-gradient(Shape, color₁, color₂, color₃ ... color_n)

like circle ellipse etc.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <title> CSS Mixed Colors </title>
```

```
<style> .box { width: 300px; height: 200px; background-color: black; }
```

```
h1 { font-size: 24px; color: white; text-align: center; }
```

```
.box::before { content: "Mixed Colors"; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); }
```

```
color: red; border: 3px solid blue; border-radius: 50%; width: 100px; height: 100px; background-color: black; }
```

Background: radial-gradient(ellipse, blue, pink, green, sky blue, black);

Border: 3px Solid blue;

Padding: 30px;

}

```
<style>
```

```
<body>
```

```
<h1>
```

:root {

```
background-color: black; color: white; font-size: 16px; }
```

```
</head>
```

```
<body>
```

```
 <h1> CSS Mixed Colors </h1>
```

```
</body>
```

```
</html>
```

HTML LINKS

```
<a href="#"> Shop in Amazon </a>
<!DOCTYPE html>
<html>
<head>
<title> HTML Links </title>
</head>
<body>
<h1> Hyperlinks </h1>
<a href="http://www.amazon.com" target="blank"> Shop in Amazon </a>
</body>
</html>
```

Pseudo class :- for anchor tag

```
<!DOCTYPE html>
<html>
<head>
<title> CSS anchor tag </title>
<style>
a {
    color: red;
    text-decoration: none;
}
a:hover {
    color: blue;
}
a:active {
    color: green;
}
a:visited {
    color: pink;
}
</style>
<body>
<h1> CSS </h1>
</body>
</html>
```

Block Level elements

Block level elements will be occupying Complete Width of the Web Page...
for ex: All heading tags are block level elements,

<div> tag also Block level element.

<div>, header, footer tags also Block level elements.

Inline level elements:

Inline level elements will be occupying the Width Much if required.

eg: " ex: <a>
 tag".

CSS display Property:

By using display Property we can Create html elements as either blocklevel (or) inline level.

```
<!DOCTYPE html>
<html>
<head>
<title> CSS elements </title>
<style>
body {
    margin: 0px;
    width: 100%
}
h1 {
    color: white;
    background-color: blue;
}
```

border: 3px solid blue;

Margin: 0px; vertical margin

Display: inline;

→ a is part of the document and it is up to the browser to match a with generated presentation blocks from the style sheet.

Σ

color: red;

background-color: yellow;

border: 3px solid green;

display: block;

margin-top: 10px;

Σ

one block of </style> has no height

<html><head></head>

<body> <div> first part </div>

<h1> Content</h1>

<h1> CSS elements</h1>

 Home

 about

</body>

</html>

→ first element is first part </div>

→ Content <div> is part of first part </div>

→ second element is second part </div>

→ CSS styles <style> </style>

CSS Combinators (Or) is used

Mixed Selector

We go for CSS Combinators When there is a Parent child relationship between the html elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> CSS Mixed Selector </title>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<span> Span one </span>
```

```
<h1> Heading One <span> Head tag child span
```

```
<h2> Heading Two </h2>
```

```
<h3> Heading Three </h3>
```

```
<header>
```

```
<div> <span> <div>
```

```
<div> <h1> Heading One </h1>
```

```
<h2> Heading Two </h2>
```

```
</div>
```

```
<footer>
```

```
<h1> heading one </h1>
```

```
<h2> heading two </h2>
```

```
<h3> heading Three </h3>
```

```
</footer>
```

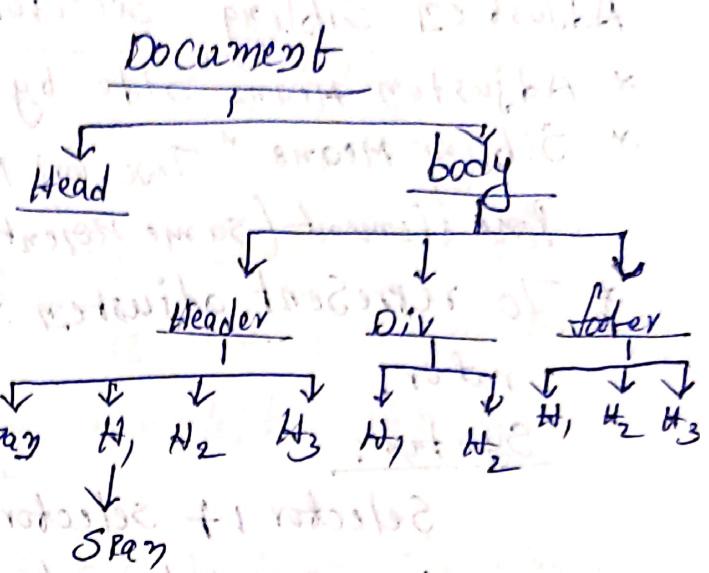
```
</body>
```

```
</html>
```

* Parent elements means

an element which contains
one or more child elements.

* Leaf element means an
element which doesn't
have any single
child element.



Descendent Selector:

* Descendent selector means any element which are having coming and under selected element (descendent element) can be child element or child child element up to leaf element)

* To represent Descendent Selector we use Space ...

Syntax:

Selector₁, Selector₂, Selector₃ ... Selector_n

<div> /* CSS styles */

Child Selector:

By using Child Selector we can target direct child elements. (>) To represent Child Selector we use greater than operator (>) before the name of selector

Syntax:

Selector₁> Selector₂> Selector₃> ... Selector_n

<div> /* CSS styles */

Adjuster Sibling Selector:

- * Adjuster Means "Side by Side".
- * Sibling Means "Two (or) More child element With Parent Parent element (same Parent)".

To represent Adjuster Sibling Selector We use "&" Operator.

Syntax:

Selector 1 & selector2 & selector3 & ... & selector n &
/* Css style */

}

<style>

:rotator {color: red;}

h1 & h2

color: red;

at q, n

<style>

(technically bad)

<head>

<body>

<header>

 Span One

<h1> heading One

 Heading one child Span

<h2> Heading Two

General Sibling Selector:

- * By using General Sibling Selector, We apply style any sibling elements of (< >) .

To represent General Sibling Selector (& tilde)

Syntax:

Selector 1 ~ Selector 2 ~ Selector 3 ~ ... ~ Selector n ~

{

<css style> & /style/ {<h1> & h2

}

<style>

color: Green;

</style>

CSS Position Property

By using Position Property We can change the place of an html elements. To change the place of an html elements we have to use Position Property along with Position Property top, left, right, bottom...

1. Position static:

By using static Position We Can't change the place of an html elements.

```
<!DOCTYPE html>
<html>
<head>
    <title> Position Property </title>
    <style>
```

div

{

Background : linear-gradient(45deg, black, yellow, green, white, blue)

width : 200px;

text-align : center;

color : white;

padding : 16px;

border-radius : 10px;

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

}

#static

{

Position : static;

top : 100px;

left : 100px;

}

</style>

<head> (will go to 22)

<body>

<div id="static">

<p> static </p> </div>

</body>

</html> it will also extend to static

so position relative will not affect static

2. Position: relative;

By using relative Position We Can change Place of an html Element from there starting place.

<style>

{

#relative

position: relative;

Position: relative;

top: 20px;

left: 100px;

<html>

<body>

<div>

<div>

100

3

(will go up) Horizontal scroll: horizontal

</style>

(will scroll) vertical

<body>

display: block

<div id="relative">

<p> relative </p>

</body>

<div>

</html>

position: absolute

top: 20px; left: 100px;

position: relative;

position: absolute

<div id="relative">

<p> static </p>

</div>

3. Position: Fixed;

By using fixed position we can change the place (position) of html elements. or well off we can make html element fixed. (given value fixed)

fixed

{

Position: Fixed;

top: 150px;

left: 300px;

}

</style>

</head> above <body> : browser side

<body>

<div id="fixed">

<p> static <p>

<div id="fixed">

<p> fixed </p>

</div>

4. Position: sticky;

By using sticky position we can make an html [position: sticky] elements to be sticky for a given values.

{ position: sticky }

sticky

width: 100% body - xed

position: sticky;

top: 0px; sheet

{

</style> browser side

<div> <head>

Programs

```
<!DOCTYPE html>
<html>
<head>
    <title> CSS Links </title>
    <style>
        body {
            margin: 0px;
            font-family: serif;
        }
        div a {
            background: linear-gradient(45deg, orange, blue, green);
            color: white;
            text-decoration: none;
            margin-right: 30px;
            display: inline-block;
            padding: 5px;
            border-radius: 20px;
            width: 200px;
            height: 40px;
            text-align: center;
            position: relative;
            z-index: 1;
        }
        div a: hover {
            background: linear-gradient(45deg, orange, blue, green);
            color: white;
            box-shadow: 0 0 10px 5px;
        }
        #header {
            background-color: blue;
            color: white;
            text-align: center;
            padding: 10px;
        }
        #header h1 {
            margin: 0px;
        }
        #header p {
            margin: 0px;
        }
    </style>
</head>
<body>
    <h1> Welcome to my website! </h1>
    <p> This is a simple website for learning CSS. </p>
    <div>
        <a href="#"> Home </a>
        <a href="#"> About </a>
        <a href="#"> Contact </a>
    </div>
</body>

```

Padding: 20px;

]

header h1

{ margin: 0px;
Main nav

{

margin-top: 0px;

position: sticky;

(bottom)

top: 0px;

</style>]

</head>

<body>

<div id="header">

<h1> JS Spiders </h1>

</div>

<div id="Main-nav">

 Gallery

 News

 about

 Content

 HOME </div>

</body>

</html>

para

```
<!DOCTYPE html>
<html>
<head>
<title> CSS PARAGRAPH </title>
<style>
div
{
    background: linear-gradient(45deg, orange, blue, white, red);
    border-radius: 10px;
    box-shadow: 0 0 10px blue;
    padding: 30px;
    width: fit-content;
    margin: auto;
    text-align: justify;
    background: linear-gradient(45deg, blue, pink, green);
    color: white;
    padding: 20px;
    border-radius: inherit;
    box-shadow: inherit;
    line-height: 35px;
    text-indent: 90px;
    word-spacing: 20px;
    letter-spacing: 20px;
    font-size: 20px;
    font-style: italic;
    font-weight: bold;
    font-family: arial, sans-serif;
}

```

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

```
<p> ..... </p>  
<p> ..... </p>
```

```
</div>
```

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

Font family

```
<link href="font.css" type="text/css" rel="stylesheet"/>
```

~~Font family~~

```
<link href="font.css" type="text/css" rel="stylesheet"/>
```

old family { monospace, sans, serif }

New family { sans-serif, cursive }

Fonts

Go to Google



Google fonts



Select font



Font family

Copy



Link Copy & Paste.

fonts - google.

* go to google.

* Type google fonts.

(To show some fonts style.)

* After pick the style & copy;

* and paste that style in font family;

* copy that link (font link) in head part (rel="cursive link tag"))

* see results.

HTML TABLE

<Table tags>

<tr>

<th>

<th>

<td>

<th>

<th>

<td>

<td>

<th>

<!DOCTYPE html>

<html>

<head>

<title> HTML TABLE </title>

<style>

Body

font-family: Comic Sans ms', Sans-Serif;

font-size: 18px;

background-color: #f2f2f2;

border-collapse: collapse;

text-align: center;

width: 100%;

border: 1px solid black;

border: 1px solid black;

border: 1px solid black;

border: 1px solid black;

width: 100%;

color: blue;

border: 1px solid black;

border: 1px solid black;

(table, th, td

border: 1px solid black;

border: 1px solid black;

border: 2px solid blue; </td>

text-align: center; </td>

border-collapse: collapse; </td>

border-radius: 20px; </td>

</td>

th, td
 <td> </td>

<td>

box-shadow: 0 0 10px blue;

<td> 0 0 10px blue;

tr
 <tr>

<td>

background: linear-gradient (45deg, orange, white, green);

<td>

td:hover

<td>

background: linear-gradient (45deg, dark slate gray, deep pink);

color: white;

cursor: pointer;

</td>

</style>

</head>

<body>

<h1> HTML TABLES </h1>

<table cellspacing="20px" cellpadding="10px">

<tr>

<th> S.NO </th>

S Name	S Marks	S Age	S Courses
Harish	95	22	HTML
Dileep	96	22	CSS

Row, Span, Column Span:-

HTML TABLE WITH COLSPAN & ROWSPAN		
HTML TABLE WITH COLSPAN & ROWSPAN	HTML TABLE WITH COLSPAN & ROWSPAN	HTML TABLE WITH COLSPAN & ROWSPAN
HTML TABLE WITH COLSPAN & ROWSPAN	HTML TABLE WITH COLSPAN & ROWSPAN	HTML TABLE WITH COLSPAN & ROWSPAN

<tr>

<th> Batch Code </th>

<th> S Name </th>

<th colspan="2" rowspan="2">Subject

</tr>

<tr>

<td rowspan="3">06WES

<td> Naveen </td>

<td> html </td>

<td> CSS </td>

</tr>

<tr>

<td> Revathi </td>

<td> html </td>

<td> CSS </td>

</tr>

<tr>

<td> Mahesh </td>

<td> html </td>

<td> CSS </td>

</tr>

</table>

</body>

</html> //

Final Output

Batch Code

S Name

Subject

06WES

Naveen

html

CSS

HTML FORMS

```
<form>
<table><tr><td><label> </td><td> </td></tr>
<input>
<table area>

<!DOCTYPE html> <html> <head>
<html/> <title> HTML FORMS </title>
<head>
<title> HTML FORMS </title>
<style>
div {background-color: #ccc; border-radius: 10px; border: 2px solid black; padding: 10px; width: 300px; margin: auto; text-align: center; font-family: sans-serif; font-size: 14px; color: #333; margin-top: 50px; }
#submit {background-color: #333; color: white; border: none; padding: 10px 20px; font-size: 16px; cursor: pointer; margin-top: 10px; }
#submit:hover {background-color: #555; transition: background-color 0.3s; }

<style>
</style>
</head>
<body>
<div>
<table border="1">
<tr>
<td> <input type="text" name="name" placeholder="Name" /> </td>
<td> <input type="text" name="email" placeholder="Email" /> </td>
</tr>
<tr>
<td colspan="2" style="text-align: center; padding-top: 10px;"><input type="submit" value="Submit" />

```

```

<td><table> user Name </table></td>
<td>< input type = "text" ></td>
</tr>

<tr>
    <td><table> Password </td>
    <td>< input type = "Password" ></td>
</tr>

<tr>
    <td align = "center" Colspan = "2" >
        <input type = "Submit" > <br/>
        <input type = "reset" >
    </td>
</tr>

```

Method :-

i) <table> (row : rowspan - browser does
 </table> (row : form - browser does
 </form> (row : 51c - browser does
 <div> (row : 51c - browser does
 </body> (row : 51c - browser does
 </html> (row : plmnot - just

. File - row2 (row : plmnot - just

Method attribute :- Method attribute is a form attribute, The default value of method attribute is "get" (get method is responsible for getting data to the server)

* Always Method attribute value must be Post (Post is the responsible form pushing the data to the server)

ii) Action attribute :- It is used for Server Side url.

Input type Values:-

color; date; datetime-local; number; text; Password; Submit;
 reset; radio; check box; week; month; range; file.... etc.

```
Input [type = "text"],   
tag name [Att Name = "Value"]
```

```
1. <html> <body> <br> <input type="text" value="Hello World" /> </body> </html>  
<!DOCTYPE html>  
<html>  
<head>  
<title> Signup form </title>  
<style> <!-- font-size: 10pt; margin: 0; -->  
Body <font-size: 10pt; margin: 0; --> HTML  
{  
background-image: url("background.jpg");  
background-repeat: no-repeat;  
background-size: cover;  
background-position: center top;  
font-family: "Comic Sans MS", sans-serif;  
  
fieldset { border: 0; }  
h1, h2, h3, h4, h5, h6, h7, h8, h9 {  
border: 2px solid blue;  
border-radius: 10px 10px 10px 10px;
```

Background-color: rgba(255, 255, 255, 0.4);

}

fieldset legend {
border: none;
background-color: transparent;

}

border-radius: inherit;

<input> font-size: inherit;
border: inherit;

background: linear-gradient(45deg, blue,

color: white, green, red);

padding: 5px 15px;

color: inherit; font-size: inherit;

border: none; border-radius: 5px; background-color: transparent;

<input> {
background-color: transparent; color: white; border: none; border-radius: 5px; width: 100%; height: 20px;}

color: white; border: none; border-radius: 5px; width: 100%; height: 20px;}

}

<input type="text">, <input type="password">

<input type="number">, <input type="checkbox" checked="checked" />

<input type="radio" checked="checked" />

color: white; border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent;

height: 20px; border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent;

<input type="text" border="1" style="border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent; outline: none;"/>

<input type="text" border="1" style="border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent; outline: none;"/>

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

border-radius: 10px;

<input type="text" border="1" style="border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent; outline: none; box-shadow: 0 0 0 10px blue;"/>

<input type="text" border="1" style="border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent; outline: none; box-shadow: 0 0 0 10px blue;"/>

<input type="text" border="1" style="border: none; border-radius: 5px; width: 100%; height: 20px; background-color: transparent; outline: none; box-shadow: 0 0 0 10px blue;"/>

```
</style>
<head>
<body>
    <h1> Sign up </h1>
    <form>
        <fieldset>
            <legend> Sign up Here </legend>
            <table>
                <tr>
                    <td><table> First name </table></td>
                    <td><input type = "text" placeholder = "enter user name" /></td>
                <td><table> Second name </table></td>
                <td><input type = "text" /></td>
            </tr>
            <tr>
                <td><table> Password </table></td>
                <td><input type = "password" /></td>
                <td><table> Confirm Password </table></td>
                <td><input type = "password" /></td>
            </tr>
            <tr>
                <td><table> Mobile number </table></td>
                <td><input type = "number" /></td>
            </tr>
            <tr>
                <td><table> Text area </table></td>
                <td><textarea></textarea></td>
            </tr>
```

<tr>

<td colspan="2"><label> Gender </label></td>
<td><input type="radio" name="x1"> Male </td>
<td><input type="radio" name="x2"> Female </td>
</tr>
<tr>
<td colspan="2"> Course </td>
<td><input type="checkbox"> Java </td>
<td><input type="checkbox"> CSS </td>
<td><input type="checkbox"> Python </td>
</tr>

<tr>

<td><label> Select a City </label></td>
<td><select>

<option> Select </option>

<option> Bangalore </option>

<option> Mumbai </option>

<option> Tirupathu </option>

</select>

</td>

</tr>

<tr><td> </td>

<td> </td>

<td> </td>

<td>

<td>

<td> </td>

<td>

<td> </td>

HTML FRAMES

<H1>

<H1> <frame> </frame> <H1> <H1> <H1>

* HTML frames are used to run multiple Web pages on a Browser.

<frame set> } These tags are used to Create frames
<frame>

<frame set> <frame> <frame> <frame> <frame>
 Frame Set
 <frame> <frame> <frame> <frame>
 <frame> <frame> <frame> <frame>
 <frame> <frame> <frame> <frame>



Num of Web pages.

<H1> <H1> <H1> <H1>

* Whenever we are using frame set tags we don't use body tag.

<html> <head> <title> </title> </head>

<!DOCTYPE html> <html>

<html> <head> <title>

<H1> HTML FRAMES </H1>

</head>

<frame set cols="50%, 50%">

<frame src="frameOne.html"></frame>

<frame src="frameTwo.html"></frame>

</frame set>

</html>

- * If we are going to create 'n' number of frames with in frame set for each frame size must be decided.
- * To occupy remaining px (or) % we need to give (*) operator.

Sample.htm

```

</head>
<frameSet cols="2" = "10px, 200px, *">
  <frame src="frame1.htm">
  <frame src="frame2.htm" width="100%" height="100%"/>
</frameSet>
</head>
<body>
  <h1>Frame one</h1>
  <h1>Frame two</h1>
</body>

```

Another file or:-

```

<!DOCTYPE html>
<html>
  <head>
    <title> HTML FRAMES </title>
  </head>
  <frameSet cols="2" = "100px, 200px, *">
    <frame src="f1.htm">
    <frame src="frame2.htm">
    <frame src="f3.htm">
  </frameSet>
</html>

```

Another file or:-

```

<!DOCTYPE html>
<html>
  <head>
    <title> FRAME ONE </title>
  </head>
  <body background="cyan">
    <h1> Frame one </h1>
  </body>

```

 target="fr2" > click here

</body>

</html>

Nested Frame Sets:

<!DOCTYPE html>

<html>

<head>

* (row1, col1) = 1st script

<title> Nested Frame Set </title>

</head>

<frameSet rows="

20%, 20%">

<frameSet cols="

20%, 20%">

<frame src="FrameOne.html"></frame>

<frameSet cols="20%, 20%">

<frame src="FrameTwo.html"></frame>

<frame src="FrameThree.html"></frame>

</frameset>

<frame src="frameFour.html">

</frame></frameSet>

</html>

<script></script>

Note:

Frame set tags are outdated. Because it's not supported to the browser.

So that reason, we are using now a days frame tags.

<frame src="frameFour.html">

```
<!DOCTYPE html>
<html>
<head>
    <title>Nested frame sets</title>
</head>
<frameset>
    <frame>
        <body>Hello friends</body>
    </frame>
</frameset>
</html>
```

verb

"i frame tags":-

- * It is a ~~an~~ HTML-5 Version tag.
- * Now a days using this type tags (i frame tags)

```
<!DOCTYPE html>
<html>
<head>
    <title>i frame tag</title>
</head>
<body>
    <h1>html frame tags</h1>
    <frame src="frameone.htm" width="500px" height="300px" border="0px"></frame>
</body>
</html>
```

* The help of i-frame tag we can add google Map:

Google Map

Search for location

↓
click on share button

↓
click on embed map

↓
Copy to the i-frame tag &

Paste in to the html file

* Playing youtube Videos:

Go to youtube

↓
Search for Video tutorial

↓
Click on share button

↓
click on embeded (privacy we enable it)

Before copying i-frame tag scroll down &
and enable the privacy enhanced mode,

↓
Copy that, & Paste in to <the html file>

* Making 360° Images:

* To check 360° Images refer "pixexid.com"

* To convert any Image into 360° deg.

1. Go to google & Search for "moments to 360".

2. Click on first link & Signup in to moments 360deg..

3. Once you Signup verify your email & Logout & login..

4. Click on Camera icon & upload the Images..

5. click on the img button & click on shape button...
6. Selected on options called as embed (or) shape alink.
7. click on Create a link... ~~copy pasting~~
8. Copy the link & Paste it in to the frame tag Source attribute....

<embed>

<object>

<object>

VBs

\$

CSS transforms

Transformation means changing html elements from one state to another state.

Change the style you want
 direction of move & state of object
`<!DOCTYPE html>` `<html>` `<head>` `<body>` `<div>`

`<div> CSS transformation </div>`

`<style>`

`div { width: 100px; height: 100px; }`

`width: 100px; height: 100px;`

`background-color: yellow; border: 1px solid black;`

`color: red; font-size: 20px; font-weight: bold;`

`padding: 30px; width: 74px; height: 74px;`

`width: 74px; height: 74px; border: 3px solid blue;`

`border: 3px solid blue; }`

`/* Transition Properties */`

Transition-Property: Padding, background-

transition-duration: 2s, 2s;

transition-timing-function: linear, linear;

transition-delay: 0s, 0s;

}

`div:hover`

{

`padding: 60px;`

`background-color: limegreen;`

}

`</style>`

`</head>`

<div>

Transer form & style

</div>

</body>

</html>

short cut properties
→ (short cut of ~~css~~ ~~color~~ bold)

Transition: Padding is linear os;
background color is Linear
os;

another short cut: all is linear os;

Transition: all is linear os; (transition
short cut of ~~css~~ ~~color~~ bold).

css transform

transformer: translate — changing the position x,y,z
rotate — rotate
scale — zoom
skew — slant

10 trans form

20 trans form

30 trans form

translate(x value)

translate(x,y)

translate 3d

translate(y value)

(x,y,z)

translate(z value)

rotate value

Note: x,y values can be in px (0);

but z value ~~can~~ must be in px;

<div>
<body>
<html>

background-color: blue;

color: white;

width: 74px;

padding: 30px;

border-radius: 10px;

border: 3px solid red;

transition: all 2s linear;

}

div: hover

{

transform: perspective(200px)

translate 3d(100px, 100px,

-100px);

}

=====

2D Transformer

2D transformer

3D transformer

rotate X(x)

rotate (x, y)

rotate 3d(x, y, z,

any degree)

rotate Y(y)

(deg)

rotate Z(z)

degrees

Note:

X, Y, Z values must be in degree (deg)

(5.6.7)

{

div: hover

{

transformer: perspective(200px) rotateY(90deg)

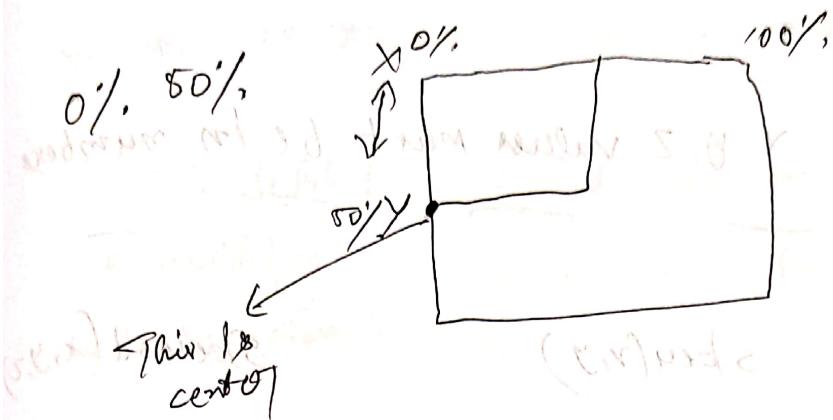
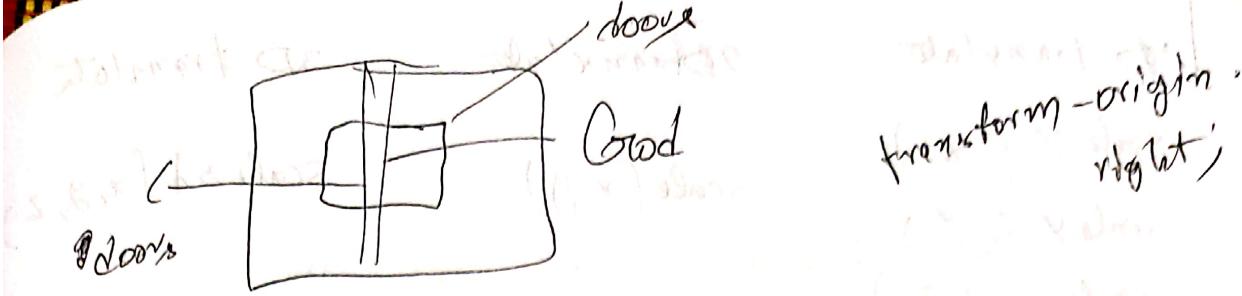
{

</style>

<head>

<body>

<div>



transform:

If used to fit one Imaginary robot.

Width get from window 500px



div: hover

Var



transform:

width: 740px

padding: 30px;

border-radius: 10px;

border: 3px solid red;

transform-origin: 100% 50%;

transition: all 2s linear;



div: hover

transform: perspective(200px) rotateZ(30deg);



</style>

<head>

<body>

1D translate

scale x (x_0)

scale y (y)

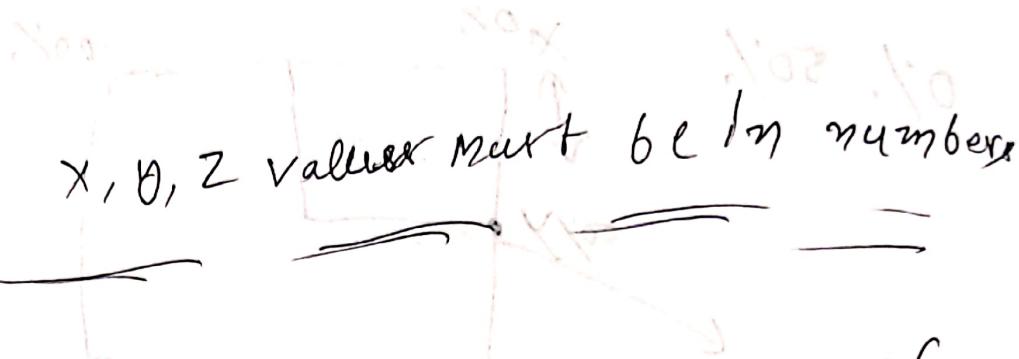
scale z (z)

2D trans (glo)

scale (x, y)

3D Trans (glo)

scale 3d (x, y, z)



skew (x)

skew (y)

skew (z)

skew (x, y)

skew 3d (x, y, z)

- transform rectangle into parallelogram
 x, y, z values must be in degrees,

=
overflow + hidden)

frames form - origin: left;

background-color

position: top-left

top: 0px; left: 0px; any deg;

width: 100%; height: 100%;

(border: none; border-radius: 0px; border: 1px solid black; background-color: white; position: absolute; top: 0px; left: 0px; width: 100%; height: 100%;)

{ background-color: -webkit-linear-gradient(white, black); }

{ background-color: linear-gradient(white, black); }

E

REVIEW: vba

Cg Animation

we can define inspace transform

we go Cg Animation over for common transformation.

To define state of element so name can be anything.

To define state of element
we represent by state variable

key frame
key frame can be
represented %

key frame must be
represented by frame
but frame can be
any transform

must be Cg
proportion

key frame
rule - rule
block

key frame block

< head >
< style >

div

{
background-color: blue;

color: white;

width: 90px;

padding: 10px;

border: 5px solid red;

position: relative;

top: 100px;
left: 200px;

/> Animation properties

~~Animation~~ - name: Anim;

Animation - duration: 1s;

Animation - timing-function: linear;

Animation - delay: 0.0s;

Animation - iteration-count: infinite;

Animation - direction: normal;

}

② key frames

{

0%:

{

transform: perspective(200px)

rotateZ(0deg);

background-

color: blue;

{

50%:

{

background-color: pink;

I

100%

E

frame form: Perspective (200px) rotateZ(360deg)

background-color: yellow; border-radius: 50%;

border-radius: 50%;

</style>

</head>

<body>

<div>

<p> Animation API </p>

</div>

</body>

</html>

short cut property for Animation:

Animation: anim as linear as infinite normal;



<style>
div

background: linear-gradient(45deg, orange, blue, green);

width: 100px;

height: 80px;

position: relative;

animation: move 2s linear infinite alternate-reverse;

}

@keyframes move

{

0%;

{

top: 0px;

left: 10px;

border-radius: 0px;

25%;

{

top: 0px;

left: 150px;

border-radius: 60px 0px 0px 0px;

50%;

{

top: 150px;

left: 150px;

border-radius: 0px 60px 0px 0px;

}

75%
top: 15px;
left: 10px; border-radius: 10px; border: 1px solid black; height: 100px; width: 100px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);

3
100% of element's width & height.

{
top: 0%; left: 0%; border-radius: 0px; border: 0px; height: 100px; width: 100px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);

100% of element's width & height.

border-radius: 50%; border: 1px solid black; height: 100px; width: 100px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);

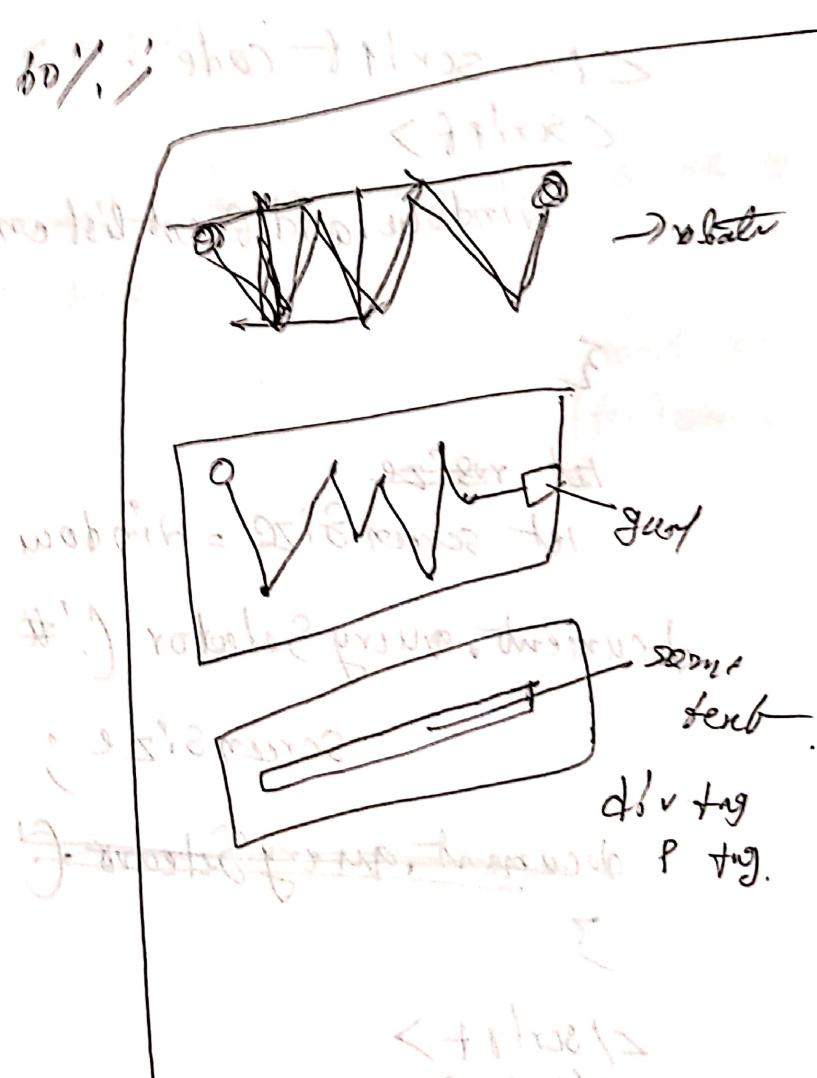
3
<style>

</head>

</html>

<body><script> window.onload = function() {
 let root = document.documentElement; let style = document.createElement('style'); style.type = 'text/css'; style.innerHTML = `#box {border: 1px solid black; border-radius: 50%; width: 100px; height: 100px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);}`; root.appendChild(style); }</script>

<body><script> window.onload = function() {
 let root = document.documentElement; let style = document.createElement('style'); style.type = 'text/css'; style.innerHTML = `#box {border: 1px solid black; border-radius: 50%; width: 100px; height: 100px; position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);}`; root.appendChild(style); }</script>



<html>
<body>
</html>

<body><div>

<div>

<div>

```
<!DOCTYPE html>
<html>
<head>
<title> Browser Pixel Counter </title>
<head>
<body>
<h1 align="center" style="font-size: 9px; id=">
    "screen"></span></h1>
```

```
<!-- script code -->
```

```
<script>
```

```
window.addEventListener('resize', function() {
```

```
{
```

```
let size =
```

```
let screenSize = window.innerWidth;
```

```
document.querySelector('.#screen').innerHTML =
```

```
    screenSize;
```

```
} document.querySelector('.#screen').innerHTML =
```

```
}
```

```
</script>
```

```
</body>
```

```
<html>
```

```
<head><title>
```

```
<style>
```

```
body
```

```
{
```

```
background-color: black;
```

HTML Layouts

to home

HTML-5 tags

header section
nav bar ~~article~~
aside footer

img

Gallery
figure
~~audio~~
figure caption

Details
summary

Audio | Video | embed

Go to google Search for Animal CSS

click on first link

<head>

<body>

<h1>

font-size
(font-size)

background-color

background-color: black;

HTML layout

<html>

<head>

</head> HTML Layout </html>

<link rel = "stylesheet"

href = "http://"

/>

<link rel = "stylesheet" href = "layout.css">

<link rel = "stylesheet" href = "fontawesome

/css/all.css">

</head>

<body>

header>

<h1 class = "Animal"

class = "Animal" style = "text-align: center; font-size: 2em; color: red; margin-bottom: 10px;"></h1>

class = "Animal" style = "text-align: center; font-size: 1.5em; color: green; margin-bottom: 10px;"></h2>

class = "Animal" style = "text-align: center; font-size: 1.2em; color: blue; margin-bottom: 10px;"></h3>

class = "Animal" style = "text-align: center; font-size: 1.0em; color: black; margin-bottom: 10px;"></h4>

class = "Animal" style = "text-align: center; font-size: 0.8em; color: grey; margin-bottom: 10px;"></h5>

class = "Animal" style = "text-align: center; font-size: 0.6em; color: grey; margin-bottom: 10px;"></h6>

class = "Animal" style = "text-align: center; font-size: 0.4em; color: grey; margin-bottom: 10px;"></h7>

class = "Animal" style = "text-align: center; font-size: 0.2em; color: grey; margin-bottom: 10px;"></h8>

class = "Animal" style = "text-align: center; font-size: 0.1em; color: grey; margin-bottom: 10px;"></h9>

class = "Animal" style = "text-align: center; font-size: 0.05em; color: grey; margin-bottom: 10px;"></h10>

class = "Animal" style = "text-align: center; font-size: 0.02em; color: grey; margin-bottom: 10px;"></h11>

class = "Animal" style = "text-align: center; font-size: 0.01em; color: grey; margin-bottom: 10px;"></h12>

class = "Animal" style = "text-align: center; font-size: 0.005em; color: grey; margin-bottom: 10px;"></h13>

class = "Animal" style = "text-align: center; font-size: 0.001em; color: grey; margin-bottom: 10px;"></h14>

class = "Animal" style = "text-align: center; font-size: 0.0005em; color: grey; margin-bottom: 10px;"></h15>

CSS Media Queries

CSS media queries are used to make screen sizes responsive.

<html>

<head>

<style> CSS MEDIA QUERIES </style>

<style>

h1

{

text-align: center;

}

div h2

{

text-align: center;

display: none;

}

@media (max-width: 500px)

{

body

{

background-color: green;

}

div-h2: nth-child(1)

{

display: block;

}

@media (min-width: 501px) and (max-width: 720px)

{

body

{

background-color: red;

}

div h2: nth-child(2)

{

display: block;

}

}

@media (min-width: 720px) and (max-width: 920px)

{

body

{

background-color:

#f0f0f0;

padding:

10px;

border:

1px solid black;

border-radius:

5px;

text-align:

center;

font-size:

16px;

font-family:

"Times New Roman";

color:

black;

border:

1px solid black;

border-radius:

5px;

text-align:

center;

font-size:

16px;

font-family:

"Times New Roman";

color:

black;

div h2: nth-child(3)

{

display: block;

}

}

body

{

margin: auto; border: 1px solid black;

background-color: #f0f0f0;

border-radius: 5px;

text-align: center;

font-size: 16px;

font-family: "Times New Roman";

color: black;

border: 1px solid black;

border-radius: 5px;

text-align: center;

font-size: 16px;

font-family: "Times New Roman";

color: black;

{

display: block;

}

</style>

<head>

<body>

div h1

<h1> screenSize :
</h1>

<div>

<h2> small screen </h2>

<h2> medium screen </h2>

<h2> Large screen </h2>

<h2> x-large screen </h2>

</div>

<script>

window.addEventListener('resize', function() {

let screenSize = window.innerWidth;
document.querySelector('#screen').innerText =
` \${screenSize} px`;

</script>

<body>

<h1>

Screen size :

Now we will add a button which will change

the screen size.

For this we will use

the document.querySelector('button')

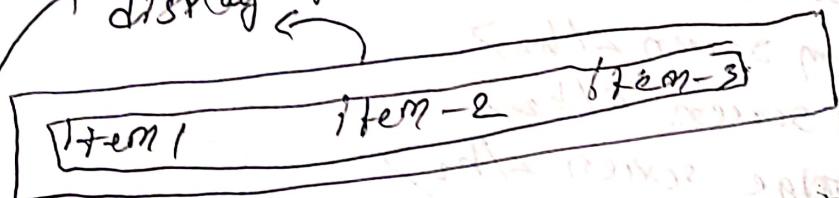
method.

So we will add a button.

CSS Flex box

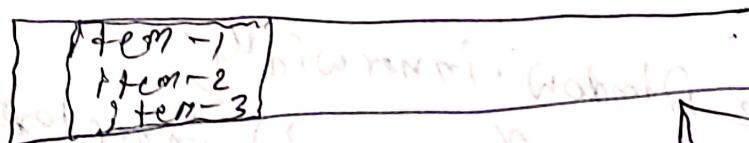
you are using

display: flex



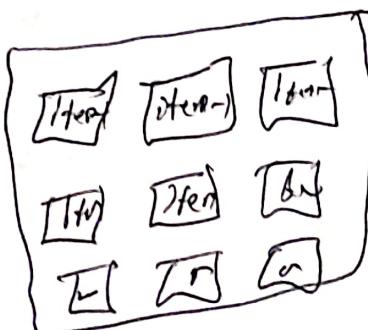
your effect.

here you are using (using)
flex-direction: row;

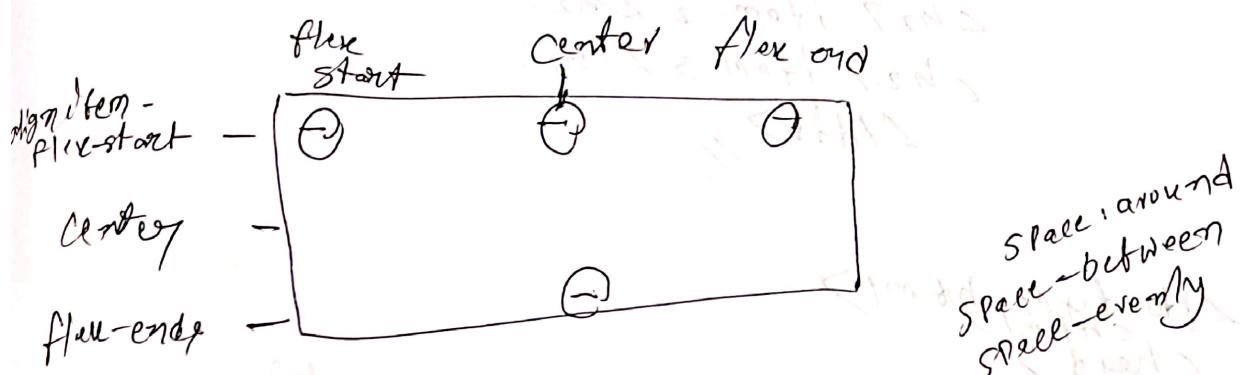


flex-wrap: nowrap;
wrap

wrapping means adjusting; you are using "wrap"
text will be multiple in the number, and
its adjusting.



align-items: properties to used to make horizontal alignments for flex box item.
just fy content property is used to make vertical alignments for flex box items.



<doctype html>

<html>

<html> CSS FLEXBOX </html>

<style>

div

{

background-color: limegreen;

padding: 20px;

display: flex;

flex-direction: row;

/* flex-wrap: wrap; */

align-items: flex-start;

justify-content: space-evenly;

height: 500px;

{

div h2

{

background-color: darkred;

color: white;

margin: 20px;

```
<style>
<head>
<body>
  <div>
    <h2> Item-1 </h2>
    <h2> Item-2 </h2>
    <h2> Item-3 </h2>
  </div> //
```

```
<doctype html>
<head>
  <style> Full Loading Register </style>
```

style

body

2

margin: 0px;

3

#main -> margin

4

background-color: black;

height: 20px;

display: flex;

align-items: center;

justify-content: space-between;

#main-new - h

margin-top: 10px;

ed vib

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        margin: 0px;
      }
      #main-nav {
        background-color: blue;
        height: 60px;
        display: flex;
        align-items: center;
        justify-content: space-between;
      }
      #main-nav li {
        color: white;
        margin: 0px;
      }
      #main-nav li nav-item {
        color: white;
        text-decoration: none;
        margin: 20px;
        padding: 6px 17px;
      }
      #main-nav li nav-item:hover {
        background-color: orange;
        border-radius: 12px;
        box-shadow: 0 0 10px white;
      }
    </style>
  </head>
  <body>
    <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid black; border-radius: 10px; width: fit-content; margin: auto;">
      <h1>Full Landing Page</h1>
      <p>This is a full landing page example showing how CSS can be used to style an entire website. It includes a header with a navigation menu, a main content area with a heading and some text, and a footer with social media links. The styling uses flexbox for layout and various CSS properties like color, font-size, and borders to create a professional look.</p>
    </div>
  </body>

```

1. --- full landing Page ~~height: 900px~~

Main - content

background: url('graph.jpg') no-repeat scroll center;

background-size: cover;

height: 100vh;

Main - content # main - content - wrapper

background-color: rgba(0,0,0,0.4);

height: 100%;

color: white;

display: flex; justify-content: center;

flex-direction: column;

text-align: center;

align-items: center;

justify-content: center;

Main - Content # Main - Content - wrapper h2

margin: 0 8px;

font-size: 30px;

Main - Content # Main - Content - wrapper p

font-size: 16px;

Main - Content # Main - Content - wrapper footer

background-color: orange;

padding: 5px 30px;

border: none;

outline: none;

border-radius: 15px;

box-shadow: 0 0 10px white;

② media (max-width: 850px)

main-nav

flex-direction: column;

nav-item

margin-top: 20px;

margin-bottom: 56px;

③ media (min-width: 500px) and (max-width: 720px)

main-nav

flex-direction: column;

nav-item a

margin: 0px;

</style>

</head>

<body>

<nav id="main-nav">

Spiderman Institute

<div id="nav-items">

 home

~~#~~> News

~~#~~> courses

~~#~~> about

</div>

variable 2

</nav>

<div id="main-content">

<div id="main-content-wapper">

<h2> Welcome to UT Courses </h2>

<hr>

<button> Read more </button>

<div>

<div>

<body> portfolio soft

<html>

→ multi-var