Web Programming

First GUI-based browser – Mosaic

Server characteristics

File structure of server consists of 2 categories

i) Document root : stores web docs
ii) Server root : Includes server & its support s/w

can be directly accessed by server. Client can access it indirectly using high-level URLs.

Virtual document Trees : Secondary areas from which documents can be accessed by the server

Proxy server : Servers are allowed to access the web documents from the document root of other server on the web.

List out the protocol supported by web servers

i) HTTP
ii) FTP
iii) Telnet
iv) gopher
v) news
vi) mailto

How do web servers interact with database systems.
→ Through CGI (Common Gateway Interface)

URL - Used to identify resources / docs on the internet

general form: scheme : Address

↙

specifies the communication protocol

http : // FQDN / pathtodoc

↙ handles documents

Created - XHTML

→ Fully Quantified Domain Name

URL Paths

- Similar to
- Directories & files are separated by separator character.

Complete & Partial Path

Complete path includes all directories along the path to a file.

Partial path: base URL path is specified in configuration path of the server.

MIME : Multipurpose Internet Mail Extension

MIME spec has the following form

i.e. type/subtype

⇓

e.g. image / jpg

video / mp4

text / plain

specifies how messages must be formatted so that they can be exchanged b/w email systems.

Experimental Document Types

→ starts with "X-"

Web providers can add an experimental subtype along with the MIME specification so that particular data can be available to others.

Helper apps are programs that are external to the browser

Code inserted into browsers – Plugins.

HTTP :

2 phases
- i) Request
- ii) Respons

Each HTTP consists of 2 parts
1. Header
2. Body

HTTP Request Phase

General forms
- i) HTTP method
- ii) Request Header
- iii) Blank line
- iv) Message Body

GET / Ex. html

HTTP Methods

- GET
- HEAD
- POST
- PUT
- DELETE

Example : Accept : text/plain

Field name — Value

GET, HEAD, DELETE doesn't include message body

Only POST & PUT does

4 categories of header files

i) General
ii) Request header
iii) Response header
iv) Entity - Included both in request header & response header

Post Method

Field content-length

↳ specifies length of response in bytes

content-type

etc.

HTTP response Phase

1. Status Line ⟶ 3 parts
2. Response header
3. Blank line
4. Response body

Ex.

HTTP Version	Status code	Textual rep.
HTTP/1.1	200	ok

1 – Information
2 – Success
3 – Redirectional
4 – Client error
5 – Server error

Response headers
Date
Server
Last modified
Content-type
Content-length
Connected close

Web Programmers Toolbox

1. XHTML – Extensible HTML – used to create layout
   – Contents & Controls → stage
2. XML – Extensible markup language
   – Simplified version of SGML (Standard generalized markup language).
   – Description of documents
3. Javascript – Can be directly embedded directly in XHTML document.
4. Java – Applets – client-side
   servlets – server-side

5. Perl – CGI (Common Gateway Interface) is used to write programs in Perl.

6. PHP – Preprocessed code
Dynamic typing

7. Ruby – Dynamic typing
Rails – framework for development of web application

8. Ajax – Asynchronous Javascript + XML
Used for development of efficient & interactive web based applications

XHTML

Latest version – HTML 5

Earlier version – HTML 1.0
HTML 2.0 → 1995
HTML 3.2 → 3.0 → 1997
HTML 4.01 → 4.0 → 1999
HTML 5

XHTML 1.0
XHTML 1.1 → 2001
XHTML 2.0 W3C

HTML V/S XHTML

- more strict
- HTML defined as an XML
- document structure is mandatory
- lowercase letters
- Properly nest
- Close the tags properly

Basic Syntax

Tags

<tagname>

Comment

<!-- . . . . >

XHTML Document structure

<? xml version = "1.0" .encoding = "utf-8" ? >

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 strict //EN": "http : www w . w3. org /TR /XHTML 1./DTD /XHTML-strict .dtd" >

Types - X HTML DTD's

1. Transitional
2. FrameSet - Frames
3. Strict <frameset> instead of body.

Displaying Headers & Paragraphs

Ordered Lists E.g. (→ After ordered lists)

```
<ol>
    <li> characters
        <ol>
            <li> Alpha
                <ol>
                    <li> A A </li>
                    <li> B 1 </li>
                </ol>
            </li>
            <li> Digits
                <ol>
                    <li> 234 </li>
                    <li> 5 78 </li>
                </ol>
            </li>
        </ol>
    </li>
    <li> others
        <ol>
            <li> Special
                <ol>
                    <li> xyz </li>
                    <li> PQR </li>
                </ol>
            </li>
        </ol>
    </li>
</ol>
```

Output

1. Character
    1. Alphabetical
        1. AA
        2. BB
2. Digit
    1. 234
    2. 576
3. Others
    1. Special
        1. XYZ
        2. PQR

Def:

Basic Syntax

Fundamental syntactic unit of HTML/XHTML → Tags

Syntax : < >

For comments :

<!-- comment -->

General form of XHTML document structure

<?XML version = "1.0" encoding = "utf-8"?>

<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1 strict//EN"

DTD = Document type dfintion

↳ Syntax rules of HTML

Type - XHTML DTDs

1. Transitional
2. Frame set
3. Strict

Use this for XHTML

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title> My webpage </title>
  </head>
  <body>
    <h1> Title Page </h1>
    <p>
      This is a paragraph
    </p>
  </body>
</html>
```

line break :
↓
Space

Block Quotation

Used to differentiate from the normal flow of text

```
<p> xxxx </p>
<p>
  <blockquote> AA </blockquote>
</p>
<p> yyyy </p>
```

Output :

```
xxxx
  AA
yyyy
```

Font Styles

 : emphasis
 : Bold
<code> : Displayed in monospace font
<sup> : superscript
<sub> : subscript → `<p> x <sup> 2 </sup> </p>` → x^2

Character entities (Ones which can't be included directly)

Characters	Entities
&	&amp
<	&lt
>	&gt
"	&quot
' (apostrophe)	&apos
¼	&frac14
°	&deg

Horizontal Rules

Used to end the current line

<hr /> ← space

Draws a horizontal line

The thickness of the line varies depending on the browser

Images

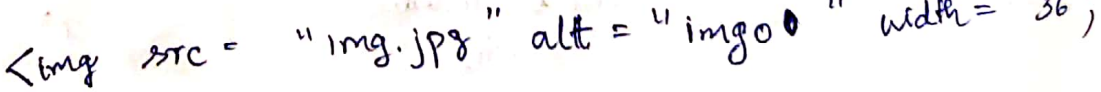
GIF → Graphical Interchange Format
Uses 8-bit colour representation
Can have max. 256 different colours

JPEG → Joint Photographic Exchange Group

PNG → Portable Network Graphics

JPEG uses 24-bit colour representation
Can have 16 million different colours
JPEG uses better compression algo compared to GIF.

PNG has advantages of both, but takes more space



(width & height are optional)

Hypertext links

Specifies address to other document

```
<a href = "a.html"> click here </a>
```

- `<a>` → anchor tag
- `href = "a.html"` → address to other document
- click here → Message that is hyperlinked to a.html

List

Used to list out items

- i) Ordered lists
- ii) Unordered lists

Unordered list

```
<ul type = "circle" or "square" or "disc">
    <li> A </li>
    <li> B </li>
    <li> C </li>
</ul>
```

Output

- A
- B
- C

Ordered lists

```
<ol>  ← type = "I" or "i" or "a" or "A"
    <li> A </li>
    <li> B </li>
    <li> C </li>
</ol>
```

Output

1. A
2. B
3. C

① To start ordering with, say 5,

```
<ol start = "5">
    ⋮
</ol>
```

Definition list

```
<dl>
  <dt>  → Term
  <ddl> → Definition
```

Example:

```
<body>
  <dl>
    <dt> JSP </dl>
    <dd> Java Server Pages </dd>
  </dl>
</body>
```

Output

JSP
Java Server Pages

Tables

<table>

i) border – width of border,
If 0 is given, no border
If negative is given, 1 is taken
Default depends on browser.
border = "0"

ii) align – "left"
"right"
"center"

~~E.g.~~

<caption> - set title of table
<th> - table header
<tr> - table row
<td> - cell value

E.g.

```
<table border = "3">
  <caption> Student </caption>
    <tr>
      <th> Name </th>
      <th> Roll </th>
    </tr>
    <tr>
      <td> XYZ </td>
      <td> 35 </td>
    </tr>
    <tr>
      <td> TUV </td>
      <td> 45 </td>
    <tr>
</table>
```

O.P.

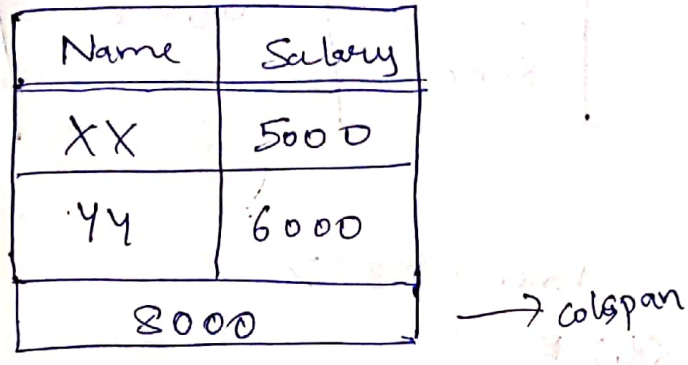
Student

Name	Roll
XYZ	35
TUV	45

Rowspan & Colspan · attributes

Rowspan – Single column spans many rows.
Colspan – Single row spans many columns

i) Colspan




```
<table border = "3">
    <tr>
        <th> Name </th>
        <th> Salary </th>
    </tr>
    <tr>
        <td> XX </td>
        <td> 5000 </td>
    </tr>
    <tr>
        <td> YY </td>
        <td> 6000 </td>
    </tr>
    <tr>
        <td colspan="2"> 8000 </td>
    </tr>
</table>
```

ii) Rowspan

Month	Salary	VacationSal
Jan	5000	10000
Feb	8000	

Rowspan

```
<table  border = "3">
    <tr>
      <th> Month </th>
      <th> Salary </th>
      <th> VacationSal </th>
    </tr>
    <tr>
      <td> Jan </td>
      <td> 5000 </td>
      <td rowspan="2"> 10000 </td>
    </tr>
    <tr>
      <td> Feb </td>
      <td> 8000 </td>
    </tr>
</table>
```

Alignment

- align – left, right or center
- valign – top, bottom

E.g.

Same table as before,
Headers are center-aligned
Others

Default – headers – center-aligned
other cells – left aligned.

Cell padding & cellspacing

Cell padding – specify spacing b/w content in cell & walls of the cell

Cell spacing – specify spacing b/w 2 cells

Forms

Common way for user to communicate information from web browser to server.

<form> → Attribute action = "____"
value of action in URL

E.g.

Write code for form with following controls

Text
Password
Checkbox
Radiobutton

```
<form>
  <p> Emp Name :
    <input type = "text", name = "Ename" size = "20"
           maxlength = "15" />
  </p>
  <p> Password :
    <input type = "password" name = "pwd" size = "25" />
  </p>
  <p>
    <input type = "
```

Emp Name :

PWD : ••••

Q.

- [x] Java
- [ ] C++
- [ ] C

```
<form action = " " >
  <p>
    <label> <input type = "checkbox" name = "lang" value = "c1" />
      Java </label>
    <label> <input type = "checkbox" name = "lang" value = "c2" />
      C++ </label>
    <label> <input type = "checkbox" name = "lang" value = "c3" />
      C </label>
  </p>
</form>
```

Radio button Control

○ AA
○ BB

```
<form action = " " >
   <P>
    <label> <input type ="radio" name = "lang" value = "r1" />
        AA </label>
    <label> <input type = " radio " name = "lang " value= "r2" />
        BB </label>
   </P>
</form>
```

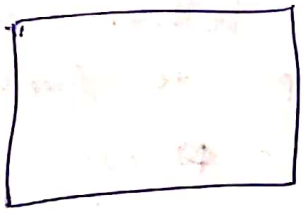
Textarea control

~~<form action= " ">~~

```
Abstract : <textarea name = " information " row = "3.0"
cols = "90" >  </textarea>
```

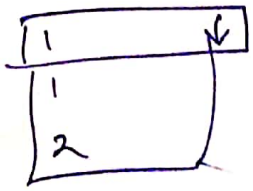
~~<form>~~

```
<P>
   Abstract:
```



Select box (Combobox)

```
<select>
   <option selected = "selected" name = "opt " value = " 1 " > 1 </option>
   <option name = "opt ". value = " 2 " > 2 </option>
</select>
```

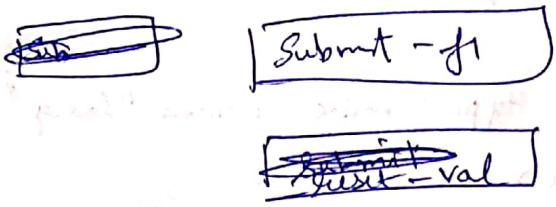


Submit Control

<input type = "submit" . Value = "submit" value = "submit" />

<input type = "reset" . Value = "reset" val "/>

O.P :



Frames

Used to display more than one document at a time

i) Window can divided into rectangular areas

Frameset tag - i) Specify no. of frames & their layout in the browser window

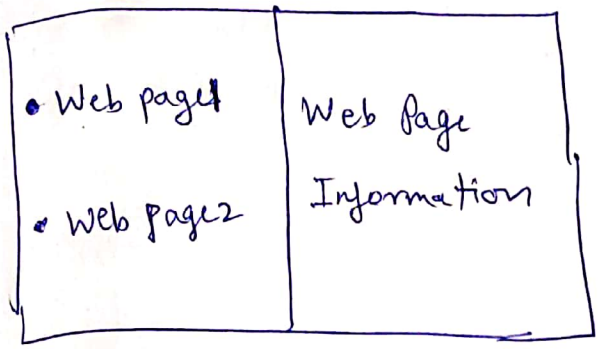
ii) Document has either <body> of <frameset> tag (not both).

iii) Set doctype to frameset

iv) Attributes - Rows, Cols } Type of values - numbers, percentages, *

<frameset rows = "33%, 33%, 33%" cols = "50%, *">

</frameset>




```
<frameset cols="50%,*">
  <frame src = "Ex1.html" />
  <frame src = "Ex2.html" name = "Info" />
</frameset> <ul>
      <li> <a href = "webprojects.html"> webpages </li>
      <li> <a href = "wp2.html"> webpages </li>
  </ul>
```

HTML 5

- better support for various media including video, audio.
- email, url, data, form, search
- Better browser compatability
- Better passing routes

Removed elements in HTML 5	Use instead
<acronym>	<abbr>
<strike>	<s> (or) <del>
<font>	css
<big>	css
	css
<applet>	<object>

Article

~~<article> </art~~

~~Se~~

Article

```
<article>
</article>
    <header>
        <h1> ——— <h1>
    </header>
</article>
```

```
<footer>
  =
  =
</footer>
```

```
<section>
  =
  =
  =
</section>
```

SVG

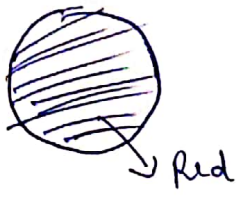
Scalable vector graphics

Used to design graphics for web & support for event handlers.

<svg> </svg>

```
<body>
<svg width="100" height="100" >
  <circle cx="50" cy="50" r="40" stroke="green"
    stroke-width="4" fill="red" />
</svg>
</body>
```

O.P.



Canvas

```
<body>
  <canvas id="mycanv" width="100" height="100"
    style="border: 1px solid: #000000" />
  </canvas>
</body>
```

O.P (Empty rectangle)

Audio

```
<audio controls>
  <source src="Ex.mp3" type="audio/mpeg">
</audio>
```

mp3, WAV, OGG

Video

```
<video width="100" height="100" controls>
  <source src="movie.mp4" type="video/mp4">
</video>
```

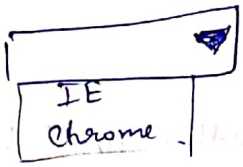
MP4, OGG, WebM

HTML 5 Form elements

1) datalist

```
<form action="">
<input list="browsers">
<datalist id="browsers">
    <option value="IE">
    <option value="chrome">
</dataset>
</form>
```

O/P



2) <output>

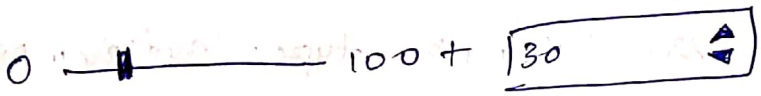
0 <input type="range" id="a" > 100

+ <input type="number" id="b"

min="10" max="50" value="30" >

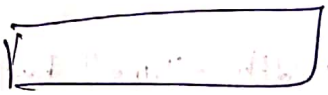
= output name = "Res" for="a b" </output>

O.P



HTML new input types

Color
date
email
Search
number
range etc



HTML form attribute

No validate

~~form~~

Autofocus.

Quatation tag

<q> & <blockquote cite="___" >

Fomatting tags

<bold>
~~<small>~~

<i> <small> —
 <sup> <sub>
<del> <ins> <mark>

Block level elements

Begin on new line
Inline elements don't begin on new line.

<address> <article> <blockquote> <footer> <dd> <dl>
<dt> <form> <h1> to <h1> <header> <hr> <col> <p>
<section> <table>

Inline elements

<a> <abbr>
  <label>
<sup> <sub> <code> <i>
<select> <img> <textarea>
<input> <small>

Unit II

CSS

Cascading Style Sheets

Advantages

- Better UI
- Seperate out web content from web presentation.
- Used to impose consistency in styles in the web content

Types of style sheets (Levels)

- Inline style sheet
- Document level style sheet
- External style sheet

1. Inline style sheet

specify style within tags itself.

Disadvantage - style & web content are mixing.

```
<body>
    <p> Text </p>
    <p style = "font-family : Arial ; font-size : 20p
        Different Text </p>
</body>
```

② Document style sheets

```
<head>
<style type = "text/css">
    h1 { color : red;
         font size : 3pt }
    p { color : blue }
</style>
</head>

<body>
  <h1> Heading </h1>
  <p> Web Page </p>
</body>
```

③ External style sheet

2 files → .css & .html

```
h1 {color : Pink}

p {color : Red}
```

```
<head>
<link rel = "stylesheet"
type = "text/css" href = "Extern" >
</head>
<body>
   <h1> ___ </h1>
   <p> ___ </p>
</body>
</body
```

Comments in CSS

```
/* Type comment */
```

Selector forms

- Can be defined in various ways

i) simple selector

```
h1 {color: red;}
```

selector

```
h2, h3 {color: green}
```

ii) Class selectors

```
<style type="text/css">
    h1.redtext {color: red;}
    p.bluetext {color: blue;}
</style>
```

Used to apply diff styles for the same element.

iii) Generic selectors

```
<body>
  <h1 class="redtext">
      Heading </h1>
  <p class="bluetext"> Text </p>
  </body>
```

iii) Generic selectors

```
<style type="text/css">
    .redtext {color: red;}
    .bluetext {color: blue;}
</style>
```

iv) id selectors

```
<style type "text/css">
    #top {
        color: red;
        font-size: 12pt;
    }
</style>
```

id

```
<body>
    <h1 id = "top">.Heading 0 </h1>
    <p id = "top"> Text </p>
</body>
```

v) Universal selector

```
<style type = "text/css"
    * {
        color: blue;
        font-size: 30pt; }
</style>

<body>
  <h2> Heading 2 </h2>
  <p> Text1 </p>
</body>
```

Categories of Properties

i) Font-family

E.g. Times New Roman

Serif — Garamond

Sans-serif — Arial, Helvetica

monospace — Carrier, prestige

ii) Font size

```
font-size : 12pt
        or
        30px;
```

small, medium,
large, x-small, xx-small
x-large, xx-large

iii) Font variants

```
font-variant : small-caps ;
        or
        normal
```

iv) Font styles

```
font-style : italic ;
        or
        oblique
```

v) Font weights

```
font-weight : bold ;
        { bolder
        { lighter
          normal
```

vi) Font shorthands

font : bold 20pt 'Times New Roman';

vii) Text ~~deletion~~ decoration

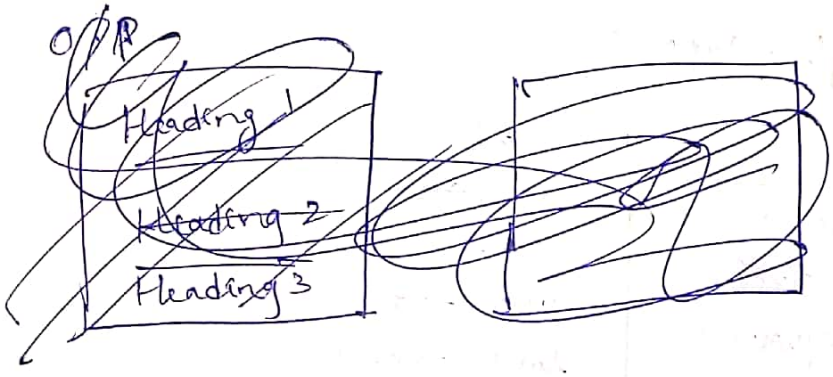
text-decoration : 'underline'

or

'line through'

or

~~overlay~~ 'overline'.




```
<head>
  <style>
    h1 { text-decoration : linethrough; }
    h2 { text-decoration : overline; }
    h3 {text-decoration : underline; }
  </style>
</head>
<body>
  <h1> Heading 1 </h1>
  <h2> Heading 2 </h2>
  <h3> Heading 3 </h3>
</body>
</html>
```

O/P.

Heading 1

Heading 2

Heading 3

viii) List Properties

list-style-type ;

ul	ol
disc ;	~~upper~~
circle ;	decimal
square ;	upper-roman
	lower-roman
	upper-alpha
	lower-alpha

To get all same

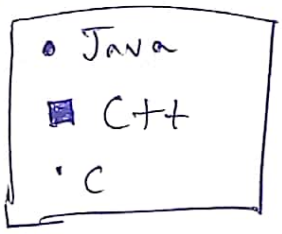
ul { list-style-type : disc }

E.g.

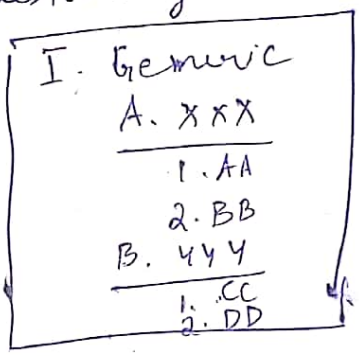
```
<html>
  <head>
    <style>
      li.red { list-style-type : 'circle' ; }
      li.gr { list-style-type : 'square' ; }
      li.blu { list-style-type : 'disc' ; }
    </style>
  </head>
```

```
<body>
  <ul>
    <li class = "r1"> Java </li>
    <li class = "r2"> C++ </li>
    <li class = "r3"> C </li>
  </ul>
</body>
```

O/P



Q. Write question for



* Image can also be used instead of index for list.

```
<style type = "text/css">
  li.image { list-style-image : url(ex1.jpg) }
</style>

<li class = "image"> Java </li>
```

Color group

Property name: color

rgb (v_1, v_2, v_3)

Name	Hexadecimal Value
black	000000
red	FF0000
green	008000
blue	0000FF
white	FFFFFF

Larger set of color - web palette (216 colors)

E.g. background - color : red ;

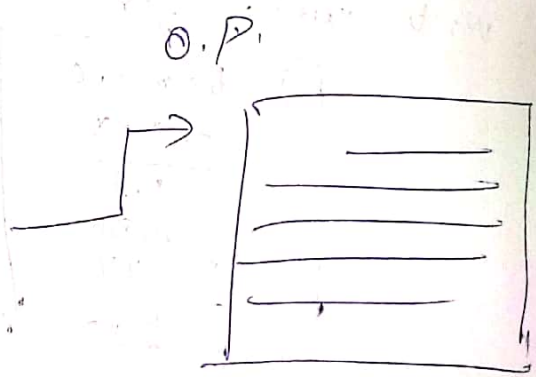
Alignment of text

text-align : text

text-align : right

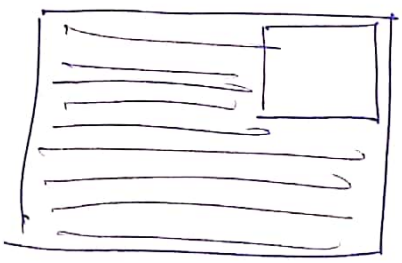
text-align : center

text-indent : 0.5in



float

```
<style type="text/css">
    img { float : right }
</style>
```



Borders

1. Border style Property

border-style (in CSS 1.0)

CSS 2.0
- border-top-style
- border-bottom-style
- border-left-style
- border-right-style

Values
- solid
- dotted
- dashed
- double
- etc.

2. Border width Property

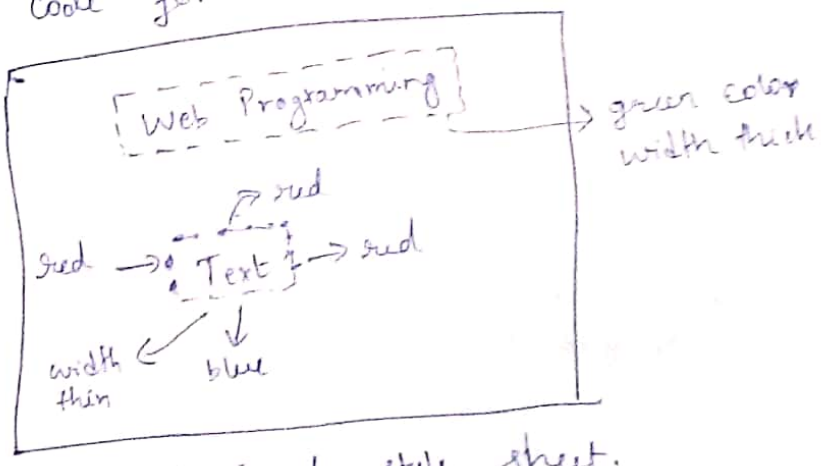
- border-top-width
- border-bottom-width
- border-left-width
- border-right-width

Values
- thick
- thin
- medium
- length value in pixels

3. Border color property

- border-color
- border-top-color
- ~~border-top-color~~
- ~~border-style-color~~
- border-bottom-color
- border-left-color
- border-right-color

Q. Write code for



using document-level style sheet.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 { text-align: center;
     border-style: dashed;
     border-color: red;
     border-width: thick;
   }
p { text-align-center;
    border-style-dotted;
    border-top-color: red;
    border-left-color: red;
    border-right-color: red;
    border-bottom-color: blue;
    border-bottom-style: dashed;
  }
</style>
<body>
<h1> Web Programming </h1>
<p> Text </p>
</body>
</html>
```

Margins & Padding

Padding - space b/w content & of element & border
Margin - space b/w border of element & neighbour element

- When there is no border, margin & padding is space b/w content of an element & its neighbour.
- When element has a background colour, it will extend into padding but not margin

Properties (Margin)

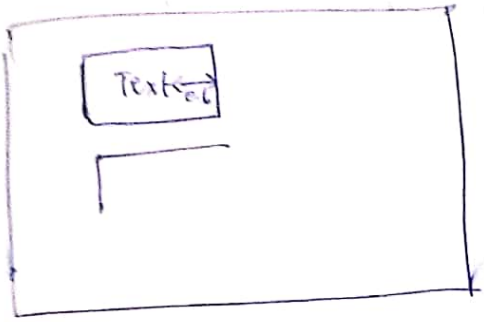
margin - left
margin - right
margin - top
margin - bottom

E.g.

```
p.para1 { margin: 0.2in;
          padding: 0.6 in;
          border-style: solid;
        }
p.para2 { margin: 0.8in;
          padding: 0.2in;
          border-style: solid;
        }
```

```
<p class="para1">Text</p>
<p class="para2">Text1</p>
```

O.P.



Properties (Padding)

padding : (value)
padding - left
padding - right
padding - top
padding - bottom

* border-style : solid → top, dashed → right, dotted → bottom, double → left

border-style : solid → top, dashed → right & left, dotted → bottom

border-style : solid ; dashed → top & bottom ; right & left

Size

1 pt = $\frac{1}{72}$ inches

1 pE = 12 pts

1 cm = 16 pixels

Background image

body { background-image : url(Ex. gif); }

1. repeat
2. no_repeat
3. repeat_x
4. repeat_y

Background_position : top
bottom
left
right
center

Span of div

Q. This is a special language

<p> This is a special
 language.

Pseudo-classes

Special effects can be applied to selectors like hover & focus

Element name : pseudoclass name


```
<style>
  input: focus { color : "green" ; }
  input : hover { color : "red" ; }
</style>
```

Conflict resolution

Resolved by precedence of 3 levels of style sheets

↓

Inline > Document > External

There are several different origins of specifications of property values.

i) Author
ii) User
iii) Browser

Property value specifications.

p. special { font_style: !important; font_size: 10pt }

Different origins has different precedences

1. Important declarations with user origin
2. Important declaration with author origin
3. Normal declaration with author origin
4. Normal declaration with user origin
5. Any declaration with browser origin

If there conflicts even after above sort, then do sort by simplicity.

1. id selectors
2. class & pseudo-class selectors
3. Contexual selectors
4. Universal selectors.

If the style conflicts, it will be resolved by giving precedence to recently seen specification.

Test 1 portions ends here.

Q Create an XHTML document that contains the 3 lines of text from a newspaper story. Every ~~word~~ in the text must be green, noun must be verb blue, preposition must be yellow.

Q Create XHTML document to display table of basketball scores in which team names are the respective color. The score of winning team should appear larger & in different font than the losing team.