**HTML**

Each screen displayed on a website is called **WEBPAGE**.

A **WEBSITE** is a collection of many web pages.

Any Website or webpages can be visited using **Web Browsers** such as Internet Explorer, Google chrome, opera safari, etc.

To Design a web page we need to learn HTML.

**What is HTML?**

**H**YPERTEXT **M**ARKUP **L**ANGUAGE is a language that is understood by a web browser for composing text and images for a webpage.

HTML is a basic building block for Web Pages.

It is a format that tells a computer how to display a web page.

**History Of HTML**

**HTML was invented by ‘Tim Berners Lee’ in 1991.**

It is an evolving language and each new definitive version is given a number.

* The first definitive version was **HTML 2.0(1991)**

->This had most of elements , but was missing netscape /Microsoft extensions

->Did not support tables

->Didn’t support align attributes.

* **HTML3** (1995)is next version

It was an ambitious(strong or determination) effort on part of **DAVE** **RAGGETT**, to upgrade the features and utility of HTML. However, it was never completed or implemented.

* **HTML 3.2(1997)** is next official version

It is an integrating support for tables, images, heading, element align attributes, etc.

* **HTML 4.01(1999)** is the next version

It support most for proprietary extensions, and extra features.

* **HTML 5** is the current official standard.- released in 2008

It includes CSS, TABLES, HYPERLINKS, JAVASCRIPT ENHANCEMENT.

2 organizations involve in development of HTML5.

1. W3C(world wide web consortium)

2. WHATWG(Web Hyper Text Application Technology working group)

* First WEBPAGE - November 1990

With no standards.

* A group called the **world wide web** consortium(w3c) was then formed and have since set the standards that are widely accepted.

**Which Software used to write HTML?**

It is only simple text editor like notepad where we write HTML code and link it to web browsers.

Let’s start HTML

HTML is the standard markup language for Web pages.

With HTML you can create your own Website.

What is HTML?

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
* **The <!DOCTYPE html> declaration defines that this document is an HTML5 document. And helps browsers to display web pages correctly. Not case sensitive**
* **The <html> element is the root element of an HTML page**
* **The <head> element contains meta information about the HTML page**
* Metadata is not displayed. Metadata typically define the document title, character set, styles, scripts, and other meta information.
* **The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)**
* **The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.**
* **The <h1> element defines a large heading**
* **The <p> element defines a paragraph**

**HTML LINKS**

The Link’s destination is specified in the **href attribute**. Attributes are used to provide Additional information about HTML elements.

* How to see web page source code ?

Right-click on webpage and select “view page source”.

* Inspect an HTMl page?

Right-click on element and s inspect to see what elements are made up of.

**HTML Elements**

THE HTML element is defined by a start tag, some content, and an end tag.

Note: <br> doesn’t have end tag. So This are called empty elements. IT is line break tag.

**NEVER SKIP THE END TAG** some HTML elements will display correctly, even if you forget end tag.

**HTML is Not Case Sensitive**

<P> == <p>

**NESTED HTML ELEMENTS**

ie. <html>

<head>

<title>

<body>

<h1> ……………….etc

**STYLE Attribute**

Used to add styles to an element, such as color, font, size …

**Lang Attribute**

It always include in inside html tag to declare language of the web page.

lang=”en” -> refers English

->Country code also be added to language code in lang attribute.

Ex: lang=”en-US”

**Title Attribute**

Defines some extra information about an element.

The value of title attribute will be displayed as a tooltip when you mouse over the element.

**NOTE**: Always Quote Attribute values.

Double Quotes are most common but single also used .

**Headings Are Important**

Search engines use the heading to index the structure and content of your web pages.

**NOTE:** use html heading for headings only. Don’t use headings to make text BIG or bold.

**BIGGER HEADINGS**

Each html heading has default size. However, you can specify it size using font-size property.

**NOTE**: Browsers automatically add some white space before and after a heading.

**HTML Paragraphs**

A paragraph always starts on new line, and is usually a block of text.

Browsers automatically add some white space before and end a paragraph.

The browsers automatically remove any extra spaces and lines when the page is displayed.

**HTML Horizontal Rules :**

The <hr> tag defines a thematic break in an html page, and is most often displayed as a horizontal rule.

It is used to separate content in HTML page.

NOTE: <hr> tag is an empty tag, which means that is no end tag.

If a poem is there as follow

<p>

My bonnie lies over the ocean.

My bonnie lies over the sea.

…….

</p>

It display in single line to overcome this we have an <pre> Element.

**<pre> Element**

It is preformatted text, text inside this element is displayed in fixed-width font and it preserves both spaces and line breaks.

**HTML-CSS**

CSS stands Cascading Style Sheets. CSS saves a lot of work. It can control the layout of multiple webpages all at once.

TIP:- cascading means style applied to a parent will also apply to all children elements within parent.

CSS can be added to HTML in 3 ways:

!Inline - using style attribute inside HTML elements.

!Internal – using <style> element in <head> section.

!External – by using <link> element to link to an external CSS file.Here external file must saved with .css extension.

TIP:- with an external style sheet, you can change the look of an entire web site, by changing one file.

**HTML**-**Links**

Links are found in all web pages . IT allows users to click their way from page to page.

HTML links are Hyperlinks

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

**Syntax:** <a href=”url”>link text</a>

**href** -> indicate link destination

**link text** ->is the part that will be visible to the reader. Clinking on link text, will send the reader to the specified url address.

Attributes in <link>tag:

**Target:** attribute specifies where to open the linked document.

It have following values

\_self:- Default. Opens document in same window/tab as it was clicked.

\_blank:- opens document in a new window/tab

\_parent:-open docu in parent frame.

\_top:-opens docu in full body of window.

**Absolute urls vs. Relative urls**

A full web address in href is absolute

A local link (a link to a page within the same website ) is specified with a Relative URL(without <http://www> part).

**HTML Links- Use an image as a link**

Here just put <img> tab inside <a> tag

**Syntax:** < a href=”deaf.asp”>

<img src=”smile.gif” alt=”HTML” style=”width:42px; height80px;”>

</a>

**Link to an Email Address**

Use **mailto:**  inside **href**  attribute to create link that opens the user’s email program

**Syntax:** <a href=<mailto:suma@gmail.com>> send mail</a>

**Button as a link**

To use HTML button as link, you have to add some javascript code,

it allows you to specify what happens at certain events, such as a click of a button.

**Syntax:** <button onclick=”document.location=’default.asp’”> HTML Tutorial</button>

**Link Titles**

The title attribute specifies extra info about an element. The information is shown as atooltip text when the mouse moves over the element.

href place **#** and it corresponded search place **id.**

**HTML Images**

Images improve design and appearance of webpage.

HTML support animate(moving images)

ABBR FileFormat fileExt

APNG - animated portable network .apng

Graphics

GIF – Graphics Interchange format .gif

ICO - Microsoft Icon .ico, .cur

JPEG- Joint Photographic Expert .jpeg,.jpg

Group Image .jfif,.pjpeg,.pjp

PNG- portable network graphics .png

SVG – scalable vector Graphics .svg

**HTML favicon**

A favicon image is small image it should be simple image with high contrast.

It is displayed to left of page title in browsers tab

**HTML Page will validate without the <html> and <body> tags.**

**HTML Versus XHTML**

XHTML is a stricter, more XML based version of HTML.

XHTML – Extensible HyperTextMarkupLang

XHTML is supported by all major browsers.

XML is a mark up Lang where all documents must be marked up correctly(well-formed).

The Most Important Differences from HTML

* <!DOCTYPE> is **mandatory**
* The xmlns attribute in <html> is **mandatory**
* <html>, <head>, <title>, and <body> are **mandatory**
* Elements must always be **properly nested**
* Elements must always be **closed**
* Elements must always be in **lowercase**
* Attribute names must always be in **lowercase**
* Attribute values must always be **quoted**
* Attribute minimization is **forbidden**

**Semantic Elements**

It clearly describes its meaning to both browser and developer.

EX: non-semantic like <div> <span -tells nothing about its content

Semantic like <form>,<table>,<article> -clearly defines its content.

**HTML Style Guide**

A consistent, clean tidy HTML code makes it easier for others to read and understand your code.

1. Always declare document type

<!DOCTYPE html>

2. use lower case element names.

Good:

<body>

<p>this is a Paragraph</p>

</body>

Bad:

<BODY>

<P>this is a Paragraph</P>

</BODY>

3.Close all HTML elements

In HTML, you don’t have to close all elements (<p>) but strongly recommend to close all elements.

4.use lowercase attribute names.

<a href=”<https://vfgf.com>”></a>

5. Always quote attribute values.

<table class=”stip”>

6. Always specify alt, width, height for images.

<img src=”html.gif” alt=”img1” style=”width:128px;height:230px;”>

7.Avoid long line codes.

When using HTML editor,it is not convenient to scroll right &left to read HTML code.

**NOTE**: for readability, add blank lines to separate large code blocks.

8.Never skip the <title> element .

The contents of a page title is very important for Search engine optimization(SEO).used by search engines algorithms to decide the order when listing pages in search result.

9.Omitting <html> and <body>

An HTML page will validate without <html> & <body> tags but strongly recommended to use . without this tags crash DOM & XML software.

10. Omitting <head>

11. close empty HTML Elements

<img src=”ksdd.gif” />

12.add lang attribute

To declare language of a web pageto assist search engines &browsers.

13. meta data

To ensure proper interpretation and correct search engine indexing, both the language and character encoding

<meta charset=”charset”>

<!DOCTYPE html>

<html lang=”en-us”>

<head>

<meta charset=”UTF-8”>

<title>Pagetitle</title>

<head>

14. Setting the viewport

The viewport is the user’s visible area of a web page. It varies with device it will smaller on mobile than computer screen.

<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>

This controls the page’s dimensions and scaling.

Width=device-width: sets width of the page to follow screen-width of device.

Initial-scale=1.0 , sets initial zoom level

When page is first loaded by browser.

15. short comments should written on one line.

<!-- this is a comment -->

16. using style sheets

Use simple syntax for linking to style sheets

<link rel=”stylesheet” href=”extern.css”>

17.loading javascript into HTML

Use simple syntax for loading external scripts

<script src=”myscript.js”>

18. Accessing HTML elements with javascript.

Using untidy HTML code result js errors.

Use follow:

getElementById(“demo”).innerHTML = “Hello”;

19.use lowercase filenames

Some web servers are case sensitive about file names.(apache,unix)

So better to use lower case.

20. File extensions.

HTML - .html or .htm

CSS - .css

Javascript - .js

21. Default filenames

Using “index.html”, “index.htm”, “default.html”, “default.htm”

**HTML-Encoding**(character Sets)

* ASCII (American Standard code for Information Interchange)was the first character encoding standard.it define 128 characters((0-9),(A-Z),some special symbols).
* ISO-8859-1 was default set for HTML-4.It support 256 characters.
* HTML-5 supports UTF-8 set which converts almost all characters&symbol

Ex: 32 for space

33 - ! 34 – “ 35 - # 36 - $

37 - % 38 - & 39 – ‘ 40 – ( 41- )

42 - \* 43 - + 44 - , 46 - . 47 - /

48 to 57 ( 0 - 9)

65 to 90( A – Z)

97 to 122 (a – z)

ASCII uses upto 127 value.

UTF-8 values same as ASCII from 0-127

UTF-8 values same as ANSI&8859-1 from 160-255.

UTF-8 continues 256 to more than 10k characters.

**HTML-URL**

A URL is another word for a web address.

A URL can be composed of words([www.html.com](http://www.html.com)) or an internet protocol(192.68.20.50)

Mostly we use names for surfing to easier remember.

->Web browsers request pages from web servers by using URL.

<https://www.w3schools.com/html/default.asp>

schema://prefix.domain:port/path/filename

scheme – defines type of internet service (http /https)

prefix – define domain prefix (www)

domain – defines internet domain name (w3scgools.com)

port – defines the port number at the host (def http is 80)

path- defines a path at the server(html)

filename- defines the name of a document

or resource. (default.asp)

scheme fullform used for

http HypertexTransfer commom web

protocol pages, not

encrypted.

https Securehypertext securewebpages

transferprotocol encrypted

ftp filetransferpro download/uploadfiles

URL-Encoding:-

URLs can only be sent over internet using ASCII characters -set.

URL-encoding converts non-ASCII characters into a format that can be transmitted over internet.

URL-encoding replace non-ASCII characters with a “%” followed by hexadecimal digits.

URL’S can’t contain spaces. URL encoding replace space with + sign or %20.

**HTML-Forms**

An HTML form used to collect user input. The user input more often sent to a server for processing.

**<form> element:-**

The HTML form element is used to create an HTML form for user input:

<form>

.

:

:

</form>

It is a container for different types of input elements, such as: text, fields, checkbox, radio buttons, submit buttons, etc.

**<input> element:-**

<input> most used form element.it can be displayed in many ways, depending on **type attribute.**

**Ex:-**

1.<input type=”text”> - displays a single-line text input field.

2.<input type=”radio”> - displays a radio button(selecting **one of many choices**)

3.<input type=”checkbox”> - displays a checkbox (selecting **0 or more of many choices)**

4.<input type=”submit”> - displays submit button(submitting form)

5.<input type=”button”> - displays clickable button.

NOTE:- The form itself is not visible. The default width of an input field is 20 characters.

**<label> element:-**

It defines a label for many form elements.

Used for screen readers.

**for attribute**  of <label> tag be equal to **id attribute** of the <input> element to bind them together.

**->name attribute** for <input>

Each input must have a **name attribute** to be submitted. If it is omitted the value of input field will not be sent at all.

**HTML-Form Attributes**

1**. Action attribute:**- defines action be performed when the form is submitted.

Usually, form data is sent to a file on the server when user clicks on the submit button.

* Ex: The form data is sent to a file called “action\_page.php” This file contains **server-side script** that handles the data:
* <form action=”/action\_page.php”>

Note:- if action is omitted, the action sent to the current page.

2.**The taget attribute:** specifies where to display the response that received after submitting the form.

1.\_blank 2.\_self 3.\_parent 4. \_top

\_self is default

* <form action=”/action\_page.php” target=”\_blank”>

3**.The Method attribute:-** specifies http method to be used when submitting the form data.

The form-data can be sent as URL variables(with method=”get” or method=”post”).

The default http method is GET.

* <form action=”/action\_page.php” target=”\_blank” method=”get”>

**get:**

1.Appends the form data to the URL, in name/value pairs.

2.NEVER use get to send sensitive data(Submitted form data visible to URL)

3.length is limited(2048 characters).

4.get is good for non-secure data like **query-strings** in google.

5.can be bookmarked.

**post:-**

1.appends the form data inside the body of HTTP request(submitted form-data not shown in URL)

2.no size limitations.

3.used to send large amounts of data.

4.cann’t be bookmarked,secured,personal information.

4.**The Autocomplete attribute:-**  specifies whether a form should have autocomplete on or off.

on -> browser automatically complete values based on values that the user has entered before.

* <form action=”/action\_page.php” target=”\_blank” method=”get” autocomplete=”on”>

5.**The Novalidate Attribute:-**

Is a Boolean attribute. It specifies that the form-data(input) should not be validate when submitted.

* <form action=”/action\_page.php” novalidate>

6.**The formenctype attribute:-**

Specifies how the form-data should be encoded when submitted(only for forms with method=”post”)

It works on submit & image input types.

* <input type=”submit” formenctype = “multipart/form-data” value=” submit as Multipart/form-data”>

**HTML Form Elements**

1.**<input>:-** displayed in several ways, depending on the **type** attribute.

2.**<label>:-** defines label for several elements.

3.**<select>:-** defines a drop-down list

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the **selected** attribute to option

<option value=”fiat” selected>Fiat</option

**Visible Values:-**

Use the **size attribute** to specify the number of visible values:

**Multiple Selections:-**

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

4.**<textarea> -**  multi-line input field(a text area).

5.**<button> -**  clickable button.

Always specify **type attribute** for button element.

6.**<fieldset> -** used to group related data in a form.

7.**<legend> -** defines a caption for <fieldset> element.

8.**<datalist> -** specifies list of pre-defined options for an **<input>** element.

Users will see a drop-down list of the pre-defined options as they input data.

The **list** attribute of the **<input>** element, must refer to the **id** attribute of the **<datalist>**  element.

9.**<output>:-** represent the result of a calculation (like one performed by a script).

10.**<optgroup>:-** defines a group of related options in a drop-down list.

**HTML Input Types:-**

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

**NOTE:-** the characters in a password field are masked.

**Input Type submit:-**

Defines button for submitting form data to a **form-handler.**

**form-handler** is typically a server page with a script for processing input data. It specified in the form’s **action attribute.**

**HTML Input attributes:-**

1.**The value attribute:-**  specifies an initial value for an input field:

2.**The readonly attribute:-** specifies input-field only readable it can’t be modifiable except user can highlight ,copy the text.It will sent when submitting form.

<input type=”text” name=”fname” value=”suma” readonly>

3.**The disabled attribute:-** specifies that input field should be disabled. It is unusable and unclickable. It will not sent when submitting the form.

<input type=”text” name=”fname” value=”suma” disabled>

4.**The size attribute:-**  specifies the

visible width, in charcters of an input field.

Default size is 20.

It works with text search,tel,url, email,

&password.

<input type=”text” name=”fname” value=”suma” size=”5”>

5.**The maxlength attribute:-** specifies max number of characters allowed in input field .

6.**The max & min attributes:-**

Specify the minimum and maximum values for an input field. It work with the: number, range, date, datetime-local, month, time and week.

Use max &min combinely to create range of legal values.

7.**The multiple attribute:-**

Specifies that user allowed to enter more than one value in an input field.

It works with email, file

8.**The pattern attribute:-** specifies a regular expression that input field value is checked against, when the form is submitted.

It works with the text, date, search, url, tel, email, password.

Use global **title attribute** to describe the pattern to help the user.

9.**The placeholder attribute:-**

Specifies a short hint that describes the expected value of input field(a sample value). It display in input field before user enter the input.

It works with text, search, url, tel, email, & password.

10.**The required attribute:-** specifies that input field must be filled out before submitting the form

11.**The step attribute:-** specifies legal number intervals for input field.

Ex:- if step=”3” legal numbers be -3,0,3,6,…

12.**The autofocus attribute:-**  specifies an input field should automatically get focus when page loads.

13,**The height & width attribute:-** specifies the height &width of an input type=”image”

14.**The list attribute:-** it refers <datalist> that contains pre-defined options.

15.**The autocomplete attribute:-** specifies whether a form or input field should hava on or off.

Means based on before typed values it will show options when putting it “on”.

**HTML Graphics**

**HTML Canvas Graphics:-**

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

The graphics to left is created with <canvas> .It shows elements like

Rectangle, gradient rectangle, multicolour rectangle, multicolour text.

**What is HTML canvas?**

The HTMl <canvas> element is used to draw graphics, via javascript.

**<canvas>** is a container for graphics. Must use javascript to actually draw the graphics.

**Differ between SVG and canvas:-**

SVG – describe 2D graphics in **xml.**

Canvas – 2D graphics in **javascript.**

**SVG** – Every element is available within the **svg DOM.** Attach js event handlers for an element.

Resolution independent.

Support event handlers

Best suited for appli’s with large rendering areas(google maps).

Slow rendering.

Not suit for game appli’s

**Canvas**- resolution dependent

No support for event handlers.

Poor text rendering

It is save as .png or .jpg

Well suited for graphic-intensive games.

**HTML Media**

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

**Format File Description**

MPEG .mpg develop by Moving

.mpeg pictures expertgroup.

Not supported in html

AVI .avi Audio-Video Interleave

Used in videocameras

& Tv hardware.develop

Ed by Microsoft

QuickTime .mov develp by apple used

Same as above purp

**HTML Plug-ins**

Plug-ins are computer programs that

Extend the standard functionality of the

Browser.

**Plug-ins:-**

This are designed to be used for many

Different purposes:

->To run java applets.

->To run Microsoft ActiveX controls.

->To display Flash Movies.

->To display maps.

->To scan for viruses.

->To verify bank id.

**<object> :=** element is supported by all browsers.

**It** defines an embedded object within an

HTML document.

It was designed to embed plug-ins(like

Java applets, PDf readers, Flash Players)

In web pages, also used to include HTML

In HTML.

<object width=”100%” height=”500px”

data=”snippet.html”></object>

**<embed>** defines embedded object within

an HTML document.

Web browsers support it for a long time.

It doesn’t have end tag.

Used to include html in html.

**HTML YouTube Videos**

The easiest way to play videos in HTML,

Is to use youtube.

Playing a youtube video in HTML:

* **Upload the video to youtube.**
* **Take a note of the video id.**
* **Define an <iframe> element in your webpage.**
* **Let the src attribute point to the video URL.**
* **Use the width and height attributes to specify dimensions of the player.**
* **Add any other parameters to URL.**

Ex:

<iframe width=”420” height=”315” src=”<https://www.youtube.com/embed/tgbNymZ7vqY>”>

</iframe>

* To start automatically use **autoplay=1**

To youtube URL.

Add mute=1 after autoplay=1 to let your

Video start playing automatically (but

Muted).

Ex:

<iframe width=”420” height=”315” src=”<https://www.youtube.com/embed/tgbNymZ7vqY>?autoplay=1&mute=1”>

</iframe>

Yutube playlist

* A comma separated list of videos to play.
* Set loop=1 let your video loop forever.
* Value 0 (def) video play only once.

Ex:

<iframe width=”420” height=”315” src=”<https://www.youtube.com/embed/tgbNymZ7vqY>?playlist=tgbNymZ7vqY&loop=1”>

</iframe>

**Youtube controls:-**

Add controls=0 to not display controls in the video player.

Value 0:player controls doesn’t display.

Value 1(def): player controls display.

Ex:

<iframe width=”420” height=”315” src=”<https://www.youtube.com/embed/tgbNymZ7vqY>?controls=0”>

</iframe>

**HTML APIs**

getCurrentPosition() method used to return the user’s position.

**HTML Web Storage API:-**

**HTML Web storage, better than cookies.**

What is HTML WEB STORAGE:-

**With Web storage, web applications can store data locally within the user’s browser.**

**Before HTML5,** application data had to be stored in **cookies,** included in every server request.

**WebStorage:**  is more secure, large amount of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger(atleast 5MB) and information is never transferred to the server.

**Webstorage** is per origin(per domain and protocol). ALL pages, from one origin, can store and access the same data.

**HTML Web Storage Objects:-**

It provides two Objects for storing data on the **client:**

* **window.localStorage:-** stores with no expiration date.
* **window.sessionStorage:-** stores data for one session (data is lost when browser tab is closed).

Before using web storage, check browser support for local storage and session storage:-

Syntax:-

if(typeof(Storage) !==”undefined”) {

//code for localstorage/session storage

}

else{

//No web storage support..

}

**IN localStorage Object :**

Stores data with no expiration date. The

data will not be deleted when the browser is closed and will be available the nextday, week, or year.

**In sessionStorage Object:-**

Equal to localstorage exceptthat it stores the data for only one session.The data is deleted when user closes the specific browser tab.

**HTML Web Workers API:-**

A web worker is a Javascript running in the background, without affecting the performance of the page.

**What is a Web Worker?**

When executing scripts in a HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things etc.., while the web worker runs in the background.

**HTML SSE API:**

Server-sent events(**SSE**) allow a web page to get updates from a server.

Ex: face book, twitter, stock price updates, news feeds, sport results.

The **EventSource** object is used to receive server-sent event notifications:

.