**Web fundamentals**

Developing + testing + hosting + maintenance

**Life cycles**- s/w development lif cycles

- Waterfall model

- agile

- spiral

-iterative

-RAD

-V-shape

URL- Uniform Resource Locator

https: hyper text transfer protocol secure

Domain: unique name/server

Sub domain: written before main domain (eg:education,aws)

Top level domain: .edu, .com, .in

Relative path: ./ -> current directory

../ -> one step back

**Shortcuts**:

Ctrl+alt+arrow keys- to edit multiple lines at the same time

**HTML**:

Tags: <>,</>

Elements: <> </>

<hr/>-horizontal line

Anchor: <a href=” “ target=”blank”> -> opens in new page

Image: <img src=” “ alt= “ “/>

**Lists:**

ol/ul- attribute “type=, start= has to be number” li-list item

ul- none, disc, square,circle (type or style)

<mark> - highlight

description list-

<dl>

Dt(description term),dd(description definition)

</dl>

**Table**:

<Table, style=”border-collapse: collapse;”>

Tr-row, td-table data/ col

Th-heading inside cell

Colspan attribute- merges columns

lly rowspan

</table>

Sup,sub

**Block scope**: div,aside,h1-h6,header,p,ol,ul,table,main

**Inline scope**: span,img,a,strong,em,br,label,button,input

**Input:** recommended to use label tags

<input type=”text/number/password” placeholder=”text displayed in input box”>-inline

Type=”file” : to upload files

Value attribute- saved stuff (helpful for backend)

Type=”radio”, name=”give common name for grouping” so that only one button can be selected at a time

Checked att- to default select

Disabled att- not working button

Required att

&nbsp-space

<textarea> : big resizeable box – attributes: rows, cols

Checkbox- multiple can be selected

Datetime-local

<select>

<option></option>

</select> - drop down

Submit, reset- type

**Forms**: can be wrapped in table for proper spacing

Uses inputs

**Video:**

<video controls autoplay loop>

<source src=” “ type=””/>

</video>

Ogg works in video as well as ogg but mp3 and mp4 cant be interchanged

iFrame – youtube videos -> Share -> embedded

**semantic**- describes its meaning (header, section, article, aside, footer) – used to distinguish groups of tags under particular section of the website. Used to reduce usage of div and span

**Css: inline, internal/embedded/ external**

For internal- style tag in head

External- link tag with href, rel=”stylesheet” in head

Selector- selecting particular tags in your file

Element selector (directly using tags like h1-6, p etc in external css)

Selector list- applying css to 2 or more elements together using commas

\* : universal selector

# : ID selector – unique

. : class selector

Units in css: px, %, vh, vw, rem, em – calc() function to convert

Descendant selector – to apply css to tag within tag

Parent child{

}

Direct child selector – “parent>child” similar to descendant but has to be direct child

Precedence / Css specificity: inline>external=internal (whatever is written first)

Selectors: id>class>element

**Css box model**

border- solid, dotted, dashed etc

Padding ( inside box) : can be given one common to all top,right, bottom, left or separately

Write as padding: 1px 2px 3px 4px

Margin(outside box) : similar to padding

Display : block/inline – inline doesn’t take additional height or width

Inline-block – will take height and width inspite of being inline

**Position in css**

Static (default)- doesn’t collide with other elements

Relative(wrt static): allows changes in top right bottom left otherwise same as static

Offset- top,right,bottom,left

Absolute- removes element from document flow and it stays in it’s position without affecting any of the other elements which act as if the absolute element doesn’t exist.

It refers to the first parent that has position other than static for offsets

Sticky : acts like relative till it reaches the specified position and then becomes fixed.

It stays fixed only till it’s parent element. After that its relative

Fixed (stays fixed on scrolling), has nothing to do with parents, uses the entire html doc as reference

Box sizing: border box (no collision in elements)

**Pseudo classes:**

Indirectly applying style to children classes:

Span: first-child/last child/.class\_name :nth-child(number,even,odd){

}

Css on states of certain elements.

eg: button- hover(present cont), active(present), focus(past)

for anchor: link,visited

button:hover{

cursor:pointer;

color:white;

background-color:black;

} //button changes colour on hovering and cursor becomes a pointer

**Pseudo elements**- to append text without access to html file

Use :: in css file

Eg: h1::be fore/after{

Content:””; - mandatory

}

To collide shapes: using pseudo elements

Use positions parent: relative, child: absolute

**Flexbox**- used for 1-D layouts

Container display:flex, then all children wrapped under will occupy only required space

Flex direction(defines main axis and cross axis): row(default)/column/row,col-reverse/ Gap

Justify-content(main axis(default)):flex-start/end/center (row)/ space-around,evenly,between

Align-items(cross axis)- col (flex-start/end, center, stretch…)

Order: number; (not recommended)

Flex:1; used to divide the parent ratio-wise (alternative for width)

Flex-wrap: wrap; content wont shrink, instead wrap into multiple lines

Animations: check code

Button{

Transition-delay, transition-duration,property

}

Or

Transition: property, timing function, delay, duration

**CSS animations**