

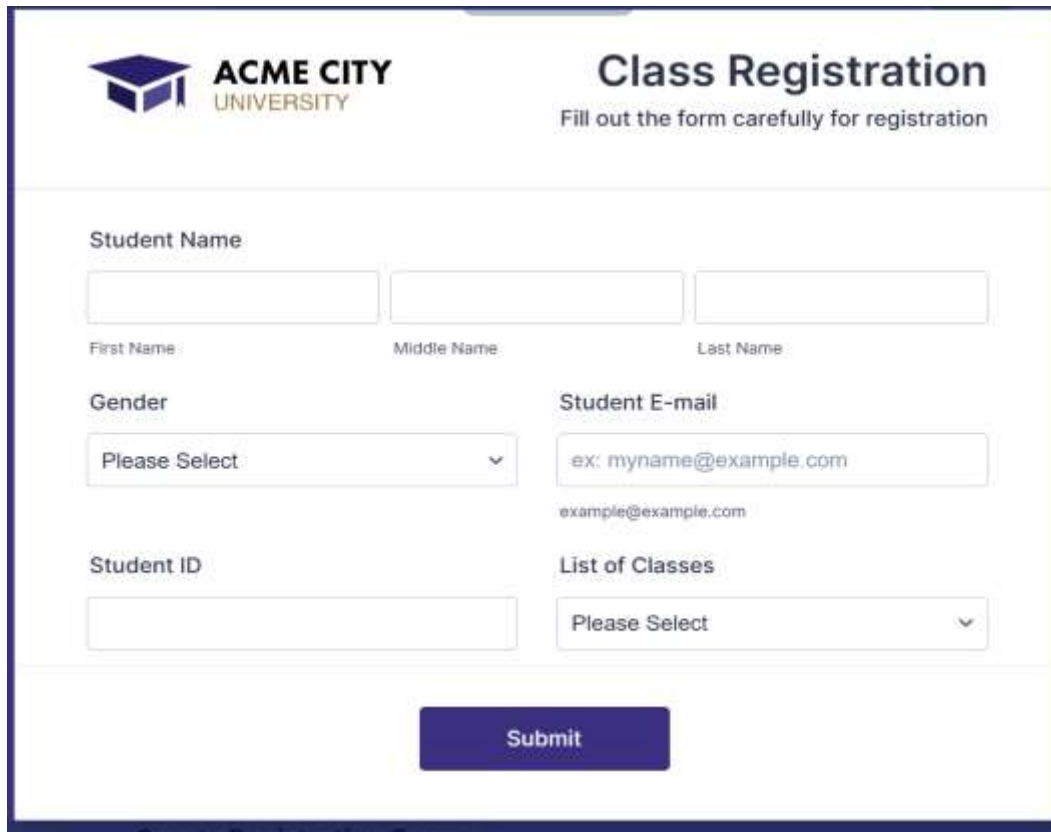
Web Application Development

Value added course

Objectives

1. Understand the fundamental concepts of **web application development**.

EX: online forms, shopping carts, video streaming, social media, games, and e-mail.



ACME CITY UNIVERSITY **Class Registration**
Fill out the form carefully for registration

Student Name

First Name Middle Name Last Name

Gender: Please Select

Student E-mail: ex: myname@example.com

Student ID

List of Classes: Please Select

Submit

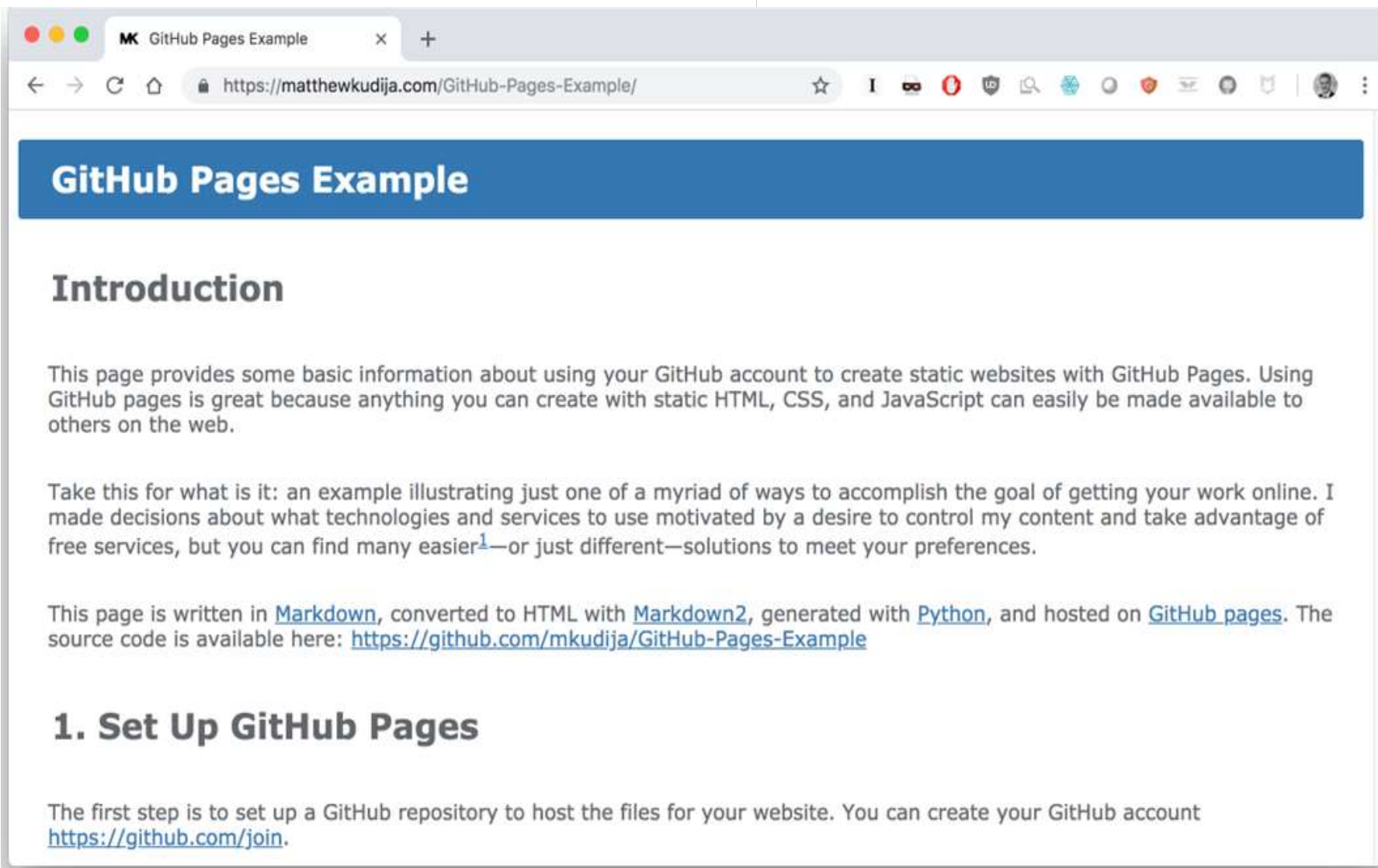


2. Apply basic Front-End Technologies to create static and dynamic web pages.

Static Web Page	Dynamic Web Page
The content and layout of a web page is fixed	The content and layout may change during run time
Static Web pages never use databases	Databases is used to generate dynamic content through queries
Static web pages directly run on the browser and do not require any server side application program	Dynamic web pages runs on the server side application programs and displays the results
Static Web pages are easy to develop	Dynamic web page development requires programming skills

Ex: HTML, CSS, Java script

Ex: JSP, ASP, PHP



Syllabus

Chapter-1: Introduction WWW,URL,HTTP,HTTPS,FTP,MIME, Basic Web Technologies, Why HTML, What is HTML, HTML Page Structure, HTML Elements Categories, HTML Block and Inline Elements, HTML Element Attributes,Comments, Meta tag In HTML, Sectioning Elements, Grouping Elements in HTML, Link Element (<a>.....), HTML Formatting Elements or Text-level semantic elements , HTML Entities or character, HTML Global Attributes, HTML Table Element, HTML Forms-Attributes, Elements, Input Type and value attribute, Embedded elements

Chapter-2: Introduction to CSS, Selectors, Cascading Order, Typography, Box Model, Layouts, Responsive Web Design, Adding Awesome Effects, CSS Security- CSS Accessibility and Best Practices

Chapter-3: Bootstrap5- Bootstrap – Introduction, Page Layout, Content Styling, Content Structuring, Navigation, Form.

Chapter-4: Java Script-Introduction To Java Script, Identifiers, Data Types, Operators, Statements and Expressions, Loops, Functions, Classes, Event Handling, Objects, Iterables, Asynchronous Programming, Modular Programming, JavaScript – Security

Unit - I

Introduction

Web technology is the establishment and use of mechanisms that make it possible for different computers to communicate and share resources.

Examples of web technologies

1. **Markup Languages** including HTML, CSS, XML, CGI and HTTP
2. **Programming Languages** and their technologies, helps in creating applications for web. Ex: perl, c#, Java and VB.Net
3. **Web servers** and server technologies, facilitate request handling on a network.
4. **Databases**, which are for data and information storage on network and
5. **Business applications** for specific executions of tasks on a network.

History of the web

- The internet is a n/w of n/ws. – users can get information from any computer if they have permissions.
- First net is ARPA (Advanced Research Project Agency) of U.S. govt. in 1969.
- In 1989 www was developed by Tim Berners-Lee in the laboratory CERN.
- The main aim of web is to facilitate communication among the members who were located in several countries

Web system Architecture

- The internet is a global networking scheme, TCP/IP assigns and uses unique addresses to communicate.
- www is a n/w of computers used to exchange text, graphics and multimedia content using standard protocols.
- For this, a) web servers – special computers set up for the distant purpose i.e. delivering content to others to access.
- B) web client – personal computers, mobile devices etc. access the server content through a browser.
- C) Browser – a specialized application for displaying web content. Ex: google chrome, Mozilla firefox etc.

- **WWW:** The World Wide Web -- also known as the web, WWW or W3 -- refers to all the public websites or pages that users can access on their local computers and other devices through the internet. These pages and documents are interconnected by means of **hyperlinks** that users click on for information.
- There are so many fields in which we can see the use of the World Wide Web. There are some real-life services or resources we are using.
- 10 uses of the World Wide Web are

- **Website:** On a website we put multiple web resources at web pages with a particular theme by using a single domain name, which can be accessed from anywhere. For example, www.pvpsiddhartha.ac.in, www.wikipedia.org, etc.
- **Email:** Email is known as electronic mail which is used to communicate with others by the message sent via the Internet. Example: gmail.com, yahoo.com.

- **Online Study Materials:** We can find study material online for learning. E.g. , [Udemy](#), [Khan Academy](#), etc.
- **Web Services:** It is a service offered by an electronic device to another electronic device.
- **Downloadable Video:** There are many examples [Youtube](#), Max player, [Netflix](#), etc.
- **Online Music:** Online mp3 files are available to download and to listen to.

- **E-Commerce:** Online shopping is a part of our life today. E.g. [Amazon](#), [Flipkart](#), Walmart, etc.
- **News Sectors:** We can see many news channels working online.
- **Online Payment:** Paying online is possible because of WWW and the Internet.
- **Online Banking:** Almost every bank provides online banking to its customers for balance check, transfer funds, receive funds within one's bank account.

WWW



Collection of files stored on thousands of servers in a form of web pages.



VIDEO



ANIMATED
GRAPHICS



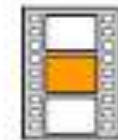
AUDIO



MAPS



TEXT



PHOTOGRAPHY

Web page: Is a kind of HTML document stored on the web server.

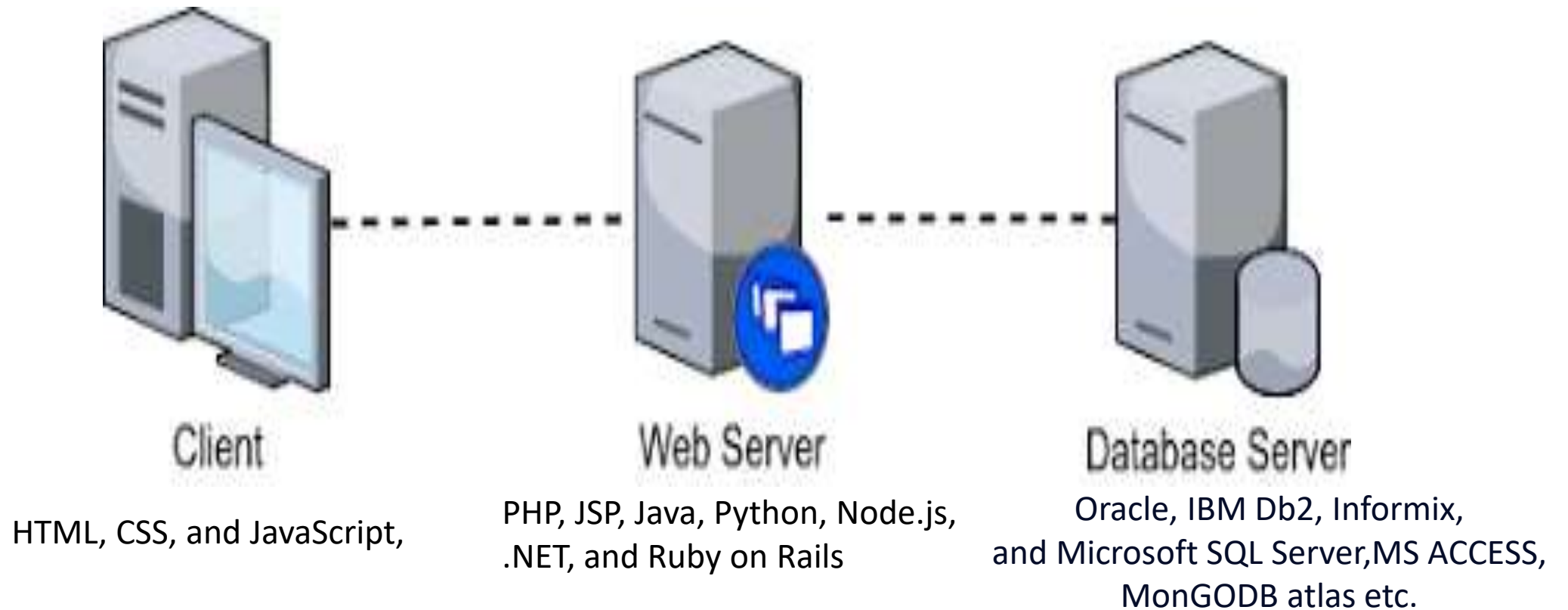
Web site: Collection of web pages belonging to particular organization.

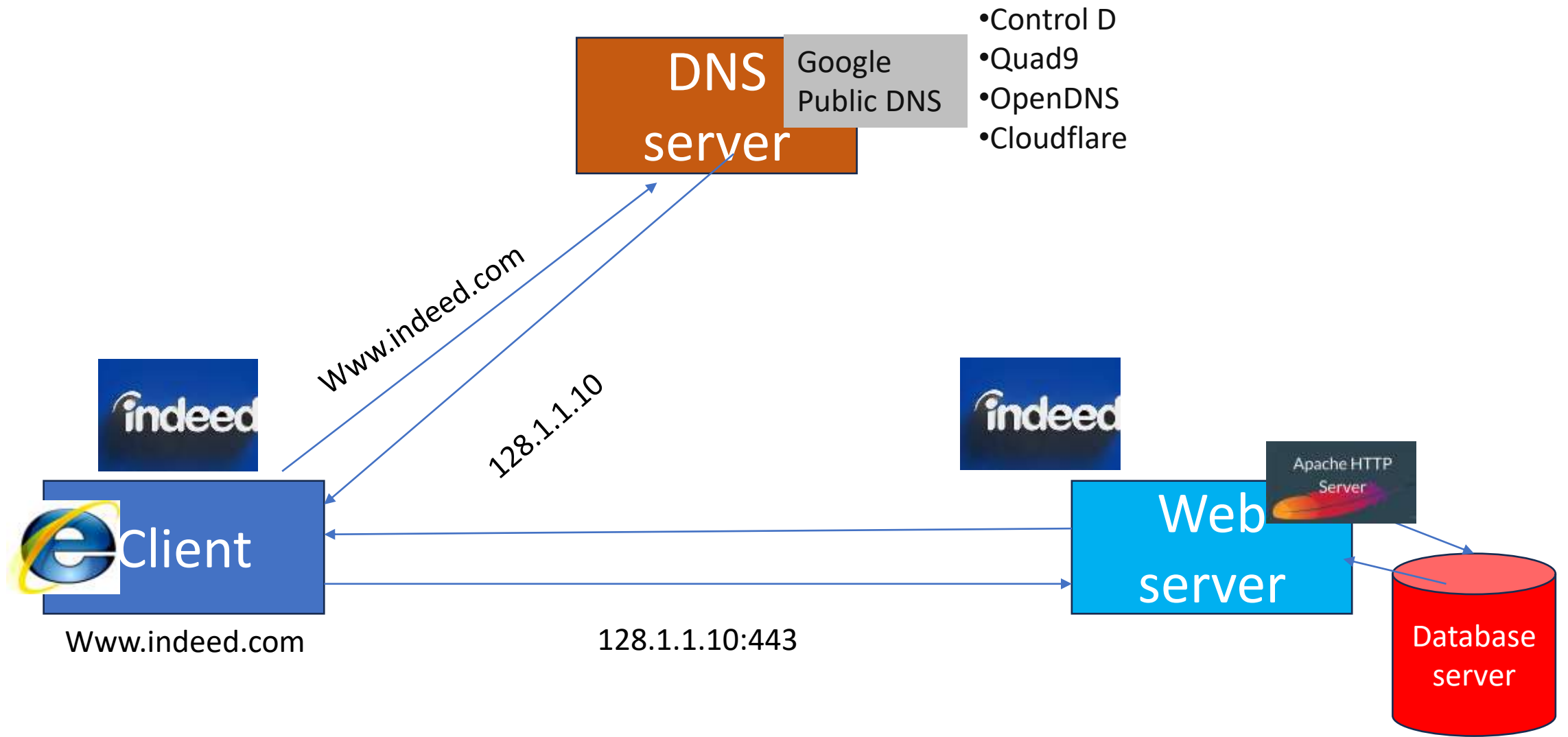
URL: Uniform Resource Locator: It is the address of a resource, which can be a specific webpage or a file, on the internet. It is also *known as web address* when it is used with http. It was created in 1994 by Tim Berners-Lee.

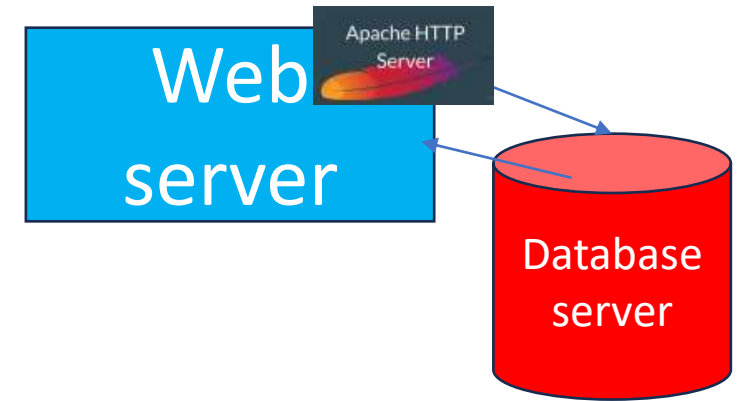
URL is a specific character string that is used to access data from the World Wide Web.

Working of WWW:

3- TIER WEB ARCHITECTURE





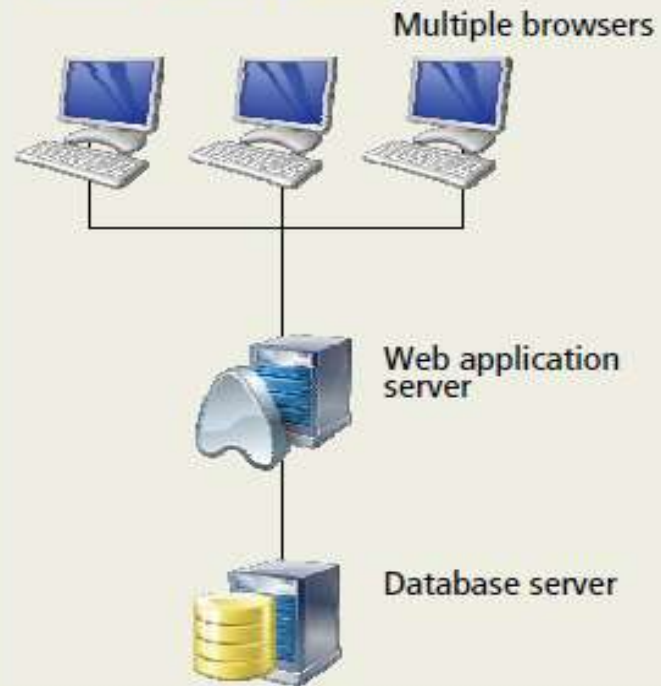


Configuration Scenarios

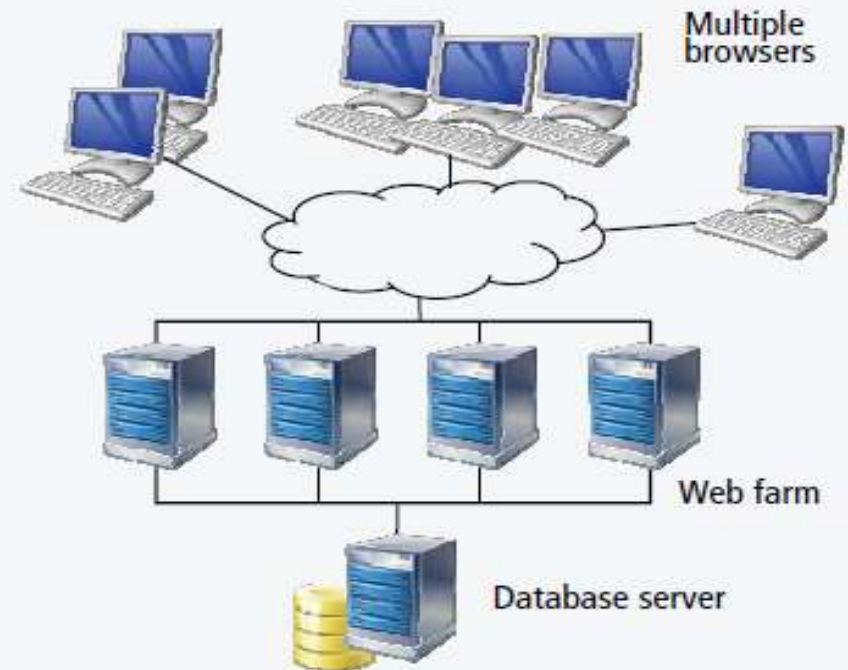
Single User



Small Enterprise



Geographically Dispersed Enterprise





InterNet and World Wide Web

Differences between Internet and WWW

Internet	WWW
The Internet allows you to link your computer to any other computer on the planet.	The World Wide Web is a collection of information accessible through the Internet.
The Internet is a worldwide network of interconnected computer networks that connect devices using the TCP/IP protocol.	The World Wide Web refers to HTML-formatted online material that may be accessed using the HTTP protocol.
The Internet can be compared to a large bookstore.	The web can be considered as a store with a collection of books.
Internet is superset of WWW.	The World Wide Web is a subset of the Internet.
It first appeared in the late 1960s.	Tim Berners-Lee, an English scientist, created the World Wide Web in 1989.
The Internet is mostly based on hardware.	In comparison to the Internet, the WWW is more software-oriented.

URL: Uniform Resource Locator

- It is the address of a resource, which can be a specific webpage or a file, on the internet.
- It is also *known as web address* when it is used with http. It was created in 1994 by Tim Berners-Lee.
- URL is a specific character string that is used to access data from the World Wide Web

Every URL contains the following information:

- The scheme name or protocol.
- A colon, two slashes.
- A host, normally called a domain name but sometimes as a literal IP address.
- A colon followed by a port number. (optional). Port80 default port number for webserver

EX: <http://www.pvpsiddhartha.ac.in/full-form>

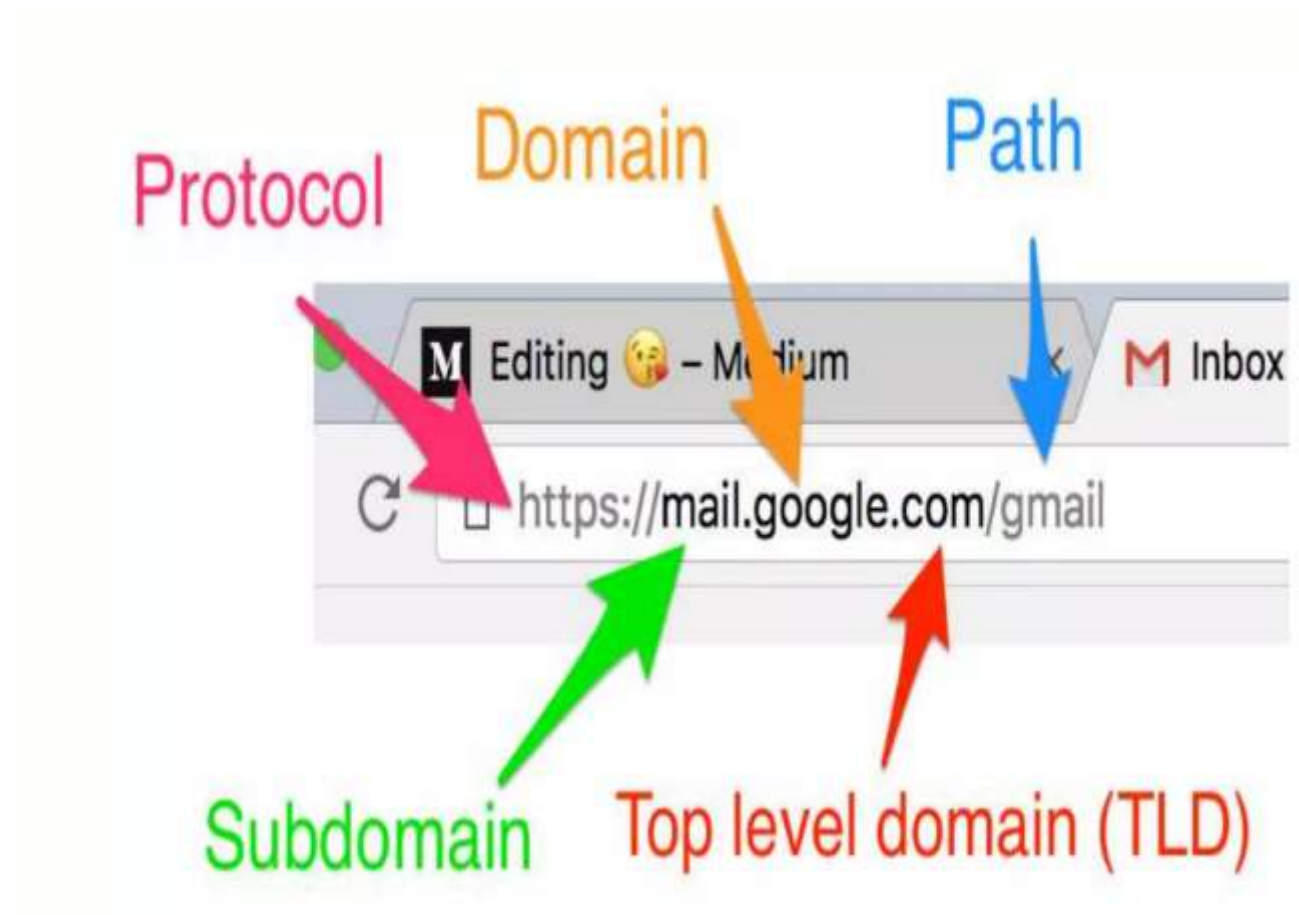
- protocol:** http:// or https:// **hypertext text protocol secure (HTTPS), FTP, Gopher**
- host or domain:** www.pvpsiddhartha.ac.in
- Path of the resource:** File or location on the webserver. /full-form

Server not found“ :

If the URL does not contain a valid server, a browser may display a "Server not found" error

"404 error".

if the path in the URL is incorrect, the browser may display a



- Domain name is a combination of IP address(4 bytes) and Port id or Service id (4 digits).
- Note:

Apache Tomcat Web Server default port number is 8080. Even Oracle has same port id and it has high preference. So, we need to change port id of Apache to access it over Oracle.

`http://localhost:8080/manager/text/deploy?config=file:/path/context.xml`

There are two types of URL:

- Absolute URL
- Relative URL

Absolute URL: This type of URL contains both the domain name and directory/page path.

An absolute URL gives complete location information.

It begins with a protocol like “http://” and continues, including every detail.

An absolute URL typically comes with the following syntax.

`protocol://domain/path`

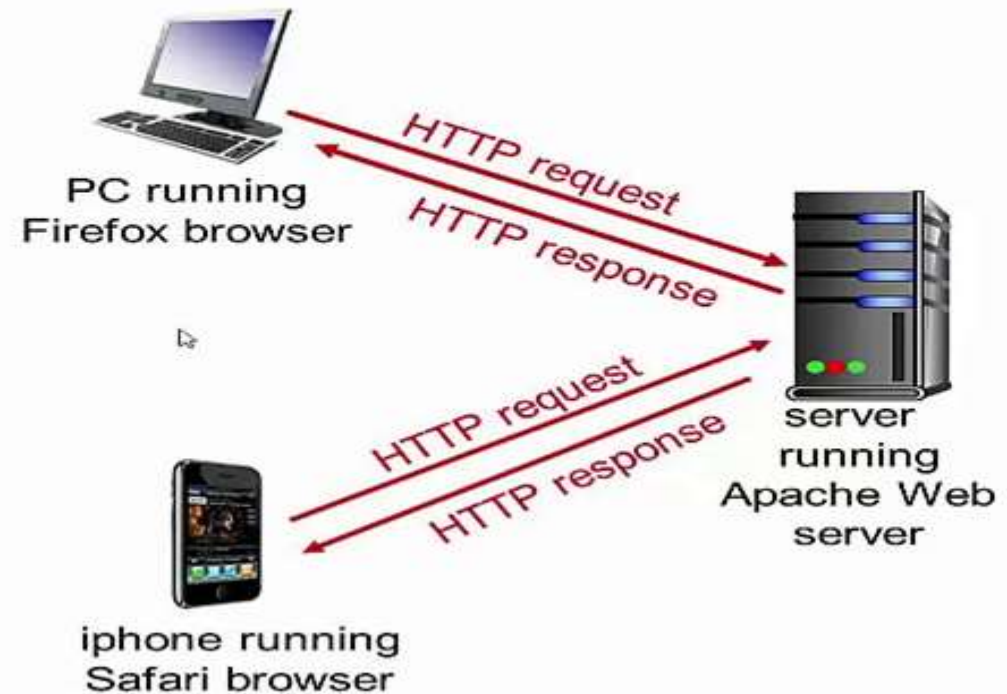
Relative URL: This type of URL contains the path excluding the domain name. Relative means “in relation to”, and a relative URL tells a URL location on terms of the current location.

``

Hypertext Transfer Protocol

HTTP: hypertext transfer protocol

- ❖ Web's application layer protocol
- ❖ client/server model
 - **client:** browser that requests, receives, (using HTTP protocol) and "displays" Web objects
 - **server:** Web server sends (using HTTP protocol) objects in response to requests



HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web

Basic Features

There are three basic features that make HTTP a simple but powerful protocol:

- **HTTP is connectionless:**
- **HTTP is media independent:**
- **HTTP is stateless:** Command is request is executed independently, without any knowledge of the requests that were executed before it

HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges

Hyper Text Transfer Protocol

- It is a **platform independent** protocol used to transfer the information on LAN and WWW.
- It is a **request/response** protocol.
- It is most commonly used protocol for web browser and web server communication.

Hyper Text Transfer Protocol Secure

- Hypertext Transfer Protocol Secure (HTTPS) is a protocol that secures communication and data transfer between a user's web browser and a website. HTTPS is the secure version of [HTTP](#).
- It uses [encryption](#) for [secure communication](#) over a [computer network](#), and is widely used on the [Internet](#).

File Transfer Protocol

- File transfer protocol (FTP) is a way to download, upload, and transfer files from one location to another on the Internet and between computer systems. FTP enables the transfer of files back and forth between computers or through the cloud. Users require an Internet connection in order to execute FTP transfers.

Multipurpose Internet Mail Extensions

- MIME (Multipurpose Internet Mail Extensions) is an extension of the original Simple Mail Transport Protocol (SMTP) email protocol. It lets users exchange different kinds of data files, including audio, video, images and application programs, over email.

Basic Web Technologies

- Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the Internet.
- **Web Technology can be classified into the following sections:**
- **World Wide Web (WWW):** The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP).
- **Web Browser:** The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and requests to the server for web documents and services.
- **Web Server:** Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).

Continue..

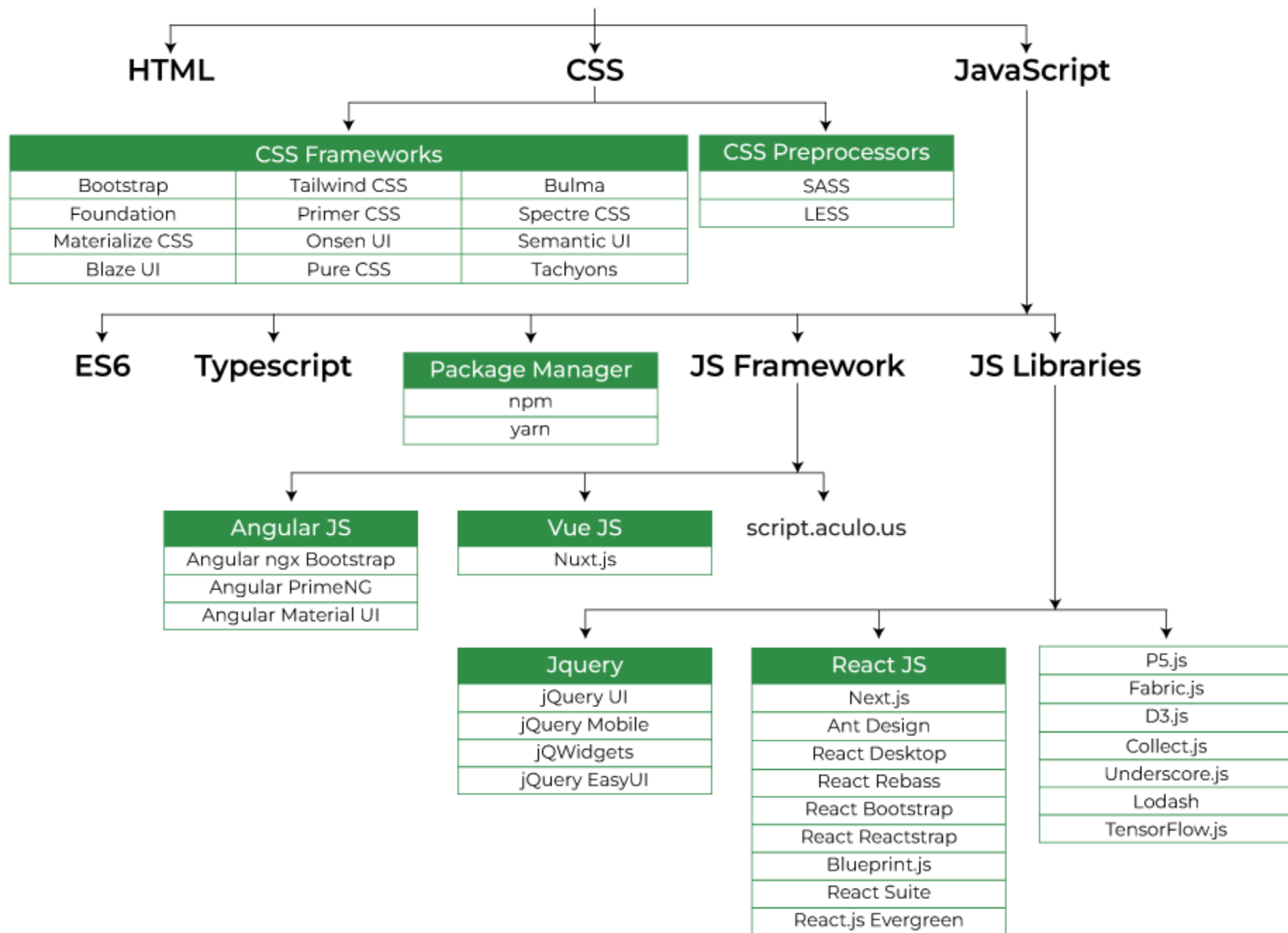
- **Web Pages:** A webpage is a digital document that is linked to the World Wide Web and viewable by anyone connected to the internet has a web browser.
- **Web Development:** Web development refers to the building, creating, and maintaining of websites.
- It includes aspects such as
 - web design,
 - web publishing,
 - web programming, and
 - database management.

It is the creation of an application that works over the internet i.e. websites.

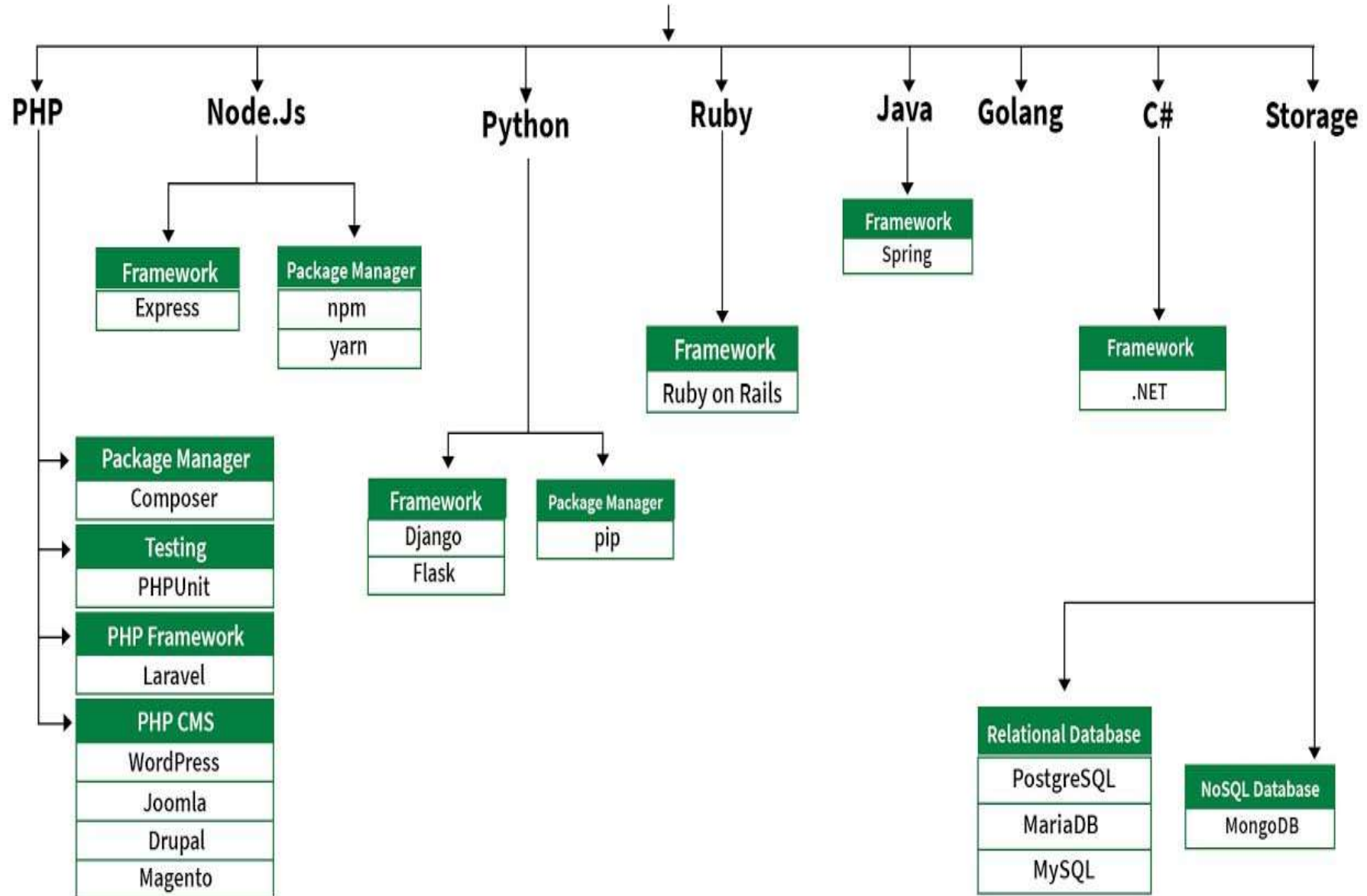
Classification of Web Development

- **Frontend Development:** The part of a website that the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.
- **Backend Development:** Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

Frontend



Backend



Frontend Languages

- The front-end portion is built by using some languages which are discussed below:
- **HTML**
- **CSS**
- **JavaScript**
- **AJAX**
- There are many other languages through which one can do front-end development depending upon the framework for example *Flutter* uses *Dart*, *React* uses *JavaScript* and *Django* uses *Python*, and much more.

Backend Languages:

- The back end portion is built by using some languages which are discussed below:
- **PHP**
- **Node.js**
- **Python**
- **Ruby**
- **Java**
- **JavaScript**
- **Golang**
- **C#**
- **DBMS**

HTML

- It is a **Markup** language.
- Used to describe web documents(web pages)
- Derived from SGML(Standard Generalization Markup Language).
- HTML stands for **Hyper Text Markup Language**
- A **hyper text** represents a link which opens text in some or other web page by clicking on it.
- A **markup language** is a set of markup tags which provides a way to describe structure of text-based information.
- HTML documents are described by HTML tags/elements.

Continue..

- HTML is used for creating web pages and web applications.
- **Hyper Text:** Hyper Text simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hyper Text is a way to link two or more web pages (HTML documents) with each other.

Continue..

- **Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.
- **Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

HTML Tags

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension
- HTML makes it possible to create static pages with text, headings, tables, lists, images, links, and so on.
- Html used predefined tags and attributes to tell the browser how to display content, means in which format, style, font size, and images to display.
- Html is a case insensitive language.
- Tag types:
 - 1.Paired Tags:** These tags come in pairs. That is they have both opening(< >) and closing(</ >) tags.
 - 2.Empty Tags:** These tags do not require to be closed.



Structure of an HTML Document

- **HEAD:** This contains the information about the HTML document.
For Example, the Title of the page, version of HTML, Meta Data, etc.
- **BODY:** This contains everything you want to display on the Web Page.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Page Title</title>
  </head>

  <body>
    <h2>Heading Content</h2>
    <p>Paragraph Content</p>
  </body>

</html>
```

Attributes are additional properties of html tags that define the property of any html tags. i.e. width, height, controls, loops, input, and autoplay.

BASIC HTML TAGS

Heading Tags: Any document starts with a heading.

You can use different sizes for your headings.

HTML also has six levels of headings, which use the elements `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

While displaying any heading, browser adds one line before and one line after that heading.

Paragraph Tag: The tag offers a way to structure your text into different paragraphs.

Each paragraph of text should go in between an opening `<p>` and a closing tag `</p>`.

Line Break Tag: Whenever you use `<\br>` the element, anything following it starts from the next line.

This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

Centering Content: Use `<center>` tag to centering contents

Write an HTML code to display your CV on a web page

Basic HTML tags

Structural Tags

`<HTML>`

These tags enclose the entire Web page document.

`</HTML>`

`<HEAD>`

These tags enclose the Head part of the document

`</HEAD>`

`<TITLE>`

These tags enclose the title of the document. This text appears in the title bar in the browser and on the bookmark list if someone bookmarks your web page.

`</TITLE>`

Sample Structure of a Web Site

```
<HTML>  
  <HEAD>  
    <TITLE> John Q. Public's Web Page </TITLE>  
  </HEAD>  
  
  <BODY>  
    This is John Public's Webpage!  
  </BODY>  
</HTML>
```


Header Tags

Header Tags -- Used for marking sections and subsections in a document.

<H1>Header 1 -- Giant-sized and bold </H1>

<H2>Header 2 -- Large and bold </H2>

<H3>Header 3 -- Normal-sized and bold </H3>

<H4>Header 4 -- Small and bold </H4>

<H5>Header 5 -- Very Small and bold </H5>

<H6>Header 6 -- Tiny and bold </H6>

Header Tags (cont.)

H1 = Giant-sized and bold

H2 = Large and bold

H3 = Normal-sized and bold

H4 = Small and bold

H5 = Very Small and bold

H6 = Tiny and bold

Breaking Lines and Paragraphs

- `<P> text </P>`
 - Paragraph tag
 - Most browsers render (process) this with blank lines between each paragraph
 - `
`
 - Line break tag
 - Used when the webmaster wants a carriage return but doesn't want a blank line to follow
-

Example:

```
<p>text a</p>  
<p>text b</p>  
<br>text c  
<br>text d
```



```
text a  
  
text b  
  
text c  
text d
```

Horizontal Rule

The <HR> tag puts a graphical line across the page.

Ex:



Horizontal Rule Attributes:

NOSHADE -- A solid line with no shading

WIDTH="xx%/xx" -- Controls the width of the line. You may specify
either percentage of the width of a page
or actual pixel length

SIZE="xx" -- Controls the height of the line. You need to specify the
dimension in pixels.

ALIGN="left/center/right" -- This allows the line to be aligned to the
left,

right, or center of the page

Text Formatting Tags

Some basic text formatting styles:

Tag	Result
<I> Italics </I>	<i>Italics</i>
 Bold 	Bold
<PRE> Preformatted Text </PRE> Text	Preformatted
 Strong 	Strong
<ADDRESS> Address </ADDRESS>	<i>Address</i>
<CITE> Citations </CITE>	<i>Citations</i>
<CODE> Source Code </CODE>	Source Code

Font modifications

Web creators can also change the way text looks by using the tag

SIZE="number" - changes size of the font; 1=smallest, 7 = largest

Big Small

Big Small

COLOR="color-name" - changes text color

This is red

This is red

FACE="font-name" - changes font

This is the verdana font; this is the chicago font.

This is the verdana font; this is chicago font.

 modifications (cont.)

One can combine font modifications:

Big, Courier & Red

Big, Courier & Red

Big & Courier - Just Big

Big & Courier - Just Big

Lists -- Unordered Lists

Unordered lists:

Item One

Item Two

Item Three

Item Four



- Item One
- Item Two
- Item Three
- Item Four

Unordered List Attributes:

type="disc/circle/square"

- Disc (default) ○ Circle ■ Square

Lists -- Ordered Lists

Ordered (Numbered) Lists:

 Item One

 Item Two

 Item Three

 Item Four



1. Item One

2. Item Two

3. Item Three

4. Item Four

Ordered List Attributes:

type="i/I/a/A/1"

(default)

i = i. Item One

I = I. Item One

a = a. Item One

A = A. Item One

1 = 1. Item One

ii. Item Two

II. Item Two

b. Item Two

B. Item Two

2. Item Two

iii. Item Three

III. Item Three

c. Item Three

C. Item Three

3. Item Three

iv. Item Four

IV. Item Four

d. Item Four

D. Item Four

4. Item Four

start="xx"

- This attribute lets you specify which number/letter will start the list

Lists -- Definition Lists

Definition Lists:

```
<DL>
```

```
  <DT>List Name One
```

```
    <DD>This is where information about List Name One would go</DD>
```

```
</DT>
```

```
<DT>List Name Two
```

```
  <DD>This is where information about List Name Two would go</DD>
```

```
</DT>
```

```
</DL>
```



List Name One

This is where information about List Name One
would go

List Name Two

This is where information about List Name Two
would go

Links

The anchor tag <A> is used to link one document to another or from one part of a document to another part of the same document.

Basic Links:

```
<A HREF="http://www.stanford.edu/">Stanford University</A>
```

Inter-document Links:

```
<A HREF="#spot">Point to 'spot' in this document</A>
```

Defining a point in a document:

```
<A NAME="spot">Spot</A>
```

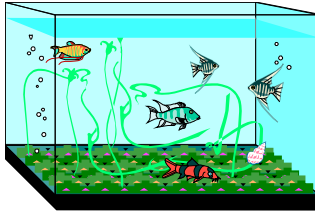
Email links:

```
<A HREF="mailto:someone@somehost.com">Email someone@somehost.com</A>
```

Graphics

To have a graphic appear on a webpage, web designers must to put the tag in with the address where the graphic "lives":

```
<IMG SRC="http://www.someplace.com/images/fish.gif">
```



Graphics attributes:

alt="text": insert a description of the graphic for those who are using browsers that cannot process images (e.g., page readers for the blind)

width="xx/xx%": width in pixels/percentage

height="xx/xx%": height in pixels/percentage

border="xx": pixel length of the border surrounding the image.

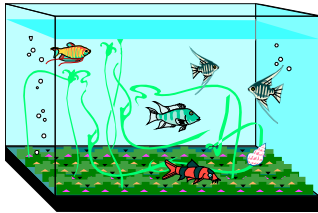
hspace="xx": places a buffer of space horizontally around the image

vspace="xx": places a buffer of space vertically around the image

align="top/middle/bottom/right/left": aligns image in relation to the text (see next 2 slides)

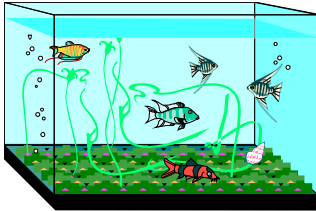
Graphics (cont.)

All about Fish



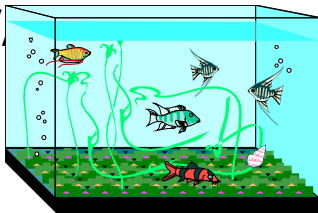
All about Fish

All about Fish



All about Fish

All about Fish

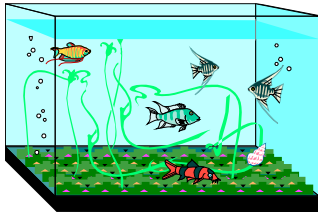


All about Fish

Graphics (cont.)

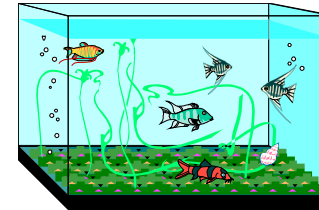
```

```



```

```



Using HTML Tables

Using HTML Tables

- You can use the `<table>...</table>` container to display information in table format.
- You can use the `<table>...</table>` container to control the layout of a page.

Using Tables for Information

Hexadecimal Conversion Chart								
DEC	0	1	2	3	4	5	6	7
HEX	0	1	2	3	4	5	6	7
DEC	8	9	10	11	12	13	14	15
HEX	8	9	A	B	C	D	E	F

Using Tables for Layout

N241 Times

All the News that's (Un)Fit to Print

Study Finds That Death Rate is Highest Among Those Who Don't Breathe

In a New York *Times* report, medical doctors have made the astounding discovery that those who do not breathe have the highest mortality rates of any known group.

Blather, blather, blather. Blather, blather, blather. Blather, blather, blather. Blather, blather, blather. Blather, blather, blather. Blather, blather, blather. Blather, blather, blather. Blather, blather, blather.

Third Debate a 'Good-Time' for all: Brew and Beernuts flow continuously throughout



At last night's presidential debate (officially sponsored by Budweiser), the beer and beernuts flowed freely. Al Gore promised to buy another round for a vote, but was turned back by memories of his time in a Buddhist Temple. George W. was three sheets to the wind before arriving at the debate, having partied all day with ex-fraternity brothers.

Today's Weather:

IT SUCKS!



The <table>...</table> Container

- Defines the beginning and end of a table
- Attributes
 - background
 - bgcolor
 - border
 - Cellspacing: setting spaces between individual cells.
 - Cellpadding: fixing the width present between a cell edge and the content that it contains
 - width
 - height

The `<tr>...</tr>` Container

- Defines a table row
- Structural tag only (no content)
- Attributes
 - align
 - valign
 - bgcolor

The <td>...</td> Container

- Defines a data cell in a table
- Can contain any content, including another, nested table
- Attributes:
 - align
 - valign
 - height
 - background
 - bgcolor
 - height
 - width
 - colspan
 - rowspan

<td colspan ...> Example


```
<tr>  
  <td colspan="2"  
    bgcolor="blue">  
  </td>  
  <td>  
    ...  
  </td>  
</tr>
```

<td rowspan ...> Example

Blue	Red	Red
	Red	Red
Red	Red	Red

```
<tr>
  <td rowspan="2"
    bgcolor="blue">
  </td>
  <td>
  ...
  </td>
</tr>
```

`<th>...</th>` Container

- Defines a header cell
- Acts just like a `<td>...</td>` container, *except* ...
- The text is normally centered vertically and horizontally AND the text is bold
- Attributes are the same as `<td>...</td>`

<th>...</th> Example

Table	Header	Row
Content 1	Content 2	Content 3
Content 4	Content 5	Content 6

<tr>

<th>

Table

</th>

<th>

...

</th>

</tr>

<caption>...</caption> Container

- Provides a summary of the table's purpose
- MUST immediately follow <table>
- Attributes:
 - align
 - summary
 - valign (MSIE only)

<caption>...</caption> Example

Hexadecimal Conversion Chart								
DEC	0	1	2	3	4	5	6	7
HEX	0	1	2	3	4	5	6	7
DEC	8	9	10	11	12	13	14	15
HEX	8	9	A	B	C	D	E	F

Figure 1: Hexadecimal Conversions

<caption align="center"
summary="Long Description goes here"
valign="bottom">
Figure 1: Hexadecimal Conversions
</caption>

HTML Code

```
<TABLE border = "1" >  
  <TR>  
    <TH>Indoor</TH>  
    <TH>Outdoor</TH>  
  </TR>  
  <TR>  
    <TD>Squash</TD>  
    <TD>Cricket</TD>  
  </TR>  
</TABLE>
```

Browser Display

Indoor	Outdoor
Squash	Cricket

<TABLE>	Table (made up of rows)
<TR>	Row (made up of data cells)
<TH>	Heading Data Cell (Can contain paragraphs, images, lists, forms, tables)
<TD>	Data Cell (Can contain paragraphs, images, lists, forms, tables)

<TABLE> Attributes

- BORDER
 - Determines the thickness of the table border
 - Example: `<TABLE BORDER = "2">`
- CELLPADDING
 - Determines the distance between the border of a cell and the contents of the cell
 - Example: `<TABLE CELLPADDING = "3">`
- CELLSPACING
 - Determines the empty spacing between the borders of two adjacent cells
 - Example: `<TABLE CELLSPACING = "1">`

HTML Code

```
<TABLE border = "1" >  
  <TR>  
    <TH>Indoor</TH>  
    <TH>Outdoor</TH>  
  </TR>  
  <TR>  
    <TD>Squash</TD>  
    <TD>Cricket</TD>  
  </TR>  
</TABLE>
```

Browser Display

Indoor	Outdoor
Squash	Cricket

HTML Code

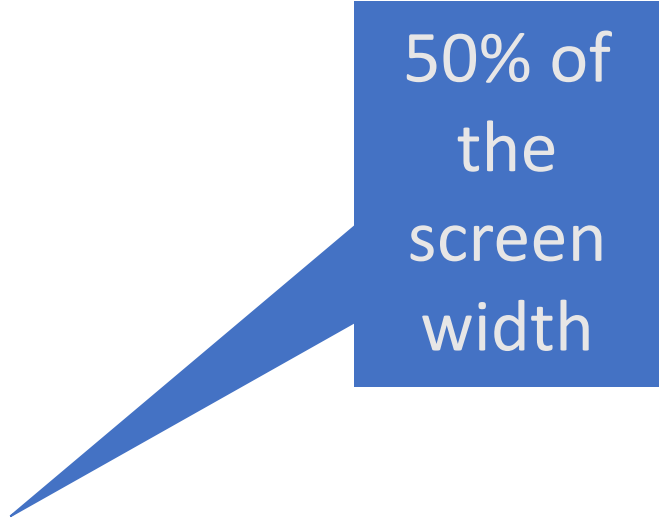
```
<TABLE>
  <TR>
    <TH>Indoor</TH>
    <TH>Outdoor</TH>
  </TR>
  <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```

Browser Display

Indoor	Outdoor
Squash	Cricket

<TABLE>,<TR>,<TH>,<TD> Attributes

- ALIGN
 - Possible values: Center, Left, Right
 - Example: `<TH ALIGN = "center">`
- BGCOLOR
 - Example: `<TD BGCOLOR = "red">`
- WIDTH
 - Example: `<TR WIDTH = "40%">`
- HEIGHT
 - Example: `<TABLE HEIGHT = "200">`



50% of
the
screen
width

<TR> Attributes

- VALIGN
 - Determines the vertical alignment of the contents of all of the cells in a particular row
 - Possible values: Top, Middle, Bottom
 - Example: `<TR VALIGN = "bottom">`

<TH> & <TD> Attributes

- NOWRAP
 - Extend the width of a cell, if necessary, to fit the contents of the cell in a single line
 - Example: `<TD NOWRAP>`
- COLSPAN
 - No. of rows the current cell should extend itself downward
 - Example: `<TD COLSPAN = "2">`
- ROWSPAN
 - The number of columns the current cell should extend itself
 - Example: `<TD ROWSPAN = "5">`
- VALIGN
 - Same as that for <TR>

HTML Code

```
<TABLE border="1" >
  <TR>
    <TH colspan="2">
      Indoor Outdoor
    </TH>
  </TR>
  <TR>
    <TD>Squash</TD>
    <TD>Cricket</TD>
  </TR>
</TABLE>
```

Browser Display

Indoor Outdoor	
Squash	Cricket

HTML Code

```
<TABLE border = "1" >  
<CAPTION>  
  My favorite sports  
</CAPTION>  
<TR>  
  <TD>Squash</TD>  
  <TD>Cricket</TD>  
</TR>  
</TABLE>
```

Browser Display

My favorite sports

Squash	Cricket
--------	---------

Must be placed
immediately after
the<TABLE> tag

Working With Frames

- HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document independently.
- A collection of frames in the browser window is known as a frameset.
- The window is divided into frames in a similar way the tables are organized: into rows and columns.
- *The <frame> tag is no longer recommended as it is not supported by HTML5. Instead of using this tag, we can use the <iframe> or <div> with CSS to achieve the similar effects.*

```
<frameset rows="50%,50%"> <frame  
name="top" src="link/to/frame1" />  
<frame name="bottom"  
src="link/to/frame2" /> </frameset>
```

Where the rows attribute of frameset defines the division of the window into horizontal sections. In this case, the window is divided into two rows, each taking up 50% of the available height.



- Creating Frames in HTML
- To make frames on a page we use <frameset> tag instead of <body> tag.
- The <frameset> tag defines how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames.
- Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.


```
<!DOCTYPE html> <html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
<frame name="top" src="/html/top_frame.htm" />
<frame name="main" src="/html/main_frame.htm" />
<frame name="bottom" src="/html/bottom_frame.htm" />
<noframes>
<body> Your browser does not support frames.
</body>
</noframes>
</frameset> </html>
```

Working With Frames

- Frames with columns

```
<html>
  <head>
    <title> Frames Example </title>
  </head>
  <frameset cols="30%,*">
    <frame src="one.html">
    <frame src="image1.html">
  </frameset>
</html>
```

Here we have created two columns to fill with two frames.

The first frame is 30% wide and will contain the navigation menubar implemented by one.html file. The second column fills in remaining space and will contain the main part of the page and it is implemented by image1.html file

- Frames with rows:

```
<html>
```

```
    <head>
```

```
        <title> Frames Example </title>
```

```
    </head>
```

```
    <frameset rows="20%,80%">
```

```
    <frame src="one.html">
```

```
    <frame src="image1.html">
```

```
    </frameset>
```

```
</html>
```

- To create frames **without border** between frames:

```
<html>
```

```
    <head>
```

```
        <title> Frames Example </title>
```

```
    </head>
```

```
    <frameset rows="30%,*" frameborder="0">
```

```
        <frame src="one.html">
```

```
        <frame src="image1.html">
```

```
    </frameset>
```

```
</html>
```

- To have multiple rows and columns in frames:

```
<html>
```

```
  <head>
```

```
    <title> Multiple Frames Example </title>
```

```
  </head>
```

```
  <frameset cols="20%,20%,20%,*">
```

```
    <frame src="1.html">
```

```
    <frame src="2.html">
```

```
    <frame src="image.html">
```

```
      <frameset rows="50%,*">
```

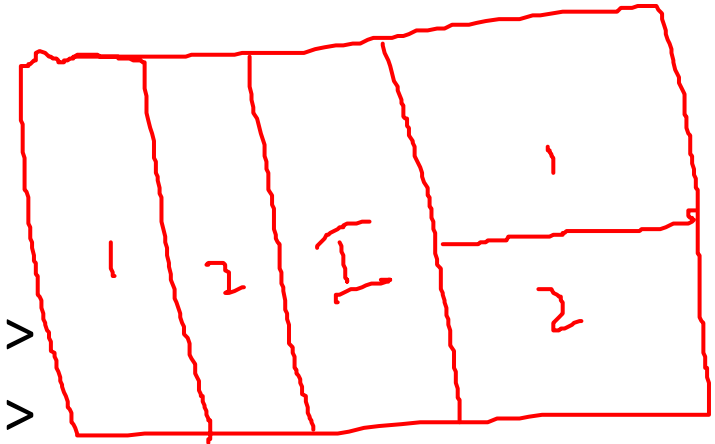
```
        <frame src="1.html">
```

```
        <frame src="2.html">
```

```
      </frameset>
```

```
    </frameset>
```

```
</html>
```



- Program to **remove scrollbar** in frames

```
<html>
```

```
<head>
```

```
    <title>Anchor Tag</title>
```

```
</head>
```

```
<frameset rows="30%,*" frameborder="1">
```

```
    <frame src="one.html">
```

```
    <frame src="image.html" scrolling="no">
```

```
</frameset>
```

```
</html>
```

- Program to **avoid resizing** the border:

```
<html>
```

```
    <head>
```

```
        <title>Anchor Tag</title>
```

```
</head>
```

```
<frameset rows= "30%,*" frameborder="1"  
    noresize="noresize">
```

```
    <frame src="one.html">
```

```
    <frame src="image.html" scrolling="no">
```

```
</frameset>
```

```
</html>
```

Iframe(Inline Frame)

- Iframe can be used to include Google Maps.

```
<html>
```

```
<body>
```

```
    <h1>Iframe Tag</h1>
```

```
    <iframe src="https://www.w3schools.com" align="right" frameborder="0">
```

```
        <p>Your browser doesn't support iframes</p>
```

```
    </iframe>
```

```
</body>
```

```
</html>
```


- To open link in a new frame:

frame.html

```
<html>
```

```
    <head>
```

```
        <title>Anchor Tag</title>
```

```
    </head>
```

```
    <frameset cols="30%,*">
```

```
        <frame src="anchor.html">
```

```
        <frame src="one.html" name="abc">
```

```
    </frameset>
```

```
</html>
```

- Anchor.html

```
<html>
```

```
<head>
```

```
    <title>Anchor tag example</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Anchor tag</h1>
```

```
    <a href="headings.html" target="abc">
```

```
Click here for headings</a><br><br>
```

```
    <a href="image2.html" target="abc">
```

```
Click here for images</a><br><br>
```

```
    <a href="tables.html" target="_blank">
```

```
Click here for tables</a><br><br>
```

```
</body>
```

```
</html>
```

What are forms?

- `<form>` is just another kind of HTML tag
- HTML forms are used to create (rather primitive) GUIs on Web pages
 - Usually the purpose is to ask the user for information
 - The information is then sent back to the server
- A **form** is an area that can contain **form elements**
 - The syntax is: `<form parameters> ...form elements... </form>`
 - Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc
 - Other kinds of HTML tags can be mixed in with the form elements
 - A form usually contains a **Submit** button to send the information in the form elements to the server
 - The form's *parameters* tell JavaScript how to send the information to the server (there are two different ways it could be sent)
 - Forms can be used for other things, such as a GUI for simple programs

The <form> tag

- The <form *arguments*> ... </form> tag encloses form elements (and probably other HTML as well)
- The arguments to form tell what to do with the user input
 - `action="url"` (required)
 - Specifies where to send the data when the Submit button is clicked
 - `method="get"` (default)
 - Form data is sent as a URL with `?form_data` info appended to the end
 - Can be used *only* if data is all ASCII and not more than 100 characters
 - `method="post"`
 - Form data is sent in the body of the URL request
 - Cannot be bookmarked by most browsers
 - `target="target"`
 - Tells where to open the page sent as a result of the request
 - `target=_blank` means open in a new window
 - `target=_top` means use the same window

The HTML `<form>` Elements

The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

```
<label for="fname">First name:</label>  
<input type="text" id="fname" name="fname">
```

The input Element

First name:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>The select Element</h2>
```

```
<p>The select element defines a drop-down list:</p>
```

```
<form action="/action_page.php">
```

```
  <label for="cars">Choose a car:</label>
```

```
  <select id="cars" name="cars">
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Saab</option>
```

```
    <option value="fiat">Fiat</option>
```

```
    <option value="audi">Audi</option>
```

```
  </select>
```

```
  <input type="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

The select Element

The select element defines a drop-down list:

Choose a car:


```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Visible Option Values</h2>
```

```
<p>Use the size attribute to specify the number of visible values.</p>
```

```
<form action="/action_page.php">
```

```
  <label for="cars">Choose a car:</label>
```

```
  <select id="cars" name="cars" size="3">
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Saab</option>
```

```
    <option value="fiat">Fiat</option>
```

```
    <option value="audi">Audi</option>
```

```
  </select><br><br>
```

```
  <input type="submit">
```


```
</form>
```

```
</body>
```

```
</html>
```

Visible Option Values

Use the size attribute to specify the number of visible values.

Choose a car: 

Submit

```
<!DOCTYPE html>
<html>
<body>
```

```
<h2>Allow Multiple Selections</h2>
```

```
<p>Use the multiple attribute to allow the user to select more than one
value.</p>
```

```
<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars" size="4" multiple>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br><br>
  <input type="submit">
</form>
```

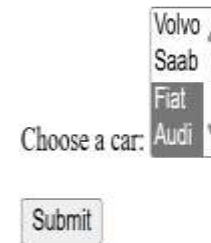
```
<p>Hold down the Ctrl (windows) / Command (Mac) button to select
multiple options.</p>
```

```
</body>
</html>
```

Allow Multiple Selections

Use the multiple attribute to allow the user to select more than one value.

Choose a car:



Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Grouping Form Data with Fieldset</h2>
```

<p>The fieldset element is used to group related data in a form, and the legend element defines a caption for the fieldset element.</p>

```
<form action="/action_page.php">
```

```
  <fieldset>
```

```
    <legend>Personalia:</legend>
```

```
    <label for="fname">First name:</label><br>
```

```
    <input type="text" id="fname"
name="fname" value="John"><br>
```

```
    <label for="lname">Last name:</label><br>
```

```
    <input type="text" id="lname"
name="lname" value="Doe"><br><br>
```

```
    <input type="submit" value="Submit">
```

```
  </fieldset>
```

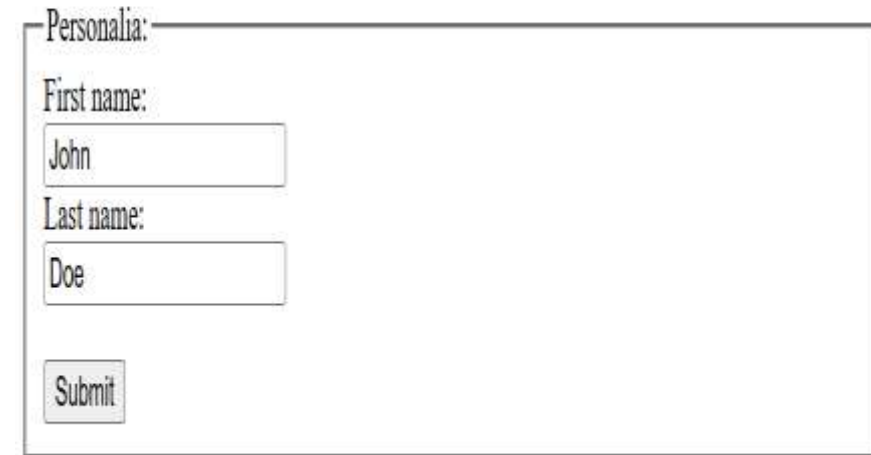
```
</form>
```

```
</body>
```

```
</html>
```

Grouping Form Data with Fieldset

The fieldset element is used to group related data in a form, and the legend element defines a caption for the fieldset element.



Personalia:

First name:
John

Last name:
Doe

Submit

```
<!DOCTYPE html>
<html>
<body>
```

```
<h2>The datalist Element</h2>
```

```
<p>The datalist element specifies a list of pre-
defined options for an input element.</p>
```

```
<form action="/action_page.php">
  <input list="browsers" name="browser">
    <datalist id="browsers">
      <option value="Edge">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
    <input type="submit">
  </form>
```

```
</body>
</html>
```

The datalist Element

The datalist element specifies a list of pre-defined options for an input element.



The image shows a web form with a dropdown menu and a 'Submit' button. The dropdown menu is open, displaying a list of browser names: Edge, Firefox, Chrome, Opera, and Safari. The 'Submit' button is located to the right of the dropdown menu.

The <input> tag

- Most, but not all, form elements use the **input** tag, with a
 - **type="..."** argument to tell which kind of element it is
 - **type** can be **text**, **checkbox**, **radio**, **password**, **hidden**, **submit**, **reset**, **button**, **file**, or **image**
- Other common **input** tag arguments include:
 - **name**: the name of the element
 - **value**: the “value” of the element; used in different ways for different values of **type**
 - **readonly**: the value cannot be changed
 - **disabled**: the user can’t do anything with this element
 - Other arguments are defined for the **input** tag but have meaning only for
 - certain values of **type**

- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="button">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text"> (default value)
- <input type="time">
- <input type="url">
- <input type="week">

Text input

A text field:

```
<input type="text" name="textfield" value="with an initial value">
```

A text field:

A multi-line text field

```
<textarea name="textarea" cols="24" rows="2">Hello</textarea>
```

A multi-line text field:

A password field:

```
<input type="password" name="textfield3" value="secret">
```

A password field:

- Note that two of these use the **input** tag, but one uses **textarea**

Button

S

- A submit button:
`<input type="submit" name="Submit" value="Submit">`
- A reset button:
`<input type="reset" name="Submit2" value="Reset">`
- A plain button:
`<input type="button" name="Submit3" value="Push Me">`

A submit button: 

A reset button: 

A plain button: 

- **submit**: send data
- **reset**: restore all form elements to their initial state
- **button**: take some action as

specified by JavaScript

- Note that the type is **input**, not “button”

Checkbox

S

- A checkbox:

```
<input type="checkbox" name="checkbox"
value="checkbox" checked>
```

A checkbox: ☒

- **type: "checkbox"**
- **name**: used to reference this form element from JavaScript
- **value**: value to be returned when element is checked
- Note that there is *no text* associated with the checkbox—you have to supply text in the surrounding HTML

Radio buttons

Radio buttons:


```
<input type="radio" name="radiobutton" value="myValue1">
```

male


```
<input type="radio" name="radiobutton" value="myValue2" checked>
```

female

Radio buttons:

☐ male

☒ female

- If two or more radio buttons have the same **name**, the user can only select one of them at a time
 - This is how you make a radio button “group”
- If you ask for the value of that **name**, you will get the **value** specified for the selected radio button
- As with checkboxes, radio buttons do not contain any text

Drop-down menu or list

- A menu or list:

```
<select name="select">  
  <option value="red">red</option>  
  <option value="green">green</option>  
  <option value="BLUE">blue</option>  
</select>
```

A menu or list: 

- Additional arguments:

- **size**: the number of items visible in the list (default is "1")
- **multiple**: if set to "true", any number of items may be selected (default is "false")

Hidden fields

- `<input type="hidden" name="hiddenField" value="nyah">`
<-- right there, don't you see it?

A hidden field: <-- right there, don't you see it?

- What good is this?
 - All `input` fields are sent back to the server, including hidden fields
 - This is a way to include information that the user doesn't need to see (or that you don't want her to see)
 - The `value` of a hidden field can be set programmatically (by JavaScript) before the form is submitted

A complete example

```
<html>
<head>
<title>Get Identity</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
<p>Name:
<input type="text" name="textfield">
</p>
<p>Gender:
<input type="radio" name="gender" value="m">Male
<input type="radio" name="gender" value="f">Female</p>
</form>
</body>
</html>
```

Who are you?

Name:

Gender: ☐ Male ☐ Female

HTML Canvas

- The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.
- The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
- Canvas is supported by all major browsers.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100"
```

```
style="border:1px solid #000000;">
```

Your browser does not support the HTML canvas tag.

```
</canvas>
```

```
</body>
```

```
</html>
```



HTML Multimedia

Multimedia on the web is sound, music, videos, movies, and animations.

Common Video Formats: The MP4, WebM, and Ogg formats are supported by HTML.

The MP4 format is recommended by YouTube.

Common Audio Formats: Only MP3, WAV, and Ogg audio are supported by the HTML standard.

Common Video Formats

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML.
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
WebM	.webm	WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube.

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```



Common Audio Formats

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. Supported by all browsers.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<audio controls>
```

```
  <source src="horse.ogg" type="audio/ogg">
```

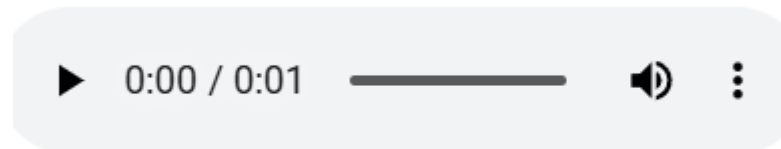
```
  <source src="horse.mp3" type="audio/mpeg">
```

```
  Your browser does not support the audio element.
```

```
</audio>
```

```
</body>
```

```
</html>
```



HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

```
<a href="url">link text </a>
```

By default, the linked page will be displayed in the current browser window.

To change this, you must specify another target for the link.

The **target** attribute specifies where to open the linked document.

The **target** attribute can have one of the following values:

- **_self** - Default. Opens the document in the same window/tab as it was clicked
- **_blank** - Opens the document in a new window or tab
- **_parent** - Opens the document in the parent frame
- **_top** - Opens the document in the full body of the window

- `Visit W3Schools!`

Unordered Lists

- Apple
- Banana
- Grapes
- Mango

Ordered Lists

1. New Delhi
2. Hyderabad
3. Mumbai
4. Chennai

Definition Lists

J2SE

Java 2 Standard Edition is for Stand-alone applications

J2EE

Java 2 Enterprise Edition is for Web-based applications

J2ME

Java 2 Mobile Edition is for developing Mobile applications

Welcome to PVPSIT Home page

Welcome to PVP College of Engineering and Technology. Our official website is pvpsiddhartha.ac.in

The college is affiliated to JNTU University, Kakinada.

We have many departments in Engineering

1. IT
2. CSE
3. ECE
4. EEE
5. IE
6. PE
7. Civil
8. Mechanical

Here's a Picture of our college



Copyrights 2023. All rights reserved.