Cascading Style Sheets

CSS 3

WITH CSS

WITHOUT CSS

- HTML :- Structure & Content
- CSS : -Presentation
- CSS is the language , use to style an HTML document.
- CSS describes how HTML elements should be displayed.

- HTML tags have properties -> use default if not specified
  E.g *<h2> have font-size*
- CSS is used to specify these properties.
- Applicable to one tag or all tags , based on how this value is set.
- Imposing consistency on the style of web pages
- Style sheet is Cascading?

```css
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Sample CSS

LEVELS OF STYLE SHEET

- 3 Levels
  - ❖ **Inline (1^)**
  - ❖ **Internal (2^)**
  - ❖ **External (3^)**

- Inline : Style to single element --In opening tag
- Document : To an entire document page --in <head>
- External :To a website– as separate txt file

- Precedence & Conflict

  inline --- document---external

E.g.. Conflict

Q: <p> Computer Science </p>

Inline : color
Document : color
External : color

o/p: Computer Science

E.g Cascading

Inline : color
Document : *Italics*
External : color, 20px

o/p : *Computer Science*

- Other features are browser default(family, weight)

- External Style sheet

  - apply to more than one document

  - simple text file with MIME type is text/css.

  -specify external style by

    1. <link> tag
    2. @import directive

  -rel attribute for relationship .

  - href for specifying path of external sheet.

  -<link> at head section

  - it can be validated : jigsaw

<link rel="*stylesheet*" type="*text/css*" href="*mycss.css*" />

- Alternative way

  - @import url(filename)

  - should come before other css declaration

  - contain markup & other styles

  - css mechanism of adding css

  - <link> is html mechanism of adding css

  - performance degrade: don’t use it

STYLE FORMAT

- Based on levels

INLINE:

1. within opening tag
2. As value of style attribute

style=“property_1: value_1; property_2: value_2; ……….”

DOCUMENT / Internal

1. With in head section
2. With in <style>

```
<style type=“text / css”>

    rule list

 </style>
```

- **Rule** :   selector {

```
                property_1: value_1;
                property_2: value_2;  ……………….;
           }
```

- Rule has 2 parts

  I. Selector : represent tag / tags affected by rule

  II. Set of values

  III. Multiple value are separated using space

```
Like   selector { property_1: value_1   value_11 value_111;
                  property_2: value_1 ;
                  property_3: value_1 ;
               }
```

EXTERNAL

i. Same as documents style sheet

ii. As separate text file

iii. No style tag - only rules

In css : /* comments */

SELECTORS -6

» Simple selector

» Class selector

» Generic selector

» id selector

» Universal selector

» Pseudo classes selector

Simple selector

- One or more Tag name like h1 or p
- Applicable for all occurrence of named element
- Comma separated tag name

```
h1  {  font-size :10 px;  }
h2, h3 {font-size :20 px; }
```

- Contextual / descendant selector :

  - Hierarchy by space

```
body  b em {font-size :24 px; }
```

Class selector

- Different style to different occurrence of same tag
- E.g. More than one <p> element

- Using . symbol & class attribute in XHTML.

p . normal {color : red}

p. abnormal { color :blue}

<p class=“normal”> CEC </p> <p class=“normal”> CS</p>

o/p : ?

CEC

CS

Generic selector

- Style applies to the content of more than one kind of tag
- Begin with • Symbol

```
.sale { color:  red;}
```

```
<h3 class=“sale” > EEE </h3>
<p class=“sale” > ECE </P>
```

O/P :

```
EEE
ECE
```

id selector

- Style to one specific element : inline style sheet
- # symbol
- Id is unique in web page, class not

```
    #section_10 { font-weight: bold ;  }
```

```
  <h2 id=“section_10 ” > CEC  </h2>
```

o/p : **CEC**

Universal selector

- To apply the style to all elements in the document
- Use * symbol

```
 * {color: green;}

/* Make all element in the document green */
```

Pseudo classes selector

- Pseudo classes are styles that apply when something happens rather than because the target element simply exists.
- Like *handover* in link , *focus* of a text field

    - hover

    -focus
- Use : symbol

```
input : hover {color: red;}
input : focus {color : brown; }
```

```
<input type="text" size="10" name="username" />
```

Property value forms

- 60 different properties into 7 category:

  1. fonts
  2. lists
  3. alignment of text
  4. margins
  5. colors
  6. background
  7. borders

- Property values can be:
  - ❖ Keywords: if few possible values – e.g. medium, bold, BOLD
  - ❖ Number value : 15 (+ -)

- Length value : number followed by unit- e.g 28px

  possible units : px,in,cm,mm,ex,em,pt,pc

  em : current font size
  ex : height of size ‘X’ } relative length

- Percentage value: 45% (+ -)
- URL values: url(dog.jpg)
- Color value: 1. named colors() -red
  2. hexadecimal -#A6B670
  3.rgb -rgb(0,255,132)

- Some property values are inherited to inner element too, some not.

  background-color: ✗

  font-size: ✓

Font properties

- Font-families: used to specify list of font families -

  :browser use the first font in the list that it support

  : precedence from high – low

  : use browser default if not support any specified

  e.g. font-families : Arial, Futura, ‘ Times New Roman’

- Font-size: as name implies

  e.g font-size: 10pt / small / xx-large / 30%

- Font- variant: specify character font,

  e.g font-variant: normal/ small-caps

- Font-style : italic / oblique

- Font- weight :degree of boldness

  e.g : font-weight: bold/ lighter / (100-900)

- Font- shorthands: used to specify ore than one font property

  e.g 1 : font: bold 14pt Arial

  e.g 2: font: italic small-caps bold 20px arial 'times new roman'

  - Last property should font family * minimum required *
  - Second last should font-size*
  - Others can be in any order

- Text-decoration: line-through / overline / underline / none

  - many browser underline links

- Letter-spacing: space between character

  e.g letter-spacing: 3pt;

List, color & Alignment of text properties

- Two types of list

  1. unordered list (default -disc)
  2. ordered list (default -sequence no)

- To change default bullet by using,

  ul {list-style-type: disc/ square/ circle/none;}

  ol {list-style-type: decimal/ upper-alpha/ lower-roman;}

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

- The same property can apply to list items (<li>) too.

- 3 levels of collection of Colors:

✔ named colors (16)- displayable by all browsers- e.g blue

✔ web palette (216)- web safe color- hex(00 33 66 99 cc ff)

✔ Six hexadecimal digit- 16 million color

- Set the foreground color of an element ,

  color : red;

- To set the background color of an element,

  background-color: blue;

- The first line of paragraph can be indented ,

  p {text-indent : 2.5in / 10% ; }

- To arrange text horizontally ,

  text-align: left (deault) /center/ right/ justify ;

- Float property is used to specify that text should flow around some element.

e.g : <img> </img>

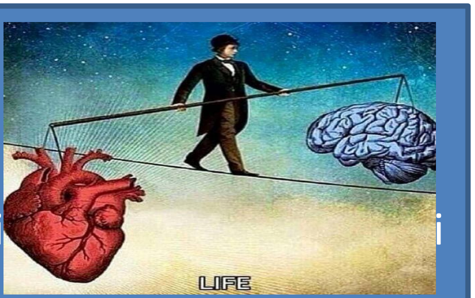
<p> </p>

Img {float: right;}

hiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii



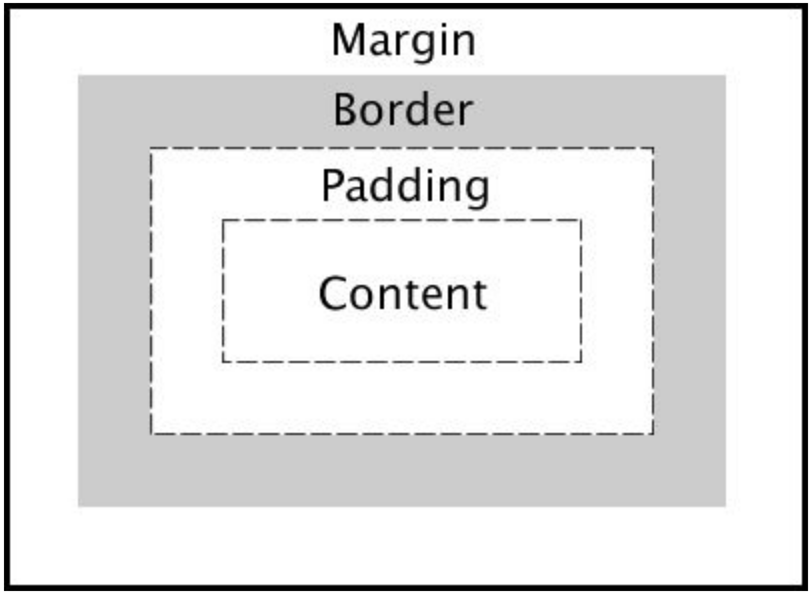
Iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii

Box model

- All xhtml element has a box structure.



- Virtually all element have border with various style(width /color).
- Padding : space b/w content & border
- Margin : space b/w border & adjacent element

- Elements content has border (4 sides), as well as style of border,

  e.g border-style :solid/ dotted /dashed/ double/none (d);

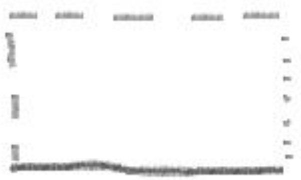
  (or)

  border-top-style: dashed;

  border-bottom-style : solid;

  border-left-style : dashed;

  border-right-style: dotted;



- Thickness of the border,

  e.g. border-width :thin/ medium(d)/thick/ 23px;

  (or)

  border-top-width : thin ;

  border-bottom-width : medium;

  border-left-width : medium;

  border-right-width: thick;

- Color of the border by,

  border-color: red;

  (or)

  border-top-color: blue;

  border-bottom-color: green;

  border-left-color : yellow ;

  border-right-color: violet;
- Background of an element filled in padding.

- padding : 0.5 in; // apply to padding in all sides

padding-left	padding-top
padding- right	padding-bottom

- Margin : 0.5in; // apply to all sides of margin

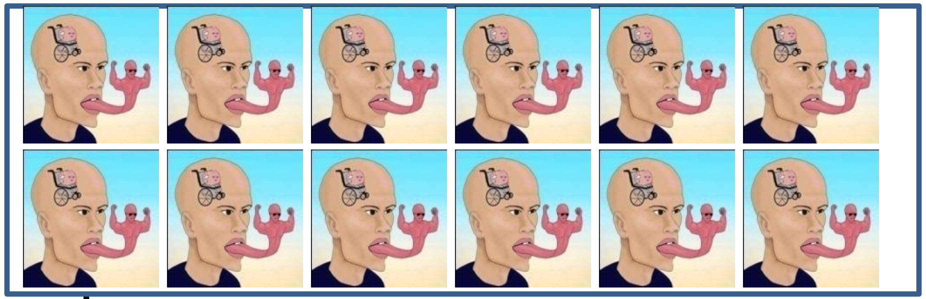
margin-left	margin-top
margin-right	margin-bottom

Background images

- To set background image by,

  background-image: url(dog.jpg)

- Tiling : The background image is replicated to fill the area of element



- Control tiling by,

  background-repeat: repeat(d)/ repeat-x/ repeat-y/no-repeat;

- Position of background image,

  background-position: left center / right bottom;

  (x-axis)(y-axis)

Div & span tag

- <span> : -no default layout for span

  : used to apply special properties to less than a whole paragraph of text.

  :change property values of part of a line of content

e.g : <p> it sure to <span> total</span> content .

css: span{color: red;

font-style: italics;

} o/p: it sure to *total* control

- <div> : like span, no implied layout for the content of div

  :it is used to specify presentation details for a section (contain many paragraph , heading etc..) of document

<div> <p>….</p> <p>….</p> <p>….</p> </div>

Conflict resolution

- Different way of occurring conflict in CSS:

  1. different value for same element in diff levels of style

     - solved by precedence

  2. with in same document

     e,g h3 { color : blue;}

     body h3 { color: green;}

  3. by inheitance

     - side 20

  4. by different origin of style specification

     - by developer
     - by user (highest precedence)
     - by browser

- By different selector (* and id)
- Style of an element is cascade of style applicable to it. That’s why it is cascade style sheet .

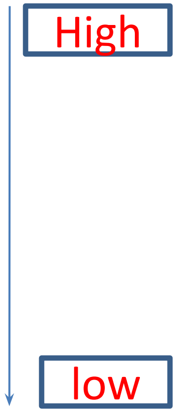
- We can use **weight**(important/ normal) of specification to avoid conflict

  e.g font-style: italics !important; color : blue;

- conflict resolution is multi stage processing .

  1. collect style specification from 3 level of style
  2. sort based on it relative precedence
  3. next sorted the specification based on origin & weight

     i. Important declaration with user origin
     ii. Important declaration with author origin
     iii. Normal declaration with author origin
     iv. Normal declaration with user origin
     v. Any declaration with browser



High

low

4. Still conflict, avoided by precedence of selector

   1. id selector
   2. class & pseudo selector
   3. contextual selector (p, h3)
   4. universal selector

- The whole sorting process that is used to resolve style specification conflict is called cascade. (look slide 3)