03/09/2022

HTML:-

HTML → Hyper Text Markup Language

HTML is the Standard markup language for creating web pages

⇒ HTML describes the structure of a web page

⇒ HTML Tells the browser how to display the content.

Doctype HTML:-

<!DOCTYPE html> declaration defines that this document is an HTML5 document.

HTML Element:-

<tagname> Content </tagname>

Empty elements:-

Some elements have no content. Ex:-
, <hr>

⇒ Empty elements do not have an end tag.

html page structure:-

```
<html>
  <head>
    <title> page title </title>
  </head>
  <body>
    <h1> some heading </h1>
  </body>
</html>
```

⇒ Save the html page:-

File name should end with .html (or) .htm and set the encoding to UTF-8.

HTML headings:-

h1 to h6

⇒ <h1> heading </h1>

⇒ HTML paragraphs:-

 some paragraph

⇒ View HTML source and Inspect

Nested HTML elements:-

Ex:- Elements can contain other elements.

Ex:- <html>, <body>, <h1>,

⇒ Never skip the end tag.

⇒ HTML is not case sensitive.

HTML attributes:-

⇒ Attributes provides additional information about elements.

⇒ Attributes are always specified in start tag.

href attribute:-

href attribute specifies the url of the page the link goes to.

The Src Attribute:-

The Src attribute specifies the path to the image to be displayed.

Absolute Url:- External links :- Ex:- https

Relative Url:- with respect to local :- C:/images/

⇒ It is always best to use relative Url's. They will not break if you change domain.

⇒ If you don't have any permission to use external images, you may be in violation of copyright laws.

Alt attributes:- (text)

If image not shown, this will display

Style attribute:-

The Style attribute is used to add styles to an element.

Ex:- para

Always Quote Attribute values:-

⇒ we can use single (or) double Quotes.

Headings are Important:-

Search engines use the headings to index the structure and content of your web pages.

⇒ with html, we cannot change the display by adding extra spaces (or) extra lines.

⇒ Browser will automatically remove any extra spaces and lines when the page is displayed.

<hr> tag:-
horizontal line.

 tag:-
defines a line break.

<pre> tag:-
used for display the spaces also Ex:- code, poems etc

attribute format:-
<tagname style = "property:value;"> content </tagname>

font family:-
style = "font-family: verdana;"

HTML text formatting:-
<b> → bold
 → important text
<i> → Italic
 → Emphasized text
<mark> → Marked text
<small> → small text ⇒ text → background color yellow
<del> → del text ⇒ ~~text~~
<ins> → inserted text ⇒ text
<sub> → subscript ⇒ H_2O
<sup> → superscript ⇒ x^2

Blockquotes:- <blockquote>
having some indentation at starting.

Quotation tag:-
defines a short Quotation.

⇒ Abbreviations ⇒ <abbr>:-
 The <abbr title = "world health"> WH </abbr> org

Address tag:- <address>
used for display in italic text

Cite tag:- <cite>
The text displayed in Italic

<bdo> ⇒ bi-directional override:-
<bdo dir = "rtl"> left to right. </bdo>

HTML Comments:-
<!-- write comments here -->

Colors:-
rgb(255,99,71), #ffffff, red

CSS:-
Cascading Style Sheets.
⇒ used for formatting the layout of a webpage.
Inline, Internal, External.
⇒ colors, fonts, sizes, Borders, padding, Margin,

link to External CSS:-
<link rel = "stylesheet" href = "index.css" />

HTML links:-
 google
Default:- 1) universal link ⇒ underlined and blue
2) visited link ⇒ underlined and purple
3) active link ⇒ " " red.

Target:-

-self, -blank.

Link to Email Address:-

 Send Email

⇒ Link titles:-

 url

Links Styles:-

a:link{ , a:visited, a:hover, a:active.

}

Bookmark in html:-

  Jump to C4

 here

Images:-

Float left and right: (float: "right" ⇒ CSS)

Background-image:-

<p style = "background-image: url("img-girl.jpg");">

Background repeat: no-repeat; repeat-x;

background-attachment: fixed; scroll;

background-position: cover; right top;

background-size: 100% 100%; (it will stretch)

HTML favicon:-

icon in the browsers tab.

<link rel = "icon" type = "image/x-icon"
href = "image.jpg" />

HTML table:-

<table>
<tr>
<td> td1 </td>
<td> td2 </td>
</tr>
</table>

th → table headers instead of td.

thead, tbody, tfoot.

Caption:- <caption> tag helps to add caption to the top of the table (center)

⇒ caption-side: bottom, makes the caption bottom.

Table borders:-

add borders to table, th, td.

Collapsed borders:-

border-collapse: collapse;

border-styles:-

dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden.

Table sizes:-

Apply inline styling to one-column, it will reflect to all columns.

Colspan:-

<th colspan = "2"> Name </th>

will occupy two colums space for that row.

cell padding and Spacing:-

add padding to table cells casually.

Spacing:

```
table {
    border-spacing : 30px ;
}
```

Table Styling:

```
. tr: nth-child(even) {

}
```

⇒ we can add tr:hover to for hovering of rows.

⇒ HTML lists:

Order lists:-

```
<ol>
  <li> . . . </li>
  <li> . . . </li>
</ol>
```

type = "1", "A", "a" "I", "i"

Ex: `<ol type = "1">` ... `</ol>`

Control list Counting:

```
<ol start = "50">
  - - -
</ol>
```

⇒ we can nest there lists also.

dl, dt, dd:-

```
Ex: <dl>
      <dt> coffee </dt>
      <dd> Black hot </dd>
      <dt> Milk </dt>
      <dd> white </dd>
    </dl>
```

un-ordered lists:-

```
<ul>
  <li> -- </li>
</ul>
```

liste-style-type : none, disc, circle, square.

Block and Inline:

Block → h1 to h6, p etc.

inline → span, input tags, buttons etc.

html classes:-

- class Name.

b html id:-

#id

Iframes:-

`<iframe src = "url" title = "desc">` `</iframe>`

set height and width.

Iframe target for a link:- add name to iframe and give the same name in target of anchor tag.

html JS:-

```
<Script src = "index.js">
</Script>
```

Entities:-

Entities are used to use symbols (Ex: <, > etc)

emojis:-

&#128512

XML & XHTML:-

HTML forms:-

<form> element is used to create an HTML form for user input

04/09/22

Input Element:

<input type = "text">

type = "radio", checkbox, submit, button.

labels ⇒ when user click on labels, corresponding input field will focus.

name attribute in input elements

⇒ Auto Complete, no-validated (validations not to be done when submitting),

html form elements:-

input, label, select, textarea, button, fieldset, legend, datalist, output, option, optgroup.

Select Element:

```
<select id = "cars" name = "cars">
    <option value = "volvo"> volvo </option>
    <option value = "fiat"> fiat </option>
</select>
```

By default, first item in the drop-down list is selected.

⇒ To define, a pre-selected option, add the selected attribute to the option.

size attribute in select tag defines no of options to be displayed. Ex: <select size = "2">

multiple:-

add multiple for multiple selects.

Text area:-

multi-line input field

```
<textarea name = "message" rows = "10" cols = "30">
</textarea>
```

⇒ we can add width and height styles to text area

fieldset and legend:-

output:


```
<fieldset>
    <legend> person </legend>
</fieldset>
```

datalist:-

Acts like select but with without preselected

Input types:-

button, checkbox, color, date, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week

⇒ datatime-local displays hours also.

⇒ html value attribute

html attributes:

value = "Shiva", readonly, disabled, size, max-length, min and max, multiple, pattern, placeholder, required, step, auto focus, heigh and width

CSS – Cascading Style sheets:-

⇒ CSS describes how elements are to be displayed on screen.

css syntax:-

CSS consists of a selector and a declaration block.

```
h1 {
   Color: blue;
   }
```

here h1 is selector
Color ⇒ property
blue ⇒ value

CSS selectors:-

1) Simple selectors ⇒ based on name, id, class

id selector:-

```
#idname
```

```
<span id="idname"> id </span>
```

class selector:-

```
.classSelector
```

```
<span class="classselector"> class </span>
```

Element selector:-

```
P {
   Color: red;
   }
```

here p is element name

⇒ we can add multiple classes by seperated with spaces between them.

Only specific html elements should be affected by a class:-

```
p.center {
   text-align: center;
   }
```

only <p> elements with class = "center" will be aligned center

Universal selector:- (*)

```
* {
   Color: red;
   }
```

⇒ it will apply to all elements in that page

Grouping Selectors:-

group selectors, seperate each selector with a comma

```
h1, h2, p {
      Color: red;
      }
```

pseudo class selectors:-

```
input:focus {
      border: 9px solid green;
   }
```

Children selectors:-

```
li:nth-child(2) {
                  Color: red;
               }
```

We can use, nth-child(even)

⇒ we can use pattern (an+b)

↓

Initially nth element = 0

⇒ (2n+1) odd

⇒ (2n+2) even

⇒ nth-first-child()

nth-last-child()

nth-child(3)

first letter selector:-

```
p::first-letter {
    color: blue;
}
```

not selectors:-

```
:not(p) {
    border: 1px solid red;
}
```

Selecting by more than one class:-

```
.s1.s2 {
    color: blue;
}
```

```
<p class="s1"> S1 </p>
<p class="s2"> S2 </p>
<p class="s1 s2"> S1 S2 </p>
```

Selecting by tag inside another tag:- (direct children and non-direct)

```
div p {
    Color: red;
}
```

only direct children:

```
div > p {
    Color: red;
}
```

⇒ If we want to select a tag immediate after another tag:-

```
div+p {
    color: red;
}
```

CSS Comments:

/* Comment to be placed here */

CSS Backgrounds:

background-color,

opacity:- Value from 0 to 1.

```
opacity: 0.3;
```

CSS Borders:-

```
p {
    border: 1px solid red;
}
```

border-radius:-

```
border-radius: 5px;
```

Margins:-

Space outside Border.

Margin inherit:-

takes the margin from parent.

Padding:-

Space inside border.

box sizing:-

~~border-box~~:- box-sizing: border-box;

height and width properties do not include padding, borders & margins! They set the height/width of the area inside the padding, border and margin of the element.

⇒ The value of width property is larger than max-width property, the max-width property is used and width property is ignored.

Box-model:

Content, padding, border, margin.

outline:-

Outline is applied above border.

CSS Text:-

Color: red;

background-color: blue;

Text Alignment:-

text-align: center, left, right, justify,

Text direction:-

direction: rtl;

Vertical align:-

vertical-align: baseline, text-top, text-bottom, sub, super;

Text-decoration:-

text-decoration: overline, line-through, underline, overline underline;

text-decoration-color: red;

⇒ text-decoration: underline solid red 5px;

⇒ text-decoration: none;

Text transform:-

text-transform: uppercase, lowercase, capitalize,

Text-Indent:-

text-indent: 50px; applies for first line.

letter spacing:-

letter-spacing: 5px;

line height:- (space between the lines)

line-height: 0.8;

word spacing:-

word-spacing: 10px;

white space:-

white-space: nowrap;

Text shadow:-

text-shadow: 2px 2px red;

cursor properties:-

Cursor: auto, crosshair, default, e-resize, help, move, pointer, progress, wait.

Lists:-

list-style-type: none, circle, square, upper-roman, lower-alpha,

list-style-image: url('image.jpg');

list-style-position: inside, outside.

Table responsive:-

overflow-x: auto;

Display:

display: inline;

display: block,

differences between display hidden and none;

CSS positions:-

position: static, relative, absolute, sticky, fixed.

Z-index:

overlap of element one above other

position: absolute;

left: 0px;

right: 0px

z-index: -1;

Z-index overlapping:-

last html element will be overlapped.

overflow:-

overflow: visible, hidden, scroll, auto.

Transition:

transition: width 0.4s ease-in-out;

linear gradients:

background-image: linear-gradient(red, yellow);

→ (to right, red, yellow)

(180°, red, yellow)

2D Tranformations:-

~~Translate~~:

Translate:-

transform: translate (50px, 100px);

Rotate:

transform: rotate(20deg);

Scale:

transform: scale(2,3); scalex(), scaley().

Skew:

transform: skewX(20deg); skew(20deg, 30deg)

3D:-

transform: rotateZ(90deg);

translate3d(x,y,z)

rotate3d(x,y,z, angle)

Transition:

```
div:hover {
    width: 300px;
}
```

```
div {
    width: 100px;
    transition: width 2s;
}
```

Transition Delay:-

```
transition-delay : 1s;
```

Animations:-

```
div {
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
@keyframes example {
    from { background-color: red; }
    to { background-color: yellow; }
```

0% { }
20% { }
80% { }

```
animation-iteration-count : 3; / infinity;
animation-direction: reverse, alternate, alternate-reverse.
animation-timing-function: ease, ease-in-out,
```

Flexbox:-

```
display: flex;
```

Justify-Content:- flex-start, flex-end, Center, space-evenly, space-between, space-around

flex-direction: row, column, row-reverse, column-reverse

Flex wrap:-

```
flex-wrap : nowrap, wrap, wrap-reverse
```

Align-items:- Center, flex-end, flex-start

Gap:-

```
gap: 50px;
row-gap : 5px; column-gap: 30px;
```

Flex Grow:- flex grow property is for child Container

```
.firstDiv {
        flex-grow: 1;
    }
.secondDiv {
        flex-grow: 2;
    }
```

flex-shrink:- (shrink, if space is not available)

```
.firstDiv {
        flex-shrink : 1;
        width: 200px;
    }
.secondDiv {
        flex-shrink: 2;
        width: 200px;
    }
```

⇒ flex Basis:-

Children will take only required space.

```
.first {
        flex-basis: 10px;
    }
.second {
        flex-basic: 200px;
    }
```

⇒ flex : 2 2 200px;

flex grow, flex shrink, flex basis

CSS Grid:-

```
display : grid;
grid - template - Column's : repeat(2, 1fr)
grid - template - rows : 200px 200px ;
```

Max Content:

```
grid-template-Columns : 200px max-content auto;
```

Min Content:

```
grid-template-Columns : 200px min-content auto,
```

Minmax functions:-

```
grid-template-columns : minmax (200px, 1fr), repeat (39, 1fr);
```

positioning elements in Grid;

```
grid-column-start : 2
grid-column-end : 3
```

grid positioning shortand:

```
grid-column : 2/4   (2 → start, 4 → end)
grid-row : 2/5
```

grid-area:

```
grid-area : 2/3/2/4   (rowstart, Columnstart, rowend, Columnend)
```

gridSpan:

```
grid-column : 2/ span 2;
```

Grid gap:-

```
Column-gap : 10px
row-gap. 20px
```

: : 10px
col...

Media Queries:-

```
@media screen and (min-width: 400px){
        .para {
            color: red;
        }
}
@media screen and (min-width: 1200px) {
        .para {
            color: blue;
        }
}
```

logical operation with media Queries:

or operations (,)
and operator (and)

```
@media ( min-width: 400px), (min-width: 200px) {
        background-color: red;
}
@media (min-width: 1200px) {
        color: blue;
}
```