

Web User Interface

HTML, CSS & JAVASCRIPT



Enabling Objectives

After completing this chapter, you will be able to explain about the basics of Web User Interface

Key Topics

- HTML
- CSS
- JavaScript

HTML

HTML

Stands for Hyper Text Mark Up Language

- It is a mark up language for designing web pages.
- HTML pages can be viewed using any web browsers.
- Cannot perform any programming logic.
- HTML pages developed using mark up tags enclosed in angular brackets(<tag>)
- Can be developed using normal text editors like notepad/wordpad or tools like dreamweaver, front page.
- File extension is “.html” (or) “.htm”

Identify the applications and benefits of HTML

<http://www.vtech-seo.com/web-design-articles/advantages-of-html.html>

What is an HTML mark up tag ?

HTML mark up tags called as HTML elements are the building blocks of any HTML document

HTML tags occurs in pairs called the opening and closing tag

Tag Syntax:

```
<tag attributes>Content</tag>
```

Attributes : Provide additional behavior to the tags

Content : The text content to be displayed in the page

Basic HTML Structure

```
<html>
```

```
  <head>
```

```
    .....  
    .....  
  </head>
```

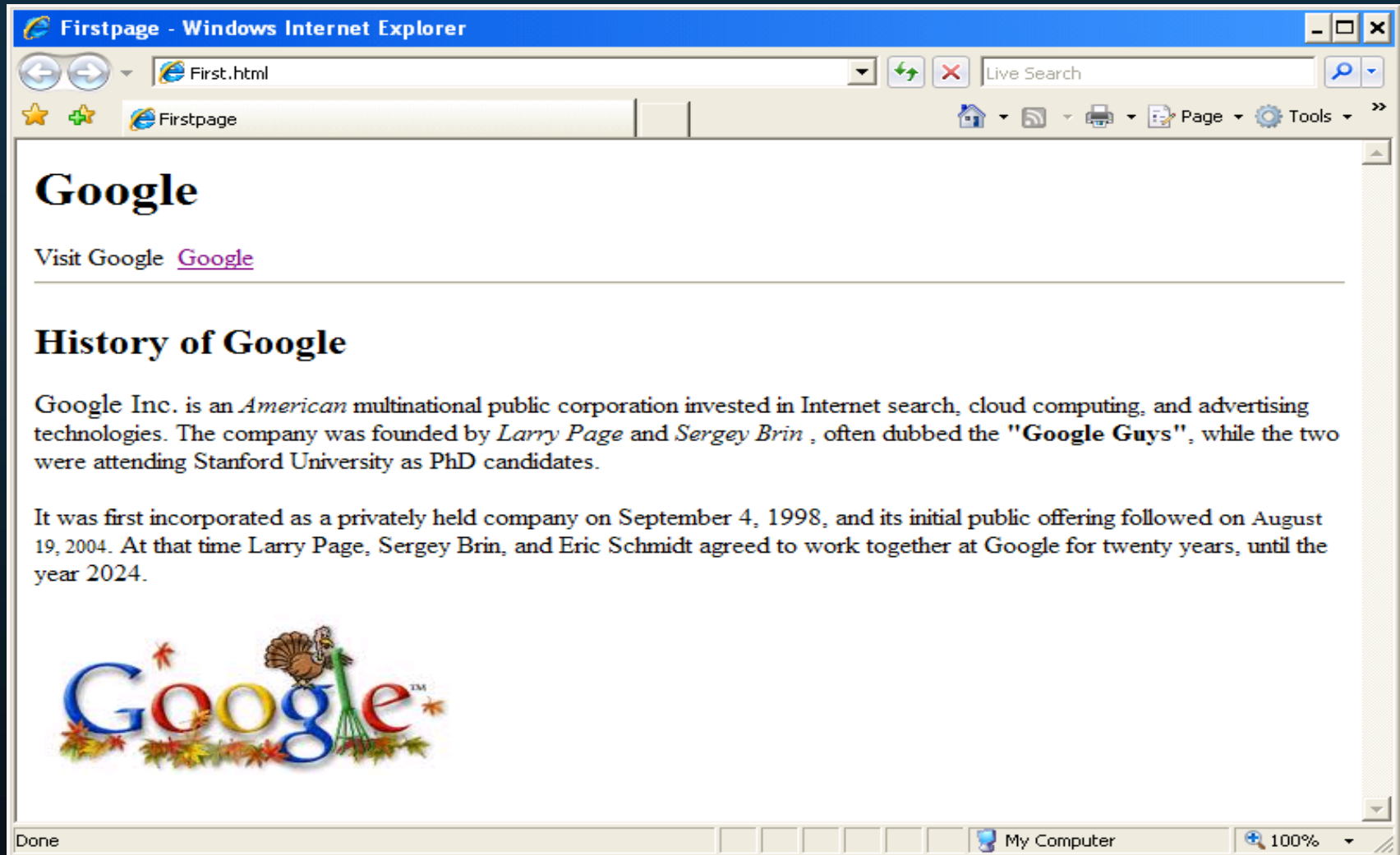
```
  <body>
```

```
    .....  
    .....  
  </body>
```

```
</html>
```

Basic Structure
of a HTML file.

Lets work together to build this page



Learn How

Ready to Develop ?

We will do some initial steps before we start

1. Open notepad, create a new text file.
2. Save it as MyFirstPage.html.

Further Steps:

As we go through each HTML tags we will add that tag in our MyFirstPage.html

Demonstrate usage of Header, Footer and Section tags appropriately

https://www.w3schools.com/html/html5_semantic_elements.asp

Root, Head and Body Tags

HTML Root Tag : Encloses the entire HTML document, this is the root element.

`<html>`

`</html>`

Head Tag : Container for all head elements like title, scripts, meta information etc

`<head>`

.....

`</head>`

Body Tag : This holds the other tags and the content which needs to be rendered

on the browser.

`<body>`

`</body>`

Heading Tag

Heading tags are used to define headings in HTML pages,
There six heading tags `<h1>` to `<h6>`

`<h1>` Denotes the most important heading

`<h6>` Denotes the least important heading.

Syntax:

`<h1 id="heading1">` This is heading 1 `<h1>`

Where `id` is the attribute used for accessing the element.

Try It out – Lets play With Headings



Add the following headings to first.html

1. Main Heading – Google
2. Sub Heading – History of Google

Google

History of Google

Paragraph tag

This tag is used to format texts as paragraph

Inserts a line before and after each paragraph tag

Apple iPhone

The iPhone is a line of Internet and multimedia-enabled smartphones marketed by Apple Inc. The first iPhone was unveiled by Steve Jobs, then CEO of Apple, on January 9, 2007,[1] and released on June 29, 2007. The 5th generation iPhone, the iPhone 4S, was announced on October 4, 2011, and released on October 14, 2011, two days after the release of iOS 5.0, the Apple operating system for handheld devices.

An iPhone can function as a video camera (video recording was not a standard feature until the iPhone 3GS was released), a camera phone, a portable media player, and an Internet client with email and web browsing capabilities, can send texts and receive visual voicemail, and has both Wi-Fi and 3G connectivity.

Paragraph 1:

One line appears between the heading and paragraph 2

Paragraph 2

Syntax :

`<p id="ID">Paragraph Content goes in here </p>`

Lend a Hand: Adding Paragraph

Add two paragraphs below the sub heading "History Of Google".
The content can be anything about Google.

History of Google

Google Inc. is an *American* multinational public corporation invested in Internet search, cloud computing, and advertising technologies. The company was founded by *Larry Page* and *Sergey Brin*, often dubbed the "**Google Guys**", while the two were attending Stanford University as PhD candidates.

It was first incorporated as a privately held company on September 4, 1998, and its initial public offering followed on August 19, 2004. At that time Larry Page, Sergey Brin, and Eric Schmidt agreed to work together at Google for twenty years, until the year 2024.

Lists

Lists are used to list items in an HTML page.

Two flavors Of List,

1. Ordered Lists
2. Unordered Lists

Ordered Lists

Definition: Delineates a list, where the items are in sequential, numerical order

1. Apple
2. Orange
3. Mango

Syntax :

```
<ol>  
<li>List Item1</li>  
<li>List Item2</li>  
<li>List Item3</li>  
</ol>
```

Example :

```
<ol>  
<li>Apple</li>  
<li>Orange</li>  
<li>Mango</li>  
</ol>
```

Un Ordered Lists

Definition: delineates a list, where the items are generally of equal importance and they are listed in bullets.

- Apple
- Orange
- Mango

Syntax :

```
<ul>  
<li>List Item1</li>  
<li>List Item2</li>  
<li>List Item3</li>  
</ul>
```

Example :

```
<ul>  
<li>Apple</li>  
<li>Orange</li>  
<li>Mango</li>  
</ul>
```

Lend a Hand - Lists

Create a set of Ordered and Unordered lists as illustrated below

Ordered Lists

Fruits

1. Apple
2. Orange
3. Mango

Unordered List

Countries

- India
- Australia
- Srilanka

Lend a Hand - Solution

```
<html>
<head>
<title>Lists</title>
</head>
<body>
<h1>Ordered Lists</h1>
<h4>Fruits</h4>
<ol>
  <li>Apple</li>
  <li>Orange</li>
  <li>Mango</li>
</ol>
<h1>Unordered List</h1>
<h4>Countries</h4>
<ul>
  <li>India</li>
  <li>Australia</li>
  <li>Srilanka</li>
</ul>
</body>
</html>
```

Anchor Tag

Anchor tag(<a>) is used for creating hyperlinks which is a word or group of words. Hyperlink when clicked opens a new web page or web site or move to a section within the same document.

Anchor tag can be used for two purposes:

1. To create a link to another document
2. To create a bookmark inside a document

Syntax :

Click me url – refers to the page which needs to open when the text "Click Me" is clicked.

Example :

Link to Google Site

[Link to Google Site](http://www.google.com)

Try It Out: Lets add a Anchor and a Image tag

Add the following to the

1. Add a link **Google** below the main heading.
2. Add an image below the paragraph.

Google

Visit Google [Google](#)

History of Google

Google Inc. is an *American* multinational public corporation invested in Internet search, cloud computing, and advertising technologies. The company was founded by *Larry Page* and *Sergey Brin* , often dubbed the "**Google Guys**", while the two were attending Stanford University as PhD candidates.

It was first incorporated as a privately held company on September 4, 1998, and its initial public offering followed on August 19, 2004. At that time Larry Page, Sergey Brin, and Eric Schmidt agreed to work together at Google for twenty years, until the year 2024.



Table tag

Table tag is used to represent data in rows and columns in an HTML page.

The `<table>` `</table>` is used for creating a table.

The following are the table tag elements that are used for creating the grid.

- `<TR>` - This is used for creating the row of the HTML table.
- `<TD>` - This is used for creating the columns of the HTML tables.
- `<TH>` - This is used to create the header row of the table.

The diagram shows an HTML table with three columns and three rows. The first row is the header, with cells labeled 'User Name', 'Contact #', and 'Column Heading 3'. The next two rows are data rows, currently empty. Annotations with lines pointing to the table components are as follows:

- A red rounded rectangle on the left contains the text: "All the three rows will be created using the `<TR>` tag." A red bracket points from this box to the three rows of the table.
- A pink box above the first column contains the text: "The heading column is created using TH." A line points from this box to the 'User Name' header cell.
- A pink box above the third column contains the text: "The data cells created using TD" (Note: the original image has a typo 'TD' instead of 'TH'). A line points from this box to the 'Column Heading 3' header cell.

User Name	Contact #	Column Heading 3

Syntax: Table tag

Syntax

```
<table id="id">
  <caption>TableName</caption>
  <tr>
    <th>ColumnName1</th>
    <th>ColumnName2</th>
  <tr>
    <td>Data1</td>
    <td>Data2</td>
  <tr>
    <td>Data1</td>
    <td>Data2</td>
  <tr>
  </table>
```

Example

```
<table id="users">
  <caption>User Details</caption>
  <tr>
    <th>Username</th>
    <th>Contact Number</th>
  <tr>
    <td>Arun</td>
    <td>9568748756</td>
  <tr>
    <td>Rose</td>
    <td>9254796541</td>
  <tr>
  </table>
```

Demonstrate usage of Blocks and Inline to align content within the document

https://www.w3schools.com/html/html_blocks.asp

HTML Forms

What are HTML Forms?

HTML Forms are user interface through which users inputs data and communicates with the web applications.

HTML Form
Sample



Registration Form

Name	<input type="text"/>
Password	<input type="password"/>
Address	<input type="text"/>
Country	<input type="text" value="Australia"/>
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
Languages Known	<input checked="" type="checkbox"/> English <input type="checkbox"/> Malayalam <input checked="" type="checkbox"/> Tamil
<input type="button" value="Clear"/>	<input type="button" value="Register"/>

Need for HTML form

Why do you need a HTML Form?

- Forms contain form fields (or) elements which are components used to accept inputs from the user.
 - Form elements example: Textbox, Label, Button etc.
- A form is simply an area where form fields can be placed.
- The data from the forms are sent to the server where it is read by the server using server side scripts/ CGI scripts Common Gateway interface.

Form Tag

Forms are created using form tags `<form></form>`.

Syntax:

```
<form method="method" action="action Name">
```

```
//Form Elements
```

```
</form>
```

The Form elements
are defined
between these tags.

You will learn more about “method” and “action” attribute in the next slides.

What is action attribute ?

Action Attribute specifies “where” to send the data when the user submits the form.

- Specifies which component is going to handle the form at the server
- This specifies the server side component which is going to process the request.

Server Side component Example : Servlets, Struts , JSF.

Example :

```
<form action="LoginServlet">
```

//Form Controls goes in here

```
</form>
```



Here when the user submits the form it will be handled by a **Login servlet** running inside a web server

Form Tag

Forms are created using form tags `<form></form>`.

Syntax:

```
<form method="method" action="action Name">  
    //Form Elements  
</form>
```

The Form elements
are defined
between these tags.

You will learn more about “method” and “action” attribute in the next slides.

What is action attribute ?

Action Attribute specifies “where” to send the data when the user submits the form.

- Specifies which component is going to handle the form at the server
- This specifies the server side component which is going to process the request.

Server Side component Example : Servlets, Struts , JSF.

Example :

```
<form action="LoginServlet">
```

```
//Form Controls goes in here
```

```
</form>
```



Here when the user submits the form it will be handled by a **Login servlet** running inside a web server

What is method attribute ?

Method attribute are used to specify “how” to send the form data to the server.

The methods are,

Get : Form data is sent via the URL with appended to the URL.

URL data is referred to as query String.

Post : Form data is sent in the body of the HTTPrequest.

The default method value is “Get”

Example:

```
<form action="LoginServlet" method="post">
```

```
//Form controls are implemented here.
```

```
</form>
```

GET Vs POST

GET	POST
As the data is sent through the URL, there are limitation in the size of data sent.	There is no limitation in the data size sent.
Since the data is sent in URL it is visible to the user, this cannot be used for sending secured data.	The data is sent along with the HTTP request body so this method is secured.

GET URL Example:

<http://www.domain.com/forms.html?username=admin&password=admin>

The form field values are sent as key/value pairs key=value. Where the userName & password are the HTML form element attributes are sent as key and the values are depicted in brown color font.

Form Controls

Form Controls or form elements are used to accept input from the user. They are the basic building blocks of forms in HTML pages.

The basic form controls are

- Text box
- Password field
- Text Area
- Drop Down Menu
- Radio Button
- Check Box
- Buttons

Lend a Hand – Create a HTML

Lets create an user registration form with controls.

To start with lets all create a empty registration.html

NOTE: On completion of each form control we will add the control to registration.html. Use table for aligning the controls, add additional columns and rows to the table wherever required

Name and ID attribute for a form Control

A form control can have a name and id attribute.

```
<formcontrol name="name" id="id"></formcontrol>
```

- **name** attribute is used to refer the control at the server side using Server side script.
- **id** attribute is used to refer the control at the client side using Client side scripts and CSS

Label Tag

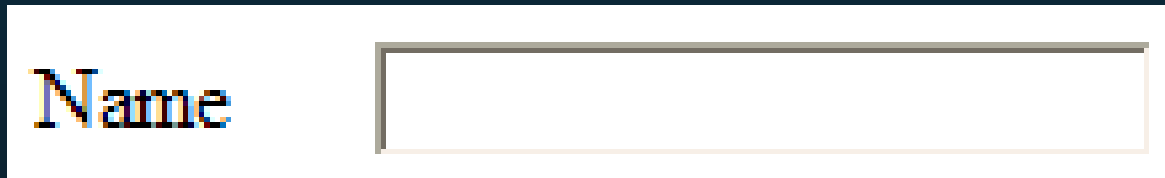
- A label tag is used to label a form control.
- Binds text to the form control.
- Uses `for` attribute to bind to a particular form control, where the value is the `id` of the form control.
- Clicking the label text activates the appropriate form control.

Usage :

- `<label for="elementID">Text</label>`
- `<formcontrol id="elementID" />`

Text Box

Used to accept single line of text from the user.

A screenshot of a web form element. It consists of a white rectangular container. On the left side of the container, the word "Name" is written in a blue, serif font. To the right of the text is a horizontal rectangular input field with a thin grey border and a light beige background.

Example of Text Fields: Name , Username, Phone number

How To Create a Text Box ?

To create a text box we use `<input>` tag

Syntax :

```
<input type="text" name="textName" id="id" value="defaultValue"/>
```

Example :

```
<input type="text" name="userName" id="userName"  
value="Enter user Name"/>
```

Optional Attributes for Textbox

- Size - Specifies the visible width, characters displayed.
- Maxlength - Specifies the maximum number of characters allowed.
- Tabindex - Sets the tab order for the control.
- Value - Default value for the textbox.

Lend A Hand – Text Fields

Lets all add a user name field to the registration form,

1. Create a label username in registration.html

```
<label>Name</label>
```

2. Create a textbox in registration.html

```
<input type =“text” name=“userName” id=“userName” />
```

3. Set the **for** attribute of the label . The value should be the text box's ID

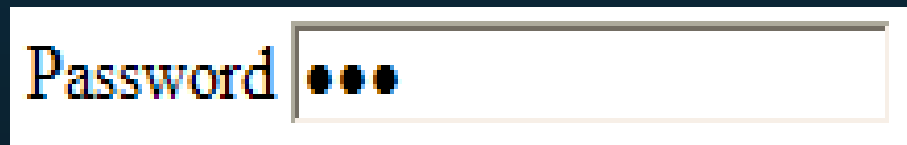
```
<label for=“userName”>Name</label>
```

4. Place the form fields inside a table tag, like

```
<table><tr>  
    <td> <label for=“userName”>Name</label></td>  
    <td> <input type =“text” name=“userName” id=“userName” /> </td>  
</tr> </table>
```

Password Field

Used to accept secure data from the user which should be displayed on the screen.

A screenshot of a password field. It consists of a label 'Password' in a serif font, followed by a rectangular text input box. Inside the input box, there are three black dots, indicating that the password is masked.

Uses : Passwords, ATM pin number etc

How To Create a password field?

To create a text box we use `<input>` tag

Syntax :

```
<input type="password" name="Name" id="id" />
```

Example :

```
<input type="password" name="password" id="password"  
value="Enter user Name"/>
```

Optional Attributes for password field

- Size - Specifies the visible width, characters displayed.
- Maxlength - Specifies the maximum number of characters allowed.
- Tabindex - Sets the tab order for the control.
- Value - Default value for the textbox.

Lend a Hand – Password Fields

Add the Password field to the registration form,

1. Create a label username in registration.html

```
<label>Password</label>
```

2. Create a password field in registration.html

```
<input type="password" name="password" id="password" />
```

3. Set the **for** attribute of the label . The value should be the password field's ID.

```
<label for="password">Password</label>
```

4. Place the form fields inside a table tag, like

```
<tr>  
  <td> <label for="password">Password</label></td>  
  <td> <input type="password" name="password" id="password"  
></td>  
</tr>
```

Hidden Fields

Hidden fields are similar to text field but does not show on the page. Therefore, the visitor cannot type anything into a hidden field.

When used?

- Used to submit information that is not entered by the visitor.
- Used as mechanism for carrying information between pages.

How to create a Hidden field ?

To create a hidden field we use `<input>` tag

Syntax :

```
<input type="hidden" name="Name" id="id" />
```

Example :

```
<input type="hidden" name="userType" id="userType"  
value="Administrator"/>
```

- The **name** setting adds an internal name to the field so the program that handles the form can identify the fields.
- The **value** setting defines what will be sent once the form is submitted.

Lend a Hand : Hidden Field

Add the Hidden field to the registration form,

1. Create a password field in registration.html

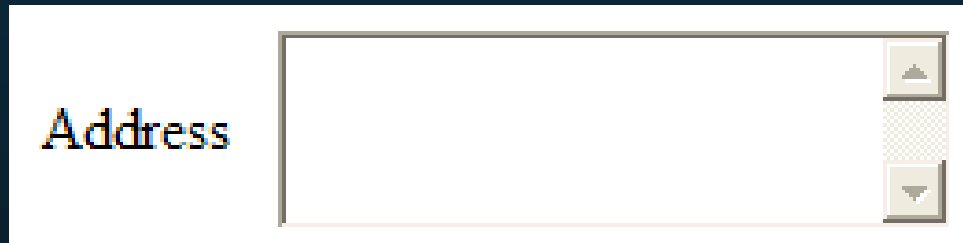
```
<input type="hidden" name="userType" id="userType"  
value="administrator" />
```

2. View the page in the browser.
3. The hidden field will not be seen in the page.

NOTE: This need not be included in the table tag.

Text Area

Used to accept multiline text input from the user

A screenshot of a web form element. On the left, the word "Address" is written in a blue, serif font. To its right is a large, empty rectangular text area with a thin grey border. On the right side of the text area, there are two small, stacked square buttons with a light grey background and a darker grey border. The top button has a small upward-pointing arrow, and the bottom button has a small downward-pointing arrow, indicating a scrollable area.

Used for Fields: Address , Description etc

How to create a Text Area ?

To create a text area we use the `<textarea>` tag

Syntax:

```
<textarea name="name" id="id" cols="columnCount"  
rows="rowCount"></textarea>
```

Example :

```
<textarea name="address" id="address" cols="10" rows="4">  
</textarea>
```

Optional Attributes for text area

- Rows - Specifies the number of rows.
- Cols - Specifies the number of columns.
- Wrapoff - Turns off line breaking.
- Tabindex - sets the tab order for the control.
- Virtual - Shows line breaking, but sends text as entered.
- Physical - Inserts line breaks when needed and even sends it.

Lend a Hand – Text Area

Add the Address field to the registration form,

1. Create a label username in registration.html

```
<label>Address</label>
```

2. Create a text area in registration.html

```
<textarea name="address" id="address" cols="10" rows="4">
</textarea>
```

3. Set the **for** attribute of the label . The value should be the text area's ID.

```
<label for="address">Address</label>
```

3. Place the form fields inside a table tag, like

- <tr>
- <td> <label for="address">Address</label></td>
- <td> <textarea name="address" id="address" cols="10" rows="4">
- </textarea>
- </td> </tr>

Check Box

Check boxes are used when you want to let the visitor select one or more options from a set of alternatives.

Languages Known	<input checked="" type="checkbox"/>	English
	<input type="checkbox"/>	Malayalam
	<input checked="" type="checkbox"/>	Tamil

How To Create a Check box?

To create a text box we use `<input>` tag

Syntax :

```
<input type = "checkBox" name = "Name" id = "id" value = "value" />
```

Example :

```
<input type = "checkbox" name = "language" id = "language1" value = "English" /> English
```

```
<input type = "checkbox" name = "language" id = "language2" value = "Tamil" /> Tamil
```

NOTE: All the check boxes should belong to the same group this is done by specifying the same name. Example: "Language".

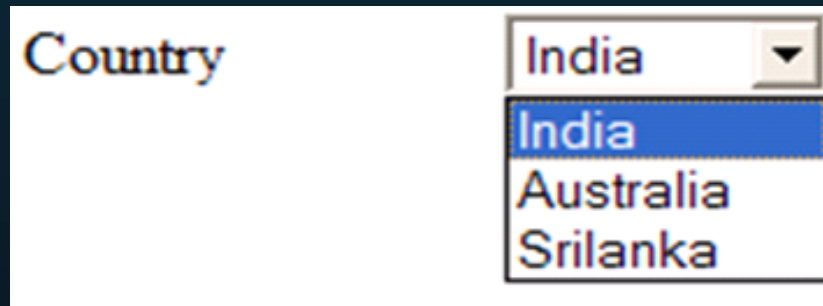
Optional Attributes for check box

- Checked - Default check this field
- Tabindex - Sets the tab order for the control.
- Value - Value that is submitted if checked

Drop Down

Drop-down menus can serve the same purpose as radio buttons (one selection only) or check boxes (multiple selections allowed).

The advantage of a drop-down menu, compared to radio buttons or check boxes, is that it takes up less space.



A screenshot of a web form. On the left, the label "Country" is displayed in a serif font. To its right is a drop-down menu. The menu is currently open, showing a list of three countries: "India", "Australia", and "Srilanka". The "India" option is highlighted with a blue background. The menu has a small downward-pointing arrow icon on its right side.

How to create a Drop Down Menu?

To create a text area we use the `<select>` tag. Each option is declared inside the `<option>` tag.

Syntax:

```
<select name="name" id="id" >
  <option value="value">
    Option1
  </option>
  <option value="value">
    Option2
  <option>
  </select>
```

Example :

```
<select name="country" id="country" >
  <option value="1" >
    India
  </option>
  <option value="2">
    Australia
  <option>
  </select>
```

Lend a Hand – Drop down

Add the Drop down menu to the registration form,

1. Create a label username in registration.html

```
<label>Country</label>
```

2. Create a drop down menu in registration.html

```
<select name="country" id="country" >  
  <option value="1" >India</option>  
  <option value="2">Australia<option>  
  <option value="3" selected="selected" >SriLanka<option>  
</select>
```

3. Place the form fields inside a table tag, like

- <tr><td > <label>Country</label> </td>
- <td > The drop down code snippet in point # 2 goes in here</td>
- </tr>

Radio Button

Radio buttons are used when you want to let the visitor select one and just one option from a set of alternatives.

Gender	<input type="radio"/> Male
	<input checked="" type="radio"/> Female

How To Create a radio button?

To create a text box we use `<input>` tag

Syntax :

```
<input type="radio" name="name" id="id" />
```

Example :

```
<input type="radio" name="gender" id="male"  
value="Male"/> Male
```

```
<input type="radio" name="gender" id="female"  
value="female"/> Female
```

Tips in Using Radio Button

- In a radio button group, all the buttons should be given the same name to make only one button in the group getting selected. Else all the buttons can be selected which does not serve the purpose of using radio button.
- The value attribute will be send to the server on page submit. It will not get printed in the html page. To display any text next to the radio button print it as normal text or using the label tag.
- When a label tag is set for a radio button clicking the label enables the corresponding radio button.

Lend a Hand – Radio Button

Add the radio button to the registration form,

1. Create a label username in registration.html

```
<label>Gender</label>
```

2. Create a radio buttons in registration.html

```
<input type="radio" name="gender" id="gender"
```

```
value="Male"/> <label for="male" >Male</label>
```

```
<input type="radio" name="gender" id="gender"
```

```
value="female"/><label for="female"> Female</label>
```

3. Clicking the label text will select the corresponding radio button.
4. Place the text and radio button in TD tags similar to the check box example.

Buttons

Buttons are used to perform some actions in an HTML page.

There are four types of buttons in HTML,

1. Submit
2. Reset
3. Button
4. Image button

Submit Button

A submit button is used to submit a form to the server.

When a visitor clicks a submit button, the form data is sent to the address specified in the action setting of the <form> tag.

Syntax :

```
<input type="submit" value="value" name="name" id="id" />
```

Where, **Value** defines what is written on the button, **Name** & **Id** for identifying the button among other form components.

Example :

```
<input type="submit" value="Register" name="register" id="register" />
```

Reset Button

Used to reset the values entered in the form by the user.

A reset button resets the form control to the default values of the form components.

Syntax :

```
<input type="reset" value="value" name="name" id="id" />
```

Where, **Value** defines what is written on the button, **Name** & **Id** for identifying the button among other form components.

Example :

```
<input type="reset" value="Clear" name="Clear" id="Clear" />
```

Button

They are used for adding normal buttons.

Used to perform action with help of client side scripts.

Syntax :

```
<input type="button" value="value" name="name" id="id" />
```

Where, **Value** defines what is written on the button, **Name** & **Id** for identifying the button among other form components.

Example :

```
<input type="button" value="Home" name="Clear" id="Clear" />
```

Lend a Hand – Submit & Reset

1. Add a submit button with value “Register” to register.html

```
<input type="submit" value="Register" name="register" id="register" />
```

2. Add a reset button with value “Clear” to register.html

```
<input type="reset" value="Clear" name="clear" id="id" />
```

3. Fill the form and check whether clicking the reset button clears the fields to the default value.

4. Fill the form and check whether the data and click submit . Check whether the fields entered by you has been appended in the URL as query string.

5. Use table tags as mentioned below

```
<tr><td> Submit button code mentioned in point # 1 goes in here</td>
```

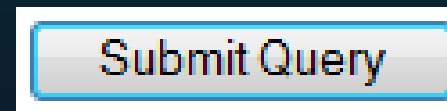
```
<td> reset button code mentioned in point # 2 goes in here</td></tr>
```

Image Button

- Image button are like the submit button are used to submit a form to the server.
- The difference is image button is shown as an image on the screen whereas submit button is shown as a normal button.



Submit Image
displayed as buttons.



Submit button

How to Create image button ?

Syntax :

```
<input type="image" src="source" name="name" />
```

Where “src” is the path of the image GIF.

Example :

```
<input type="submit" src="submit.gif" name="search" />
```

Hands On Exercise

Perform the following tasks,

1. Create a form as shown in the next slide alignment should be done proper using table tags.
2. Title should be “Apply for Credit Card”.
3. The labels should be properly bind to the controls.
4. The “First/Last name” text fields should display max of 12 characters and should not allow more than 10 characters.
5. Only one radio button option should be selected for sex.
6. Should be able to select multiple options in the “Language Known” drop drown.
7. The default value in dropdown should be “select”
8. The values in drop down are “English, Hindi, Tamil, Malayalam, Telugu”.
9. The address field should have 4 rows and 10 columns
10. The check box group should have four hobbies Singing, Dancing, Sports and Others.
11. There should one submit button named “Apply Card” and one reset button named “Clear Fields”

Required Screen

Apply for Credit Card

First Name

Last Name

Address

Languages Known

Sex ☒ Male ☐ Female

Hobbies ☐ Dancing ☐ Singing ☐ Sports ☐ Others

List the options available in HTML5 form fields

https://www.w3schools.com/html/html_form_input_types.asp

https://www.w3schools.com/html/html5_new_elements.asp

CSS

What is CSS?

CSS Stands for **C**ascading **S**tyle **S**heets.

- It is used for applying styles to HTML elements
 - Example: Font bold, Italics, Page back ground color.
 - It also isolates the page content from the style, thus helps the developers
-
- Advantages:
 - To easily maintain the HTML pages.
 - To reuse the same style in multiple pages thus reduces the development effort.

Explain the need and benefits of CSS

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

How CSS is applied ?

Syntax:

```
selector { property :value ; property :value; }
```

Where Selector can be a,

- Tag : Applies the style to all the tags of the specific type.

```
h1{color:red;} -- makes all Level 1 headings red
```

- Tag id : For applying to a single/unique element. Starts with '#'.

- ```
#m1 {color:red;} -- makes color of element with id "m1" as red.
```

- ```
<h1 id="m1">Center-aligned heading</h1>
```

- Tag class : For applying styles to more than one element. Starts with '.'
Element should be declared with the class attribute.

```
.headings{color:red;} --makes the color of elements with class  
name "headings" as red.
```

```
<h1 class="headings">Center-aligned heading</h1>
```

```
<p class="headings">Center-aligned paragraph.</p>
```

Three Ways to apply CSS

Three ways to apply CSS

- Inline Style
 - Internal Style
 - External Style
- Inline Style: The Style is declared inline in the html tag.

Example

```
<h1 style="color: red;">Red Heading </h1>
```

```
<body style="background-color: red;">Red Heading </body>
```

Disadvantages:

This mixes the content with the style , should be avoided

Internal Style Sheet

The styles are written inside the head tag.

```

<html>
  <head>
    <style type="text/css">
      body{
        background-color:Gray;
      }
      #main{
        color:green;
      }
      .subheadings{
        color:brown;
      }
    </style>
  </head>
  <body>
    <h1 id="main">Main Heading</h1>
    <h2 class="subheadings">SubHeading 1</h2>
    <h3 class=" subheadings ">SubHeading 2</h3>
  </body>
</html>

```

Diagram illustrating the internal style sheet structure and its effects:

- Page Back ground changed to gray** (applied to `body`)
- Main Heading color changed to green** (applied to `#main`)
- All sub headings color changed to brown** (applied to `.subheadings`)

Lend a Hand – Internal Style Sheet

Create an HTML file named CSSDemo.html and add the following code and run it.

```
<html>
  <head>
    <style type="text/css">
      body{
        background-color:Gray;
      }
      #main{
        color:green;
      }
      .subheadings{
        color:brown;
      }
    </style>
  </head>
  <body>
    <h1 id="main">Main Heading</h1>
    <h2 class="subheadings">SubHeading 1</h2>
    <h3 class=" subheadings ">SubHeading 2</h3>
  </body>
</html>
```

External Style Sheet

The styles will be written in a separate .css file and included in the HTML file. This is the ideal way as the same style can be reused in multiple pages

```
<html>
```

```
<head>
```

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

```
</head>
```

```
<body>
```

```
<h1 id="main">Main Heading</h1>
```

```
<h3 class="subheadings">SubHeading 1</h3>
```

```
<h3 class=" subheadings ">SubHeading 2</h3>
```

```
</body>
```

```
</html>
```

CSS File
included in the
HTML page.

Lend a Hand – External Style Sheet

Steps :

Step 1: Move the inline style from CSSDemo.html to styles.css

Step 2: Import styles.css into CSSDemo.html

Style.css

```
body{  
  
background-color:Gray;  
}  
  
#main{  
color:green;  
}  
  
.subheadings{  
color:brown;  
}
```

CSSDemo.html

```
<html>  
  
  <head>  
  
    <link rel="stylesheet" type="text/css" href="styles.css" />  
  
  </head>  
  
  <body>  
  
    <h1 id="main">Main Heading</h1>  
  
    <h3 class="subheadings">SubHeading 1</h3>  
  
    <h3 class=" subheadings ">SubHeading 2</h3>  
  
  </body>  
  
</html>
```

What happens if multiple styles are applied?

When multiple styles are applied in a single HTML the priority in which the styles will be applied as follows.

Priority 1: Inline Style

Priority 2: Internal Style

Priority 3: External Style

Priority 4: Browser default

Try it Out

Refer this URL for the styles to be applied,
http://www.w3schools.com/css/css_background.asp

Create a HTML and apply the styles

1. Create a page “home.html” with gray back ground
2. Create a Main Heading “My First HTML” with Blue Color
3. Place an image in the page. The image size should be 200x200.
4. Create links home, profile, logout.

The non visited links should be in brown color.

The visited links should be green color.

When the mouse moves over the link color should be orange .

5. Create two paragraphs with brown color and font-style italic. The second paragraph should have the text underlined.
6. Use external CSS for link, image and paragraph styles.

Demonstrate usage of various selectors

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

Explain Box Model in CSS

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

Demonstrate formatting text alignment, fonts and colors

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

JAVASCRIPT

90 The need of client side scripting?

A client side script is a small program which is executed by the browsers.

Examples : Javascript , VBscript

Need of client side script?

- For validating the user data before it reaches the server.

Example: Login name and password fields are mandatory in a login page. Validation can be done by java script.

- Perform some dynamic functionality based on user navigations.

Example:

- User hovers above a image, dynamically changes the image.
- Disable the salary text field if user selects unemployed radio button.

How client side script works?

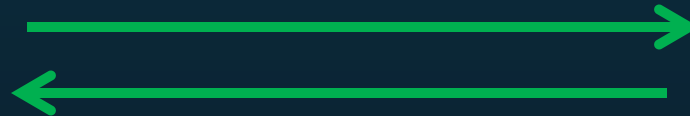
A user from India is trying to login a mail application residing in US. Assume that the user forgets to enter the login name and clicks submit.

Java Script in Browser validates the data.



User
(India)

User submits the login page.



Server throws an error
"login name Mandatory"



Server
(UK)

Server validates the login page

No Client side scripting : If each transaction (arrow depicted) takes 3 seconds the total transaction takes 9 seconds.

Client side script implemented: server calls will not happen if the user data is wrong. Assuming script validation takes 1 second for the validating the login name the transaction could be completed in 1 second (reducing the response time by 90%).

What is Java Script?

JavaScript is a client side scripting language which can be used to perform client side operations. Java scripts are executed by the browser engine.

Example:

- Validations in fields
- Show tool tips
- Perform some operations on mouse click.
- Change a font color on mouse over.

Advantages of Java Script

- Java Script is used for performing the basic sanity checks on the user request before it is submitted to the server, thus improving the transactions response time.
- This also reduces the load on the server as some logic is executed by the client browser.
- This also reduces the network traffic by ensuring that no invalid user requests are transmitted over the network.

Uses Of Java Script?

- Client side validation
 - Examples: Check for mandatory fields, numeric validation in salary field.
- Dynamic application of styles
 - Examples: Change the font color of a text dynamically.
- JavaScript can react to events like *onclick* , *onblur*, *onfocus* etc
 - Examples: Perform a validation on Click. On focus of a text box clear the values.

Demonstrate handling events for validations

https://www.w3schools.com/jsref/event_onclick.asp

How to develop a java script function ?

Java Script are developed using the `<Script> </Script>` tag. *Functions* are like java methods which executes a specific logic.

A simple function to print the date:

- `<script type="text/javascript">`
- `function printDate(){`
`document.write("<p>" + Date() + "</p>");`
- `}`
- `</script>`

Use functions to handle form validations

https://www.w3schools.com/js/js_functions.asp

Values & Variables

- JavaScript variables does not have explicit data types.
Example: Variables need not be declared as integer, a string or a real.
- All JavaScript variables are declared using the keyword *var*.
Example: `var x=7` // *defines a variable x with value 7.*
- JavaScript recognizes the data type based on the *value*
 - `var pi = 3.14` – pi is considered a number by the java script engine.
 - `var flag = true` – flag is considered a boolean variable.
 - `var countryName = "India"` – This will be considered as a string.

Values & Variables (Cont)

- A JavaScript identifier or name must start with a letter or underscore ("_") subsequent characters can also be digits (0-9)
- Variables Scope:
 - Local variable are declared inside a function
 - Global variable are declared outside a function
- Applying *var* to declare a global variable is optional but *var* is mandatory for local variables.

JavaScript DOM

- JavaScript is designed on a simple object-based paradigm.
- An object is a construct with properties (variables), methods or other objects.
- JavaScript uses an object model known as Document Object Model (DOM) to navigate through the HTML document in an hierarchy.

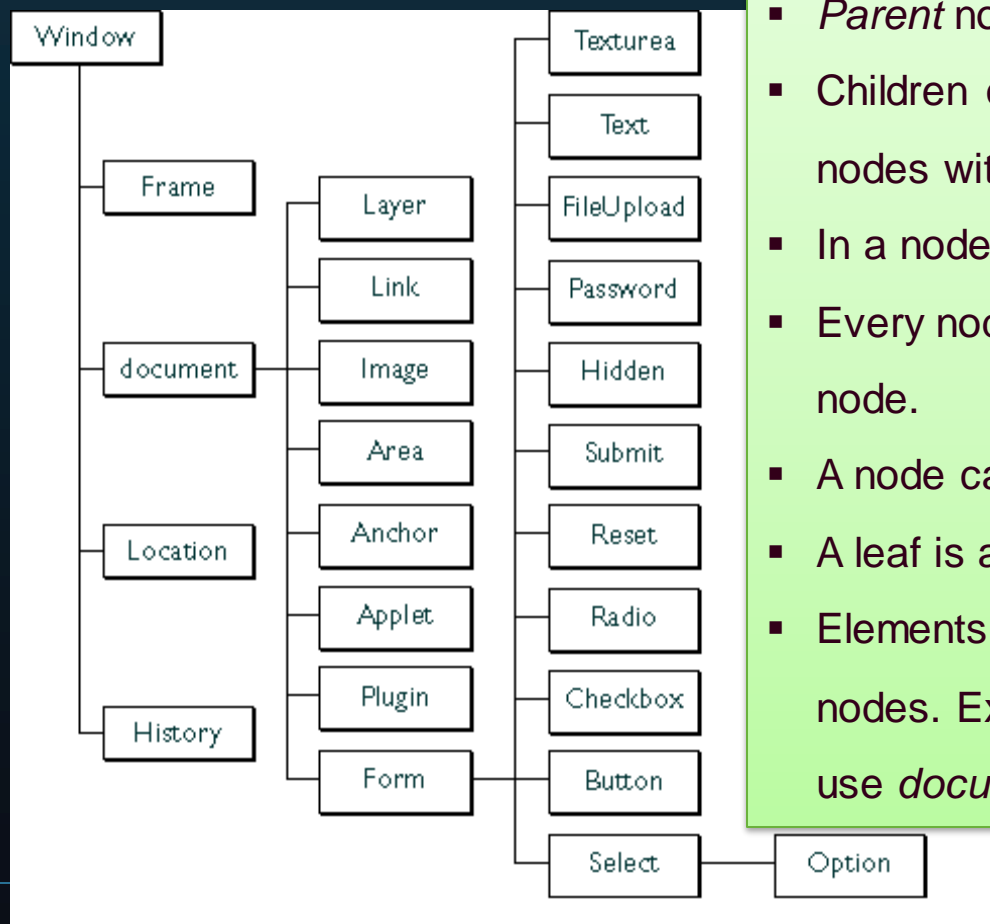
Document Object

- Each HTML document loaded into a browser window becomes a Document object.
- DOM represents HTML elements in a Tree structure.
- Every elements in a HTML document is represented as a node.
- DOM can be used to change, add, or delete HTML elements.
- Java Script access elements of the HTML document using the document object.
- DOM nodes accessed through the following methods,
 1. getElementById() : Returns the element specified by the ID
 2. getElementsByTagName() : Returns the list of elements with the specified tag name.

For more Details on DOM visit

<http://www.w3schools.com/html/dom/default.asp>

HTML DOM Structure



- The nodes in the node tree are placed in a hierarchy.
- The terms parent, child, and sibling are used to describe the relationships.
- *Parent* nodes have children.
- Children on the same level are called *siblings*, are nodes with the same parent.
- In a node tree, the top node is called the *root*.
- Every node, except the root, has exactly one parent node.
- A node can have any number of children .
- A leaf is a node with no children.
- Elements can be accessed by moving through the tree nodes. Example: To access the text field element we use *document.form.text*.

Accessing Elements Using getElementsByTagName

getElementsByTagName() retrieves all the elements with a *specified tag name*.

Scenario: Assume that there is a HTML with many paragraph elements `<P></P>`.

How can one access the first paragraph tag in the document?

```
document.getElementsByTagName("p")[0];
```

Where,

`document.getElementsByTagName("p")` return a list of all the paragraph tags in the page and index `[0]` returns the first tag in the list.

How read the contents inside the first paragraph tag in the document?

```
var content=document.getElementsByTagName("p")[0].innerText;
```

Accessing Elements Using `getElementsByTagName`

`getElementsByTagName()` method used for accessing a specific element using its *name*.

`var choice=document.getElementsByTagName("hobby");` Gets all the elements with the name `"hobby"`.

Individual elements can be accessed from the collection using the index.

`hobby[0]` returns the first element in the collection `choice`

- Length property gives the number of elements in the collection can be used to iterate through the entire list of elements.
- Value of the individual elements can be retrieved using the value property.

Scenario : Assume `"hobby"` is the name of a collection of check boxes with values "Singing, Dancing, Cricket" in a html page.

How to identify which hobbies are selected?

```
var choice=document.getElementsByTagName("hobby");  
for(var i=0;i<choice.length;i++){  
    alert(choice[i].checked);}
```

This iterates through the *hobbies* check boxes and using the *checked* attribute and alerts whether it is checked or not.

How To Access Fields using getElementById() method

To access the form fields we use the method *document.getElementById()*.

Example of Fields: Text box, Radio button, Check box , text area etc.

Syntax : `document.getElementById(elementID)` // This retrieves the element whose id matches the id passed as parameter. "Id" here refers to the ID attribute of the form element.

Example: How to access a text field "*user name*" whose id is "`username`".

- `document.getElementById("username")`

How to access and set values of form elements

To retrieve the value of an element we use the attribute *“value”*.

Syntax:

- *// retrieves the value*
- `var variableName=document.getElementById(elementId).value`
- *// Sets the value*
- `document.getElementById(elementId).value = value To Be Set.`

Example:

- *//This retrieves the value from the text field “userName”*
- `var userName=document.getElementById(“username”).value`
- *//This sets the value of username as “Arun” in the text field “userName”*
- `document.getElementById(“username”).value=“Arun” ;`

String Object

The String Object type provides a set of methods for manipulating strings.
Example: `var string1="Hello World"` and `var string2="JavaScript"`

Method	Description	Example
<code>charAt(index)</code>	Returns the character at the specified index	<code>string1.charAt(0)</code> returns "H"
<code>charCodeAt(index)</code>	Returns the Unicode of the character at the specified index	<code>string1.charAt(0)</code> returns 72 which is the Unicode value for "H"
<code>fromCharCode(unicode1,unicode2)</code>	Converts Unicode values to characters	<code>String.fromCharCode(72,69,76,76,79)</code> returns "HELLO" where the numbers 72,69,76,79 are the Unicode's of the letters H,E,L and O respectively.
<code>indexOf(string)</code>	Returns the position of the first found occurrence of a specified value in a string	<code>string1.indexOf("World")</code> returns 6
<code>lastIndexOf(string)</code>	Returns the position of the last found occurrence of a specified value in a string	<code>string1.lastIndexOf("o")</code> returns 7

String Object Methods

Method	Description	Example
<code>match(regEx)</code>	Searches for a match between a regular expression and a string, and returns the matches	Visit the below link for details on using regEx. http://www.w3schools.com/js/js_obj_regexp.asp
<code>replace(stringA,StringB)</code>	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring	<code>string1.replace("World","John")</code> returns "Hello John"
<code>search(regEx)</code>	Searches for a match between a regular expression and a string, and returns the position of the match	Visit the below link for details on using regEx. http://www.w3schools.com/js/js_obj_regexp.asp
<code>substr(start,length)</code>	Extracts the characters from a string, beginning at a specified start position, and through the specified number of characters. length is optional in that case the rest of the string from the starting index is returned.	<code>string1.substr(3,4)</code> returns "lo w"
<code>substring(start,stop)</code>	Extracts the characters from a string, between two specified indices excluding the stop index. stop index is optional in that case the rest of the String starting from the start index is returned. Negative values become zero. if start > end, the arguments are swapped	<code>String1.substring(3,5)</code> returns "lo"

String Object Methods

Method	Description	Example
slice()	Returns a portion of the string from position start to, but not including position end. Normally Works similar to substring() method but for negative arguments it means go backwards from the tail.	String.slice(4,-2) returns “o wor” that is from 4 th index to 2 nd position from the tail.
toLowerCase()	Converts a string to lowercase letters	string1.toLowerCase() returns “hello world”
toUpperCase()	Converts a string to uppercase letters	String1.toUpperCase() returns “HELLO WORLD”
concat(string)	Joins two or more strings, and returns a copy of the joined strings	string1.concat(string2) returns “Hello WorldJavaScript”

Use methods in String object viz., length(), trim()

[https://www.w3schools.com/js/js_string_methods.as
p](https://www.w3schools.com/js/js_string_methods.asp)

111 Conditional Statements: If..else

Definition: Used to execute some code if the condition is true and another code if the condition is false.

Syntax

```
if (condition){  
  //code to be executed if condition is  
  true  
}  
else  
{  
  //code to be executed if condition is  
  not true  
}
```

Example

```
if (a>b){  
  document.write("a is greater than b");  
}  
else  
{  
  document.write("b is greater than a");  
}
```

Conditional Statements: If..else if ..else

Syntax

```
if (condition){  
  //code to be executed if condition is true  
}  
else if (condition2) {  
  //code to be executed  
}  
else {  
  //code to be executed  
}
```

Example

```
var time=16;  
if (time<11){  
  document.write("Good Morning");  
}  
else if (time<15) {  
  document.write("Good Noon");  
}  
else {  
  document.write("Good Evening");  
}
```


Conditional Statements: Switch

Definition : Used to select a particular block of statements based on the user choice. Similar to if..Else..If Statement.

Syntax

```
switch(n) {  
  case 1:  
    //execute code block 1  
    break ;  
  case 2:  
    //execute code block 2  
    break ;  
  default:  
    //code to be executed if n is different  
    from case 1 and 2  
}
```

Example

```
Var n=2;  
switch(n) {  
  case 1:  
    document.write("one");  
    break ;  
  case 2:  
    document.write("Two");  
    break ;  
  default:  
    document.write("invalid choice");  
}
```

Conditional Statements: For Loop

Definition : Used for iterating through a block for the specified number of times.

```
var initval;  
for(initval=startvalue;initval<=endalue;initval=initval+incrval)  
{  
    //code to be executed in loop  
}
```

```
var i;  
for(i=1;i<=10;initval=initval+1)  
{  
    document.write(i);  
}
```

Conditional Statements: while and do while Loop

While Loop: The while loop iterates through a block of code till a specified condition is true.

```
while (initval<=endvalue)
{
  //code to be executed in loop
}
```

```
var i=1;
while (i<=10)
{
  document.write(i);
}
```

Do..While : The do...while loop is a variant of the while loop. This loop will execute the block of code *atleast* ONCE, and then it will repeat the loop as long as the specified condition is true.

```
do {
  //code to be executed in loop
} while (var<=endvalue);
```

```
var i=1;
do {
  document.write(i);
} while (i<=10);
```

Expressions

JavaScript *expressions* are similar to expressions in any other programming language.

Three types of expressions:

1. Arithmetic: evaluates to a number. *Example* : `var sum = a + b;`
2. String: evaluates to a character string. *Example* : `var name = "Fred";`
3. Logical: evaluates to true or false. *Example* : `var flag = a > b;`
4. Conditional Expression: A conditional expression can have one of the two values based on a condition.

Syntax: `var value=(condition) ? val1 : val2`

If condition is true, the expression has the value of val1, Otherwise it has the value of val2.

Example: `status = (age >= 18) ? "adult" : "minor"`

Assigns the value "adult" to the variable status if age is eighteen or greater. Otherwise, it assigns the value "minor" to status.

Operators

JavaScript *operators* are similar to the operators in any other programming language.

JavaScript supports the following types of operators

1. Arithmetic – *Example:* +, *
2. Bitwise – *Example:* ^, &
3. Comparison – *Example:* >, < and !=
4. Assignment - *Example:* =
5. Logical - *Example:* &&, ||

Pop up windows

Pop ups are small windows used for either getting user input (or) displaying alert messages. There are three types of popup boxes

1. Alert Box

Generally, the `alert()` method is used to warn the user on something.

Example of usage:

- Displayed when incorrect information input in a form.
- An invalid result from a calculation.
- A warning that a service is not available on a given date.



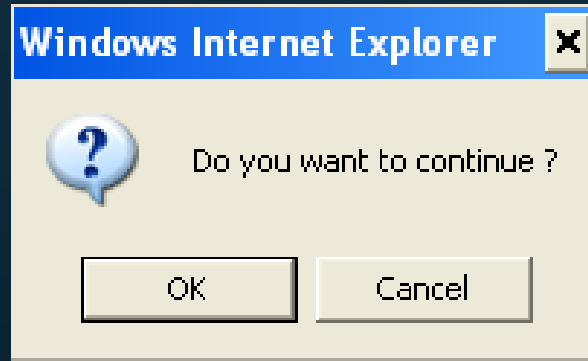
Syntax:

```
alert("Hello ! I am an alert box!")
```

When an alert box pops up, the user will have to click "OK" to proceed.

Pop Up Window – Confirm Box

A *confirm box* is often used if you want the user to verify or accept something.



- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
- If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax:

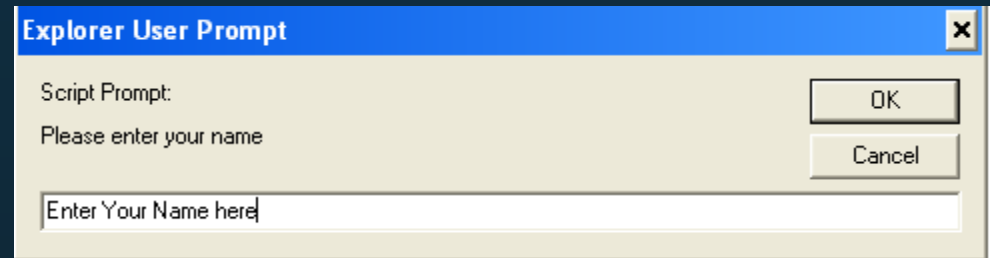
```
var value=confirm("Confirm Text ?")
```

Example:

```
var value=confirm("Do you wish to continue ?")
```

Pop Up Window- Prompt Box

A *prompt box* is often used if you want the user to input a value before entering a page.



- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax:

```
var variable= prompt("Prompt Text", "Default Value") ;
```

Example:

```
var name=prompt("Please Enter your name", "Enter Your Name Here") ;
```


Use form object to submit the form

https://www.w3schools.com/jsref/dom_obj_form.asp

Recap

In this chapter, we have learnt about:

- HTML
- CSS
- Javascript

You have successfully completed –
Learning on basics of Web User Interface

