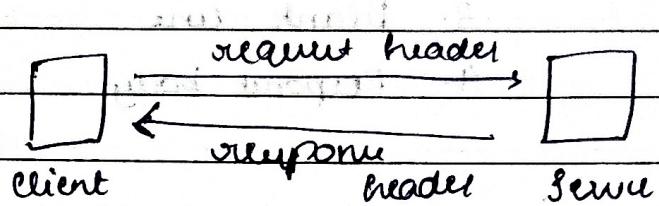


Unit-1

Fundamentals of Web

- * HTTP response and request header
- * HTTP - to communicate b/w client & server



HTTP - method domain part HTTP version
q.e.

e.g.: GET /abc.html HTTP/1.1

Fieldname : Field value

Accept : text/html

Host : hostname

If modified since : date

Content length : 200

Blank line

4 characteristics of header fields

- 1) General or Request
- 2) Response
- 3) Priority

Date.....

Response header

General form

1. Status line
2. Response header fields
3. Blank line
4. Response body

1. Status line

HTTP version 3 digit status code

Message text
Expansion
of status code

Eg:- HTTP/1.1 200 OK

Date :

Name of Server:

Last modified:

Content length:

Content type: text/plain

Blank line → follows body part

* well formed
Consistent
compatible }
 } xhtml

Date:

Web Programmer's Tool Box

→ XHTMl

- HTML and XHTML are only used to display data/output
- XML is used to organising & transferring data
- JS is embedded in XHTML
- JS is high level interpreted
- JS is both client & server side

HTML → XHTML

A stricter, XML-Based version of HTML. It enforces well-formed code (e.g. all tags must be closed, lowercase, properly nested) which makes it more consistent and compatible with XML tools.

→ CSS

A styling language used to control how web pages look. It defines layout, colors, fonts, spacing, responsiveness, and overall design.

→ JS

A programming language for the web that makes

Date:

pages interactive and dynamic (e.g. form validation, animations, real-time updates without reloading)

→ PHP

A server-side scripting language mainly used for back-end development. It processes data requests with database, and generates dynamic web content.

Aspect	HTML	XHTML
- well formed	not strict → allows mixing tags, unclosed elements, uppercase/lowercase	strict → requires proper nesting, lowercase tags, all elements closed, attr values quoted
- consistent	less consistent → different b/w - where may interpret std rules make code portable across browsers	Highly consistent → strict - by code differently
- compatible	compatible with most browsers but may break with XML parsers	compatible with both browsers and XML parsers

Date:.....

→ Rail

- Rail is a web application framework written in the Ruby language. It follows the MVC architecture and is designed to make web development easier & faster.

→ AJAX

- AJAX is a technique used in web development to create fast & dynamic web applications.
- Instead of reloading the whole webpage when data is needed from the server, AJAX follows partial page updates.

→ Perl

- Perl is a general purpose high level language.
- It is evaluated by an interpreter rather than being compiled into machine code.
- It is dynamically typed, meaning no need to declare variable type repeatedly.
- Perl syntax is similar that to C.

→ Ruby

- It is an object oriented interpreted scripting language.

Date:

- operator is very an syntactic mechanism to specify method of corresponding operation.
- no need of declaring variable because of dynamic type

list

1) Ordered

1. —

2. —

3. —

Unordered

• —

• —

• —

Obj

html \rightarrow dt
= dd

PHP =

CSS =

ordered

1. TSE

• ISB

2. CSE

• CSB

3. MCA

• MCP

 TSE

 CSB

 TSE

 MCA

 CSB

<ol type="a">

ul style="type: circle"

Date:

Hot drinks

- Coffee
- Tea
 - Black Tea
 - Green Tea
- ~~Milk~~ coffee

<ok> order hot = "hot drinks">

 coffee

 tea

 Hot drinks

Coffee

 coffee

 Tea

 Black Tea

 Green Tea

 coffee

Student information

Tables

First name	Last name	Age
ABC	abc	20
XYZ	xyz	40
PQR	pqr	10

Date:.....

<table>

<caption>

student information

</caption>

<thead><tr> <th> <td> First Name </td> <th> </th> <th> </th>
<th> <td> <td> Last name </td> <th> </th> <th> </th>
<th> <td> <td> Age <td> <th> </th> <th> </th>

<tr>

<td> ABC </td>

<td> abc </td>

<td> 20 </td>

</tr>

colspan

Name	Telephone
abc	123

rowspan

Name	abc
Telephone	123 456

<table>

<tr>

1st

<th> Name </th>

<th> colspan="2" > Telephone </th>

</tr>

<tr>

2nd

<td> abc </td>

<td> 123 </td>

<td> 456 </td>

</tr>

Date:

</table>

• <table>

<tr>

<th> name </th>
<td> abc </td>
<th> rowspan="2" > telephone </th>
<td> 123 </td>
<td> 456 </td>

</tr>

$$x_2^3 + y_1^2$$

$$2 <\sub> 2 <\sub> <\sup> 3 <\sup> +$$

Q) what are character entities? give example

align & valign

align = "left" center right

left	center	right
bottom Default	Top	Bottom

Imp

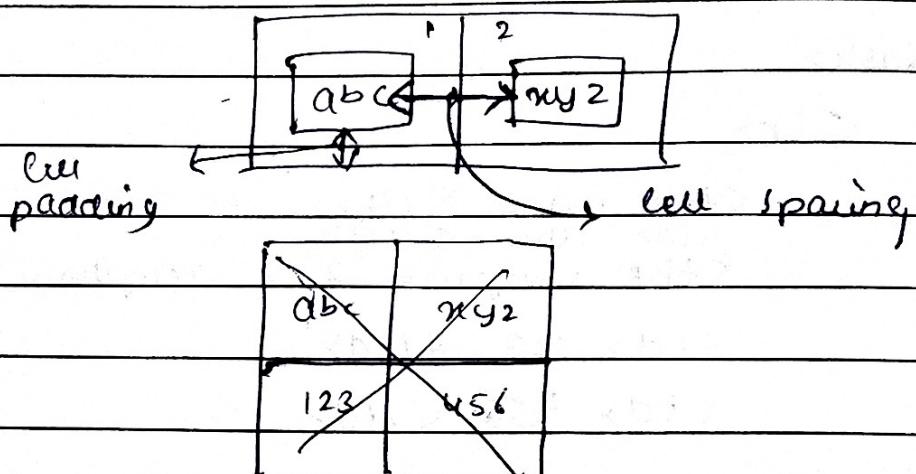
cell padding & cell spacing



spacing b/w content of a cell & lines written
q. a cell

Date:

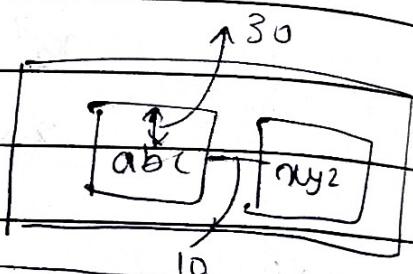
* Cell spacing - distance b/w cells in a table



< table

cellpadding = "30"

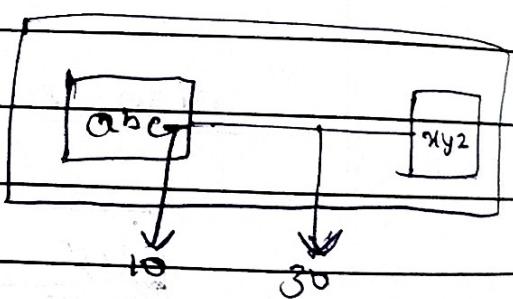
cellspacing = "10" >



< table

cellpadding = "10"

cellspacing = "30" >



Form elements

* Easy way to communicate with web page

< form >

Date:

Name

password

About yourself
(Briefly)

Gender male female

Hobbies reading playing

Subject preferred

Math

Science

Send data

Clear

Code

<form>

<input type="text" name="name">

<input type="text" name="name">

<input type="password" name="password">

<text area name="about" rows="4" cols="6">

<input type="radio" name="gender" value="male">

<input type="radio" name="gender" value="female">

<input type="checkbox" name="hobby" value="reading">

<input type="checkbox" name="hobby" value="playing">

Date:

CSS (Cascading Style Sheets)

3 levels (lowest to highest)

- 1) inline level → apply to the content of single element
- 2) document level → apply to the body of a whole document
- 3) external level → " " " bodies of any no of documents

`<h2 style="font-size: 20px;">` we see is in class `<h2>`

inline → have precedence over
document
external → have precedence over

Style Specification Format

① inline level

`style = "property1: value1; property2: value2; ...
... propertyN: valueN;"`

e.g:-

`<h2 style="font-size: 20px; font-color: red; font-`

optional

Date:

face : Arial" > I am in my class </h2>

2) document level

<head> <title> </title>

<style type = "text/css">

rule-set → set of rules

</style>

form } </head>

Selector { property1 : value1 ; property2 : value2 ; ...
propertyN : valueN }

eg:-

h2 { font-color : red ; font-size : 20px }

Comments

/* ----- */

3) External level

<head>

<link rel = "stylesheet"

type = "text/css"

href = "http:// / / style1.css"

</head>

Selectors form

① Simple selector form

```
h1 { font-size: 20pt; }
h2, h3 { font-size: 18pt; }
```

② class selector → diff occurrences of same tag to have different styles specifications

<style>

```
p.normal { font-size: 20pt; color: blue; }
```

```
p.warning { font-size: 30pt; color: red; }
```

</style>

<body>

```
<p class="normal"> - - -
```

```
- - -
```

```
- - -
```

</p>

```
<p class="warning"> - - -
```

```
- - -
```

```
- - -
```

</p>

3) generic selector

Date:.....

→ styles apply to more than one kind of tag

• `.sale { font-size: 10pt; color: green; }`

`<h3 class = "sale">`

This is h3 heading `</h3>`

`<p class = "sale">`

This is a para

`</p>`

4) **id selector**

→ style applies to one specific element

`# specific id { properties-value list }`

`# section14 { font-size: 40pt; }`

`<h2 id = "Section24"> I am h2 header</h2>`

5) **universal selector**

→ applies to all elements in whole document

↳ style `* { properties-value list }`

`* { color: red; }`

Date:

6

Pseudo classes

Style applied when something happens

⑤ 9

<style type="text/css">

input : hover { color: red; }

input : focus { color: green; }

</style>

=

<body>

<input type="text">

<h1>

=

<style>

p { text-align: center; color: red; }

</style>

<body>

=

<p> Every para will be affected </p>

<p id="flower"> This too </p>

<p> Sun this </p>

</body>

Date:

<html>

=

<style type = "text/css">

* centered { text-align: center; color: red; }

<body>

<h1 class = "center" > This ^{is} an ^{h1} centered red
page </h1>

<p class = "center" > This ^{is} a centered
red </p>

Property value forms

1. Keyword property values :- large, medium, small, small, small
2. Number property values :- integer, sequence of numbers, +ve, -ve, decimal
Ex:- +20, -20.5
3. length values :- length values are NO values immediately followed by unique name 2 characters
Ex:- 20pt