13.6.18

* **<html lang = “en” >**

Means the language of the html file will be English. This is important for SEARCH ENGINE OPTIMISATION.

* Inside head we write <meta> tags , for SEO i.e. information regarding the page is given in this section.

Eg : <meta charset = “UTF-8”> has UTF-8 as the encoding.

* The data structure used in making a document object model (DOM) is B+ Tree.
* **SEMANTIC TAGS:  
  For things like accessibility and SEO.**

1. <a> </a> is an anchor tag which is used to reference one page to another. It is a span type tag.
2. Target is used to redirect the page.   
   Eg : target = “\_blank” opens a new tab
3. <header> gives info to the Search Engine that the following text are part of the header. Header is important for SEO

Eg: Things like CONTACT US etc. are generally part of HEADER or FOOTER.  
Hence we do need a header.

1. The **HTML <nav> element** represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.
2. <main> </main> is the starting of the MAIN BODY of the page. We have headers, footers , main etc.
3. <section> represents a standalone section in the document.  
   Note : Section is a block element.
4. Footer does not mean that text will come at bottom. It just tells the SE that the following text are part of the footer. Also, sections can have their own footers.
5. <address> indicates that the following text provides contact information for the person, organization etc.
6. The **HTML <aside> element** represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes. **NOTE: doesn’t mean ki text side mai aa jayega.**
7. <article> is used on text. As the name suggests.
8. **<figure> tag** is used to enclose images inside it.   
   With this tag we have **<figcaption> G LOGO </figcaption> ,** this gives a caption to the image.

**CONCLUSION:**

**Semantic Tags are:**

1. **<main>**
2. **<section>**
3. **<nav>**
4. **<header>**
5. **<footer>**
6. **<address>**
7. **<aside>**
8. **<article>**
9. **<figure>**

**NON SEMANTIC TAGS ARE:**

1. **<div>**
2. **<span> etc.**

* **Difference between <b> and <strong> i.e. making text bold :**

While accessing the document, for SEO and for screen readers, <b> is not differentiated from other texts as bold, but <strong> differentiates the text as bold from other text.  
Hence, use <strong> instead of <b>

Similarly, use <em> instead of <i> to make text italic.

* **USING LISTS :** 
  1. <li> uses 100% width
  2. <ul> is used to make unordered lists.
  3. Definition List : Used when definitions are to be given to each title.  
     <dl>  
      <dt> WORD </dt>  
      <dd>  
      MEANING OF WORD  
     </dd>  
     </dl>
* &nbsp; gives space
* **FORM**

Forms are used in HTML to get user input and send the data to the server.   
we can only design the form in HTML. We can’t set the server.

**<form action = “#”>** means when the work is done in the page i.e. form ka kaam ho gaya toh the page is directed to itself (if we use hash) or the page is directed to the link put in action.  
using

**<form action = “#” method = “post”** > makes sure that the ID and PASSWORD are not appended in the URL.

* **Read about Iframe**
* **Read about <video> <audio> and preload in these tags**
* **DOM – DOCUMENT OBJECT MODEL** is the sequence of steps followed before a page is loaded.

All the tags <head> <body> etc are passed as TOKENS to DOM PARSER.

A DOM TREE is made in DOM PASSER using the tokens as nodes.

* The tree made by CSS is **CSSOM.** Both DOM Tree and CSSOM Tree combine to make **Render Tree.**

**16.6.18**

**CSS INTRODUCTION:**

* it is used to style boxes
* follows the box model
* if we need to style only a specific box, we give it a box id and then write the CSS code for that id only.
* When you write <div> \_\_\_\_ </div> in different lines you get text spacing i.e. there is a small gap between each div even if you put them in-line;  
  Another method to remove text spacing, use font-size = 0 in the container and then overwrite the font size in box class so that there is an actual font size in each box.
* **Calc(33.33% - 40 px) , if we need to subtract 40 px from the total, we don’t need to calculate % ourselves. We can use calc() and subtract 40 px from the total %.**
* **box-sizing: border-box ;** when we give padding, the box size does not increase, instead the content moves inside i.e. the borders of the box are fixed.  
  **box-sizing:content-box;**  leads to increase in box size while padding.
* **Hover** is a pseudo selecter that exists at Runtime only.