**HTML**

html - Parent tag holds the complete content of pagehead - > Holds header content of pagetitle - to add title to pagebody -> holds the actual content of pagep -> Paragraph tag holds multiline text contentdiv -> Block level element to group relative set of elements as blockspan -> Inline element to add content in same lineh1 to h6 -> Heading tagsbr -> to add line breaks to contentul, ol -> List itemshr - > To add horizontal line etc.Basic Structure of a HTML Page: <html> <head> <title> Page title </title>

</head> <body> Actual content of page... </body> </html>

**Attributes:**

Follwing are the predefined attributes being supported in Html.

1. id:- Through which we can add a unique reference to elements
2. name:- name value can be added to html elements
3. class:- to inject single or multiple classes to an element
4. style:- To inject single or multiple inline CSS
5. href:- To set external link to a or ling tag
6. src:- To specify external resource path to script or img tag

Syntax:

<tagname attributename1=”Value1” attributename2=”Value2” attributename3=”Value3”

……..

</tagname>

**Note:**

* For each single html element we can enject or add any no. of attributes.
* The same attribute cannot be repeated within the same tag
* For every html element we can add attributes
* html also provides a feature of creating custome attributes with user defined values.

**12/01/2021**

**CSS Class**

A group of required CSS properties being binded as a module, assigned with a user defined name is called a CSS class

* “Class” is a predefined html attribute through which we can add or inject any no. of CSS classes to html elements.
* A single class attributes is used to inject any no. CSS class.
* “style” is a predefined tag under which we can define any no. of CSS classes.
* Class name always start with “.” operator, while defining a class.

Syntax to create a class

.<classname>{

….

…..// set of CSS properties

}

Eg:

<style>

.abc{

font-size:20px;

text-decoration:underline

}

.newsContent{

color:blue;

border:2px solid green;

}

</style>

**18/01/2021**

**Different ways of creating a CSS class:**

1. Through a class name (User Defined).
2. Through an ID of an element
3. Through the tag name

**Creating a class using ID of an Element:**

A CSS class can be created using id of an existing element where the difference is it always starts with ‘#’ symbol

Syntax:

#<element ID>{

……..

……… //set of css properties

}

Example:

<style>

#container {

……..

……..

}

<p id=”Container”>

………

………

</p>

**Defining a class using tag name**

If some set of html tags of same type should have the same type of look and fill within the page, we can make is the particular tag name itself to define the CSS class. While defining a CSS class using tag name it doesn’t need to start with any symbol.

syntax:

<tagname> {

…..

……// set of CSS properties

}

Example:

<style>

h3{

…..

……

}

</style>

<h3>

……

……

</h3>

**19/01/2021**

**Debugging tools:**

Every browser now a day comes with a default debugging tool using which we could de-bug the HTML/CSS/JS code of current or any web page.

Following are the default features being supported or execute through debugging tools.

1. Through debugging tools, we can explore DOM structure of any webpage running within the browser.
2. Debugger tool provides an options to add or delete or update any HTML element or its corresponding CSS property dynamically when the page is running.
3. Debugger tool provide an options to debug JS code by adding break points, adding that we want to monitor.
4. Through debugging tool it is very easy to refer to a particular HTML code of a corresponding view within the page.
5. It also provides an option to monitor list of all network calls happened within the current page.
6. We can monitor any REST service communication calls (the Header information, the response, total time it has taken, response states code of the server)
7. Through debugging tool we can also monitor the performance of the page by monitoring list of resources getting loaded along with the total time get taking to a loaded.
8. We can also monitor the memory management of the current web page.
9. Using debugger tool we can also monitor the storage component like local storage, Session storage, Indexed DB, Web SQL and Cookies.

**Note:** F12 is the option to open the Debugging tool in any browser or Right click on right most click on the page and choose inspect option to open debugging tool.

**DOM(Document Object Model):**

DOM indicates the tree structure of the current web page by specifying the no. of elements participating within the page, relationship b/w the elements, the attributes that element and holding etc.

**23/01/2021**

**CSS Properties:**

Among all the 4 different ways applying to all CSS property, if we change the same CSS property using with a different value using all the four different ways browser by default follows the below priority order while applying CSS to html elements.

* Among all the different ways of applying CSS to html, inline CSS takes the higher priority in order
* CSS being added through ID takes second priority in order.
* CSS being added through a class takes third priority in order.
* CSS being added through tag name always take least priority in order

**Note:**

* If at all user want to override the default priority order we go with the property ‘!important’.
* Any CSS property being added with ‘!important’ always takes higher priority, higher property, override the default priority order.

[webdev.prasad@gmail.com](mailto:webdev.prasad@gmail.com)

**25/01/2021**

**Adding images to a container using ‘img’ tag**

‘img’ predefined tag supported in html using which we can add an image to the webpage. It takes in mandatory ‘src’ tag through which we can specify the absolute or relative path of external image resource. One ‘img’ tag is used to add a single image, multiple images can be added to the page using multiple ‘img’ tag.

Syntax:

<img src = “..absolute / relative path”/>

Eg:

<img src = “<http://sampe.com/data/image/abc.jpeg/>>

<img src = “…../data/sample/test.png”/>

**CSS Pseudo Classes:**

Following are predefined Pseudo classes that supported in CSS using which we could able to apply the CSS to an element not while loading on the page but based on its current states.

:active -> It selects the active links

:checked -> to select the checked input element with type checkbox

:disabled -> disabled input elements

h3:empty -> selects every h3 tag which is empty

:enabled -> selects every input element which is in enable state

:first-child -> selects tag which is in first child position

:last-child -> selects tag which is in last child positon

:nth-child(2) -> selects element in 2nd position

:nt-last-child(2) -> selects element in 2nd position from last

:only-child -> selects element which is the only child

:focus -> selects element which is in focus state

:visited -> anchor tag which is already visited

etc.

**Pseudo Elements:**

Following are the predefined Pseudo elements are supported in html using which we could able to add the CSS properties not to the complete content but partially we can apply.

::after

::before

::first-letter

::first-line

::selection

Syntax:

class:pseudo class{

……

……

}

Eg:

.abc::first-letter{

font-size:20px; color:red;

}

**26/01/2021**

**Difference b/w block level element and inline elements**

All the predefined HTML elements are being characterized into block level or inline type

Any HTML element comes under block level category holds following properties:

1. By default it occupies 100% width of the container
2. While rendering on the page it automatically comes to the new line and gets rendered, the next element following the block level elements also automatically comes to new line.
3. Even though it occupies 100% width of the container by default, we can always control the width and height of the block level element using CSS ‘width’ and ‘height’ properties.
4. Block level elements are mainly used to group relative set of element as a individual block.
5. ‘div’ tag is best example of block level elements

**Inline Elements:**

Any HTML element comes under inline categories holds following properties:

1. While rendering on the page it always tries to renders within the same line doesn’t come to new line.
2. By default it occupies the width based on the total content it is holding.
3. We can’t control the dimension of an inline element, even though we provide the width and height properties it simply skips them.
4. Inline elements are mainly used to add CSS to a particular content to a group of content
5. ‘span’ is the best example of inline elements.

**27/01/2021**

**CSS padding and Margin properties:**

Any time the elements get rendered on the page it always tries to randers from top left corner without maintaining any space. In order to add required space within or b/w the DOM elements we make is of the CSS properties padding or margin.

**CSS padding properties:**

using which we could able to add the space b/w the border of the container and content of the container.

Syntax:

padding: 10px; //Adds padding for four directions

padding: 10px 20px; // Adds 10px padding for top and bottom, 20px for left and right

padding: 10px 5 px 30px 10px;

top right bottom left

padding-top: 10px;

padding-right:15px;

padding-bottom:5px;

padding-left:10px;

etc.

**CSS margin properties:**

Margin is a CSS properties through which we can add space or gap above the border of the container

Syntax:

margin: 10px; //Adds margin for four directions

margin: 10px 20px; // Adds 10px margin for top and bottom, 20px for left and right

margin: 10px 5 px 30px 10px;

top right bottom left

margin-top: 10px;

margin-right:15px;

margin-bottom:5px;

margin-left:10px;

etc.

**CSS Box model**

While an element is rendered on the page it has to calculate the dimension or the space being occupied by the previous elements so that accordingly the new elements can occupied within the page. In order to calculate the actual dimension of DOM elements, browser users the concept of CSS box model.

In CSS box modeling in order to calculate the total dimension being occupied by an element within the page, it considers all the following properties:

1. Actual dimensions of the element(width and height)
2. Total border space being occupied on all four direction
3. Total margin and padding space being occupied on all four direction