

Hyper text Mark Up language(HTML)

Every defined language which uses set of tags through which we could be able to display content on webpage is called as markup languages.

Eg. HTML,XML.

HTML is a predefined markup language comes with a set of predefined tags using which we could able to render required content on the webpage.

Note: HTML is the only language through which we can render content within the browser.browser only understands html,css,javascript.

Following are set of pre defined html tags using which we could be able to render required content in different ways on the page.

1. Html :Predefined tag to hold the complete content of the page
2. Body: Used to hold the actual body part of page
3. Head: Used to hold extra content of page like title,meta info,external css/js files etc.
4. Title :Used to add title to webpage
5. Div :Block level element to hold block type of content on page
6. Span :Inline element to add inline content of page
7. a: Anchor tag used to create an external reference from current page
8. P: Paragraph tag
9. h1 to h6: predefined heading tag
10. img: used to add a single image resource to page
11. br : break tag used to single line break in content
 - self closing tags
 - doesn't hold content,so no closing tag eg:

 - bold
 - block level elements
12. UL/OL: Ordered and unordered list tag to hold the group of relative items.

What is tag?

An predefined text binded between open and close angular braces is called as tag.

Eg. <div></div>

Html attributes

- Attributes are used to inject or add extra properties(additional information) to any html element.
- Attributes provide additional information about html elements.

Following are set of predefined attributes can be added for html elements.

1.Id: used to add unique identity to elements ,we cannot have same id been assigned to multiple elements

2.Name: used to add name value to any html elements

Eg: <meta name="description" content="...">

here the name attribute specifies a name for the content attribute value

3.Class: using which we could be able to inject any number of css classes to any html elements

```
<style type="text/css">
```

```
.className
```

```
{
```

```
Properties;
```

```
}
```

```
</style>
```

`<html-element class=".className">`

4.title: used to add title to an element, which will be shown on mouse over of particular html element

5.alt: using which we can specify an alternate text which will be shown automatically when the resource is not available.

6.href: using which we could be able to specify path of external resource for anchor tag

7.Src: using which we can specify external resource path for img and script tag.

- Syntax for adding attributes to html elements

```
<tagName attrName1="value1 (user-def)" attrName2="value2"...>
```

```
....
```

```
</tagName>
```

Note:

1. for a single tag we can inject any number of attributes, with space as separator
2. for a single tag, same attribute should not be added for more than once.
3. Same id shouldn't be assigned to multiple elements

8.Style: Using which we can add css properties to an element.

Different ways of applying css to html elements:

Following are the different ways we can inject required properties to html elements

1. Inline css. (using style attributes)
2. Internal css. (defining a css class under <style>tag.
3. External css. (Creating an external css file which holds all the required css classes)

1. Inline css :

Through style attribute we can inject css properties to a single element.

Note: Inline css is mostly not recommended due to maintenance issue and duplication behavior.

Syntax : `<tagname style="cssProperty1:value; cssProperty2:value2;...>`

```
.....
```

```
.....
```

```
</tagname>
```

Eg. `<p style="color:green; font-size:15px>`

```
.....
```

```
.....
```

```
</p>
```

2.Internal css.(defining a css class under <style>tag.

Css classes (to avoid rewriting code):

- The process of grouping & defining set of css properties as an individual block and assigning user defined name to it,making use of it by injecting its name to a required html element is called a css class
- While defining class names,it is always user defined name.
- Every class name should always start with dot operator while defining it.
- Style is a predefined tag under which we can define any no of css classes.
- Class is an predefined html attribute using which we can inject any no of css classes to any html element.
- Once we define a css class,it can be injected to any no of html elements.
- The same css class can be injected for multiple html elements
- In a single page we can define any no of css classes but it is recommended to create a single style tag throughout the page.(inside head tag)

Syntax:

```
//define a css class

.<className> {

...

//set of css properties

}

//Injecting a css class to html elements

<div class="className">

...

</div>
```

- Css classes → code reusability & maintainence
- Also for single html any no of classed can be added.

Class = “.class1 .class2”

Different ways of defining CSS class:

1. Using CSS classname.(already described in the above “css classes” topic)
- 2.Using elements ID
- 3.Using tag name of an element.

2.Defining css classes using element ID:

In a case,an html element need to be injected with set of css properties.The set of properties will not be used by any other elements once the element is having an ID property assigned.The elements holds ID attribute,in this case instead of css class using the element ID itself,it is almost like defining a normal css class where the only difference is that css class with an ID starts with hash operator.

Syntax:

```
#<elementId> {

...

...//set of css properties;

}
```

Eg:

```
<style>
```

```
#sample {
...
...//set of css properties;
}
</style>
<p id="sample">
...
</p>
```

3. Defining css classes using tag name(not recommended)

If multiple same type of element(eg. Multiple 'p' tags) need to be rendered with same look and feel, then we can define just by using the <tagname> instead of ID or class name.

Note: While defining css with <tagname> it should not be start with any operator or symbol just a <tagname>.

Syntax:

```
tagName {
.....
.....//set of css properties;
}
```

Eg.

```
<p>
.....
.....
</p>
<style>
P{
    Font-size:20px;
    Color:green;
}
</style>
```

Frameworks (set of rules and regulations for handling large codeBase)

1. Angular js

- designed for client side
- high performance(only deals with client side)

2. Angular 7: clientSide + server side

- Angular rules and regulations using typescript (superScript of javascript)
- interpreter/compiler(browser cannot process typeScript) → fed to browser

4. Reactjs:

designed for client side

Adding images within the page

“img” tag is a predefined tag been supported in html using which we could able to inject a global or a local image resource to the webpage.

It takes a mandatory attribute src using which we can specify the relative or absolute path of an image resource.

Syntax:

```

```

Eg:

```

```

```

```

Note:“img” is an inline block element, gets rendered in the same line, but we can control the dimensions using width and height properties.

Interview Questions:

1. Give eg for Self closing tags?
2. Give eg for predefined html attributes?
3. What are the different ways we can apply css to html elements?
4. Why inline and tag based css are not recommended?
5. Difference between a css class and id based questions?
6. Difference between absolute and relative path?

DOM(document object model)

- Which specifies the complete tree structure of the current page.
- Specifies relationship between elements(child,parent,sibling etc)
- It holds list of attributes and its values.
- Every webpage can not have the same DOM structure.
- The dom structure holds list of all the elements,relation between the elements,contents of the elements,attributes it is holding and their corresponding values.

CSS priority order :

Following is the default css priority order any browser maintains while adding css properties to elements.

1. Among all the different ways Inline CSS has the highest priority in the order.
2. CSS being applied through Id of an element takes the second priority in the order.
3. CSS being applied through a CSS class takes third priority in the order.
4. CSS being applied through the <tagName> takes the least priority in the order.

Note: Irrelevant of all above css priority order any css property being added with “!important” takes the highest precedence among all above.

Eg. Font-size:20px !important;

Debugger tool :

We can easily identify corresponding css and html particular look and feel of the webpage.

Using debugger tool we can easily perform CRUD operations on any html element or css properties.

Using this tool we can debug not just the local page but even global pages(amazon,flipkart etc)

There may be different tabs in the debugging tools of different browsers but all do same job i.e to debug the page.

It is also responsible for making dynamic changes to the webpage but those are temporary changes.

You can debug html,css,js code.

Anchor tag

Is a predefined tag been supported in html using which we could able to create reference from current page to some external web page or any other element within the same page.

It takes a mandatory attribute 'href' ,using which we can specify the external webpage link,which need to be linked.

Target is an optional attribute.

Syntax:

```
<a href="absolute/relative path of external page">
    ...link content
</a>
```

Eg:

```
<a href=http://www.google.com>
    Click to visit google page
</a>

Click <a href=http://www.amazon.in>amazon link</a> to visit page
```

Note:

Using anchor tag, we can refer to either local or external webpages.

```
<a href="filePath/webPageAdress" target="_blank">
```

Adding Image to a page: (img) tag

- It is html predefined tag using which we can inject img resource to a webpage.
- It takes the mandatory attribute "src" through which we can provide url or path of the img.

Eg. (global image)

Eg. (local image)

Inline elements and block level elements:

All the elements under html,are been categorized into two types

- 1.block level elements
- 2.inline elements

Block level elements:

Any dom element,which comes under the block level category holds following properties.

- While rendering on the page, a block level element automatically comes to a new line ,irrelevant of whether break tag been added or not.
- Any dom element which is following a block level element,also automatically comes to a new line.
- By default , every block level element tries to occupy 100% width of its container.
- Even though it occupies 100%width by default,we can still control the dimensions of it using css width and height properties.
- "Div" tag is the best example for block level elements.

Inline elements:

Any dom element,comes under category of inline elements holds following properties

- inline elements always tries to render within the same line
- it occupies the space(width and height) on the page based on the content it is holding
- we cannot control the dimensions of an inline element with css

- even though ,we try to control the dimensions using css width and height properties,it simply skips and doesn't consider.
- span tag is the best example for an inline element.

Pseudo classes and pseudo elements

Pseudo classes:

Following are set of predefined pseudo classes been supported in css using which we could able to apply css properties to dom elements,not on load of the page but apply the css properties based on the current state of it

1. :hover
 2. :active
 3. :link
 4. :focus
 5. :first-child
 6. :last-child
 7. :nth-child(n)
 8. :nth-last-child(n)
 9. :checked
 10. :empty
- etc

Example

```
.sample: hover {
...
...// set of css properties gets applied to an element not on load of page,but while a mouse hover happens on the element.
}
```

Pseudo elements:

Through which we can apply css to elements but not to full content only to partial content.

1. ::after
2. ::before
3. ::first-line
4. ::first-letter etc.

Eg. . 1 .abc::first-letter

```
{
.....
.....
}
```

Eg. 2 #test::after

```
{
Content:"continue ... ";
}
```

CSS float property

- By default when an element is rendered ,it tries to render from top left top corner of page,it follows top to bottom approach if it is block level.left to right approach if it is inline.
- In order to change the default rendering direction of the page or to make multiple block level elements to get rendered within the same line,We make use css property float which take possible values.

css float property

- left
- right

any dom element with float:left property, makes the dom element to get rendered to the complete left direction

float:right will make the dom element to get rendered extreme right direction of the container

multiple continuous dom elements having set the float property to either left or right, all the elements will try to render within the same line irrelevant of whether the elements belongs to inline or block level category.

When a dom element, changes its rendering direction to either left or right, the dom element which is following the floated element, will also try to follow the previous elements direction.

CSS clear property:

To make the dom elements, to not follow previous floated element direction, and to follow its own default direction we make use of the css property.

“clear”, which takes following possible values.

- Left
- Right
- Both(recommended)