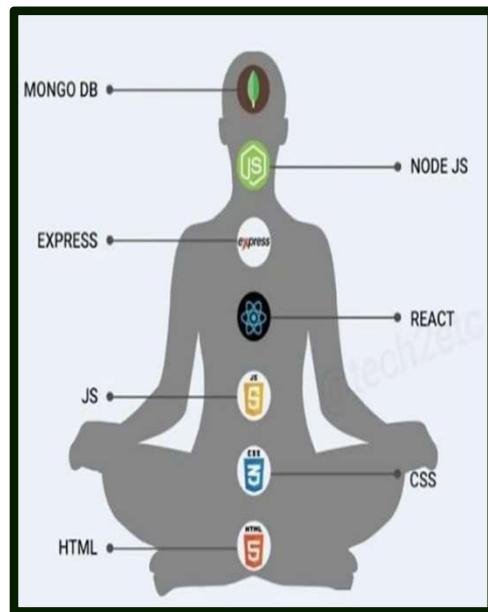


22ITC08

ENTERPRISE APPLICATION



DEVELOPMENT

UNIT-I

Web Fundamentals



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

Full Stack Development:

It refers to the development of both **front end**(client side) and **back end**(server side) portions of web application.

Full Stack Web Developers:

A full stack developer is one who is comfortable working with both backend and frontend technologies.





UNIT-I

Introduction:

Web Fundamentals, **HTML 5.0**: Basic tags, Images, Tables, Lists, Forms, Layout, Graphics, span and div tags.

Introduction to Cascading Style Sheets:

Types of CSS, **text and font, color**, CSS Selectors, CSS BOX Model, CSS Positioning, and CSS floating, CSS Grid layout Module.

22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

UNIT-II

Java Script: Data Types & Type Conversion, JSON, Events, String and Date Functions, Object Oriented Programming (OOP) in JS, Document Object Model, JavaScript Regular Expressions.

Bootstrap: Introduction of Bootstrap, Container and Container-fluid, Connectivity of Bootstrap in page.

Bootstrap Component: Jumbotron, Button, Grid, Table, Form, Alert, Panels, Image, Progress Bar, Tabs/Pill, Navbar, Modals.



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

UNIT-III

React JS: Basics, State, Props, ComponentsLifecycle, , Events, Router, Forms, Tables, Portals, ES6, CSS, Hook, and Back End Integration.

Express JS: The model-view-controller pattern, Defining EJS template Engine Building a front-end controller, defining routes, creating actions, Configuring Express to use EJS, Using REST, Reading POST data Adding middleware .



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

UNIT- IV

Node JS Modules: Functions, Buffer, Modules, Modules Types, Core Module, Local Modules and Modules Exports

Node Package Manager: What is NPM? Installing Packages Locally, installing package globally, adding dependency in package JSON and Updating packages.

Creating Web Server: Creating Web Server, Sending Requests and Handling HTTP requests.

File System: Read File, writing a File, opening a File Deleting a File, Writing a file asynchronously and Other I/O Operations.

Events: Event Emitter class, Inheriting Events and Returning event emitter.



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

UNIT-V

Mongo DB

Introduction, Importance of NoSQL databases, JSON features, Data types and examples. CRUD Operations, Data Modelling & Schema Design, Indexing and Aggregation, Mongo Import/Export and Master/Slave Replication.



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT



Text Books:

1. Vasan Subramanian, "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", second Edition, Apress Publications, 2019.
2. David Hows, Peter Membrey, EelcoPlugge – “MongoDB Basics”, Apress, 2014.

Suggested Reading:

1. Ethan Brown, “Web Development with Node and Express”, Oreilly Publishers, First Edition, 2014.

22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

Full Stack Development:

It refers to the development of both **front end**(client side) and **back end**(server side) portions of web application.

Full Stack Web Developers:

A full stack developer is one who is comfortable working with both backend and frontend technologies.



22ITC08 – ENTERPRISE APPLICATION DEVELOPMENT

POPULAR TECH STACKS

MERN



MEAN



MEVN

-MongoDB - ExpressJS - VueJS -NodeJS



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

Stage 1: HTML

Stage 2: CSS

Stage 3: JavaScript

Stage 4: ReactJs

Stage 5: NodeJs

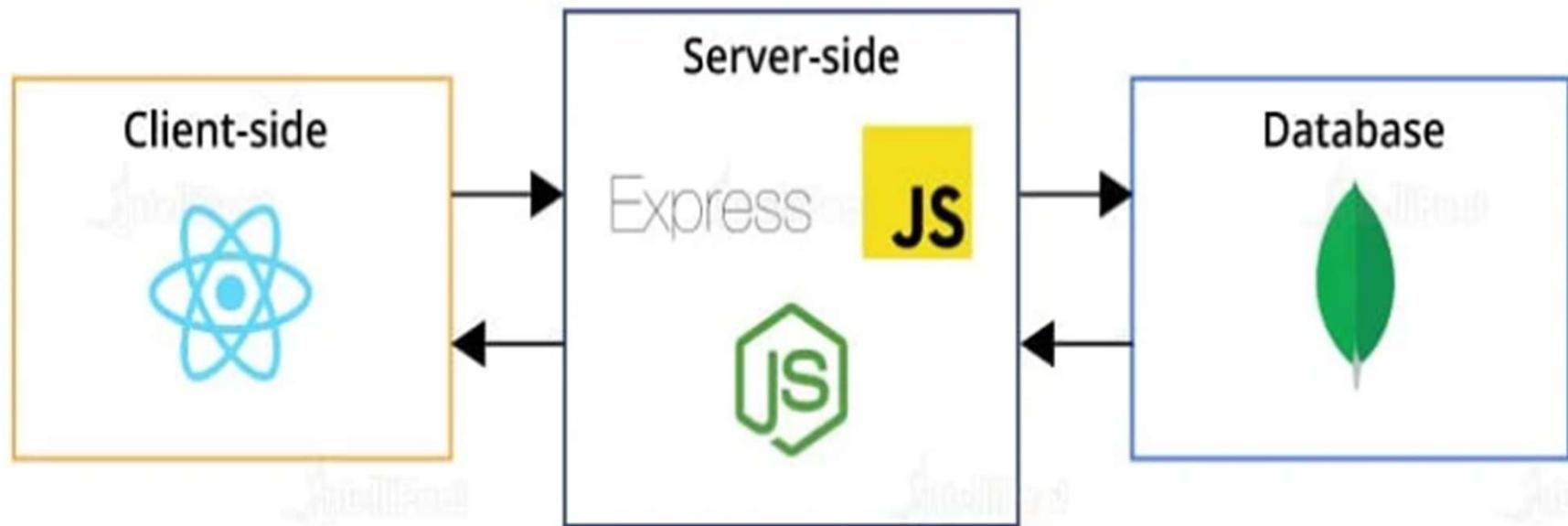
Stage 6: ExpressJS

Stage 7: MongoDB



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

3- Tier Architecture of MERN Stack



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "MERN-STACK-CRUD". The "routes" folder is selected.
- Code Editor:** Displays the "package.json" file. The "start" script is defined as "nodemon server.js". The "dependencies" section includes "body-parser": "^1.20.0", "cors": "^2.8.5", "express": "^4.18.1", and "mongoose": "^6.5.2". The "devDependencies" section includes "nodemon": "^2.0.19".
- Terminal:** Shows a Windows PowerShell window with the following output:

```
(2437:3) autoprefixer: Replace color-adjust to print-color-adjust. The color-adjust shorthand is currently deprecated.  
webpack compiled with 1 warning  
* History restored  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
PS C:\Users\Admin\mern-stack-crud>
```
- Status Bar:** Shows "Ln 7, Col 33" and "Spaces: 2" and "UTF-8".



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

IDE'S Required for FSD-MongoDB Compass

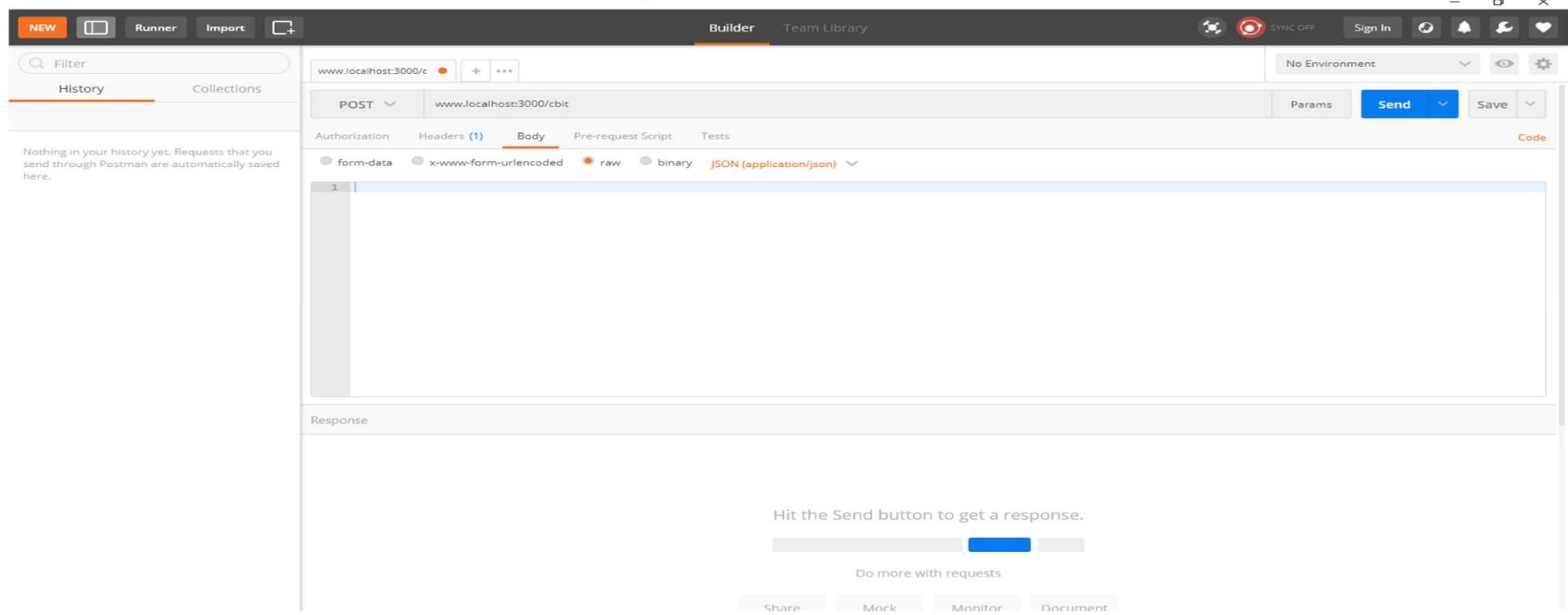
The screenshot shows the MongoDB Compass application interface. The left sidebar displays connection information for 'localhost:27017' with 3 DBs and 3 Collections, and a 'FAVORITE' button. Below this are sections for 'My Queries', 'Databases', and a search bar. The main area is titled 'Databases' and contains three entries:

Database	Storage size	Collections	Indexes
admin	20.48 kB	1	1
config	24.58 kB	1	2
local	36.86 kB	1	1



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT

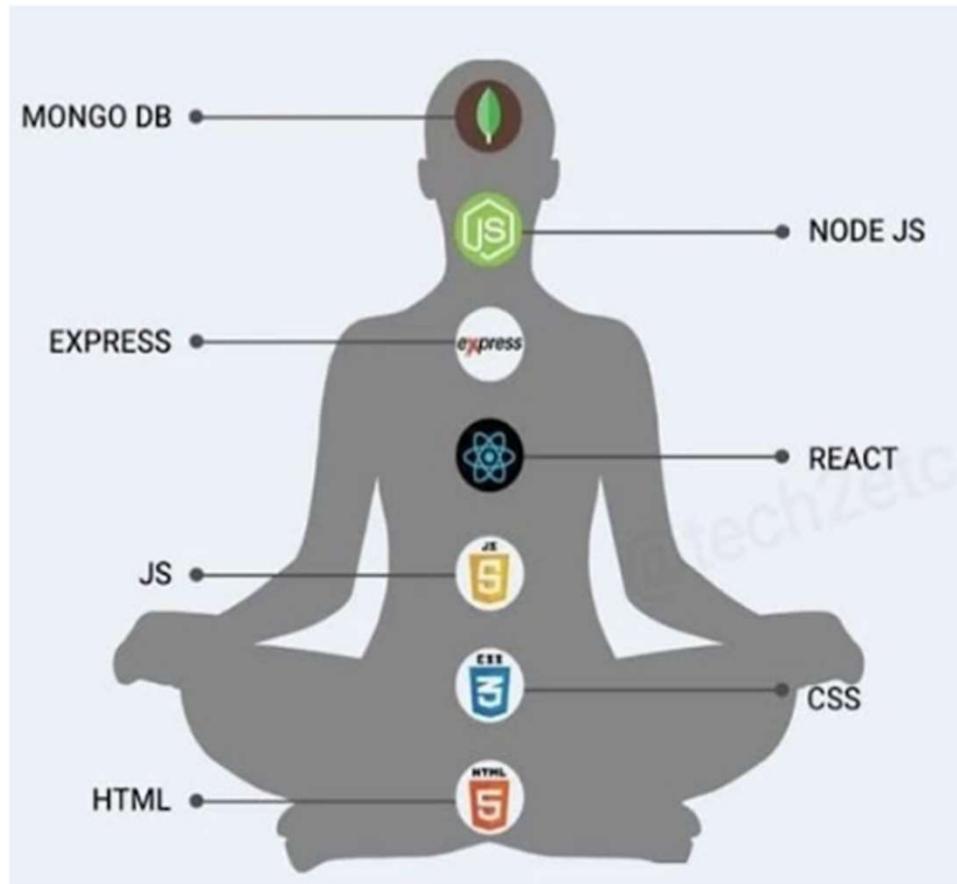
IDE'S Required for FSD-Postman



The screenshot shows the Postman application interface. At the top, there are tabs for "Builder" and "Team Library". Below the tabs, the URL is set to "www.localhost:3000/cbit" and the method is "POST". The "Body" tab is selected, showing options for "form-data", "x-www-form-urlencoded", "raw", and "binary", with "JSON (application/json)" selected. A large text area for the request body is present. The "Headers" section contains one header entry. Below the request area, there is a "Response" section with the placeholder text "Hit the Send button to get a response." At the bottom, there are buttons for "Share", "Mock", "Monitor", and "Document". The left sidebar shows a "History" tab with a message stating "Nothing in your history yet. Requests that you send through Postman are automatically saved here." and a "Collections" tab.



22ITC08 –ENTERPRISE APPLICATION DEVELOPMENT



Introduction: Web Fundamentals

- U.S. Department of Defense(**DoD**) interested in developing a new large-scale computer network in the 1960's, which led to **ARPAnet** (Advanced Research Projects Agency net).
- Later in 1970's and 1980's **BITNET**(Because It's Time Network) and **CSNET**(Computer Science Network) were developed for servicing the institutions of the Universities.
- In 1986 **NSFnet** (National Science Foundation) was created and replaced ARPAnet for non-military uses.



Introduction: Web Fundamentals

- By 1992, a wide variety of organizations had established nodes on this network and had connected more than 1 million computers around the world.
- In 1995 a small part of NSFnet returned to being a research network and the rest became as **Internet**.



What The Internet Is?

- The *internet is a huge collection of computers* connected in a communication network.
- Transmission Control Protocol/Internet Protocol (**TCP/IP**), became the *standard for computer network* connections in 1982.
- It can be used directly to allow a program on one computer to communicate with a program on another computer via the Internet.



What The Internet Is?

- Rather than connecting every computer on the Internet directly to every other computer on the Internet, normally the individual computers in an organization are connected to each other in a local network. (***LAN***)
- One node on this local network is physically connected to the Internet. So, the ***Internet is actually a network of networks*** rather than a network of computers.



Internet-Based Services



- **Email** – A fast, easy, and inexpensive way to communicate with other Internet users around the world.
- **Telnet** – Allows a user to log into a remote computer as though it were a local system.
- **FTP** – Allows a user to transfer virtually every kind of file that can be stored on a computer from one Internet-connected computer to another.
- **UseNet news** – A distributed bulletin board that offers a combination news and discussion service on thousands of topics.
- **World Wide Web (WWW)** – A hypertext interface to Internet information resources.

Internet Protocol Address

- Internet nodes are identified by numeric addresses.
- The *Internet Protocol (IP) address* of a machine connected to the Internet is a unique **32-bit** number.
- IP addresses usually are written (and thought of) as four 8-bit numbers, separated by periods.
- The four parts are separately used by Internet-routing computers to decide where a message must go next to get to its destination.

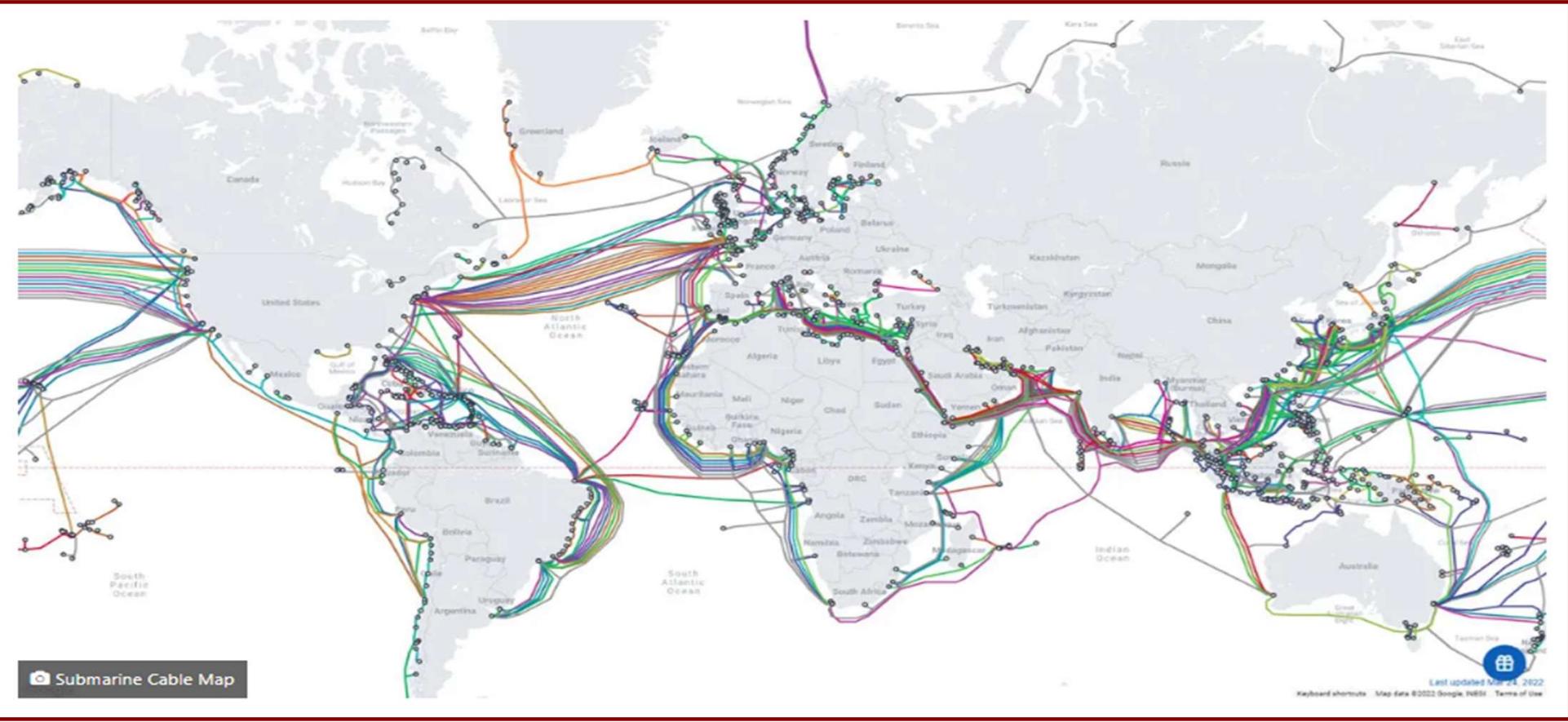


Internet Protocol Address

Class	Address Range	Supports
Class A	0.0.0.0 to 127.255.255.254	Supports 16 million hosts on each of 127 networks.
Class B	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 networks.
Class C	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networks.
Class D	224.0.0.0 to 239.255.255.255	Reserved for Multicast Groups.
Class E	240.0.0.0 to 254.255.255.254	Reserved for future use, or Research and Development Purposes.



Submarine Cable World Map



Domain Names

- Because people have difficulty dealing with and remembering numbers, machines on the Internet also have textual names.



- The host name along with the domain names are together called a *fully qualified domain name*.
- These domain names are converted into IP addresses by *Domain Name System(DNS)*



Domain Name System (DNS)



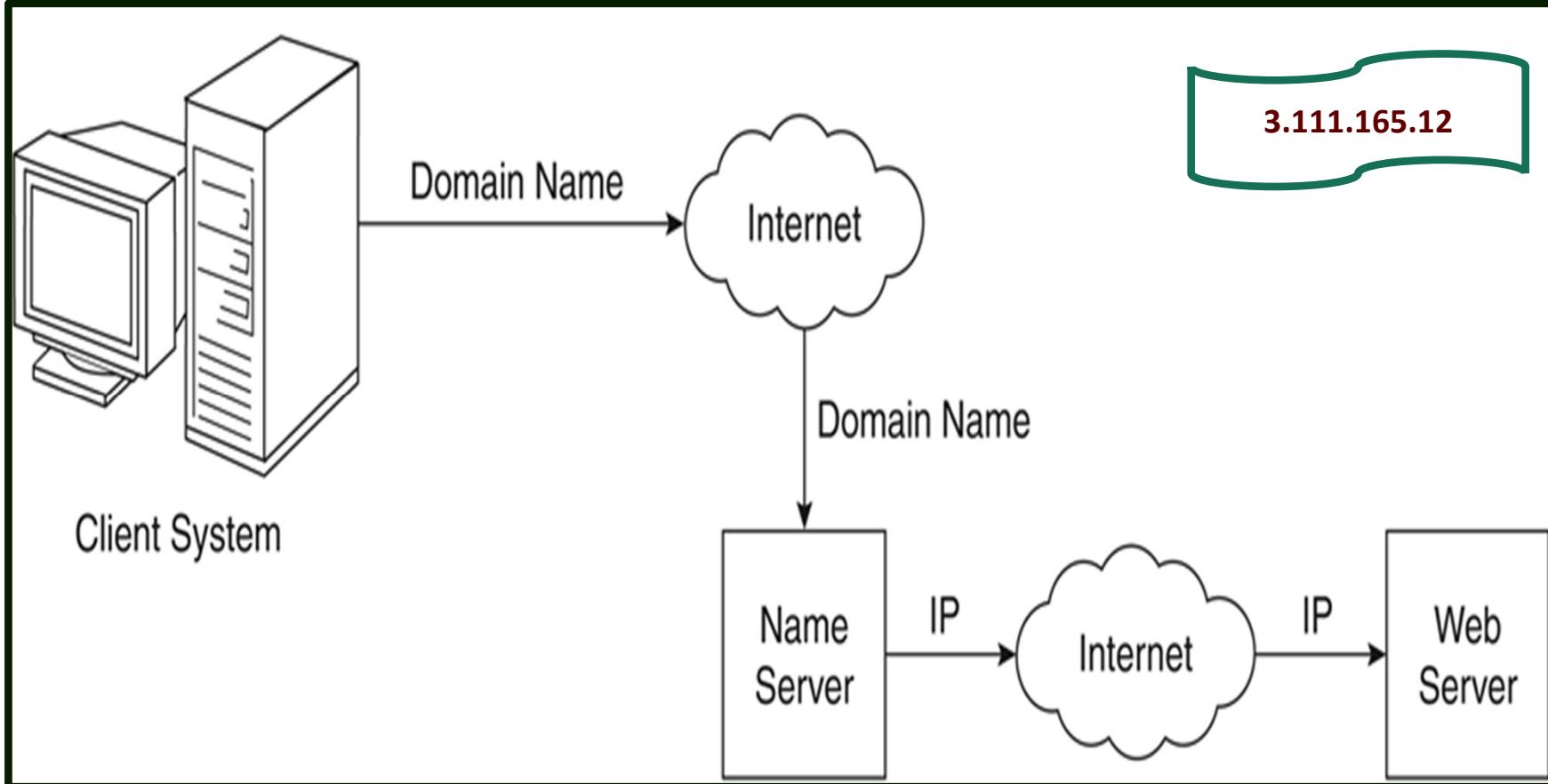
- IP addresses are the addresses used internally by the Internet, But *FQDN* which is given by a browser user, must be *converted to an IP address* before the message can be transmitted over the Internet to the destination.
- These conversions are done by software systems called name servers, which implement the **Domain Name System (DNS)**.

Domain Name System (DNS)

- Name servers serve a collection of machines on the Internet and are operated by organizations that are responsible for the part of the Internet to which those machines are connected.
- All document requests from browsers are routed to the nearest name server.
- If the name server can convert the fully qualified domain name to an IP address, it does so.
- If it cannot, the name server sends the fully qualified domain name to another name server for conversion.
- Like IP addresses, fully qualified domain names must be unique.



Domain Name Conversion



World Wide Web

- In 1989, a small group of people led by Tim Berners-Lee at **CERN** (Conseil Européen pour la Recherche Nucléaire, or **European Organization for Particle Physics**), proposed a new protocol for the Internet, as well as a system of document access to use it.
- The intent of this new system, which the group named the **World Wide Web**, was to allow scientists around the world to use the Internet to exchange documents describing their work.



Web or Internet?

- The *Internet is a collection of computers* and other devices connected by equipment that allows them to communicate with each other.
- 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*.



Web Browser and Web Server



- A ***browser is a client on the Web*** because it initiates the communication with a server, which waits for a request from the client.
- ***Web servers are programs*** that provide documents to requesting browsers.
- ***Servers are slave programs:*** They act only when requests are made to them by browsers running on other computers on the Internet.
- The most ***commonly used Web servers are Apache***, which has been implemented for a variety of computer platforms, and Microsoft's Internet Information Server(IIS), which runs under Windows operating systems.



Web Server Operation

- Web browsers initiate network communications with servers by sending them URLs
- A URL can specify one of two different things:
 1. The address of a data file stored on the server that is to be sent to the client.
or
 2. A program stored on the server that the client wants executed, with the output of the program returned to the client.
- All the communications between a Web client and a Web server use the standard Web protocol, Hypertext Transfer Protocol (HTTP).

Web Server Operation

- The primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by the commands.
- All HTTP commands include a URL, which includes the specification of a host server machine.
- When the URL is received, it is translated into either a file name (in which case the file is returned to the requesting client) or a program name (in which case the program is run and its output is sent to the requesting client).



Uniform Resource Locators

- Uniform (or universal) resource locators (URLs) are used to identify documents (resources) on the Internet.

scheme:object-address

- In the case of HTTP, the form of the object address of a URL is as follows:

//fully-qualified-domain-name/path-to-document

- When file is the protocol, the fully qualified domain name is omitted, making the form of such URLs as follows:

file://path-to-document



Multipurpose Internet Mail Extensions(MIME)

- A browser needs some way of determining the format of a document it receives from a Web server.
- The forms of these documents are specified with Multipurpose Internet Mail Extensions (**MIME**).
- **MIME specifications** have the following form:

type/subtype



Hyper Text Transfer Protocol(HTTP)

- All Web communications transactions use the same protocol:
Hypertext Transfer Protocol (HTTP) formally defined as RFC 2616
- HTTP consists of **two phases**:
Request and Response.
- Each HTTP communication (request or response) between a **browser** and a **Web server** consists of **two parts**:
Header and a Body.
- The header contains information about the communication; the body contains the data of the communication if there is any.



The Request Phase



- The general form of an HTTP request is as follows:
 1. HTTP method Domain part of the URL HTTP version
 2. Header fields
 3. Blank line
 4. Message body
- The following is an example of the first line of an HTTP request:

GET /storefront.html HTTP/1.1

HTTP REQUEST METHODS

Method	Description
GET	Returns the contents of the specified document
HEAD	Returns the header information for the specified document
POST	Executes the specified document, using the enclosed data
PUT	Replaces the specified document with the enclosed data
DELETE	Deletes the specified document

There are four categories of header fields:

1. General: For general information, such as the date.
2. Request: Included in request headers.
3. Response: For response headers.
4. Entity: Used in both request and response headers.



The Response Phase

- The general form of an HTTP response is as follows:

1. Status line

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

2. Response header fields

3. Blank line

4. Response body

- The status line includes the HTTP version used, a three-digit status code for the response, and a short textual explanation of the status code.

HTTP/1.1 200 OK

- The status codes begin with 1, 2, 3, 4, or 5.



22ITC08

EAD

UNIT-I

HTML 5.0



General Structure of HTML

```
<html>
  <head>
    <title>
      Title of the page
    </title>
  </head>
  <body>
    Content of the page
  </body>
</html>
```

The diagram illustrates the general structure of an HTML document. It consists of nested tags forming a tree-like structure:

- The outermost tag is `<html>`, represented by a yellow vertical arrow pointing downwards.
- Inside `<html>` is the `<head>` tag, represented by a red double-headed vertical arrow.
- Inside `<head>` is the `<title>` tag, which contains the text "Title of the page".
- Below `<head>` is the closing tag `</head>`.
- Inside `<html>` is the `<body>` tag, represented by a red double-headed vertical arrow.
- Inside `<body>` is the text "Content of the page".
- Below `<body>` is the closing tag `</body>`.
- The innermost tag is `</html>`, represented by a yellow vertical arrow pointing upwards.



What is HTML?

- HTML stands for Hyper Text Markup Language.
- HTML is the standard markup language for creating Web pages.
- HTML elements are the building blocks of HTML pages.
- The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>
- HTML elements are represented by tags.
- Browsers do not display the HTML tags, but use them to render the content of the page.



HTML

- The <!DOCTYPE html> declaration defines this document to be HTML5.
- The <html> element is the root element of an HTML page.
- The <head> element contains meta information about the document.
- The <title> element specifies a title for the document.
- The <body> element contains the visible page content.
- The <h1> element defines a large heading.
- The <p> element defines a paragraph.

.html / .htm



HTML Tags



- HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

- HTML tags normally come **in pairs** like <p> and </p>
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

STRUCTURAL TAGS

<HTML>

These tags enclose the entire Web page document.

</HTML>

<HEAD>

These tags enclose the Head part of the document

</HEAD>

<TITLE>

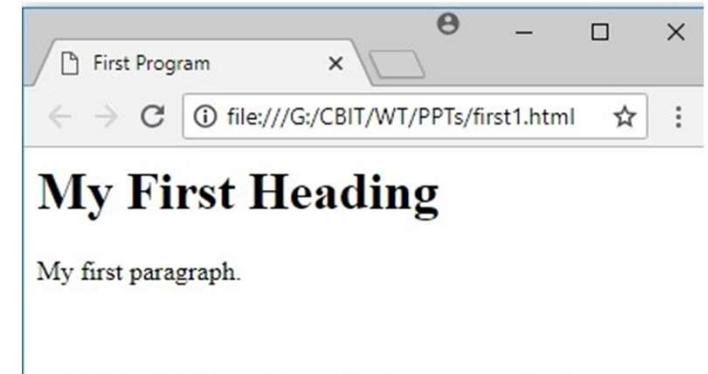
These tags enclose the title of the document. This text appears in the title bar in the browser and on the bookmark list if someone bookmarks your web page.

</TITLE>



A SIMPLE HTML Document

```
<!DOCTYPE html>
<html>
    <head>
        <title>First Program</title>
    </head>
<body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
</body>
</html>
```



Save it as first1.html



HTML Element Syntax

- HTML Elements represent semantics, or meaning.
- For example, The title element represents the title of the document.
- Most HTML elements are written with a *start tag* and an *end tag* with the content in between.

<tagname>Content goes here...</tagname>

- Elements can also contain **attributes** that define additional properties of an element



Empty HTML Elements

- Empty elements (also called **self-closing or void elements**) are not container tags — that means, you can not write

<hr>*some content*</hr>

or

**
some content</br>.**

- A typical example of an empty element, is the **
** element, which represents a line break.

**<p>This paragraph contains
 a line break.</p>**



HTML5 Tags

- HTML5 tags are not case sensitive:
- **<P> means the same as <p>.**

```
<!DOCTYPE html>

<head>
    <title>welcome</title>
</head>

<body>
    <p>welcome</p>
</body>

</html>
```



HTML Comment Tags

- You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

- With comments you can place notifications and reminders in your HTML.





Block-Level Elements

- Block-level elements are those elements that are formatted visually as blocks with a line break before and after the element.

```
<!DOCTYPE html>
<html>
    <body>
        <p>This is some text.</p>
        <div style="background-color:lightblue">
            <h3>This is a heading in a div
element</h3>
            <p>This is some text in a div
element.</p>
        </div>
        <p>This is some text.</p>
    </body>
</html>
```

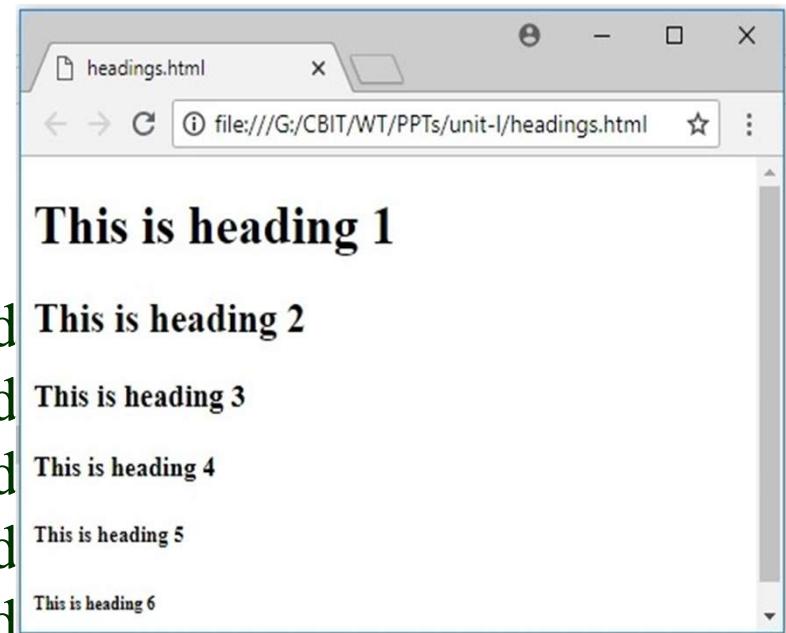
Inline-Level Elements & Headings

- Inline-level elements are those elements of the source document that do not form new blocks of content; the content is distributed in lines.
- For example, emphasized pieces of text within a paragraph (****), strong element (****) etc.
- **HTML headings** are defined with the **<h1> to <h6>** tags.
- **<h1>** defines the most important heading.
- **<h6>** defines the least important heading.



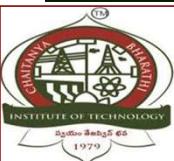
HTML Headings

```
<!DOCTYPE html>
<html>
  <body>
    <h1>This is heading 1
    <h2>This is heading 2
    <h3>This is heading 3
    <h4>This is heading 4
    <h5>This is heading 5
    <h6>This is heading 6</h6>
  </body>
</html>
```



HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in **name/value pairs** like: **name="value"**



The Style Attribute

- The style attribute is used to specify the styling of an element, like **color, font, size** etc.

```
<!DOCTYPE html>
<html>
    <body>
        <p style="color:red">I am a paragraph.</p>
        <p>Use the style attribute to control an element's layout.</p>
    </body>
</html>
```

I am a paragraph

Use the style attribute to control an element's layout.



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:

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

text d



USAGE OF



```
<!DOCTYPE html>

<html>

    <body>

        <p>This is<br>a paragraph<br>with line
breaks</p>

        </body>

    </html>
```



This is
a paragraph
with line breaks

Text Formatting Tags

Tag

Result

<I> Italics </I>

Italics

 Bold

Bold

<PRE> Preformatted Text </PRE> Preformatted Text

 Strong

Strong

<ADDRESS> Address </ADDRESS>

Address

<CITE> Citations </CITE>

Citations

<CODE> Source Code </CODE>

Source Code



HTML Text Formatting Tags

Tag	Description
	Defines bold text.
	Defines deleted text.
	Defines emphasized text.
<i>	Defines italic text.
<ins>	Defines inserted text.
<mark>	Defines marked/highlighted text.
<small>	Defines small text.
	Defines strong text.
<sub>	Defines subscripted text.
<sup>	Defines superscripted text.



HTML Text Formatting Tags

This text is bold.

This text is important!.

<i>This text is italic</i>

This text is emphasized

<small>This is some smaller text.</small>

<p>Do not forget to buy <mark>milk</mark> today.</p>

<p>My favorite color is blue red.</p>

<p>My favorite color is blue <ins>red</ins>.</p>

<p>This is _{subscripted} text.</p>

<p>This is ^{superscripted} text.</p>

This text is bold. This text is important!. This text is italic This text is emphasized This is some smaller text.

Do not forget to buy milk today.

My favorite color is blue red.

My favorite color is blue red.

This is subscripted text.

This is superscripted text.



HTML Styles (Inline)

```
<!DOCTYPE html>
```

```
  <html>
```

```
    <body>
```

```
      I am normal
```

```
I am red
```

```
I am blue
```

```
I am big
```

```
      <p style="color:red;">I am red</p>
```

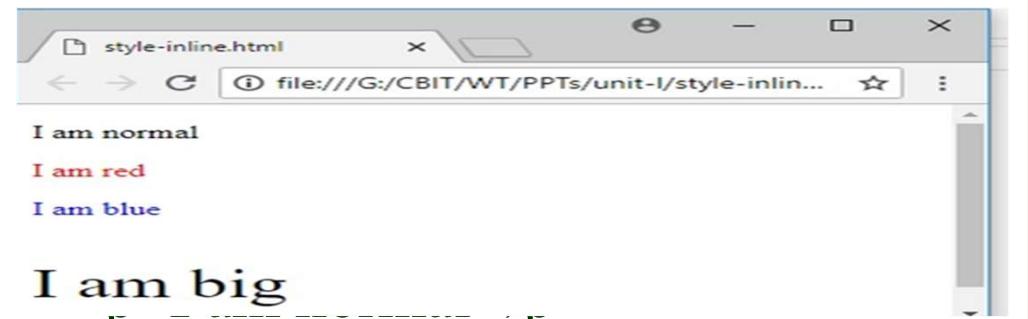
```
      <p style="color:blue;">I am blue</p>
```

```
      <p style="font-size:36px;">I am
```

```
    big</p>
```

```
  </body>
```

```
</html>
```



HTML Background Color

```
<!DOCTYPE html>
<html>
    <body style="background-color:powderblue;">
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```



FONT MODIFICATIONS



- Web creators can also change the way text looks by using the tag.
- Changes size of the font: SIZE="number" 1=smallest, 7 = largest
`Big Small`
Big Small
- Changes Text Color: COLOR="color-name"
`This is red` This is red
- Changes Font: FACE="font-name"
`This is the verdana font;`
`This is the chicago font.`
This is the verdana font;
This is chicago font.

HTML LINKS

- **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.

Syntax:

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

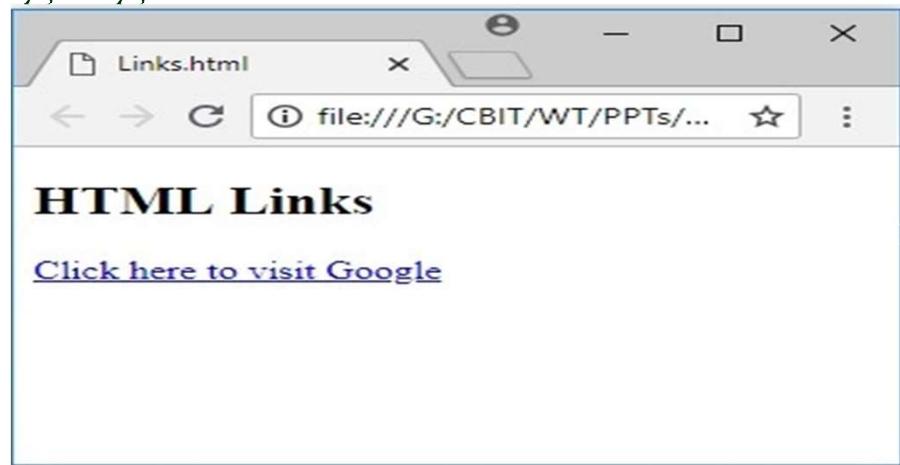


HTML LINKS

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

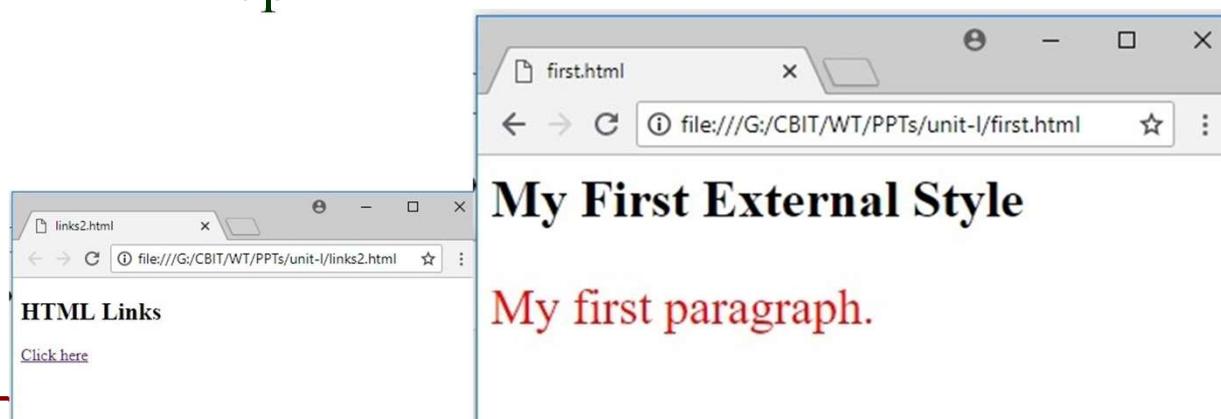
    <h2>HTML Links</h2>
    <p>
      <a href="https://www.google.com/">Click here to visit Google
    </a>      </p>

    </body>
  </html>
```



Navigation in HTML

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Links</h2>
    <p>
      <a href="first.html">Click here
    </a>
    </p>
  </body>
</html>
```



HTML IMAGES

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Image</h2>
    
  </body>
</html>
```



```

<!DOCTYPE html>
<html>
  <body>
    <h2>Basic HTML Table</h2>
    <table>
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
      </tr>
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
      </tr>
    </table>
  </body>
</html>

```



The screenshot shows a web browser window with the title "tables.htm". The page displays a heading "Basic HTML Table" followed by a table with three columns: Firstname, Lastname, and Age. The data rows are Jill Smith (50), Eve Jackson (94), and John Doe (80).

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80



Lists -- Ordered Lists

Ordered (Numbered) Lists:

 Item One
 Item Two
 Item Three



1. Item One
2. Item Two
3. Item Three
4. Item Four

Three

 Item Four

Ordered List Attributes:

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

i = i. Item One
One

I = I. Item One

(default)

a = a. Item One

A = A. Item

ii. Item Two II. Item Two
2. Item Two

iii. Item Three III. Item Three
3. Item Three

b. Item Two

B. Item Two

c. Item Three

C. Item Three

iv. Item Four IV. Item Four
4. Item Four

d. Item Four

D. Item Four

Lists -- Unordered Lists

Unordered lists:

```
<UL>
```

```
    <LI>Item One  
    <LI>Item Two  
    <LI>Item Three  
    <LI>Item Four
```

```
</UL>
```



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

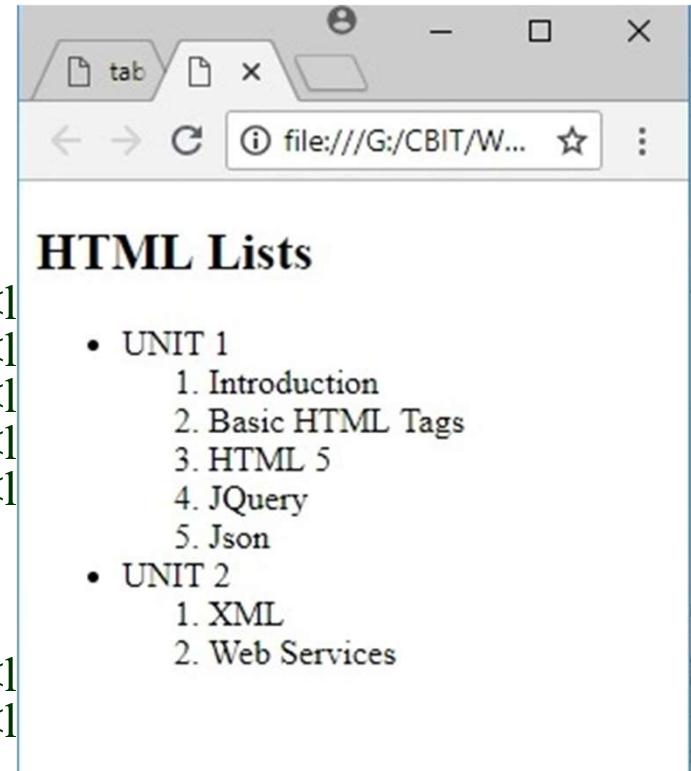
Unordered List Attributes:

type="disc/circle/square"

- Disc (default) Circle Square

HTML Lists

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Lists</h2>
    <ul>
      <li>UNIT 1</li>
      <ol>
        <li><!-->
        <li><!-->
        <li><!-->
        <li><!-->
        <li><!-->
      </ol>
      <li>UNIT 2</li>
      <ol>
        <li><!-->
        <li><!-->
      </ol>
    </ul>
  </body>
</html>
```



HTML 5 Elements

- Form controls,
- Layout Management,
- Graphics,
- Media,
- Span and div tags.



HTML Form Controls

1. Text input(one line or multiple lines)
2. Checkboxes
3. Radio buttons
4. Dropdowns
5. Submit and reset buttons



<input> Tag

- Form elements use the <input> tag, with a type="..." attributes to tell which kind of element it is.

Input tag attributes

1. Type - specifies type of the input control
2. Name – name of the control

Textfield

```
<form>  
    Rollno<input type="text" name="rollno" size="20" maxsize="10"/><br/>  
    stuname<input type="text" name="studentname" size="30"/><br/>  
</form>
```



HTML Form Controls

Checkbox

```
<form>  
    <input type="checkbox" name="branch" value="cse" checked="checked"/>CSE<br/>  
    <input type="checkbox" name="branch" value="it"/>IT<br/>  
</form>
```

Radio buttons

```
<input type="radio" name="radiobutton" value="1"/>male<br/>  
<input type="radio" name="radiobutton" value="2" checked="checked"/> female
```

Radio buttons:
 male
 female



HTML Form Controls-Buttons



A submit button:

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

A submit button:

A reset button:

```
<input type="reset" name="Submit2" value="Reset">
```

A reset button:

A plain button:

```
<input type="button" name="Submit3" value="Push Me">
```

A plain button:

HTML Form Controls- Textarea

Textarea: To insert a text area we use the tag <textarea>

Syntax:

```
<textarea name="string" rows="n" cols="n"> ... </textarea>
```

using text area

```
<form>
```

```
    <center>
```

```
        <h1>using text area</h1>
```

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

```
    </center>
```

```
</form>
```



HTML Form Controls- Drop Down List

<Select> : To create a drop down list we use **<select>** tag

Syntax:

```
<select name="string"> ..... </select>
```

<option> : To insert the values in the dropdown list we use the tag

Syntax:

```
<option value="string" [selected]> .... </option>
```

```
<select name="countries">
<option value="1" selected="selected"> India</option>
<option value="2"> Srilanka</option>
<option value="3"> Bangladesh</option>
</select>
```



NEW INPUT TYPES

```
html>
<body>
    <form action="">
        Birthday: <input type="date" name="bday"><br><br>
        Select your favorite color:
        <input type="color" name="favcolor" value="#ff0000">
        <br><br>
        E-mail:<input type="email" name="email">
        <br><br>
        Quantity (between 1 and 5):
        <input type="number" name="quantity" min="1" max="5">
        <input type="submit">

        <br><br>
    </form>
</body>
</html>
```

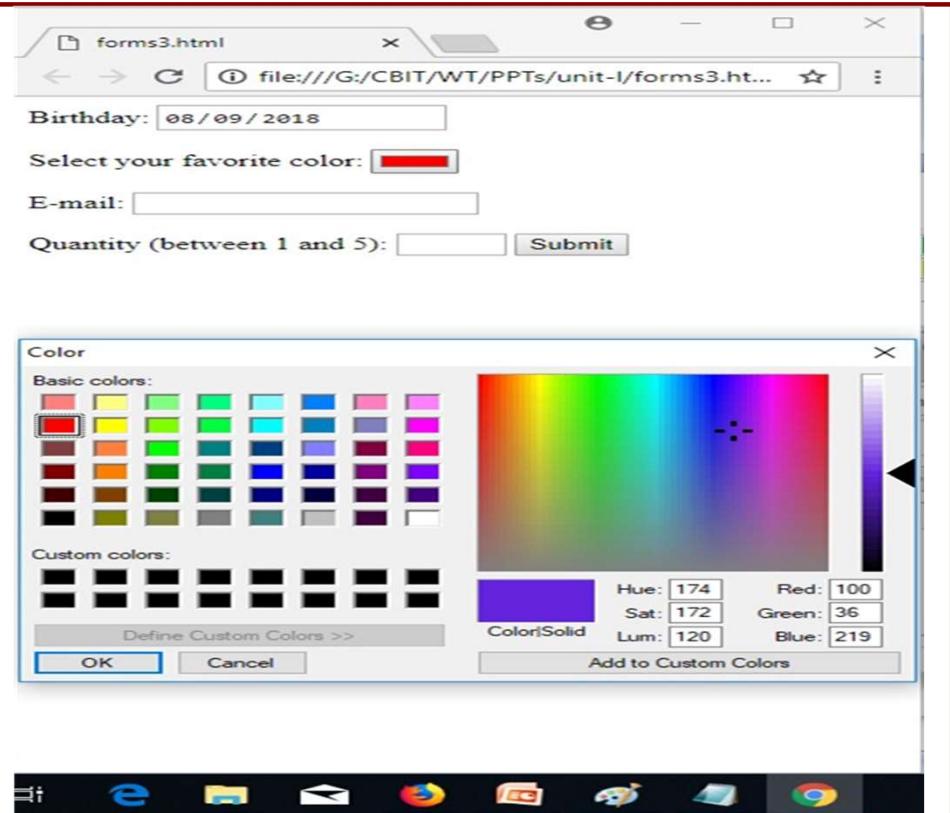


NEW INPUT TYPES

A screenshot of a web browser window titled "forms3.html". The page contains several input fields:

- Birthday: A date input field showing "08/09/2018".
- Select your favorite color: A color picker input field.
- E-mail: An email input field.
- Quantity (between 1 and 5): A numeric input field with a range from 1 to 5.

The browser toolbar at the top includes icons for back, forward, search, and refresh.



HTML Forms

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

    <h2>HTML Forms</h2>

    <form>
      First name:<br>
      <input type="text" name="firstname" value="Mickey"><br>
      Last name:<br>
      <input type="text" name="lastname" value="Mouse"><br>
      User password:<br>
      <input type="password" name="psw"> <br>
```



HTML Forms

Gender :

```
<input type="radio" name="gender" value="male"> Male<br>
<input type="radio" name="gender" value="female" checked> Fem<br>
<input type="radio" name="gender" value="other"> Other<br><br>
```

CheckBox:


```
<input type="checkbox" name="vehicle1" value="Bike">I have a bike<br>
<input type="checkbox" name="vehicle2" value="Car">I have a car<br>
```

Country

```
<select name="country">
    <option value="India">India</option>
    <option value="Australia">Australia</option>
    <option value="USA">USA</option>
    <option value="Canada">Canada</option>
</select> <br><br>
```

Address :


```
<textarea name="message" rows="10" cols="30">
    Enter your Address
</textarea><br><br>
```

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

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

The screenshot shows a web browser window with the title "form1.html". The page content is titled "HTML Forms". It contains the following form elements:

- First name: Mickey
- Last name: Mouse
- User password: (empty input field)
- Gender: Female (radio button selected)
- CheckBox:
 - I have a bike (checkbox selected)
 - I have a car (checkbox not selected)
- Country: India (dropdown menu selected)
- Address: Enter your Address (text area)

A "Submit" button is located at the bottom of the form.



HTML Canvas

- The HTML **<canvas>** element is used to **draw graphics via JavaScript**.
- The **<canvas>** element is only a container for graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
- A canvas is a rectangular area on an HTML page.
- By default, a canvas has no border and no content.

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

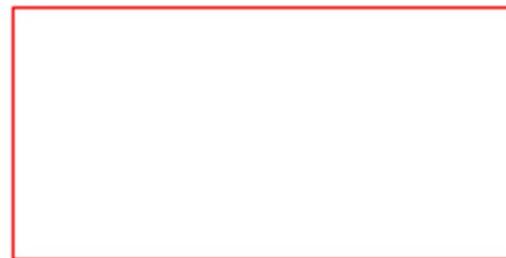


HTML Canvas

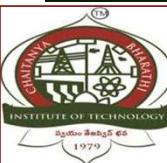
```
<!DOCTYPE html>
<html>
    <body>

<canvas
id="myCanvas" width="200" height="100" style="border:1px solid red;">
</canvas>

    </body>
</html>
```



---> A rectangle will be displayed in the output

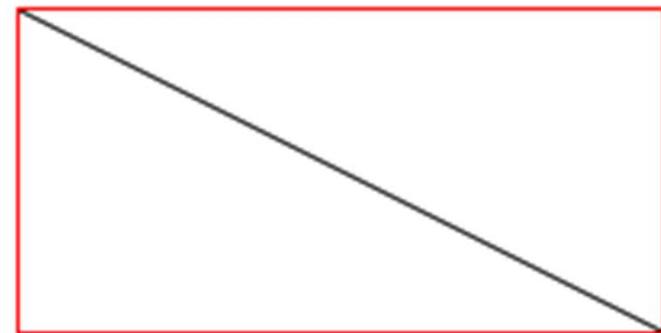


HTML Canvas

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

<canvas id="myCanvas" width="200" height="100" style="border:1px solid red;">
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>

</body>
</html>
```



What is SVG?

- SVG stands for **Scalable Vector Graphics**
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation
- The HTML <svg> element is a container for SVG graphics.
- SVG has several methods for drawing paths, boxes, circles, text, and graphic images.



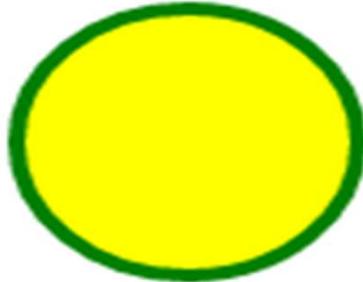
What is SVG?



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

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

</body>
</html>
```

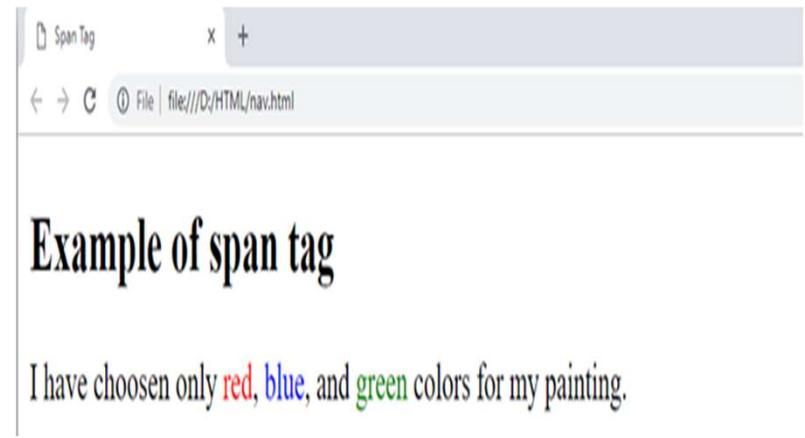


HTML Tag

- A element used to color a part of a text.
- The tag is used to group inline-elements in a document.
- The tag provides no visual change by itself.
- The tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
- The tag is much like the <div> element, but <div> is a block-level element and is an inline element.



```
<!DOCTYPE html>
<html>
<head>
<title>Span Tag</title>
</head>
<body>
    <h2>Example of span tag</h2>
    <p>I have chosen only
        <span style="color: red;">red</span>,
        <span style="color: blue;">blue</span>, and
        <span style="color: green;">green</span> colors for my painting.
    </p>
</body>
</html>
```



Example of span tag

I have choosen only red, blue, and green colors for my painting.

Tag <div>

- The <div> tag defines a division or a section in an HTML document.
- The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.
- By default, browsers always place a line break before and after the <div> element.



Tag <div>

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 { width:500px; margin: auto; border: 3px solid #73AD21; }
div.ex2 { max-width:500px; margin: auto; border: 3px solid #73AD21; }
</style>
</head>
<body>
<div class="ex1">This div element has width: 500px;</div><br>
<div class="ex2">This div element has max-width: 500px;</div>
<p><strong>Tip:</strong> Drag the browser window to smaller than 500px wide, to see the difference between the two divs!</p>
</body>
</html>
```



What is Multimedia?

- Multimedia comes in many different formats. It can be almost anything you can hear or see.
- Examples: Images, music, sound, videos, records, films, animations, and more.

Multimedia Formats

- Multimedia elements (like audio or video) are stored in media files.
- Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.





What is Multimedia?

Playing Videos in HTML

```
<video width="320" height="240" controls>  
    <source src="movie.mp4" type="video/mp4">  
</video>
```



Playing audios in HTML

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



<iframe> in HTML

- The iframe in HTML stands for **Inline Frame**.
- The ” iframe ” tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.
- An inline frame is used to **embed another document** within the current HTML document.
- The ‘ **src** ‘ attribute is used to specify the URL of the document that occupies the iframe.

<iframe src="URL"></iframe>



<iframe> in HTML

```
<!DOCTYPE html>

<html>
    <head>
        <title>HTML Iframes</title>
    </head>
    <body>
        <p>Document content goes here...</p>
        <iframe src = "C:/Users/ganga/Desktop/login.html" width = "555" height = "200">    </iframe>
        <p>Document content also go here...</p>
    </body>
</html>
```



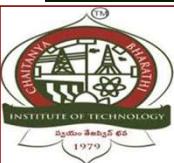
HTML YouTube Videos

```
<!DOCTYPE html>

<html>
<body>

<iframe width="420" height="345" src=" https://youtu.be/rzjgMKTPlis "></iframe>

</body>
</html>
```



What is CSS?

- CSS stands for **Cascading Style Sheets**.
- CSS describes **how HTML elements are to be displayed on screen or in other media.**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.
- External style sheets are stored in **CSS files**.



CSS SELECTORS

- Element Selector
- ID Selector
- Class Selector
- Group Selector
- Universal Selector



CSS Selectors



- CSS selectors are used to "find" HTML elements based on their element name, id, class, attribute, and more.

The element Selector

- The element selector selects elements based on the element name.
- You can select all `<p>` elements on a page.
- In this case, all `<p>` elements will be center-aligned, with a red text color.

Element Selector

```
<!DOCTYPE html>
  <html>
    <head>
      <style>
        p {
          text-align: center;
          color: red;
        }
      </style>
    </head>
    <body>
      <p>Every paragraph will be affected by the style.</p>
      <p id="para1">Me too!</p>
      <p>And me!</p>
    </body>
  </html>
```



The Id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a **hash (#) character**, followed by the id of the element.



The Id Selector

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #para1
      {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>
    <p id="para1">Hello World!</p>
    <p>This paragraph is not affected by the style.</p>
  </body>
</html>
```



The Class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a **period (.) character**, followed by the name of the class.

syntax:

```
.classname{ CSS-Property: value; ..... }
```



The Class Selector

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .center
            {
                text-align: center;
                color: red;
            }
        </style>
    </head>
    <body>
        <h1 class="center">Red and center-aligned heading</h1>
        <p class="center">Red and center-aligned paragraph.</p>
    </body>
</html>
```





The Class Selector

Header 1

A paragraph.

```
<!DOCTYPE html>
  <html>
    <head>
      <style>
        .intro
        {
          color: blue;
          text-align: center;
        }

        .important
        {
          background-color: yellow;
        }
      </style>
    </head>
    <body>
      <h1 class="intro important">Header 1</h1>
      <p>A paragraph.</p>
    </body>
  </html>
```

Group Selectors

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1, h2, p
      {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>

    <h1>Hello World!</h1>
    <h2>Smaller heading!</h2>
    <p>This is a paragraph.</p>

  </body>
</html>
```



Universal Selectors

```
<html>
  <head>
    <title>Simple Example of CSS universal selectors </title>
    <style>
      *
      {
        color: blue;
        background: silver;
      }
    </style>
  </head>
  <body>
    <h1>Example of CSS.</h2>
    <h2>Example of grouping of CSS universal selectors.</h2>
    <p>CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, established in the Year 1979, esteemed
as the Premier Engineering Institute in the States of Telangana and Andhra Pradesh.</p>
  </body>
</html>
```



Cascading Style Sheets

➤ There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style



External Style Sheet

- With an external style sheet, you can **change the look of an entire website** by changing just one file!
- Each page must include a reference to the external style sheet file inside the `<link>` element.
- The style sheet file must be saved with a **.css** extension.

```
<link rel="stylesheet" type="text/css" href="cbitexternal.css">
```



External Style Sheet

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css"
  href="cbitexternal.css">
  </head>
  <body>

    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>

  </body>
```

"cbitexternal.css"



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

h1
{
    color: navy;
    margin-left: 20px;
}
```

Internal Style Sheet

- An internal style sheet may be used if one **single page has a unique style**.
- Internal styles are defined within the **<style>** element, inside the **<head>** section of an HTML page.

<head>

<style>

Selector

{

property:

}

value

</style>

</head>



Internal Style Sheet

```
<html>
  <head>
    <style>
      body
      {
        background-color: linen;
      }
      h1
      {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



Inline Styles

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

<tag style=“property:value>content</tag>



Inline Styles

```
<!DOCTYPE html>

<html>
<body>

<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



CSS Background Color

```
<html>
  <head>
    <style>
      h1
      {
        background-color: green;
      }

      div
      {
        background-color: lightblue;
      }

      p
      {
        background-color: yellow;
      }
    </style>
  </head>
  <body>

    <h1>CSS background-color example!</h1>
    <div>
      This is a text inside a div element.
      <p>This paragraph has its own background color.</p>
      We are still in the div element.
    </div>

  </body>
</html>
```



CSS Background Image



```
<html>
  <head>
    <style>
      body
      {
        background-image: url("paper.gif");
      }
    </style>
  </head>

  <body>
    <h1>Hello World!</h1>
    <p>This page has an image as the background!</p>
  </body>
</html>
```

CSS Background

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

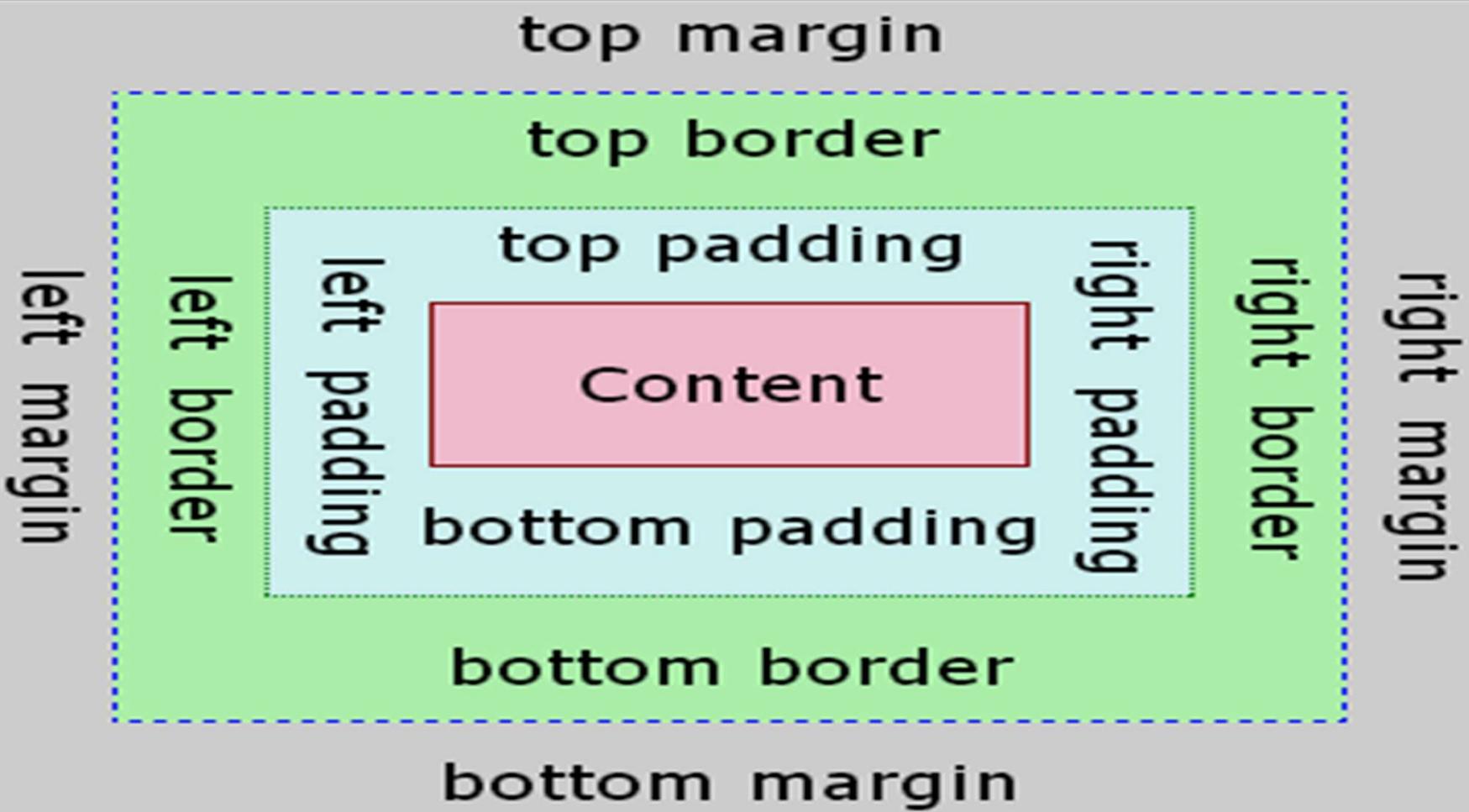


The CSS Box Model

- All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element.
- Each Rectangular Box contains 4 edges.
 - Content Edge
 - Padding Edge
 - Border Edge
 - Margin Edge



The CSS Box Model



The CSS Box Model

- **Content** - The content of the box, where text and images appear.
- **Padding** - Clears an area around the content. The padding is transparent.
- **Border** - A border that goes around the padding and content.
- **Margin** - Clears an area outside the border. The margin is transparent.



The CSS Box Model

The `padding` property is a shorthand property for the following individual padding properties:

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

So, here is how it works:

If the `padding` property has four values:

- **`padding: 25px 50px 75px 100px;`**
 - top padding is 25px
 - right padding is 50px
 - bottom padding is 75px
 - left padding is 100px

```
div {  
  padding: 25px 50px 75px 100px;  
}
```

```
div {  
  padding: 25px 50px 75px;  
}
```

```
div {  
  padding: 25px 50px;  
}
```

```
div {  
  padding: 25px;  
}
```





The CSS Box Model

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div
      {
        background-color: lightgrey;
        width: 300px;
        border: 25px solid green;
        padding: 25px;
        margin: 25px;
      }
    </style>
  </head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
<div>This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>
</body>
</html>
```

This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CSS POSITIONING

- The CSS position property defines the position of an element where generally the top, right, bottom, and left properties will determine the position of the element.
- However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

position: value;



CSS POSITIONING

Value	Description
static	Normal position for the element (where top, right, bottom, and left have no effect) <code>div { position: static; }</code>
relative	Position the element relative to where its normal position would have been <code>div { position: relative; top: 10px; left: 15px; }</code>
absolute	Position the element absolutely relative to its container <code>div { position: absolute; top: 10px; left: 15px; }</code>
fixed	Position the element relative to the screen's viewport and stay fixed on screen when scrolling <code>div { position: fixed; top: 10px; left: 15px; }</code>



STATIC POSITION

position: static;

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.



STATIC POSITION

```
<html>
  <head>
    <style>
      div.static
      {
        position: static;
        border: 3px solid #73AD21;
      }
    </style>
  </head>
  <body>

    <h2>position: static;</h2>

    <p>An element with position: static; is not positioned in any special way;</p>

    <div class="static"> This div element has position: static; </div>

  </body>
</html>
```



Relative Position



position: relative;

- An element with **position: relative;** is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element.

FIXED POSITION

position: fixed;

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.



FIXED POSITION

```
<html>
  <head>
    <style>
      div.fixed
      {
        position: fixed;
        bottom: 0;
        right: 0;
        width: 300px;
        border: 3px solid #73AD21;
      }
    </style>
  </head>
  <body>

    <h2>position: fixed;</h2>

    <p>An element with position: fixed;</p>
    <div class="fixed">
      This div element has position: fixed;
    </div>
  </body>
</html>
```



ABSOLUTE POSITION

position: absolute;

- Elements with **position: absolute;** are positioned relative to their parent elements.
- That means that the parent element has to have a position value other than **position: static.**
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.



ABSOLUTE POSITION

```
<html>
  <head>
    <style>
div.relative { position: relative; width: 400px; height: 200px; border: 3px solid #73AD21;}
div.absolute { position: absolute; top: 80px; right: 0; width: 200px; height: 100px; border: 3px solid #73AD21;}
    </style>
  </head>
  <body>
    <h2>position: absolute;</h2>
    <p>An element with position: absolute; is positioned relative to the nearest positioned ancestor.</p>
    <div class="relative">This div element has position: relative;
      <div class="absolute">This div element has position: absolute;</div>
    </div>
  </body>
</html>
```



CSS Layout - Float and Clear

- The float property specifies whether or not an element should float.
- The clear property is used to control the behavior of floating elements.
- The float property can be used to wrap text around images.



CSS Layout - Float and Clear



Float Right

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, established in the Year 1979, esteemed as the Premier Engineering Institute in the States of Telangana and Andhra Pradesh, was promoted by a Group of Visionaries from varied Professions of Engineering, Medical, Legal and Management, with an Objective to facilitate the Best Engineering and Management Education to the Students and contribute towards meeting the need of Skilled and Technically conversant Engineers and Management Professionals, for the Country that embarked on an Economic Growth Plan.



CSS Layout - Float and Clear

- The clear property is used to control the behavior of floating elements.
- Elements after a floating element will flow around it. To avoid this, use the clear property.
- The clear property specifies on which sides of an element floating elements are not allowed to float.



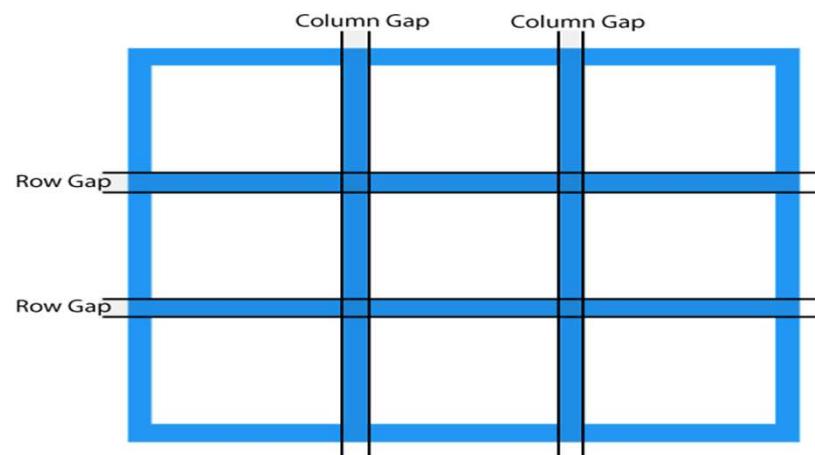
CSS Grid Layout

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
- **Display property is set to grid or inline-grid.**
- The **gap property** is a shorthand property for the **row-gap and the column-gap properties**.



CSS Grid Layout

1	2	3
4	5	6
7	8	9



CSS Grid Layout

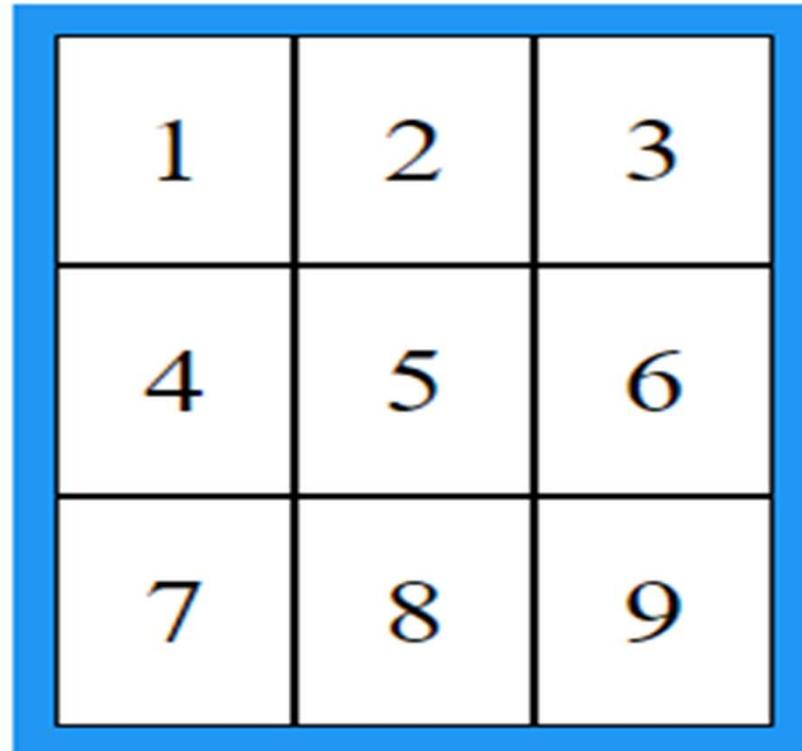
```
.grid-container
{
    display: inline-grid;
    grid-template-columns: auto auto auto;
    background-color: #2196F3;
    padding: 10px;
}
```

```
.grid-item
{
    background-color: rgb(255, 255, 255);
    border: 1px solid rgb(0, 0, 0);
    padding: 20px;
    font-size: 30px;
    text-align: center;
}
```



CSS Grid Layout

```
<div class="grid-container">  
  <div class="grid-item">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
  <div class="grid-item">8</div>  
  <div class="grid-item">9</div>  
</div>
```



CSS Based CHESS Board & VIBGYOR

