HTML 50% CSS 40% JS 10%

HTML → (body)

→ HTML stands for Hypertext markup language

→ It is basically standard markup language for giving a static skeleton to web applications & websites

→ Its a well standardized system

CSS (colour, design, Hair, size, font)

→ Cascading style sheets ~~which is known as css, is a style sheets language~~

→ It is used to handle the presentation of the web page containing HTML

→ It makes our websites beautiful & modern looking

Java script (JS) [Brain]

→ It is a high level dynamic interpreted programming language

→ It allows client side scripting to create completely dynamic web applications & websites

Ex: Let's go there

#3

<!DOCTYPE html> // It is html document

<html lang="en"> // language ~~eng~~ english

<head>

→ meta tags are present

↓

Information about information

example: keywords, discription, external sheet link, title.

→ Meta tage include the title, base, ~~link~~, style & script elements, keywords, discription, external sheet link or link

Comment:

ex:

→ <!-- our .html code starts here-->

→ ~~This co~~

→ comment can be ~~made~~ make directly used [control + /]

<body>

→ Here present the content

#4

Head

→ meta tags it is used in SEO (search in the optimization) (8)

```
<meta charset="UTF-8">
→ <meta name="viewport"
      content="width, initial
<meta name="description"
      content="This is
               description">
<meta name="keywords" content="
<meta name="robots" content="Index,follow
<title>Document</title>
<!--This is external css-->
<link rel="stylesheet" href="style.css">
<!--This js-->
<script src="harray.js"></script>
```

meta
- → charset
- → viewport (
- → description
- → keywords
- → robots
- → title
- → link (css)
- → script (js)

emt.

10/9/20

#5

HTML body

Lorem 4.

→ heading tag. Ex: <h1> </h1>

→ ~~para~~ paragraph.

→ paragraph ex <P> </P>

<P> Lorem 4 </P> → This can be make in paragraph using view. toggle word wrap.

→ If we want 4 paragraph then we write p*4 then it gives 4 paragraphs

→ → This makes bold.

→ → This is makes looks different like italic.

→
 → It is break the spaces.

→ <hr> → horizantal tag.

→ → bold [strong]

→ → italic [em]

* <h1> </h1>
  <P> </P>
    or   // bold.
    or   // italic
  <hr> // horizantal line

 // break the space.

HTML body: Adding images & links!

<body>

 goto google

attributes

↓ links

links

window - alt
mac - option

 goto google

It is makes it open in new tap

 body

open internal link

<image

<img src = "harry.jpg" alt = "Error">

when image is not found

Source splash

Html body: lists, <ul>, <ol> & Tables

→ <dl> datalist
<dt>
<dd>

* lists & Tables

<title> Tables & lists </title>

~~<body>~~

~~<lists ...>~~

* lists has two type.

(1) unordered list (ul) → It is in '•' form

(2) order list (ol) → It is in 'number' form

```
<ul>
    <li> first </li>
    <li> Second </li>
</ul>
```

o/p
=> • first
• second

```
<ol>
    <li> first </li>
    <li> second </li>
</ol>
```

o/p
1. first
2. second

and

~~<ol type="A">~~

* ~~(ol)~~ ol attributes are 'A', 'a', 'I', 'i'

* ul attributes are disc, square, circle

Eg: <ul type = 'disc' or 'square' or 'circle'>

<ol type = "A" or "a" or "I" or "i">

* ~~We can~~ nested lists also create it

```
<ul>
    <li> first </li>
    <li> second </li>
        <ul>
            <li> another first </li>
        </ul>
</ul>
```

o/p

- first
- second
  - another first

★ Table:

```
<h3> Html table </h3>
<table>  // table head & table body
    <thead>
        <tr>
            <th> name </th>
            <th> RollNo </th>
            <th> marks </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td> Amruta </td>
            <td> 08 </td>
            <td> 20 </td>
        </tr>
```

name	RollNo	marks
Amruta	08	20
Radhika	28	30

thead, th, tbody

Ad table data

```
        <tr>
            <td> Radhika </td>
            <td>  28     </td>
            <td>  30     </td>
        </tr>
    </tbody>
```

#8

* Html body: forms & input tags.

* Forms:

```
<h3> Forms </h3>
<form action="backend.php">
    <div>
        Name: <input type="text" name="myname">
    </div>
    <div>
        role: <input type="text" name="myrole">
    </div>
    <div>
        email: <input type="email" name="my email">
    </div>
    <div>
        <input type="submit" value="submithere">
    </div>
    <div>
        Dob: <input type="Date" name="mydate">
```

```
<div>
Bonoes: <input type="number" name="mybonaes">
</div>
<div>
Are you eligible?: <input type="checkbox" name="myeligibility">
</div>
<div>
Gender: male <input type="radio" name="mygender">
Female <input type="radio" name="mygender">
</div>
<div>
<input type="submit" value="Submit Now">
</div>
<div>
<input type="Reset" value="Reset Now">
</div>
<div>
Write about yourself: <textarea name="mytext" cols="30" rows="10">
</textarea>
```

* label

```
<div>
  <label for="name">Name</label>
  <input type="text" name="myname" id="name">
</div>
<label for="car">car</label>
<select name="mycar" id="car">
  <option value="ind">Indica</option>
  <option value="swf">swift</option>
</div>
  <input type="submit" value="submitnow">
</div>
<div>
  <input type="Reset" value="Resetnow">
</div>
```

id.1

id

same name.

12/9/20

#9

* Html body: Inline & Block elements:

→ paragraph is block element

 this is paragraph   this is also paragraph

=

this is paragraph
 this is also paragraph

o/p this is paragraph
this is also paragraph.

} Because paragraph is block element

~~∴ use instead of ~~

∴ Use instead of

 this is paragraph
 " " also "

o/p this is paragraph this is also paragraph

Because span is inline element.

*

* ~~Inline~~ Inline element:

→ <a> [~~anchor~~]
→ ,
→
→ <img>

~~li, en, div, img.~~

Block element:

→
→
→

→
→
→

10

* Html body : Ids & classes.

```
<head>
  <title> Ids & classes </title>
</head>
  <body>
    <h3> Ids & class </h3>
    <div id = "main Box"  class = "red Bg">
        this is main box.
    </div>
    <div id = "mainBox"  class = "red Bg blackborder">
        this is main box
    </div>
        <span class = "redBg"> </span>
```

id = "main Box" → identifier

class = "red Bg" → if we have more element that makes changes, size, color, font. (Ex: 100)

id = "mainBox" → only one element

class = "red Bg blackborder" → class can be gives to more time.

* span .red Bg [class] ⇒
* span #mainspan [id] ⇒

```
<div id = "mainbox"  class = "redBg blackborder">
    this is main box
</div>
<span cla
<!-- . is for class & # is for id -->
<span class = "redBg"> </span>
<span id = "mainspan"> </span>
```

<!-- emmet takes div tag as default>

• blackbackground

⇓

<div class= blackbackground">

<!-- creating multiple elements using emmet>

<!-- span.myclass.myclass2.myclass*+<Tab> ⟹
to print 4 similar element using emmet-->

first

<span class= "myclass myclass2 myclass3>second

<span class= "myclass myclass2 myclass3> third>

* Html body: Html entities [It is use when we want space]

```
<title> HTml Entities </title>
<div class="container">
    <p> this is paragraph </p>
</div>

<div class="container">
    <p> this is  paragraph </p>
</div>
```

o/p this is paragraph.

* If we want spaces then we use

example: this is   paragraph

non breaking space

* paragraph is   o/p is paragraph is

 paragraph is <p> // o/p => paragraph is

Html entities are:

- →   → — (space)
- → <p> →
- → £ → £
- → © → ©
- → → empty character

#112

Html body: Semantic tags:

meaning

```
<table> Semantic </table>
</heads>
  <h3> Semantic element </h3>
```

Semantic tags:

→ <header>, <nav>, <section>, <article>, <aside>, <footer>

nav → navigation → preview →

→ Semantic element mozilla.

```
<details>
    Lorem 34
</details>
```

CSS

13/9/20

#13

→ Cascading style sheets   p, h2, etc

→ CSS gives style to raw html

→ CSS is used to give style to our web pages.

→ CSS is used to make websites responsive.

→ Html is used to structure a website.

→ **CSS** takes responsibility of design.

→ CSS is beautiful & modern looking

~~CSS syntax:~~

select

~~p{ color: blue;}~~

CSS syntax:

selector → p, property → color, value → blue

one selector ★ `p { color: blue;}` ⇒ change color of paragraph

- p → where?
- color → This
- blue → set this value

multiple selector

group selector / multiple properties ★

```
header, p.intro {
    background-color: red;
    border-radius: 3px;
}
```

- header, p.intro → Group of selector (intro → class)
- { … } → Declaration block

* Three ways to add css to the markup: (HTML)
↓ html code

(1) Inline css: css is added to the elements directly using style attribute.

~~(2) Inter~~ Ex: first

(2) Internal css: css is kept inside the 'head' tags in <style> tags.

(3) External css: css is kept separately inside a '.css' style sheet.

→ step 1: write css in .css
→ step 2: include that css file to markup.

#14 ★ Inline, Internal & External css:

Inline css

```
<title> css tutorial </title>
</head>
<body>
    <h3> This is css tutorial </h3>
    <p style= color: red; > this is paragraph </p>.
```

↓ Inline css

* Internal css: add <style> inside <head> tags.

```
<head>
    <style>
        P { color: purple;
            background: yellow;
          }
    </style>
</head>
```

html file.

```
P { color: greenyellow; }
```

css file

* External css:

```
<head>
    <link rel="stylesheet" href="14.css"> // link: css enter
    <style>
        P { color: purple;
            background: yellow;
          }
    </style>
</head>
```

include cssfile to html.

html.

```
P { color: greenyellow; }
```

.css

o/p: shows o/p

* <head>

<style>

p{ color: red;

background-color: yellow !important;

}

</style>

<link rel = "stylesheet" href = "14.css">

</head>

.html

o/p: shows as o/p

.css → p{ color: ~~red~~ green yellow; }

#15

* selectors in css:

selector property value.

p {color: blue; }

→ css selectors are used to find the element whose property will be set

→ selectors are used to target the html element

→ selectors makes it easy for us to easily target single / multiple HTML elements in the markup.

* Types of basic css selectors:

(1) css element selector

(2) css id(#) selector

(3) css class(.) selector

(4) css grouping selector

*(1) element selector [using ~~inline css~~ internal css]

```
<head>
    <title> css selector </title>
    <style>
        p{ color:red;}
    </style>
</head>
<body>
    <h3> css selectors </h3>
    <p> This is paragraph </p>
    <p> This is another paragraph <p>
</body>
```

only this part is changes

all color is red

*(2) css id selector:

```
<head>
    <title> css selector </title>
    <style>
        #redelement {color:red;}
    </style>
</head>
```

```
<body>
    <h3> CSS selectors </h3>
    <p> paragraph </p>
    <p id="redelement"> another paragraph </p>
</body>
```

only this element is changes

* (3) CSS class selector:

```
<head>
    <title> CSS selector </title>
    <style>
        P.secondpara { color: red; }
    </style>
</head>
<body>
    <h3> CSS selector </h3>
    <p> paragraph </p>
    <p id="redelement" class="secondpara">
        another paragraph
    </p>
</body>
```

*(4) CSS grouping selector: [multiple selector]

```
<head>
    <title> css selector </title>
    <style>
        p, footer { color: red; }
    </styles>
</head>
<body>
    <h3> css selector </h3>
    <p> paragraph </p>
    <footer> footer </footer>
```

→ CSS Comment as /* this */

```
*   p {
        border: 2px solid red; // element selector.
    }
```

#16

* Chrome developer tools:

```
<title> developertools </title>
<style>
    p{ color: purple;
       font-style: italic;
       background-color: rosybrown;
    }
</style>
</head>
<body>
    <h3> Developer tools</h3>
    <p> chrome developer tools </p>
</body>
```

#17

* fonts in css:

→ free install

→ web safe font
→ google font

```
<head>
    <title> css font </title>
    <style>
        p{ font-family: ... copy paste
        }
    </style>
</head>
```

```
<body>
    <h3> fonts </h3>
    <p> playing with fonts is very exciting </p>
</body>
```

*Fonts are:

→ font-family
→ font-size : 33px ; /* 1/96th of an inch */
→ line-height : 1.3 em;
→ font-weight: bold ;
→ font-style: italic ;

#18 *color in css :

```
<head>
  <style>
    p # first para {
          color: blue;
        }
    p # secondpara {
          color: rgb(0, 255, 0);
        }
    p # thirdpara {
          color: #ff88ff;
          background-color: rgb(0,0,0);
        }
  </style>
```

HSL - Hav saturation lightness.
↓ (0-100%) (0-100%)
(0-360°)

(hex color picker)

* ~~Borders & backgrounds:~~

* Height, width, border & backgrounds;

```
<head>
  <title> Height, width, borders & backgrounds </title>
  <styles>
      p {
          background-color : red;
          width: 499px;
          height: 900px;
          border_width: 4px;
          border-color: green;
          border-style : solid;
          border-radius: 30px;
      }
      Span{
          border_top : 2px solid green
          border-right
          border-bottom
          border- left
          border-top-left-radius: 4px
      }
```

border: 4px solid green;

#thirdpara{

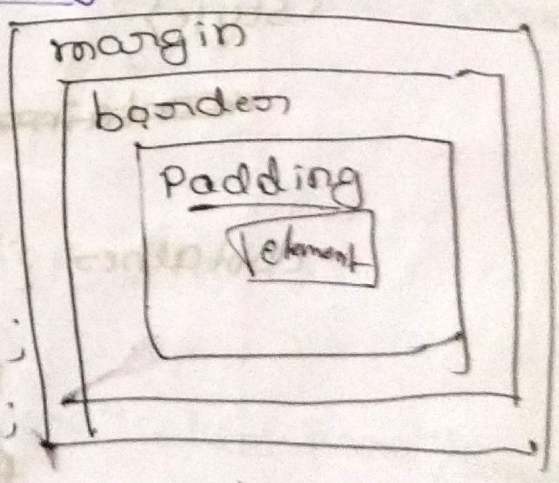
height: 500px
width: 455px
background-image: url( );
border: 2px solid red;
background-repeat: repeat-x or repeat-y,
no repeat;
background-position: center center;
top center;
12px center;

</style>

#20

box-sizing: border-box;

* Box model, margin and padding:



<style>

p{

background-color: rgb ( );
border: 2px solid rgb( );
padding: 9px;
margin: 14px;
padding-top :
-bottom:

padding: 23px 56px 6px 78px /* padding: top, right, bottom, left */

padding: 33px, 55px /* y (top / bottom)
x (left / right) */

}

universal selector (*)

```
*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}
```

// inspect

14/9/20

#21

* Float and clear:

* Alignment: [float & clear]

```
<head>
    <link href="" rel="stylesheet">
    <style>
        * {
            box-sizing: border-box;
        }

        .container {
            width: 900px;
            border: 3px solid purple;
            margin: 33px auto; // (y, x)
        }

        .item {
            border: 3px solid rgb( );
            margin: 12px 3px;
            padding: 12px 3px;
            background: red;
        }

        body {
            font-family: 'ubuntu', sans-serif;
        }
```

auto → (center)

```
<body>
    <div class = "container">
        <h2> welcome to my store </h2>
        <div id = "fruit"  class = "item">
            <h3> fruit </h3>
            <p  id = "fruitpara" class = "para"> Lorem34
            </p>
        </div>
        <div id = "computer"  class = "item">
            <h3> computer </h3>
            <p  id = "computerpara"  class = "para"> Lorem34
            </p>
        </div>
        <div  id = "stationary"  class = "item">
            <h3> stationary </h3>
            <p  id = "stationarypara"  class = "para"> Lorem34
            </p>
        </div>
</body> </div>
```

```
#fruit {
    float = left;
    width = 48%
}
#computer {
    float = right;
    width = 48%.
}
```

```
#stationary {
    clear = left;
    width = 100%.
    clear = right;
}
```

It means any float left element are not overlap

↓ not all

* float, clear, alignment:

float is used in layout

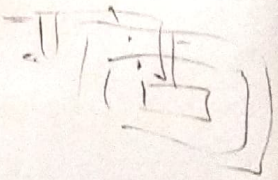
```
<head>
<style>
#fruit{ float=left;
        width=48%.;
      } float

#computer{ float=right;
           width=48%.;
         }

#stationary{ clear:both; // right & left clear
             width:100%.
           }

P,h2 {
      text-align:right;
      text-align:left;
      text-align:center;
      text-align:justify;
      width:
     }
```

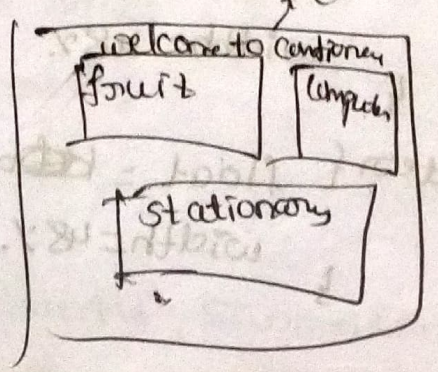
alignment



```
h1{
    margin: 23px auto;
    width: 455px;
  }
```

center

container



#22

~~Pseudo selectors & more designing:~~

* Pseudo selectors & more designing: [styling links & buttons]

<head>

```
.container{
    border: 2px solid red;
    background-color: rgb( );
    padding: 34px;
    margin: 34px auto;
    width: 666px;
}
.btn {
    background-color: crimson;
    padding: 6px;
    border: none;
    cursor: pointer; // show cursor on both
    font-size: 18px;
}
a{
    text-decoration: none; // to remove underline
    color: black;
}
```

a:hover{ color: background-color: }

<body>

```
<div class="container" id="cont1">
    <h3>
    <p> Lorem 34 </p>
//link  <a href="" class="btn"> Read more </a>
//button  <button class="btn"> Contact Us </button>
```

* link

Read more

* button.

<button class="btn">Contant us</button>

* Pseudo selector

Ex:

a:hover{ ← ~~when it is use then~~

when we cursor on link & button then what change will happen

color: rgb( );

background_color: rgb( );

a:visited{

background_color: yellow;

}

a:active{

background_color: blue;

}

bootstrap → Documentation → Components