INTRODUCTION TO WEB DESIGNING

SUMAN GURJAR
SCIENTIST D
ISSD

KEYWORDS

- ❑ Internet
- ❑ World wide web
- ❑ File
- ❑ Web browser
- ❑ Website
- ❑ Web page
- ❑ Home page

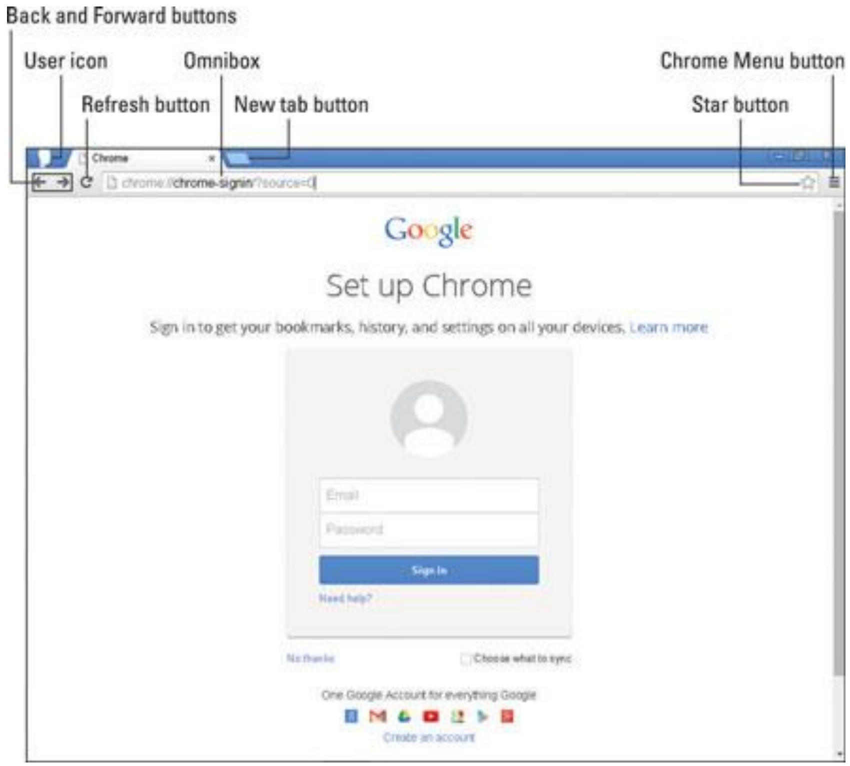
- Internet, hardware such as computers, cables, and telephone wires that is connected to create a massive world wise network
- World wide Web, is a system of interlinked hypertext documents that are accessed via internet
- Files, contains information, such as text, graphics, video, animation that is stored on computer
- A website is made up of web pages
- A home page is generally first page a user sees when visiting a site
- Hypertext Markup language (HTML) is the code used to create web pages

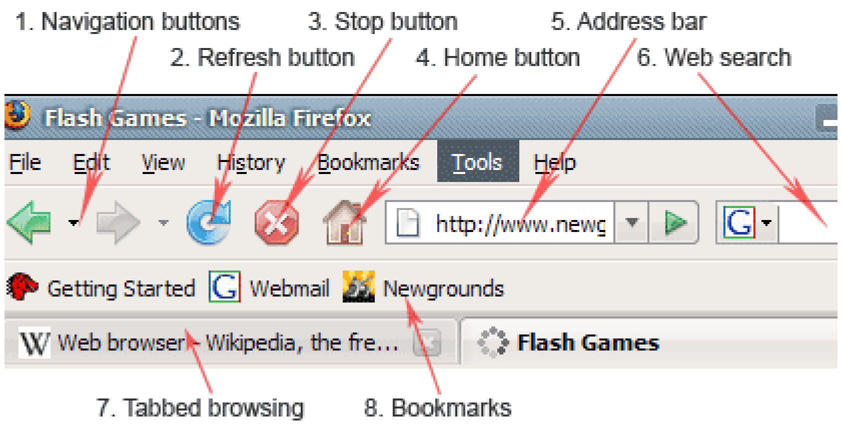
WEB BROWSER

A web browser access the web page from internet and display the web page on computer screen.

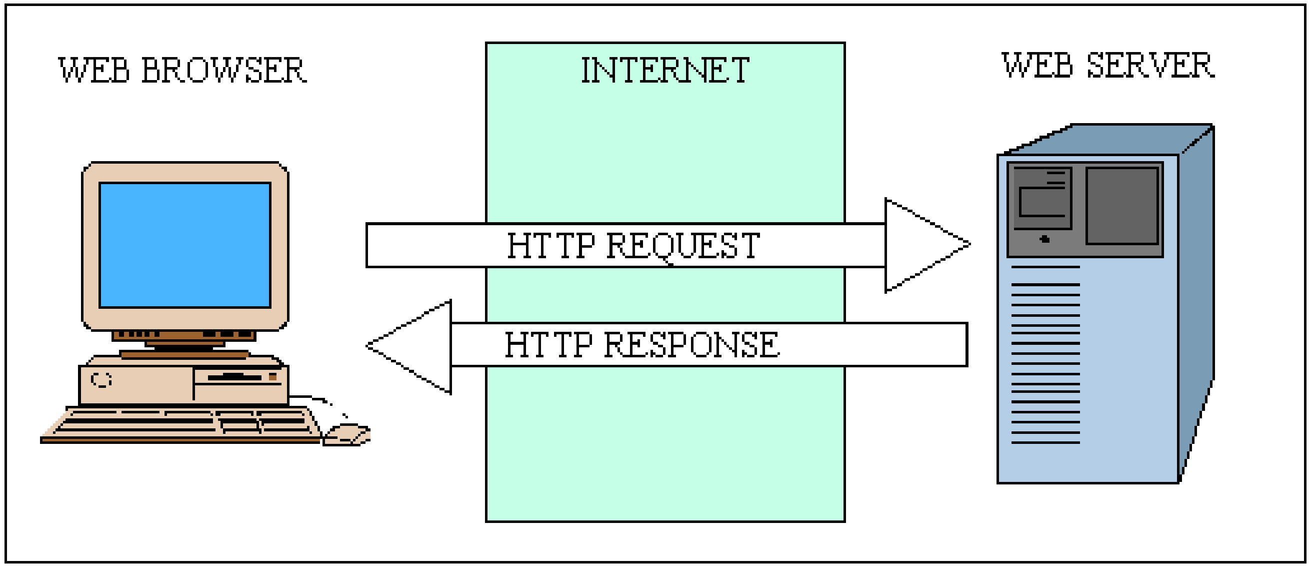
Mozilla Firefox
Google Chrome
Microsoft Internet Explorer
Apple Safari
Opera

Basic Internet Browser Features

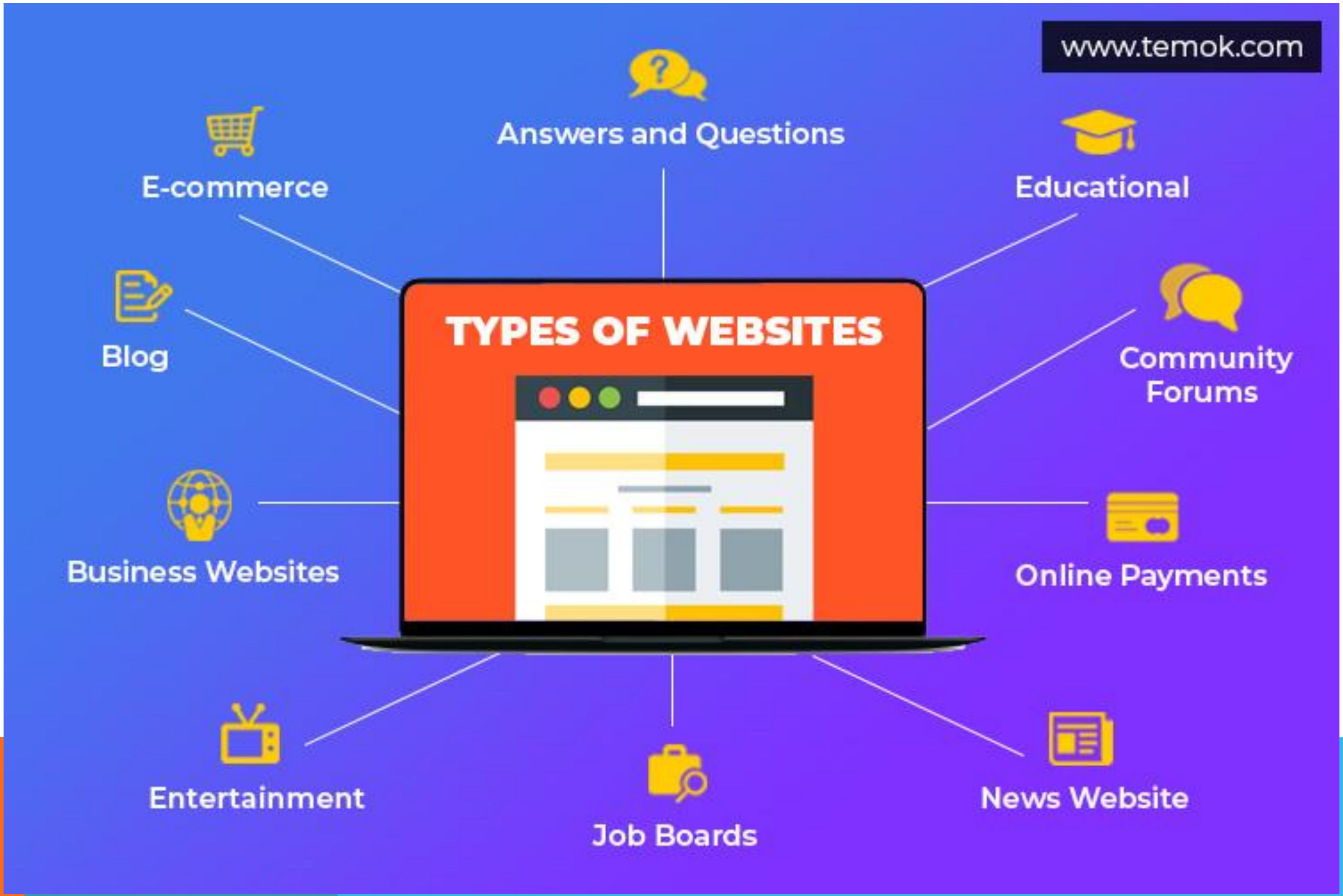




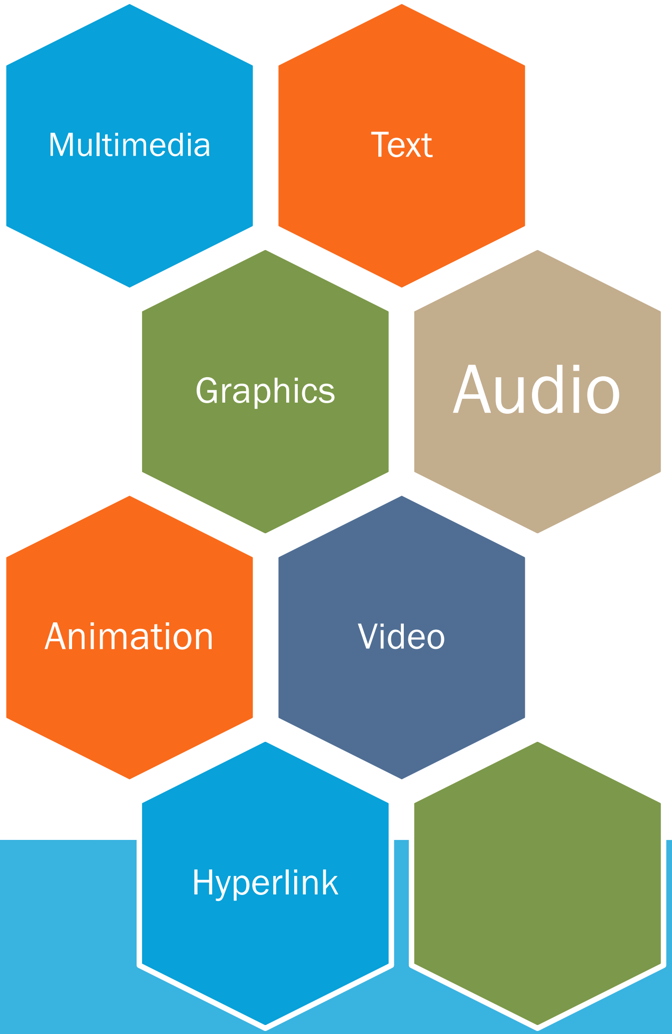
How Web Browser Works



Types Of Web Sites



Elements Of Web Page

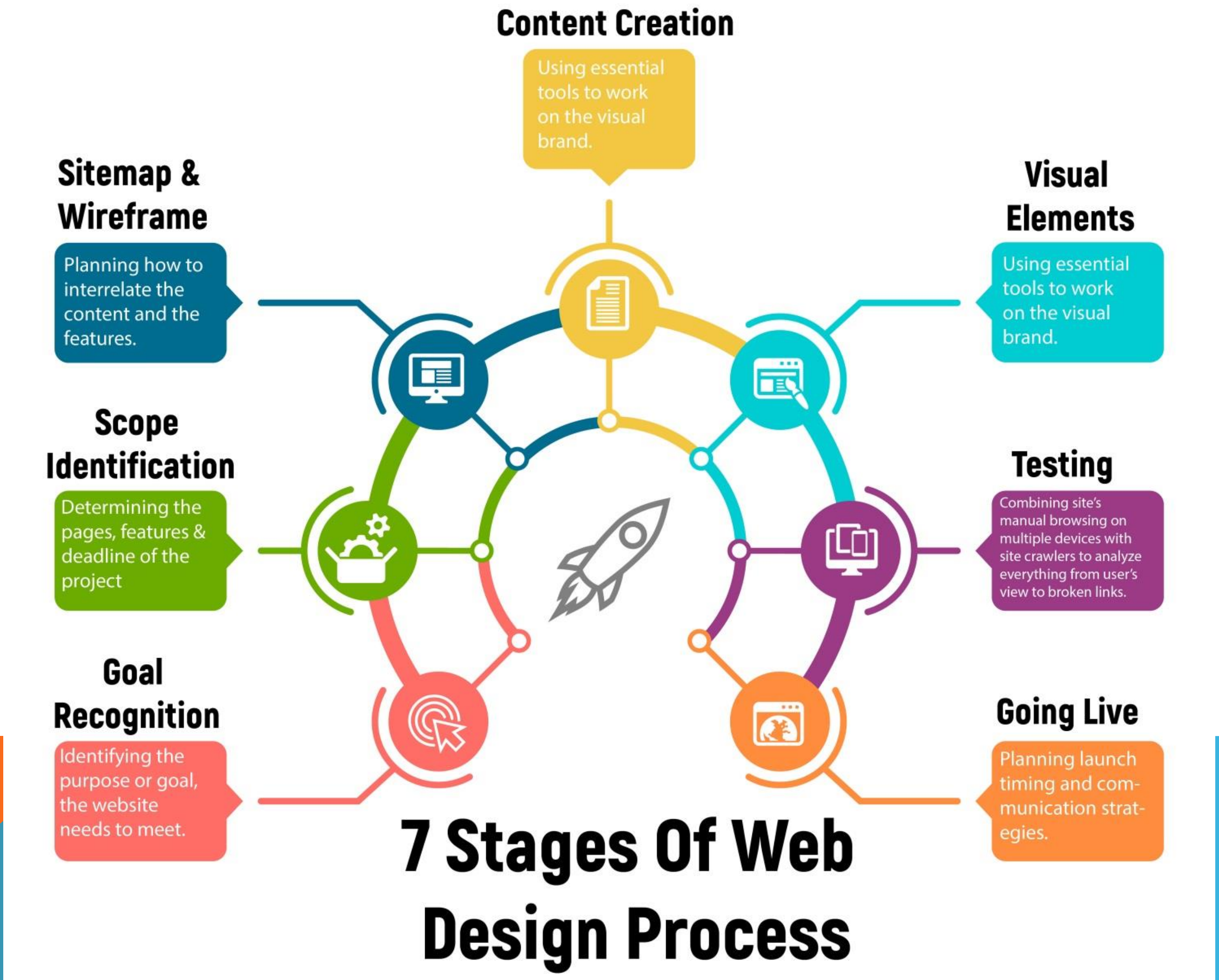


What Is Web Designing

- Web design is the planning and creation of websites. This includes the information, user interface, site structure, navigation, layout, colors, fonts and imagery.

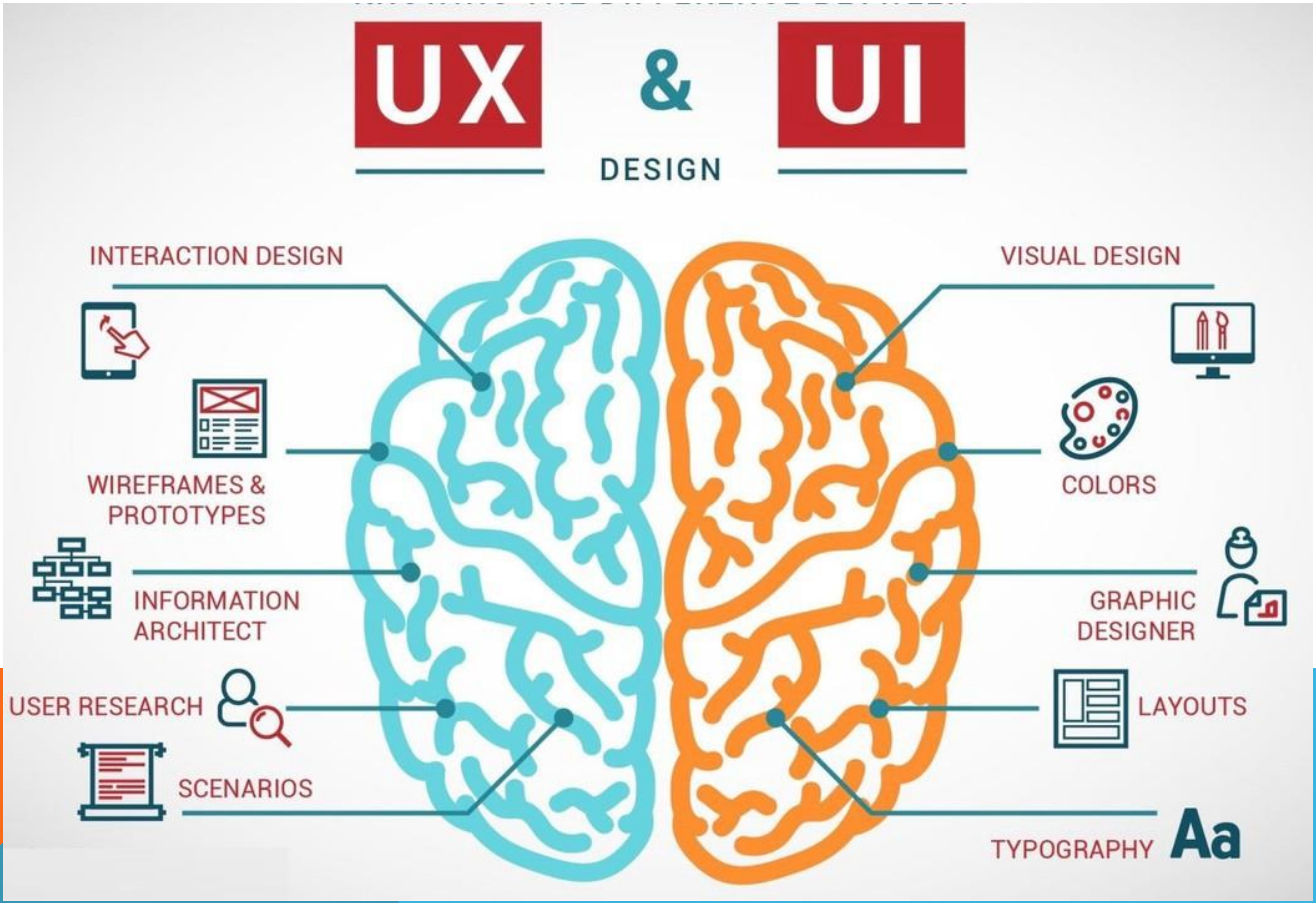
- All of these are combined with the principal of web designing to create a website that meets the goals of the owner and designer

Web Designing Process


Content Creation
Using essential tools to work on the visual brand.
Sitemap & Wireframe
Planning how to interrelate the content and the features.
Visual Elements
Using essential tools to work on the visual brand.
Scope Identification
Determining the pages, features & deadline of the project
Testing
Combining site's manual browsing on multiple devices with site crawlers to analyze everything from user's view to broken links.
Goal Recognition
Identifying the purpose or goal, the website needs to meet.
Going Live
Planning launch timing and com-munication strat-egies.
7 Stages Of Web Design Process

Web Development Technologies

USER EXPERIENCE (UX) AND USER INTERFACE (UI)

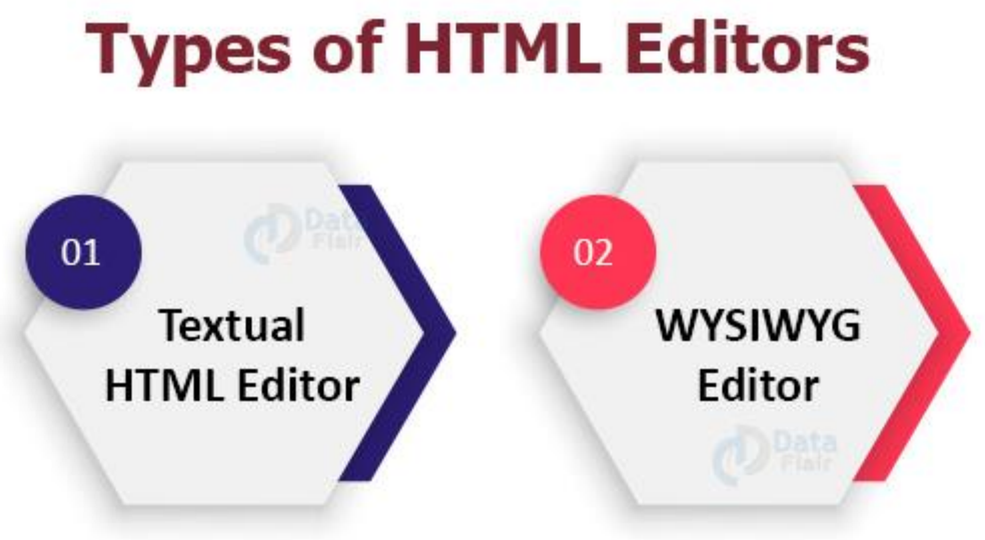


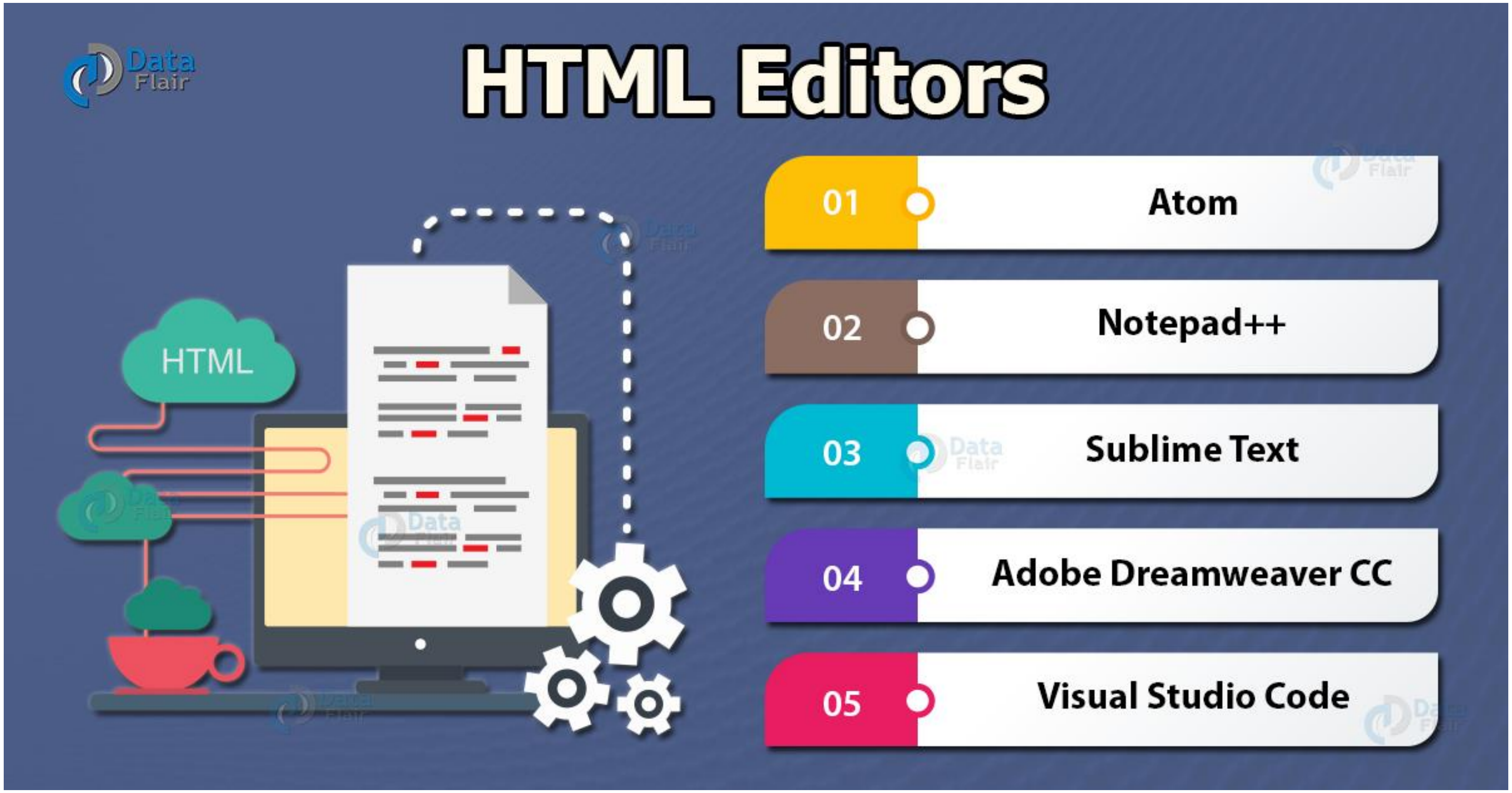
INTRODUCTION TO HTML

What Is HTML?

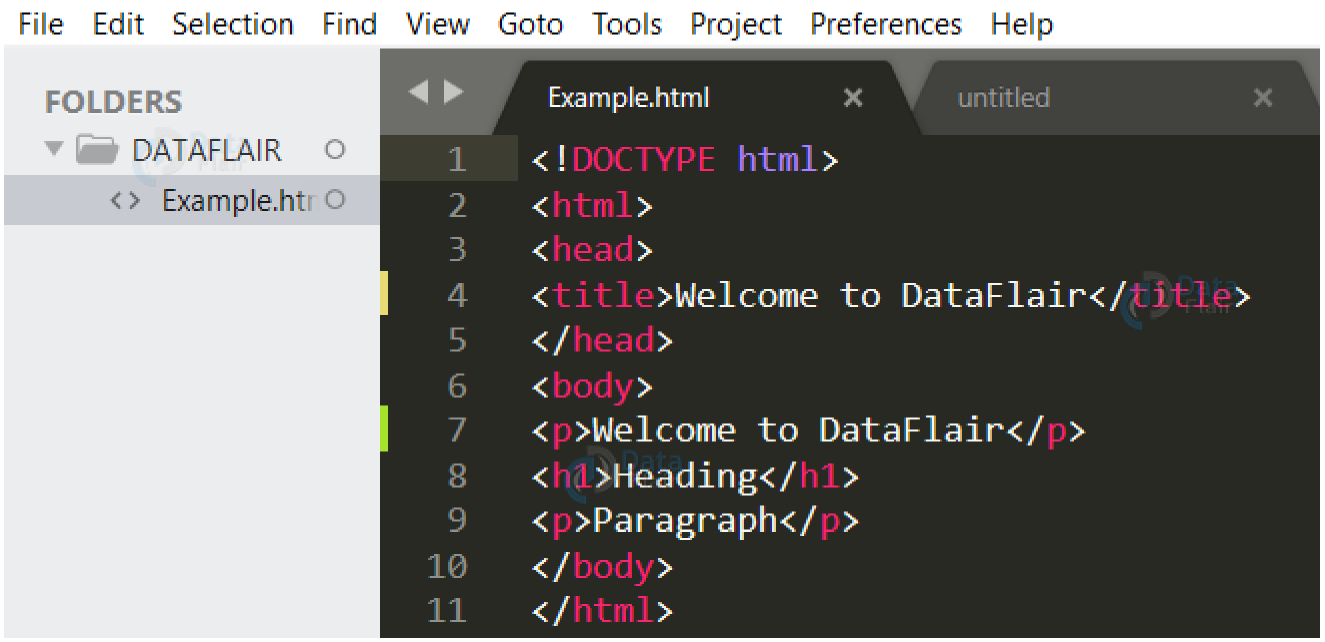
- HTML, also known as HyperText Markup Language, is the language used to create Web pages
- Using HTML, you can create a Web page with text, graphics, sound, and video

HTML Editors

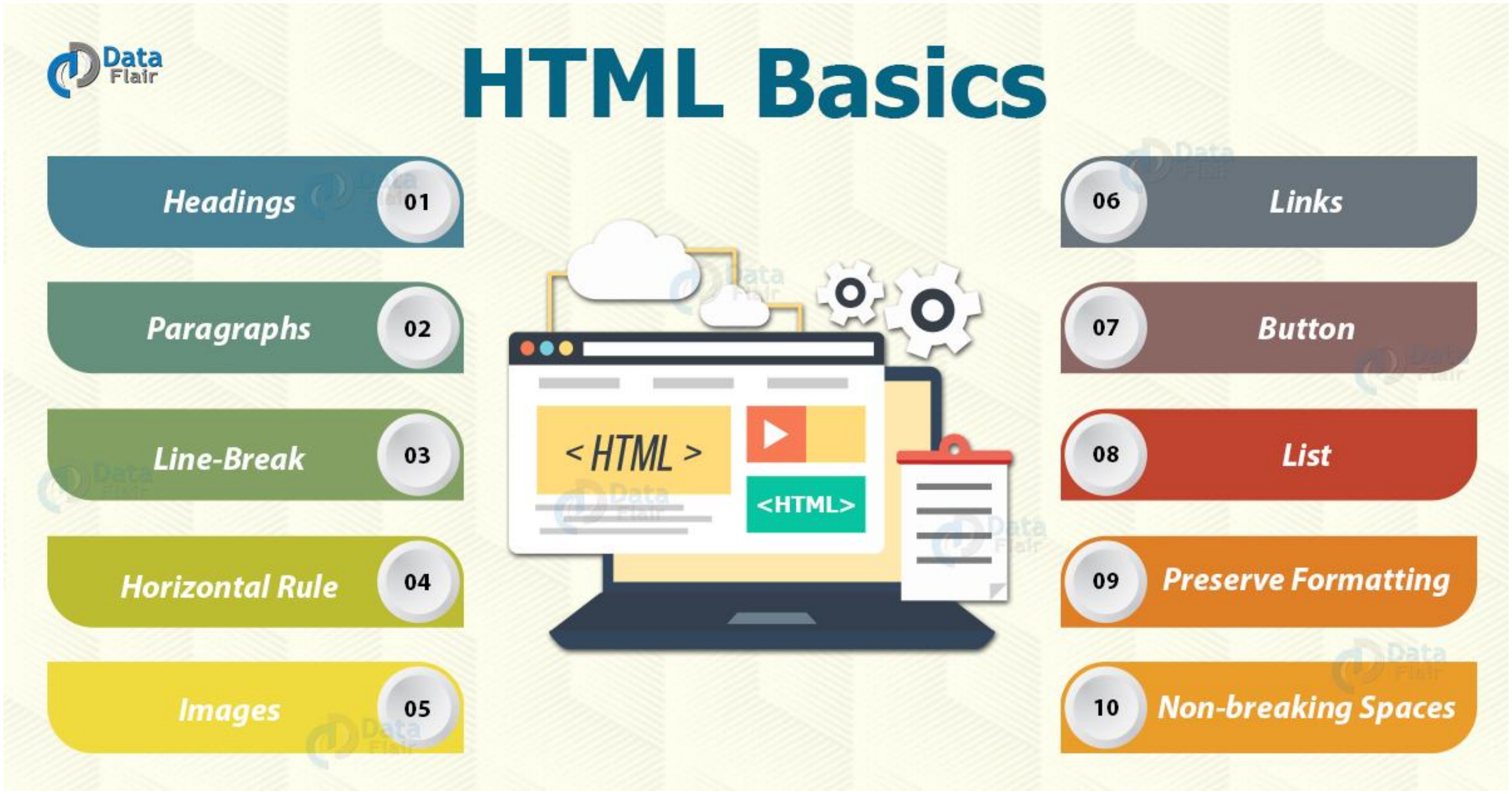



Data Flair
HTML Editors
HTML
01 Atom
02 Notepad++
03 Sublime Text
04 Adobe Dreamweaver CC
05 Visual Studio Code

Basic HTML Page



```html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to DataFlair</title>
</head>
<body>
<p>Welcome to DataFlair</p>
<h1>Heading</h1>
<p>Paragraph</p>
</body>
</html>
```


Data Flair
HTML Basics
Headings 01
Paragraphs 02
Line-Break 03
Horizontal Rule 04
Images 05
06 Links
07 Button
08 List
09 Preserve Formatting
10 Non-breaking Spaces
< HTML >
<HTML>

HTML ELEMENTS

The HTML **element** is everything from the start tag to the end tag:

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

Examples of some HTML elements:

<h1>My First Heading</h1>

<p>My first paragraph.</p>

Tags

- ❑ The essence of HTML programming is tags
- ❑ A tag is a keyword enclosed by angle brackets (Example: <I>)
- ❑ There are opening and closing tags for many but not all tags; The affected text is between the two tags
- ❑ The opening and closing tags use the same command except the closing tag contains and additional forward slash /
- ❑ For example, the expression <B> Warning </B> would cause the word ‘Warning’ to appear in bold face on a Web page

Nested Tags

- ❑ Whenever you have HTML tags within other HTML tags, you must close the nearest tag first

Example:

```
    <H1> <I> The Nation </I> </H1>
```

HTML ATTRIBUTES

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

- **The href Attribute**

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

- **The src Attribute**

```
<img src="img_girl.jpg“ />
<br>
```

- **The width and height Attributes**

```
<img src="img_girl.jpg" width="500" height="600">
```

- **The style Attribute**

```
<p style="color:red;">This is a red paragraph.</p>
```

THE <TITLE> TAG

- Choose the title of your Web page carefully; The title of a Web page determines its ranking in certain search engines

- The title will also appear on Favorite lists, History lists, and Bookmark lists to identify your page

TEXT FORMATTING

- ❑ Manipulating text in HTML can be tricky; Oftentimes, what you see is NOT what you get

- ❑ For instance, special HTML tags are needed to create paragraphs, move to the next line, and create headings

Text Formatting Tags

- <B> **Bold Face** </B>
- <I> *Italics* </I>
- <U> Underline </U>
- <P> New Paragraph </P>
- <BR> Next Line

CHANGING THE FONT

- The expression <FONT FACE = “fontname”> … </FONT> can be used to change the font of the enclosed text

- To change the size of text use the expression <FONT SIZE=n> …. </FONT> where n is a number between 1 and 7

CHANGING THE FONT

- To change the color, use <FONT COLOR="red">.... </FONT>; The color can also be defined using hexadecimal representation (Example: #ffffff)

- These attributes can be combined to change the font, size, and color of the text all at once; For example, <FONT SIZE=4 FACE="Courier" COLOR="red"> .... </FONT>

HEADINGS

- ❑ Web pages are typically organized into sections with headings; To create a heading use the expression <Hn>….</Hn> where n is a number between 1 and 6

- ❑ In this case, the 1 corresponds to the largest size heading while the 6 corresponds to the smallest size

ALIGNING TEXT

- The ALIGN attribute can be inserted in the <P> and <Hn> tags to right justify, center, or left justify the text

- For example, <H1 ALIGN=CENTER> Times of India</H1> would create a centered heading of the largest size

COMMENT STATEMENTS

- Comment statements are notes in the HTML code that explain the important features of the code
- The comments do not appear on the Web page itself but are a useful reference to the author of the page and other programmers
- To create a comment statement use the <!--- .... --> tags

PAGE FORMATTING

- To define the background color, use the BGCOLOR attribute in the <BODY> tag
- To define the text color, use the TEXT attribute in the <BODY> tag
- To define the size of the text, type <BASEFONT SIZE=n>

EXAMPLE

```
<HTML>
<HEAD>
<TITLE> Example </TITLE>
</HEAD>
<BODY BGCOLOR=“black”  TEXT=“white”>
<BASEFONT SIZE=7>
    This is where you would include the text and images on your Web
    page.
</BODY>
</HTML>
```

INSERTING IMAGES

- Type <IMG SRC = “image.ext”>, where image.ext indicates the location of the image file

- The WIDTH=n and HEIGHT=n attributes can be used to adjust the size of an image

- The attribute BORDER=n can be used to add a border n pixels thick around the image

ALTERNATE TEXT

- Some browsers don’t support images. In this case, the ALT attribute can be used to create text that appears instead of the image.

- Example:
- <IMG SRC=“satellite.jpg” ALT = “Picture of satellite”>

LINKS

- A link lets you move from one page to another, play movies and sound, send email, download files, and more….

- A link has three parts: a destination, a label, and a target

- To create a link type

```
<A HREF="page.html"> label </A>
```

EXAMPLE: LINKS

- **To create a link to IMD type:**

```
<A HREF=“https://mausam.imd.gov.in/”>IMD</A>
```

- **To create a link to IITM pune type:**

```
<A HREF=“https://www.tropmet.res.in/”>IITM Pune</A>
```

ANCHORS

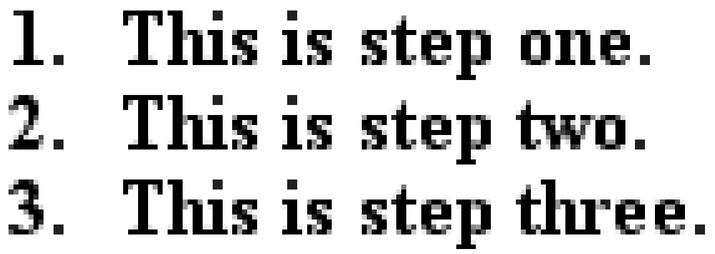
- Anchors enable a user to jump to a specific place on a Web site
- Two steps are necessary to create an anchor. First you must create the anchor itself. Then you must create a link to the anchor from another point in the document.
- To create the anchor itself, type <A NAME=“anchor name”>label</A> at the point in the Web page where you want the user to jump to
- To create the link, type <A HREF=“#anchor name”>label</A> at the point in the text where you want the link to appear

ORDERED LISTS

- Ordered lists are a list of numbered items.
- To create an ordered list, type:

```
<OL>
   <LI> This is step one.
   </LI>
   <LI> This is step two.
   <LI> This is step three.
</OL>
```

Here's how it would look on the Web:



1. This is step one.
2. This is step two.
3. This is step three.

MORE ORDERED LISTS….

The TYPE=x attribute allows you to change the the kind of symbol that appears in the list.

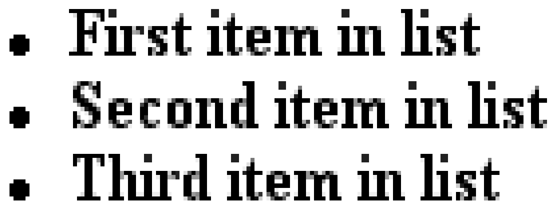
- A is for capital letters
- a is for lowercase letters
- I is for capital roman numerals
- i is for lowercase roman numerals

UNORDERED LISTS

- An unordered list is a list of bulleted items
- To create an unordered list, type:

```
<UL>
   <LI> First item in list
   <LI> Second item in list
   <LI> Third item in list
</UL>
```

Here's how it would look on the Web:



- First item in list
- Second item in list
- Third item in list

MORE UNORDERED LISTS…

The TYPE=shape attribute allows you to change the type of bullet that appears

- *circle* corresponds to an empty round bullet
- *square* corresponds to a square bullet
- *disc* corresponds to a solid round bullet; this is the default value

FORMS

What are forms?

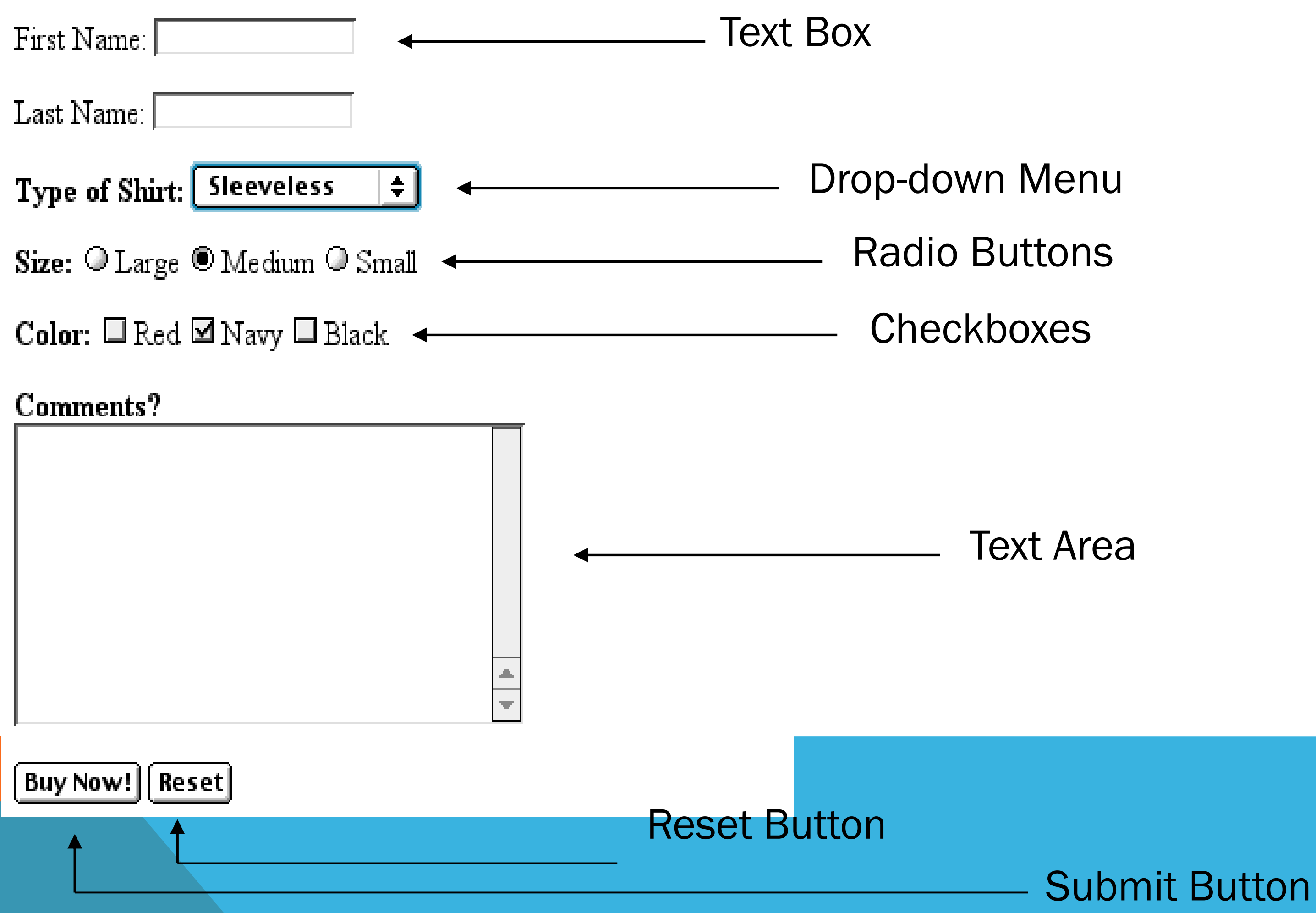
- An HTML form is an area of the document that allows users to enter information into fields.
- A form may be used to collect personal information, opinions in polls, user preferences and other kinds of information.

FORMS

- There are two basic components of a Web form: the shell, the part that the user fills out, and the script which processes the information

- HTML tags are used to create the form shell. Using HTML you can create text boxes, radio buttons, checkboxes, drop-down menus, and more…

EXAMPLE: FORM


First Name:

Last Name:

Type of Shirt: Sleeveless

Size: Large [✓ Medium] Small

Color: Red [✓ Navy] Black

Comments?

THE FORM SHELL

A form shell has three important parts:

- the <FORM> tag, which includes the address of the script which will process the form
- the form elements, like text boxes and radio buttons
- the submit button which triggers the script to send the entered information to the server

CREATING THE SHELL

- ❑ To create a form shell, type <FORM METHOD=POST ACTION="script_url"> where "script_url" is the address of the script
- ❑ Create the form elements
- ❑ End with a closing </FORM> tag

CREATING TEXT BOXES

- To create a text box, type <INPUT TYPE=“text” NAME=“name” VALUE=“value” SIZE=n MAXLENGTH=n>

- The NAME, VALUE, SIZE, and MAXLENGTH attributes are optional

TEXT BOX ATTRIBUTES

- The NAME attribute is used to identify the text box to the processing script
- The VALUE attribute is used to specify the text that will initially appear in the text box
- The SIZE attribute is used to define the size of the box in characters
- The MAXLENGTH attribute is used to define the maximum number of characters that can be typed in the box

EXAMPLE: TEXT BOX

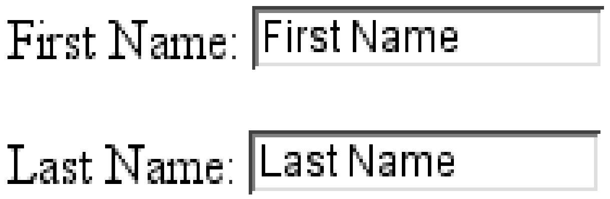
```
First Name: <INPUT
TYPE="text"
NAME="FirstName"
VALUE="First Name"
SIZE=20>

<BR><BR>


Last Name: <INPUT
TYPE="text"
NAME="LastName"
VALUE="Last Name"
SIZE=20>

<BR><BR>
```

Here’s how it would look on the Web:


First Name: First Name

Last Name: Last Name

CREATING RADIO BUTTONS

To create a radio button, type <INPUT TYPE="radio" NAME="name" VALUE="data">Label, where "data" is the text that will be sent to the server if the button is checked and "Label" is the text that identifies the button to the user

EXAMPLE: RADIO BUTTONS

```
 <B> Size: </B>
 <INPUT TYPE="radio" NAME="Size"
VALUE="Large">Large
 <INPUT TYPE="radio" NAME="Size"
VALUE="Medium">Medium
 <INPUT TYPE="radio" NAME="Size"
   VALUE="Small">Small
```

CREATING CHECKBOXES

- To create a checkbox, type <INPUT TYPE=“checkbox” NAME=“name” VALUE=“value”>Label

- If you give a group of radio buttons or checkboxes the same name, the user will only be able to select one button or box at a time

EXAMPLE: CHECKBOXES

```
          <B> Color: </B>
          <INPUT TYPE="checkbox" NAME="Color"
    VALUE="Red">Red
          <INPUT TYPE="checkbox" NAME="Color"
         VALUE="Navy">Navy
        <INPUT TYPE="checkbox" NAME="Color"
         VALUE="Black">Black
```

CREATING DROP-DOWN MENUS

- To create a drop-down menu, type <SELECT NAME="name" SIZE=n MULTIPLE>
- Then type <OPTION VALUE= "value">Label
- In this case the SIZE attribute specifies the height of the menu in lines and MULTIPLE allows users to select more than one menu option

EXAMPLE: DROP-DOWN MENU

```
<B>WHICH IS FAVOURITE FRUIT:</B>
<SELECT>
<OPTION VALUE="MANGOES">MANGOES
<OPTION VALUE="PAPAYA">PAPAYA
<OPTION VALUE="GUAVA">GUAVA
<OPTION VALUE="BANANA"> BANANA
<OPTION VALUE="PINEAPPLE">PINEAPPLE
</SELECT>
```

CREATING A SUBMIT BUTTON

- To create a submit button, type <INPUT TYPE="submit">
- If you would like the button to say something other than submit, use the VALUE attribute
- For example, <INPUT TYPE="submit" VALUE="Buy Now!"> would create a button that says "Buy Now!"

TABLES

- Tables can be used to display rows and columns of data, create multi-column text, captions for images, and sidebars

- The <TABLE> tag is used to create a table; the <TR> tag defines the beginning of a row while the <TD> tag defines the beginning of a cell

ADDING A BORDER

- The BORDER=n attribute allows you to add a border n pixels thick around the table

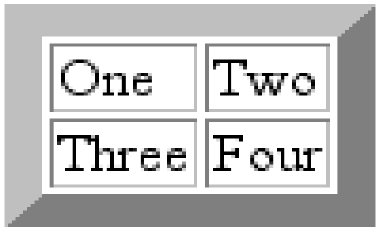
- To make a solid border color, use the BORDERCOLOR="color" attribute

- To make a shaded colored border, use BODERCOLORDARK="color" and BORDERCOLORLIGHT="color"

CREATING SIMPLE TABLE

```
<TABLE BORDER=10>
    <TR>
        <TD>One</TD>
        <TD>Two</TD>
    </TR>
    <TR>
        <TD>Three</TD>
        <TD>Four</TD>
    </TR>
</TABLE>
```

Here's how it would look on the Web:



One	Two
Three	Four

ADJUSTING THE WIDTH

- When a Web browser displays a table, it often adds extra space. To eliminate this space use the WIDTH =n attribute in the <TABLE> and <TD> tags

- Keep in mind - a cell cannot be smaller than its contents, and if you make a table wider than the browser window, users will not be able to see parts of it.

CENTERING A TABLE

There are two ways to center a table

- Type <TABLE ALIGN=CENTER>

- Enclose the <TABLE> tags in opening and closing <CENTER> tags

WRAPPING TEXT AROUND A TABLE

- ❑ It is possible to wrap text around a table. This technique is often used to keep images and captions together within an article.

- ❑ To wrap text around a table, type <TABLE ALIGN = LEFT> to align the table to the left while the text flows to the right.

- ❑ Create the table using the <TR>, <TD>, and </TABLE> tags as you normally would

ADDING SPACE AROUND A TABLE

To add space around a table, use the HSPACE=n and VSPACE=n attributes in the <TABLE> tag

Example:

```
<TABLE HSPACE=20 VSPACE=20>
```

ALIGNING CELL CONTENT

- By default, a cell’s content are aligned horizontally to the left and and vertically in the middle.

- Use VALIGN=direction to change the vertical alignment, where “direction” is top, middle, bottom, or baseline

- Use ALIGN=direction to change the horizontal alignment where “direction” is left, center, or right

CONTROLLING CELL SPACING

- Cell spacing is the space *between* cells while cell padding is the space *around* the contents of a cell

- To control both types of spacing, use the CELLSPACING =n and CELLPADDING=n attributes in the <TABLE> tag

NESTING TABLES

- Create the inner table
- Create the outer table and determine which cell of the outer table will hold the inner table
- Test both tables separately to make sure they work
- Copy the inner table into the cell of the outer table
- Don’t nest too many tables. If you find yourself doing that, find an easier way to lay out your Web page

CHANGING A CELL'S COLOR

To change a cell's color, add the BGCOLOR="color" attribute to the <TD> tag

Example:

```
   <TD BGCOLOR="blue">
```

HTML GRAPHICS

HTML Canvas Graphics

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

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

HTML SVG Graphics

SVG defines vector-based graphics in XML format.

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

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

INTRODUCTION TO XML

- XML stands for EXtensible Markup Language.
- It is a markup language much like HTML.
- It was designed to describe data, not to display data.
- It’s tags are not predefined. We must define our own tags.
- If we need to display dynamic data in our HTML document, it will take a lot of work to edit the HTML each time the data changes.
- With XML, data can be stored in separate XML files.
- This way we can concentrate on using HTML for display, and be sure that changes in the underlying data will not require any changes to the HTML.
- With a few lines of JavaScript code, we can read an external XML file and update the data content of our web page.

It has following advantages:

☐ Simplifies data sharing.

☐ Simplifies data transport.

☐ Simplifies platform changes.

☐ Makes the data more available.

INTRODUCTION TO CONTENT MANAGEMENT SYSTEM (CMS)

- ❑ Content Management System (CMS) can be defined as a tool or software program containing a set of interrelated programs used for creating and managing different digital or online content.
- ❑ Some famous examples of CMS software are **Joomla**, **Drupal**, **WordPress**, **TYPO3**, etc.

Components of CMS

- ❑ A **content management application (CMA)** is a graphical user interface that allows its users to create, delete, modify, and publish content even without the knowledge of HTML or other programming languages that are necessary to create web pages.
- ❑ A **content delivery application (CDA)** is responsible for the back-end services. It manages as well as delivers content after framed in the CMA.

FEATURE OF CMS

- **User Management**: This permits the management of user information like the roles of different users allotted to work simultaneously, such as creating or deleting the user, change the username, password, and other related information.
- **Theme System**: This allows us to modify the site view as well as functionality using stylesheets, images, and templates.
- **Extending Plugins:** Different plugins are offered, which gives custom functionalities and features to create the CMS site.
- **Search Engine Optimization**: It is embedded with a lot of search engine optimization (SEO) tools making content SEO more straightforward.
- **Media Management**: is used for managing the media files and folder, with uploading media contents easy and effortless.
- **Multilingual**: Translation of the language, as preferred by the user, is possible through CMS.

ADVANTAGE OF CMS

- Most of the CMS is open source and is available for free.
- Easy and quick uploading of media files can be done.
- Several SEO tools make on-site SEO simpler.
- Easy customization is possible as per the need of the user.
- It can modify CSS files as per the design needed by the user.
- Many templates and plugins are available for free. Customization of plugins is also possible.
- Content editing is also more comfortable as it uses the WYSIWYG editor.

INTRODUCTION TO CSS

WHAT IS CSS

- Cascading Style Sheets
- Contains the rules for presentation of HTML
    - HTML + CSS = Web page
- CSS was introduced to keep the presentation information separate from HTML markup (content)

WHY CSS?

- ❑ **CSS saves time:** You can write CSS once and reuse the same sheet in multiple HTML pages.
- ❑ **Easy Maintenance:** To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
- ❑ **Search Engines:** CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.
- ❑ **Superior styles to HTML:** CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- ❑ **Offline Browsing:** CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

CSS SYNTAX:

- A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document.
- A style rule set consists of a selector and declaration block.
  - Selector -- h1
  - Declaration -- {color:blue; font size:12px;}
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
  For Example:
  –; color is property and blue is value.
  –; font-size is property and 12px is value.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

USING CSS IN HTML

Inline - by using the style attribute inside HTML elements

Internal - by using a <style> element in the <head> section

External - by using a <link> element to link to an external CSS file

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-
color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

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

</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

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

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

</body>
</html>
```

"styles.css":

```
body {
  background-
color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS SELECTORS

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

- **THE UNIVERSAL SELECTORS:** Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type

```
* {
        color: #000000;
  }
```

This rule renders the content of every element in our document in black.

- **THE ELEMENT SELECTOR:** The element selector selects elements based on the element name. You can select all p elements on a page like this (in this case, all p elements will be center-aligned, with a red text color)

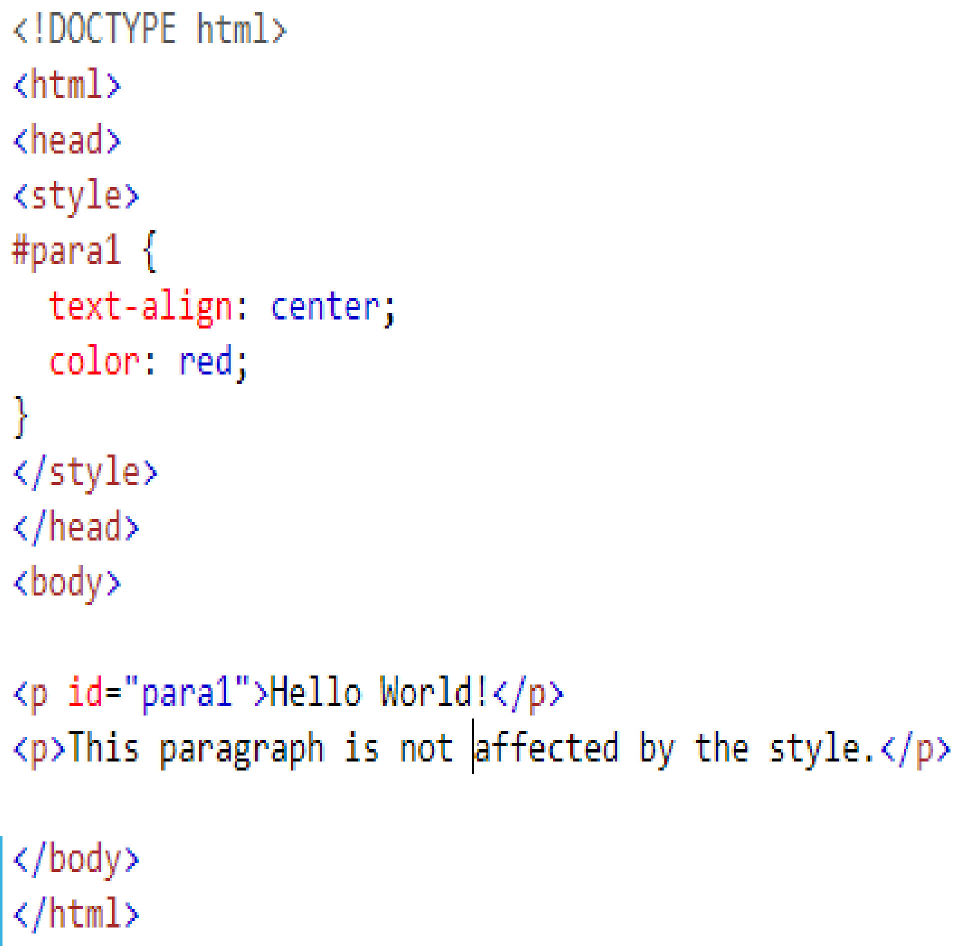
```
p {
        text-align: center;
        color: red;
}
```

- **THE DESCENDANT SELECTOR:** Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to the em element only when it lies inside the ul tag.

```
ul em {
        color: #000000;
}
```

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 style rule below will be applied to the HTML element with id="para1":



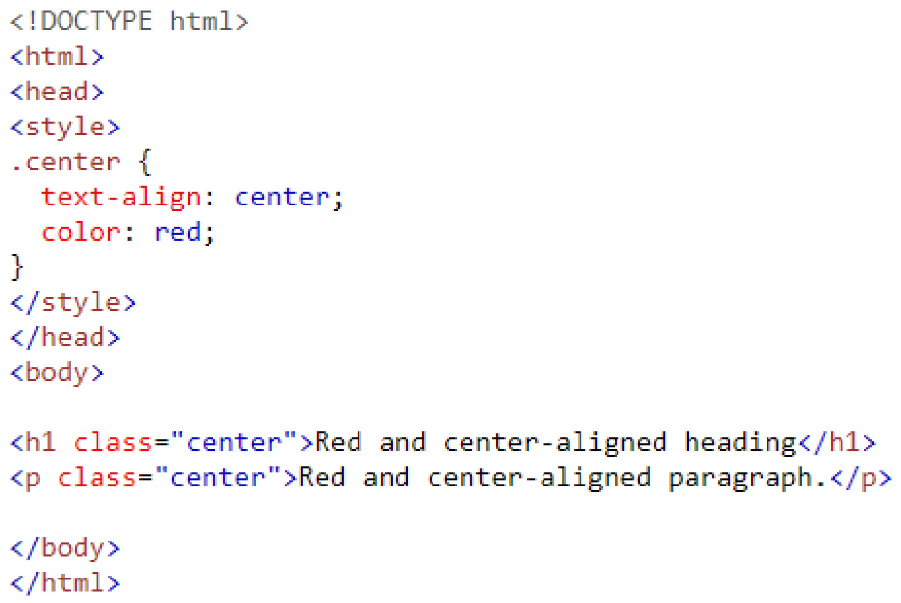
```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

THE CLASS SELECTORS

- ❑ 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.
- ❑ In the example below, all HTML elements with class=”center” will be red and center-aligned:
- ❑ You can apply more than one class selector to a given element.



```
<!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>
```

GROUPING SELECTORS

```
h1 {
   text-align: center;
   color: blue;
}

h2 {
   text-align: center;
   color: blue;
}

p {
   text-align: center;
   color: blue;
}
```

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma.

```
h1, h2, p {
    text-align: center;
    color: blue;
}
```

List of CSS attributes:
https://www.w3schools.com/cssref/default.asp

CSS Style for body :

https://www.w3schools.com/tags/tag_body.asp

CSS Style for image :

https://www.w3schools.com/css/css3_images.asp

CSS Style for hn :

https://www.w3schools.com/tags/tag_hn.asp

CSS Style for list

https://www.w3schools.com/css/css_list.asp

CSS Style for link

https://www.w3schools.com/css/css_link.asp

CSS Style for link

https://www.w3schools.com/css/css_table.asp

Introduction To JavaScript

WHAT IS JAVASCRIPT (JS)?

- A lightweight programming language ("scripting language")
- Used to make web pages interactive
- Insert dynamic text into HTML (ex: user name)
- React to events (ex: page load user click)
- Get information about a user's computer (ex: browser type)
- Perform calculations on user's computer (ex: form validation)

WHAT IS JAVASCRIPT (JS)?

- JS is interpreted
- JS is relaxed about syntax, rules, and types
- JS is case-sensitive
- JS is more object-oriented
- JS focuses on user interfaces and interacting with a document
- JS code runs on the client's browser

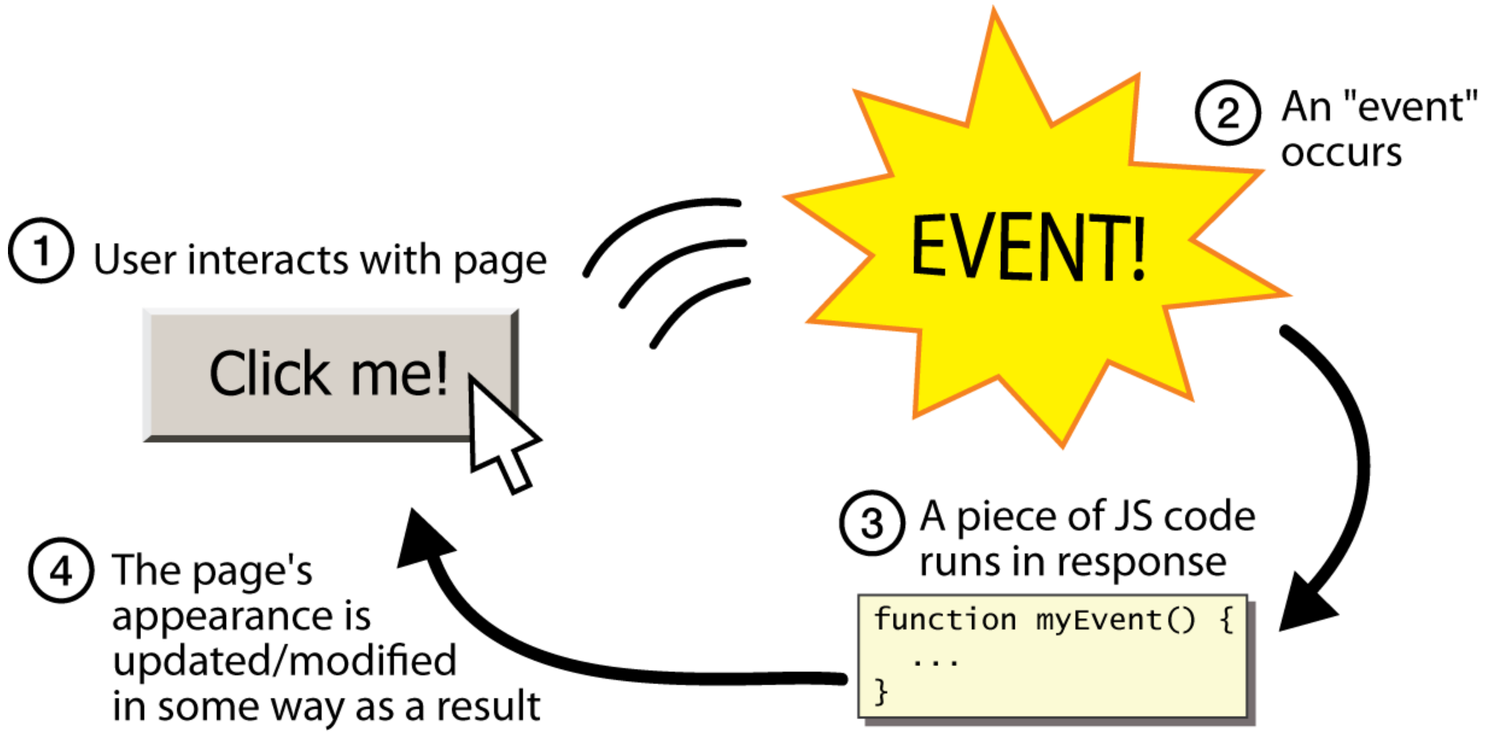
LINKING TO A JAVASCRIPT FILE: `SCRIPT`

```
<script src="filename" type="text/javascript"></script>
```
HTML

- script tag should be placed in HTML page's head
- script code is stored in a separate .js file
- JS code can be placed directly in the HTML file's body or head (like CSS)

but this is bad style (should separate content, presentation, and behavior

EVENT-DRIVEN PROGRAMMING


1 User interacts with page
Click me!
EVENT!
2 An "event" occurs
3 A piece of JS code runs in response
4 The page's appearance is updated/modified in some way as a result

```
function myEvent() {
  ...
}
```

- JavaScript programs instead wait for user actions called events and respond to them
- event-driven programming: writing programs driven by user events
- Let's write a page with a clickable button that pops up a "Hello, World" window...

BUTTONS

```
<button>Click me!</button>                                   HTML
```

- ❑ button's text appears inside tag; can also contain images
- ❑ To make a responsive button or other UI control:
  1. choose the control (e.g. button) and event (e.g. mouse 1. click) of interest
  2. write a JavaScript function to run when the event occurs
  3. attach the function to the event on the control

JAVASCRIPT FUNCTIONS

```
function name() {
statement ;
statement ;
...
statement ;
}                                                          JS
```

```
function myFunction() {
      alert("Hello!");
      alert("How are you?");
}                                                          JS
```

- the above could be the contents of example.js linked to our HTML page
- statements placed into functions can be evaluated in response to user events

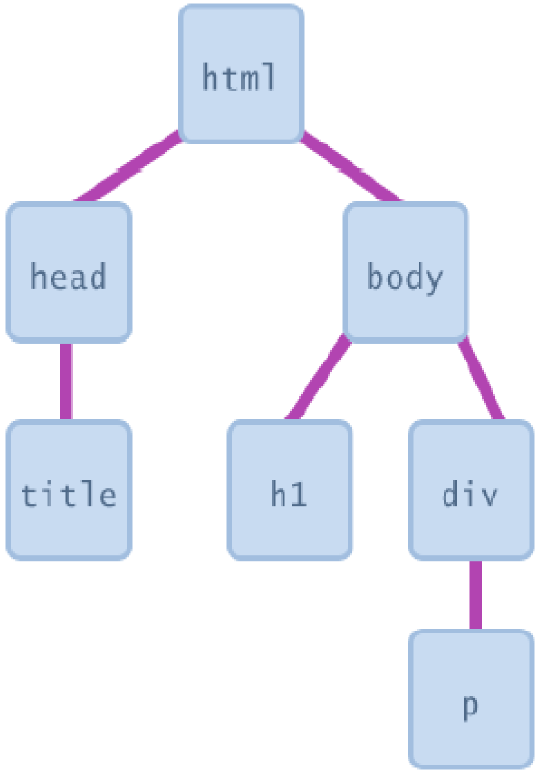
EVENT HANDLERS

```
<button onclick="myFunction();">Click me!</button>
```
HTML

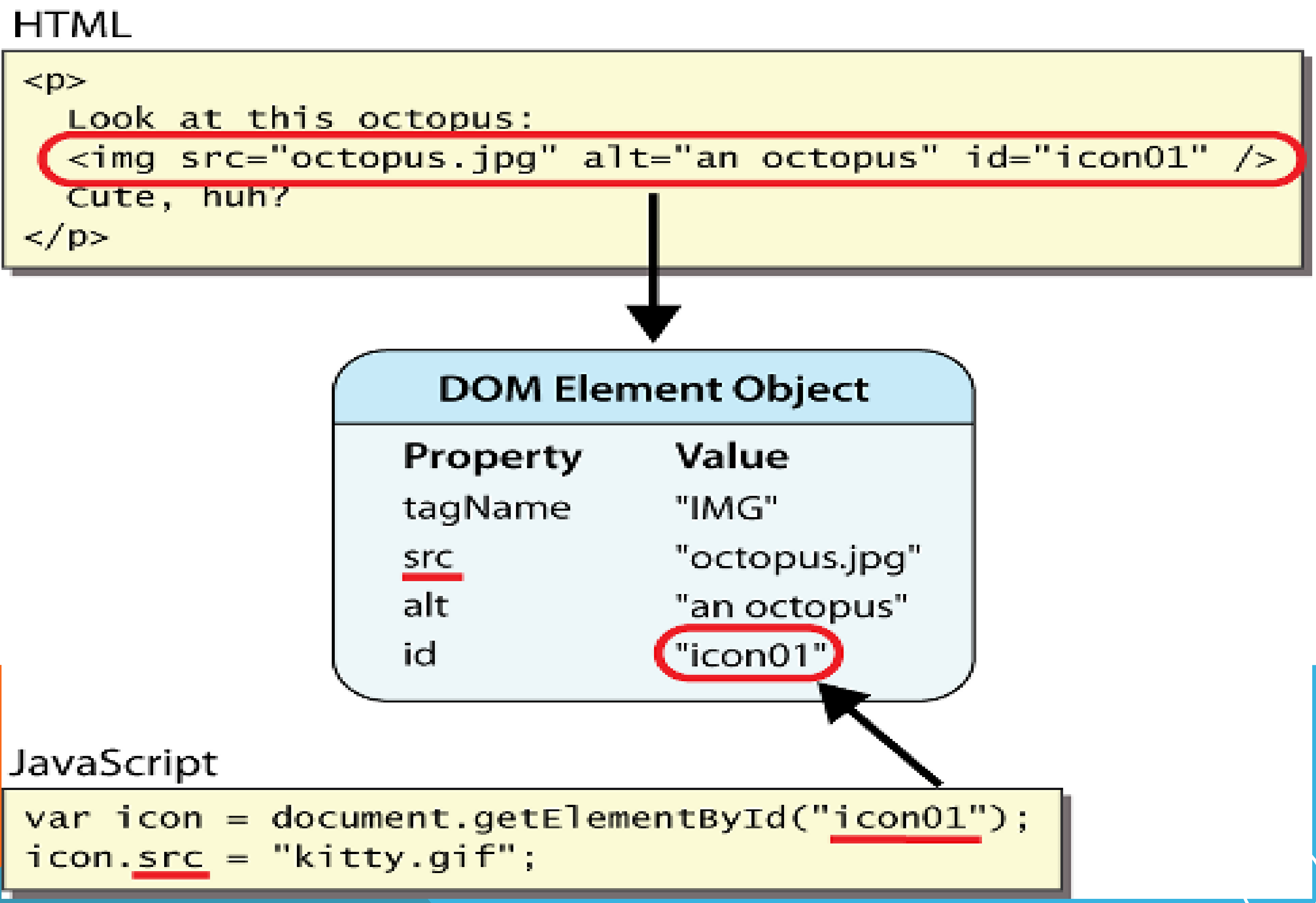
- JavaScript functions can be set as event handlers
- when you interact with the element, the function will execute
- onclick is just one of many event HTML attributes

DOCUMENT OBJECT MODEL (DOM)

- most JS code manipulates elements on an HTML page
- we can examine elements' state
  e.g. see whether a box is checked
- we can change state
  e.g. insert some new text into a div
- we can change styles
  e.g. make a paragraph red



DOM ELEMENT OBJECTS


HTML

```
<p>
  Look at this octopus:
  <img src="octopus.jpg" alt="an octopus" id="icon01" />
  Cute, huh?
</p>
```

Property	Value
tagName	"IMG"
src	"octopus.jpg"
alt	"an octopus"
id	"icon01"

JavaScript

```
var icon = document.getElementById("icon01");
icon.src = "kitty.gif";
```

ACCESSING ELEMENTS:

- document.getElementById returns the DOM object for an element with a given id
- can change the text inside most elements by setting the innerHTML property
- can change the text in form controls by setting the value property

ACCESSING ELEMENTS:

```
var name = document.getElementById("id");
                                                       JS
```

```
<button onclick="changeText();">Click me!</button>
<span id="output">replace me</span>
<input id="textbox" type="text" />                   HTML
```

```
function changeText() {
       var span = document.getElementById("output");
       var textBox = document.getElementById("textbox");

        textbox.style.color = "red";

}                                                      JS
```

INTRODUCTION TO PHP

CLIENT/SERVER ON THE WWW

- Standard web sites operate on a request/response basis
- A user requests a resource E.g. HTML document
- Server responds by delivering the document to the client
- The client processes the document and displays it to user

SERVER SIDE PROGRAMMING

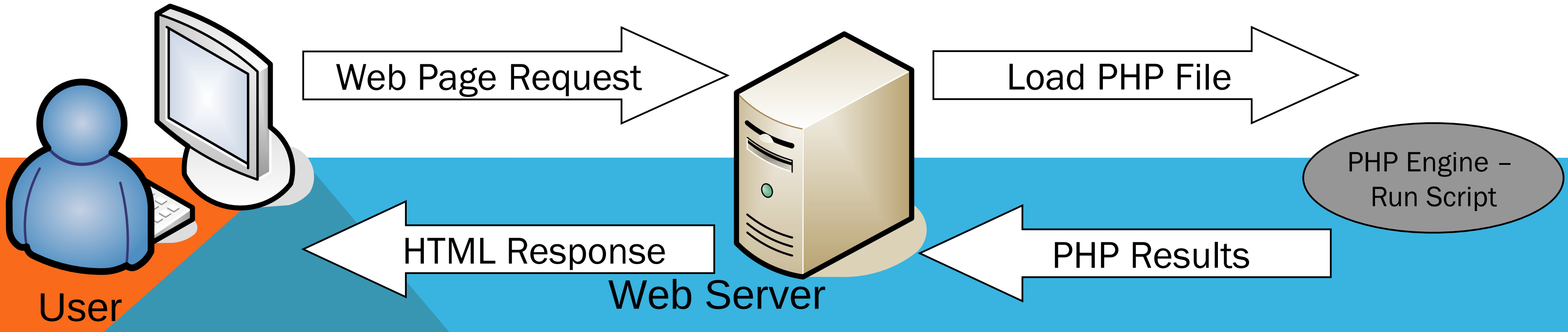
- ❑ Provides web site developers to utilise resources on the web server
- ❑ Non-public resources do not require direct access from the clients
- ❑ Most server side programming script is embedded within markup.

PHP - WHAT IS IT / DOES IT DO?

- PHP: <u>P</u>HP <u>H</u>ypertext <u>P</u>re-processor
- Programming language that is interpreted and executed on the server
- Execution is done before delivering content to the client
- Contains a <u>vast</u> library of functionality that programmers can harness
- Executes entirely on the server, requiring no specific features from the client

PHP - WHAT IS IT / DOES IT DO?

- ❑ Static resources such as regular HTML are simply output to the client from the server
- ❑ Dynamic resources such as PHP scripts are processed on the server prior to being output to the client
- ❑ PHP has the capability of connecting to many database systems making the entire process transparent to the client



PHP LANGUAGE BASICS

- Syntax and structure
- Variables, constants and operators
- Data types and conversions
- Decision making IF and switch
- Interacting with the client application (HTML forms)

PHP - SYNTAX AND STRUCTURE

All scripts start with <?php and with with ?>
Line separator: ; (semi-colon)
Code block: { //code here } (brace brackets)
White space is generally ignored (not in strings)
Comments are created using:

- // single line quote
- /* Multiple line block quote */

Precedence

- Enforced using parentheses
- E.g. $sum = 5 + 3 * 6; // would equal 23
- $sum = (5 + 3) * 6; // would equal 48

PHP - VARIABLES

Prefixed with a $

Assign values with = operator

Example: $author = “Trevor Adams”;

No need to define type

Variable names are <u>case sensitive</u>

- $author and $Author are different

PHP - EXAMPLE SCRIPT

<?php

- $author = “Trevor Adams”;
- $msg = “Hello world!”;
- echo $author . “ says ” . $msg;

?>

PHP - CONSTANTS

Constants are special variables that cannot be changed

Use them for named items that will not change

Created using a define function

- define(‘milestokm’, 1.6);
- Used without $
- $km = 5 * milestokm;

PHP - OPERATORS

Standard mathematical operators

- +, -, *, / and % (modulus)

String concatenation with a period (.)

- $car = “SEAT” . “ Altea”;
- echo $car; would output “SEAT Altea”

Basic Boolean comparison with “==”

- Using only = will overwrite a variable value
- Less than < and greater than >
- <= and >= as above but include equality

PHP - DATA TYPES

PHP is not strictly typed

- Different to JAVA where all variables are declared

A data type is either text or numeric

- PHP decides what type a variable is
- PHP can use variables in an appropriate way automatically

E.g.

- $vat_rate = 0.175; /* VAT Rate is numeric */
- echo $vat_rate * 100 . “%”; //outputs “17.5%”
- $vat_rate is converted to a string for the purpose of the echo statement

PHP - EMBEDDED LANGUAGE

PHP can be placed directly inside HTML E.g.

<html>

- <head><title>Basic PHP page</title></head>
- <body>
- <h1><?php echo “Hello World!; ?></h1>
- </body>

</html>

DECISION MAKING - BASICS

Decision making involves evaluating Boolean expressions (true / false)

If($catishungry) { /* feed cat */ }

“true” and “false” are reserved words

Initialise as $valid = false;

Compare with ==

AND and OR for combinations

- E.g. if($catishungry AND $havefood) {/* feed cat*/}

PHP - IF STATEMENT

Used to perform a conditional branch

If (Boolean expression) {

- // one or more commands if true

} else {

- // one or more commands if false

}

PHP - SWITCH STATEMENTS

Useful when a Boolean expression may have many options E.g.

switch($choice) {

- case 0: { /* do things if choice equal 0 */ }
- Case 1: {/* do things if choice equal 1 */ }
- Case 2: {/* do things if choice equal 2 */ }
- Default: {/* do if choice is none of the above */}

}

PHP - DEALING WITH THE CLIENT

Text fields

Checkbox

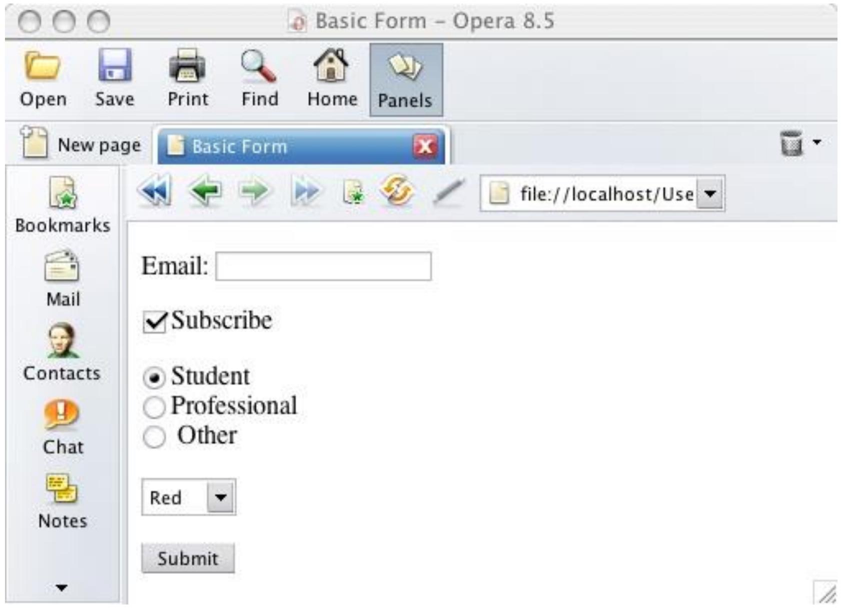
Radio button

List boxes

Hidden form fields

Password box

Submit and reset buttons



Email:

✓ Subscribe

● Student
○ Professional
○ Other

Red

PHP - DEALING WITH THE CLIENT

<form method="post" action="file.php" name="frmid" >

- Method specifies how the data will be sent
- Action specifies the file to go to. E.g. file.php
- id gives the form a unique name

Post method sends all contents of a form with basically hidden headers (not easily visible to users)

Get method sends all form input in the URL requested using name=value pairs separated by ampersands (&)

- E.g. process.php?name=trevor&number=345

  Is visible in the URL shown in the browser

PHP - DEALING WITH THE CLIENT

All form values are placed into an array

Assume a form contains one textbox called "txtName" and the form is submitted using the post method, invoking process.php

process.php could access the form data using:

- $_POST['txtName']

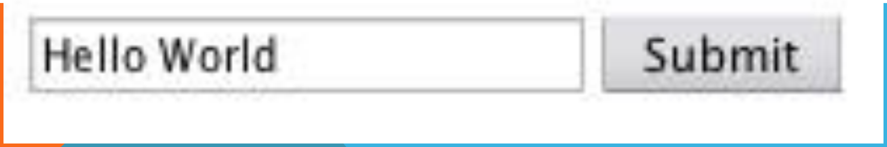
If the form used the get method, the form data would be available as:

- $_GET['txtName']

PHP - DEALING WITH THE CLIENT

For example, an HTML form:

- <form id="showmsg" action="show.php" method="post">
  - <input type="text" id="txtMsg" value="Hello World" />
  - <input type="submit" id="submit" value="Submit">
- </form>



INSTALLATION OF WEB SERVER

https://www.c-sharpcorner.com/article/how-to-install-wamp-server-in-windows-10/

https://www.instructables.com/How-to-Run-a-PHP-Script-With-Wamp-Server/

PRE-REQUISITE

Web Server

PHP

Database

DETAILED DESCRIPTION OF PHP AND ITS FUNCTION

https://www.w3schools.com/php/