Unit-1
Introduction to Web and HTML

Marwadi University

Department of
Computer Engineering

Web Technology-
01CE0306

Prof. Kajal Tanchak

Outlines

- Introduction to Web Servers,
- HTTP request and Response Model,
- Structure of HTML,
- Doctypes in HTML,
- HTML Tags, Elements and attributes
- **HTML 5**
  - HTML 5 Layout and syntax,
  - Attributes,
  - Events,
  - Web forms and validations,
  - Audio & Video,
  - SVG.

CO Mapping

- **CO1: Understand various web servers and HTTP request response method. (Understand)**
- **CO2: Select various HTML 4, HTML 5 and CSS 3 tags to create an interactive pages. (Apply)**

Introduction to WWW

- The World Wide Web (WWW) is a network of online content that is formatted in HTML and accessed via HTTP.
- The term refers to all the interlinked HTML pages that can be accessed over the Internet.
- A properly designed hypertext document can help users to locate desired type of information rapidly.
- Hypertext documents enable this by using a series of link.
- A link is a special type of item in a hypertext document connecting the document to another document.
- Hypertext documents on internet are known as Web Pages.
- The World Wide Web was originally designed in 1991 by Tim Berners-Lee .


WELCOME
TO THE
WORLD WIDE WEB

Internet vs. WWW

- It is all the Web pages, pictures, videos and other online content that can be accessed via a Web browser.
- The Internet, in contrast, is the underlying network connection that allows us to send email and access the World Wide Web.
- **Internet**
- Global system of interconnected computer networks to share information
- Use the standard Internet Protocol suite (TCP/IP)
- *network of networks*

Internet vs. www

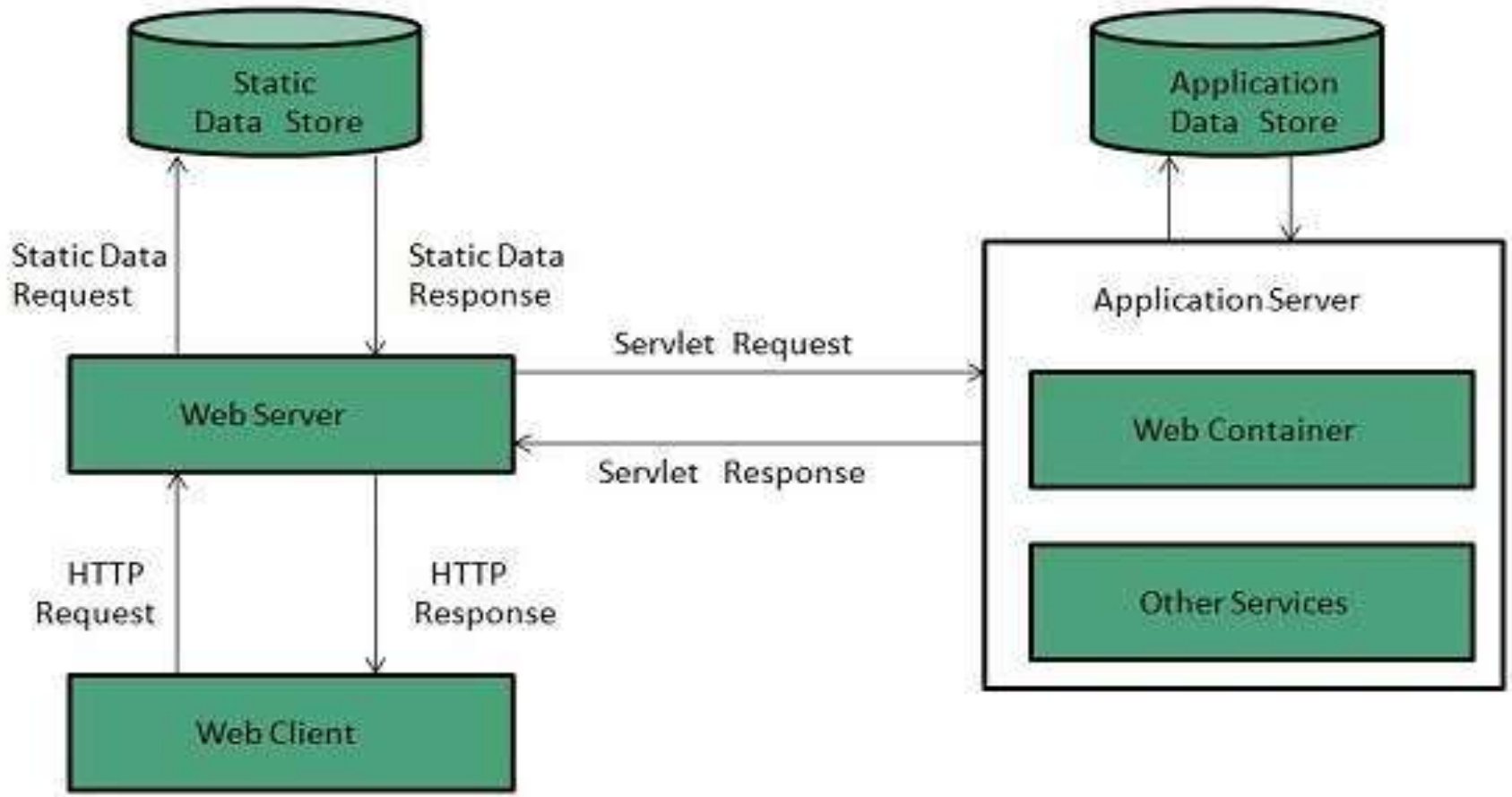
Internet	WWW
global network of networks	collection of information which is accessed via the Internet.
Internet is infrastructure	Web is service on top of that infrastructure
viewed as a big book-store	viewed as collection of books on that store.
Internet as hardware	Web as software

WWW Structural & Semantic components

- The structural components of a web application basically refer to the functionality of the web application with which a user interacts, the control and the database storage.
- The web browser or client
- The web application server
- Cache
- Internet
- **Semantic Components**
- A system of globally unique identifiers for resources on the Web, known as Uniform Resource Locator (URL)
- The publishing language Hypertext Markup Language (HTML)
- The Hypertext Transfer Protocol (HTTP)

Introduction to Web Servers

- Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.
- Web site is collection of web pages while web server is a software that respond to the request for web resources.
- **Web Server Working**
- Web server respond to the client request in either of the following two ways:
- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database


Static
Data Store
Application
Data Store
Static Data
Request
Static Data
Response
Application Server
Servlet Request
Web Server
Web Container
Servlet Response
HTTP
Request
HTTP
Response
Other Services
Web Client

Cont..

- **Key Points**

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.

- If the requested web page is not found, web server will the send an HTTP response:Error 404 Not found.

- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

- **Architecture**

- Web Server Architecture follows the following two approaches:
- Concurrent Approach
- Single-Process-Event-Driven Approach.

Cont..

- **Concurrent Approach**
- Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:
- Multi-process
- Multi-threaded
- Hybrid method.
- **Multi-processing**
- In this a single process (parent process) initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.
- It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

Cont..

- **Multi-threaded**
- Unlike Multi-process, it creates multiple single-threaded process.
- **Hybrid**
- It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Examples

S.N.	Web Server Description
1	**Apache HTTP Server** This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	**Internet Information Services (IIS)** The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	**Lighttpd** The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	**Sun Java System Web Server** This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	**Jigsaw Server** Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

Introduction to Web Browser

- A browser is a software program that is used to explore, retrieve, and display the information available on the World Wide Web.
- This information may be in the form of pictures, web pages, videos, and other files that all are connected via hyperlinks and categorized with the help of URLs (Uniform Resource Identifiers). For example, you are viewing this page by using a browser.
- A browser is a client program as it runs on a user computer or mobile device and contacts the webserver for the information requested by the user.

Cont..

- The web server sends the data back to the browser that displays the results on internet supported devices. On behalf of the users, the browser sends requests to web servers all over the internet by using HTTP (Hypertext Transfer Protocol). A browser requires a smartphone, computer, or tablet and internet to work.

- **Features of Web Browser**

- Most Web browsers offer common features such as:
- **Refresh button:** Refresh button allows the website to reload the contents of the web pages. Most of the web browsers store local copies of visited pages to enhance the performance by using a caching mechanism. Sometimes, it stops you from seeing the updated information; in this case, by clicking on the refresh button, you can see the updated information.

Cont..

- **Stop button:** It is used to cancel the communication of the web browser with the server and stops loading the page content. For example, if any malicious site enters the browser accidentally, it helps to save from it by clicking on the stop button.
- **Home button:** It provides users the option to bring up the predefined home page of the website.
- **Web address bar:** It allows the users to enter a web address in the address bar and visit the website.
- **Tabbed browsing:** It provides users the option to open multiple websites on a single window. It helps users to read different websites at the same time. For example, when you search for anything on the browser, it provides you a list of search results for your query. You can open all the results by right-clicking on each link, staying on the same page.
- **Bookmarks:** It allows the users to select particular website to save it for the later retrieval of information, which is predefined by the users.

Cont…

- **What is the URL (Uniform Resource Locator)?**
- A **uniform resource locator** is the address of a resource on the internet or the World Wide Web.
- It is also known as a web address or uniform resource identifier (URI).
- A URL represents the address of a resource, including the protocol used to access it.
- A URL includes the following information:
- It uses the protocol to access the resource.
- It defines the location of a server by IP address or the domain name.
- It includes a fragment identifier, which is optional.
- It contains the location of the resource in the directory of the server.

- A URL forwards user to a particular online resource, such as a video, webpage, or other resources.
- For example, when you search information on Google, the search results display the URL of the relevant resources in response to your search query. The title which appears in the search results is a hyperlink of the URL of the webpage.
- It is a **Uniform Resource Identifier**, which refers to all kinds of names and addresses of the resources on the webservers. URL's first part is known as a **protocol identifier**, and it specifies the protocol to use, and the second part, which is known as a resource name, represents the IP address or the domain name of a resource.
- Both parts are differentiated by a colon and two forward slashes like **http://www.javatpoint.com.**

Component of a Web browser

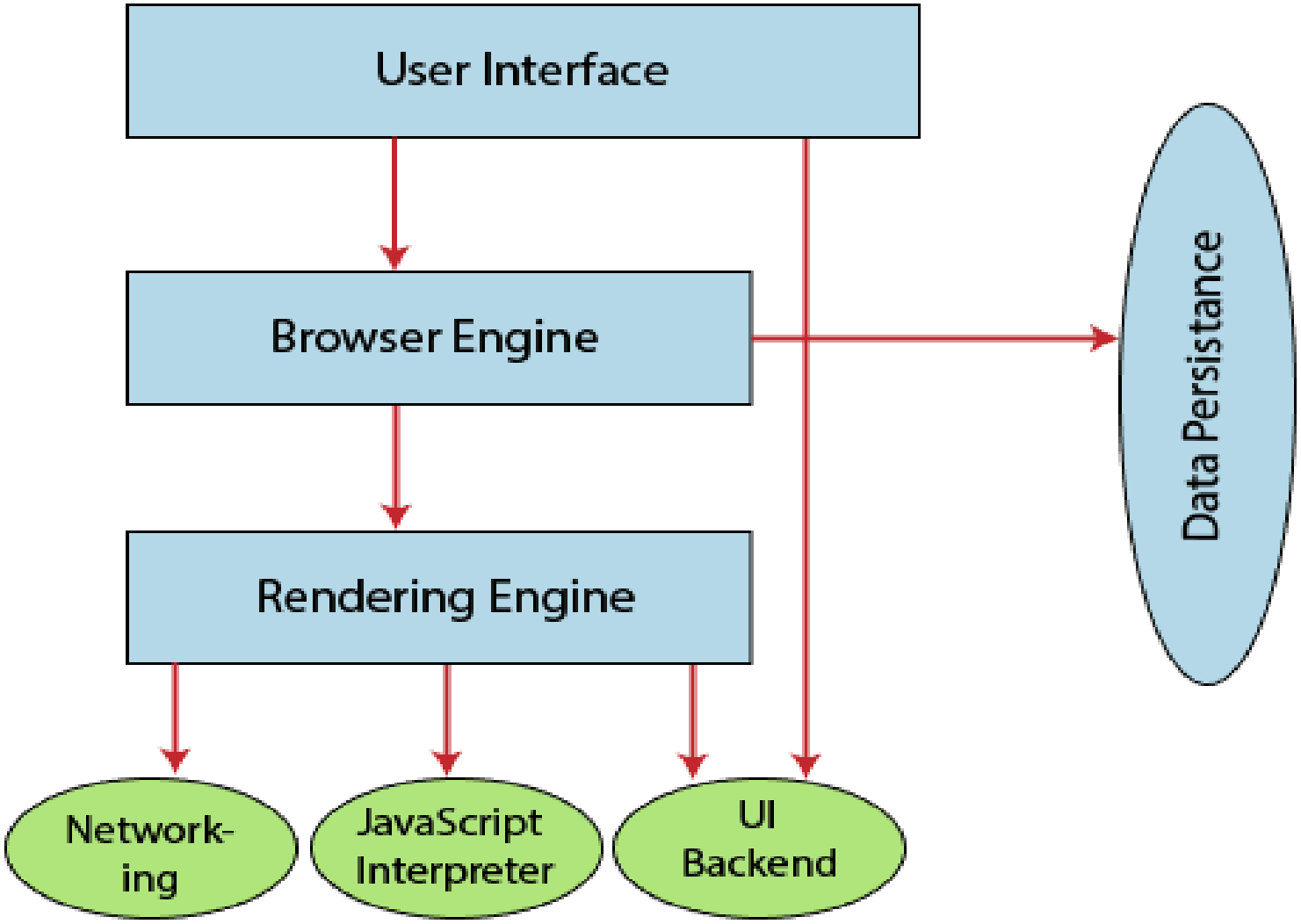
- **User Interface:** The user interface is an area where the user can use several options like address bar, back and forward button, menu, bookmarking, and many other options to interact with the browser.
- **Browser Engine:** It connects the UI (User Interface) and the rendering engine as a bridge. It queries and manipulates the rendering engine based on inputs from several user interfaces.
- **Rendering Engine:** It is responsible for displaying the requested content on the browser screen. It translates the HTML, XML files, and images, which are formatted by using the CSS. It generates the layout of the content and displays it on the browser screen.

- Although it can also display the other types of content by using different types of plugins or extensions. such as:

- Internet Explorer uses Trident
- Chrome & Opera 15+ use Blink
- Chrome (iPhone) & Safari use Webkit
- Firefox & other Mozilla browsers use Gecko

- **Networking:** It retrieves the URLs by using internet protocols like HTTP or FTP. It is responsible for maintaining all aspects of Internet communication and security. Furthermore, it may be used to cache a retrieved document to reduce network traffic.
- **JavaScript Interpreter:** As the name suggests, JavaScript Interpreter translates and executes the JavaScript code, which is included in a website. The translated results are sent to the rendering engine to display results on the device screen.

- **UI Backend:** It is used to draw basic combo boxes and Windows (widgets). It specifies a generic interface, which is not platform-specific.
- **Data Storage:** The data storage is a persistence layer that is used by the browser to store all sorts of information locally, like cookies. A browser also supports different storage mechanisms such as IndexedDB, WebSQL, localStorage, and FileSystem. It is a database stored on the local drive of your computer where the browser is installed. It handles user data like cache, bookmarks, cookies, and preferences.


User Interface
Browser Engine
Rendering Engine
Data Persistance
Network-
ing
JavaScript
Interpreter
UI
Backend

How does a browser work?

- When a user enters a web address or URL in the search bar like javatpoint.com, the request is passed to a **domain name servers** (DNS). All of these requests are routed via several routers and switches.
- The domain name servers hold a list of system names and their corresponding IP addresses. Thus, when you type something in the browser search bar, it gets converted into a number that determines the computers to which the search results are to be displayed.
- The browser acts as a part of the client-server model. A browser is a client program that sends the request to the server in response to the user search queries by using Hypertext Transfer Protocol or HTTP. When the server receives the request, it collects information about the requested document and forwards the information back to the browser.
- Thereafter, the browser translates and displays the information on the user device.

HTTP request and Response Model

- HTTP means HyperText Transfer Protocol.
- HTTP is used by the World Wide Web and this protocol.
- defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
- when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page.
- stateless protocol

Basic Features

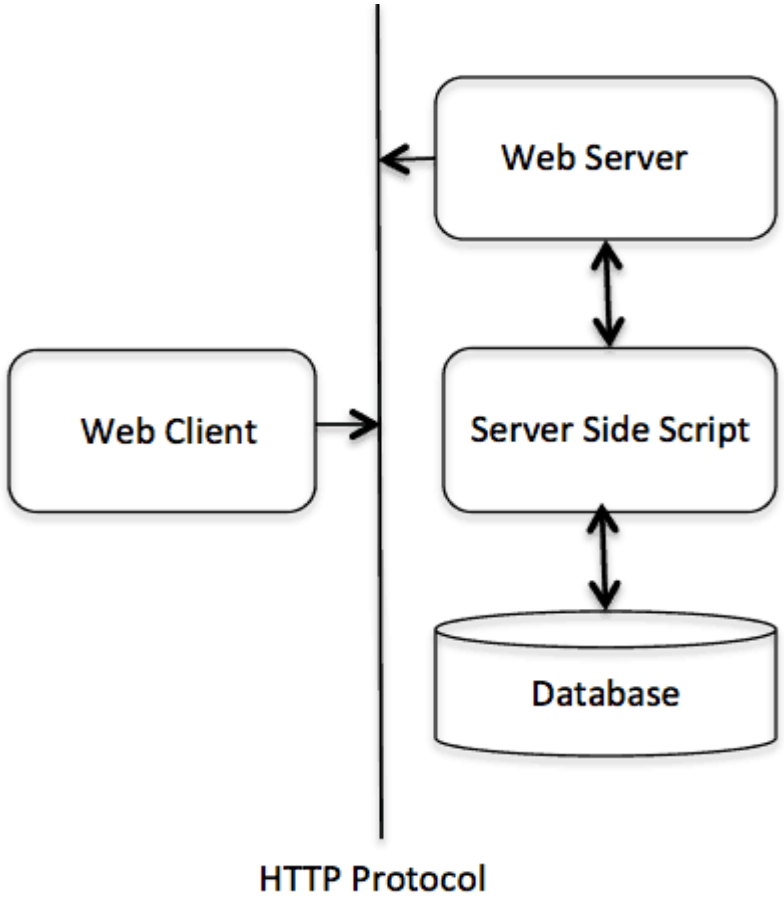
- There are three basic features that make HTTP a simple but powerful protocol:
- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.

- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.

- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

Basic Architecture

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



HTTP Request Method

- The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.
- Request message:

```
GET /guide/index.html HTTP/1.1
Host: www.xxxx.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE
6.0; Windows NT 5.1)
(blank line)
```

HTTP Request Methods

- GET
- POST
- PUT
- HEAD
- DELETE
- OPTIONS

Get Method

- **GET Method**
  - **used to request data from a specified resource.**
  - query string (name/value pairs) is sent in the URL of a GET request
  - /test/demo_form.php?name1=kamal&name2=nisha
- Notes
  - GET requests can be cached
  - GET requests remain in the browser history
  - GET requests can be bookmarked
  - GET requests should never be used when dealing with sensitive data
  - GET requests have length restrictions

Post Method

- used to send data to a server to create/update a resource.
- data sent to the server with POST is stored in the request body of the HTTP request:
- POST /test/demo_form.php HTTP/1.1
- Host:w3schools.com?name1=value1&name2=value2

- Notes
  - POST requests are never cached
  - POST requests do not remain in the browser history
  - POST requests cannot be bookmarked
  - POST requests have no restrictions on data length based o server configuration.

Put method

- **PUT is used to send data to a server to create/update a resource.**
- The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result.
- In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.

- **HEAD Method:**
- HEAD is almost identical to GET, but without the response body.
- HEAD requests are useful for checking what a GET request will return before actually making a GET request - like before downloading a large file or response body.
- **DELETE Method:**
- The DELETE method deletes the specified resource.
- **OPTIONS Method**
- The OPTIONS method describes the communication options for the target resource.

HTTP Headers

- **General-header:** These header fields have general applicability for both request and response messages.
- **Client Request-header:** These header fields have applicability only for request messages. This type of headers contains information about the fetched request by the client.
- **Server Response-header:** These header fields have applicability only for response messages. This type of headers contains the location of the source that has been requested by the client.
- **Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request. This type of headers contains the information about the body of the resources like MIME type, Content-length.

HTTP Response Message

HTTP/1.1 200 OK

Date: Sun, 18 Oct 2015 08:56:53 GMT

Server: Apache/2.2.14 (Win32)

Last-Modified: Sat, 20 Nov 2015 07:16:26 GMT

Accept-Ranges: bytes

Content-Length: 44

Connection: close

Content-Type: text/html

X-Pad: avoid browser bug

<html><body><h1>Guide</h1> This is guide on HTTP protocol </body></html>

Sr. No.	Code and Description
1	**1xx: Informational:** It means the request was received and the process is continuing.
2	**2xx: Success:** It means the action was successfully received, understood, and accepted.
3	**3xx: Redirection:** It means further action must be taken in order to complete the request.
4	**4xx: Client Error:** It means the request contains incorrect syntax or cannot be fulfilled.
5	**5xx: Server Error:** It means the server failed to fulfill an apparently valid request.

Status Code	Reason Phrase	Description
200	OK	Standard response for successful request
401	Unauthorized	Resources are password protected
404	Not found	Requested resource is not present currently
509	Bandwidth Limit Exceeded	The server is temporarily unable to respond your request due to the site owner reaching bandwidth limit.
500	Internal server error	Software internal failure

Status Code	Reason Phrase	Description
101	Switching Protocols	The protocol SHOULD be switched only when it is advantageous to do so. For example, switching to a newer version of HTTP is advantageous over older versions.
200	OK	Standard response for successful request.
307	Temporary Redirect	User agent MUST NOT automatically redirect the request unless it can be confirmed by the user.
401	Unauthorized	Resources are password protected.

Difference

http	https
Transfers data in hypertext (structured text) format	Transfers data in encrypted format
Uses port 80 by default	Uses port 443 by default
Not secure	Secured using SSL technology
Starts with `http://`	Starts with `https://`

Structure of HTML

- HTML is the standard markup language for creating Web pages
- stands for Hyper Text Markup Language
- describes the structure of a Web page
- consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
- HTML is not case sensitive

Cont..

```
<html>
    A little bit of html
</html>
```

- Anything written in triangular bracket is known as tag.
- <html> defines that it is HTML.
- First and last line is same except / it denotes closing tag is matching with opening tag.

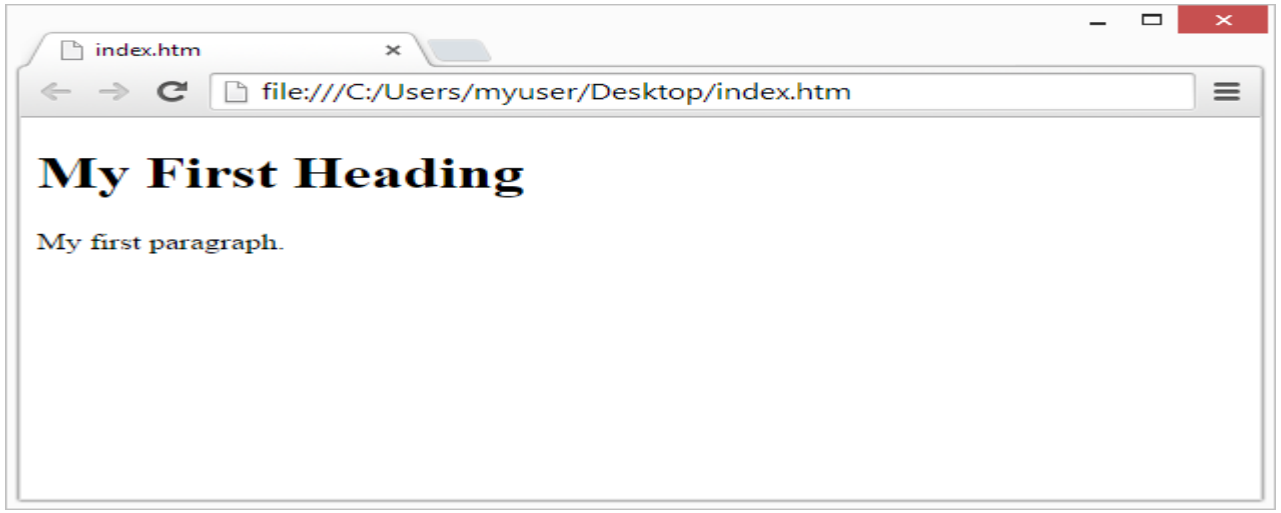
Note:

- Html tags are in lower case.
- Html is not case sensitive

```
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>

                <h1>My First Heading</h1>
                <p>My first paragraph.</p>

        </body>
</html>
```

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

- The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them.
- The browser does not display the HTML tags, but uses them to determine how to display the document:



HTML Page Structure

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Doctype in HTML

- The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.
- The <!DOCTYPE> declaration is not an HTML tag.
- It is an instruction to the web browser about what version of HTML the page is written in.
- The DTD specifies the rules for the mark up language, so that the browsers render and display the content correctly.
- There are different document types on the web.
- To display a document correctly, the browser must know both type and version.
- The doctype declaration is not case sensitive. All cases are acceptable:
- <!DOCTYPE html>
- <!DOCTYPE HTML>
- <!doctype html>
- <!Doctype Html>

Common DOCTYPE Declarations

- **HTML 5**
- <!DOCTYPE html>
- **HTML 4.01 Strict**
- This DTD contains all HTML **elements and attributes**, but does NOT INCLUDE presentational or deprecated elements (**like font**). **Framesets** are not allowed.
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- **HTML 4.01 Transitional**
- This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font). Framesets are not allowed.
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Common DOCTYPE Declarations

- **HTML 4.01 Frameset**
- This DTD is equal to HTML 4.01 Transitional, but allows the use of frameset content.
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

HTML Elements, tags and attributes

- **HTML Elements**:
- An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:
- <tagname>Content goes here...</tagname>
- The HTML **element** is everything from the start tag to the end tag:
- My first paragraph.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	

- **Empty element**:
- HTML elements with no content are called empty elements.
- Empty elements do not have an end tag, such as the
 element (which indicates a line break).
- **Nested HTML Elements:**
- HTML elements can be nested (elements can contain elements).
- All HTML documents consist of nested HTML elements.

- This example contains four HTML elements:
- <!DOCTYPE html>
<html>

<body>

<h1>My First Heading</h1>
My first paragraph.

</body>
</html>

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an HTML element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**
- **The href Attribute:**
- HTML links are defined with the <a> tag. The link address is specified in the href attribute:
- This is a link

- **The src Attribute:**
- HTML images are defined with the <img> tag.
- The filename of the image source is specified in the src attribute:
- <img src="img_girl.jpg">
- **The width and height Attributes**
- HTML images also have width and height attributes, which specifies the width and height of the image:
- <img src="img_girl.jpg" width="500" height=“500">
- The width and height are specified in pixels by default; so width="500" means 500 pixels wide.

- The alt attribute specifies an alternative text to be used, if an image cannot be displayed.
- The value of the alt attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a vision impaired person, can "hear" the element.
- <img src="img_girl.jpg" alt="Girl with a jacket">
- **The style Attribute**
- The style attribute is used to specify the styling of an element, like color, font, size etc.
- This is a paragraph.

- **The lang Attribute:**
- The language of the document can be declared in the <html> tag.
- The language is declared with the lang attribute.
- Declaring a language is important for accessibility applications (screen readers) and search engines:

- <!DOCTYPE html>
  <html lang="en-US">
  <body>

  ...

  </body>
  </html>
- The first two letters specify the language (en) If there is a dialect, add two more letters (US).
- **The title Attribute**
- Here, a title attribute is added to the element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:
-
  This is a paragraph.

Formatting and Fonts

- **HTML Headings**
- Headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.
- **Note:** Browsers automatically add some white space (a margin) before and after a heading.
- Headings Are Important for Search engines to index the structure and content of your web pages.
- Users often skim a page by its headings. It is important to use headings to show the document structure.

```
<html>
    <body>

        <h1>Heading 1</h1>
        <h2>Heading 2</h2>
        <h3>Heading 3</h3>
        <h4>Heading 4</h4>
        <h5>Heading 5</h5>
        <h6>Heading 6</h6>

    </body>
</html>
```

Bigger Headings

- Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute, using the CSS font-size property:

```
<html>
        <body>

                <h1 style="font-size:60px;">Heading 1</h1>
                <p>You can change the size of a heading with the
style attribute,        using the font-size property.</p>

        </body>
</html>
```

HTML Horizontal Rules

- **HTML Horizontal Rules:**
- The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.
- The <hr> element is used to separate content (or define a change) in an HTML page

```
<html>
        <head>
                <title>1st webpage</title>
        </head>
        <body>
                <h1>My First Heading</h1>
                <hr/>
                <p>My first paragraph.</p>

        </body>
</html>
```

HTML Paragraphs

- **HTML Paragraphs:**
- The HTML <p> element defines a paragraph.
- **Note:** Browsers automatically add some white space (a margin) before and after a paragraph.
- You cannot be sure how HTML will be displayed. Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.
- The browser will remove any extra spaces and extra lines when the page is displayed:

```
<html>
    <body>

        <p>
            This paragraph
            contains a lot of lines
            in the source code,
            but the browser
            ignores it.
        </p>
        <p>
            This paragraph
            contains     a lot of spaces
            in the source    code,
            but the    browser
            ignores it.
        </p>
        <p>
            The number of lines in a paragraph depends on the size of the browser window.
            If you resize the browser window, the number of lines in this paragraph will
            change.
        </p>
    </body>
</html>
```

- **Br tag:**
- The HTML
 element defines a line break.
- Use
 if you want a line break (a new line) without starting a new paragraph.
- The
 tag is an empty tag, which means that it has no end tag.

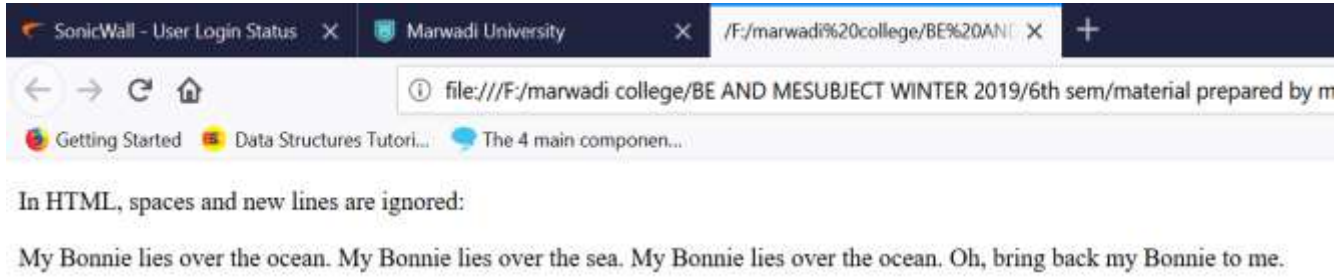
```
<html>
        <body>

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

        </body>
</html>
```

PRE tag

- **The Poem Problem:**

```
<html>
        <body>
                <p>In HTML, spaces and new lines are ignored:</p>
                <p>
                        My Bonnie lies over the ocean.
                        My Bonnie lies over the sea.
                        My Bonnie lies over the ocean.
                        Oh, bring back my Bonnie to me.
                </p>
        </body>
</html>
```

- This poem will display on a single line.



- The HTML <pre> element defines preformatted text.
- The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks

```
<html>
        <body>

            <p>In HTML, spaces and new lines are ignored:</p>

            <pre>

                My Bonnie lies over the ocean.

                My Bonnie lies over the sea.

                My Bonnie lies over the ocean.

                Oh, bring back my Bonnie to me.

            </pre>

        </body>
</html>
```

HTML Styles

- Setting the style of an HTML element, can be done with the style attribute.
- <*tagname* style="*property*:*value;*">
- The ***property*** is a CSS property. The ***value*** is a CSS value.
- **style attribute:** for styling HTML elements
- **background-color:** for background color
- **color :** for text colors
- **font-family:** for text fonts
- **font-size:** for text sizes
- **text-align:** for text alignment

- **Background Color**
- The CSS **background-color** property defines the background color for an HTML element.

```
<body style="background-color:powderblue;">

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

</body>
```

- **Font family**
- The CSS **font-family** property defines the font to be used for an HTML element

```
<body>

        <h1 style="font-family:verdana;">This is a heading</h1>
        <p style="font-family:courier;">This is a paragraph.</p>

</body>
```

- **Font color**
- **t**he CSS **color** property defines the font to be used for an HTML element.

```
<body>

        <h1 style="color:red;">This is a heading</h1>
        <p style="font-family:courier;">This is a paragraph.</p>

</body>
```

- **Text Size**
- The CSS **font-size** property defines the text size for an HTML element.

```
<html>
        <body>

                <h1 style="font-size:300%;">This is a heading</h1>
                <p style="font-size:160%;">This is a paragraph.</p>

        </body>
</html>
```

- **Text Alignment:**
- The CSS **text-align** property defines the horizontal text alignment for an HTML element.

```
<html>
        <body>

        <h1 style="text-align:center;">Centered Heading</h1>
        <p style="text-align:right;">Centered paragraph.</p>

        </body>
</html>
```

- Alignments can be:
- Right
- Left
- Center
- Justify

```
<html>
        <body>

                <p>I am normal</p>
                <p style="color:red;">I am red</p>
                <p style="color:blue;">I am blue</p>
                <p style="font-size:50px;">I am big</p>

        </body>
</html>
```

HTML Text Formatting

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

- **HTML <b> and Elements:**
- **Bold:**
- The HTML <b> element defines bold text, without any extra importance.

```
<html>
        <body>

        <p>This text is normal.</p>
        <p><b>This text is bold.</b></p>

        </body>
</html>
```

- **HTML <i> and Elements**
- **Italic:**
- The HTML <i> element defines italic text, without any extra importance.

```
<html>
        <body>
                <p>This text is normal.</p>
                <p><i>This text is italic.</i></p>
        </body>
</html>
```

- **Emphasize:**
- The HTML element defines emphasized text, with added semantic importance.

```
<html>
        <body>
                <p>This text is normal.</p>
                <p><em>This text is emphasized.</em></p>
        </body>
</html>
```

Note: Browsers display as <b>, and as <i>. However, there is a difference in the meaning of these tags: <b> and <i> defines bold and italic text, but and means that the text is "important".

- **HTML <small> Element:**
- The HTML <small> element defines smaller text.

```
<html>
        <body>
        <h2>HTML<small>Small</small>Formatting</h2>
        </body>
</html>
```

HTML <mark> Element

- The HTML <mark> element defines marked/highlighted text

```
<html>
        <body>
                <h2>HTML <mark>Marked</mark> Formatting</h2>
        </body>
</html>
```

HTML <del> Element

- The HTML <del> element defines deleted/removed text.

```
<html>
        <body>
                <p>The del element represents deleted (removed) text.</p>
                <p>My favorite color is <del>blue</del> red.</p>
        </body>
</html>
```

- **HTML <ins> Element**
- The HTML <ins> element defines inserted/added text.

```
<html>
        <body>
                <p>The del element represents deleted (removed) text.</p>
                <p>My favorite color is <ins>blue</ins> red.</p>
        </body>
</html>
```

HTML <sub> Element

- The HTML <sub> element defines subscripted text.

```
<html>
        <body>
                <This is <sub>subscripted</sub> text.</p>
        </body>
</html>
```

HTML <sup> Element

- The HTML <sup> element defines superscripted text.

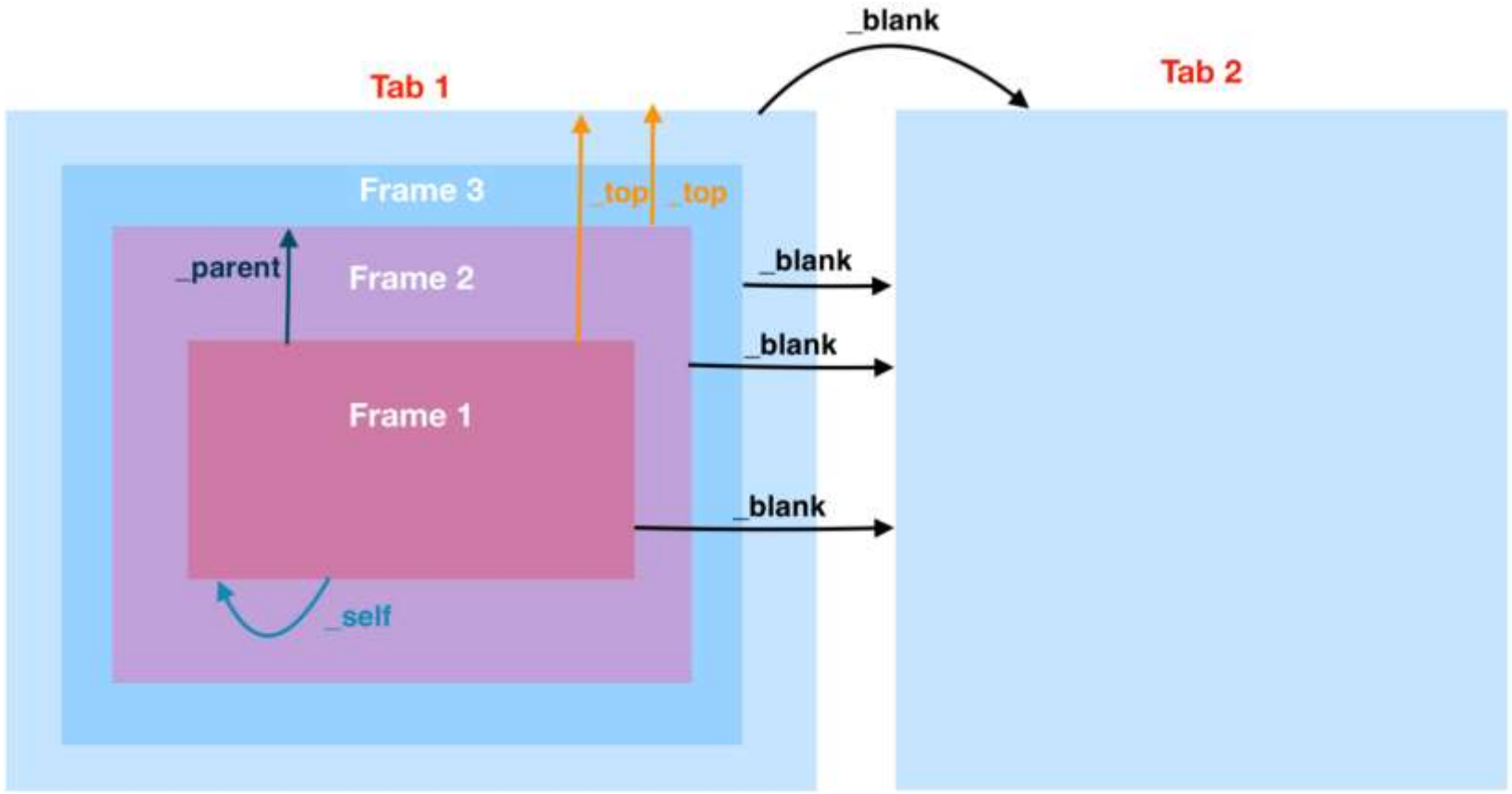
```
<html>
        <body>
                <This is <sup>subscripted</sup> text.</p>
        </body>
</html>
```

HTML Links

- Links are found in nearly all web pages. Links allow users to click their way from page to page.
- HTML links are hyperlinks. You can click on a link and jump to another document.
- ***link text***
- Hyperlinks are defined with the HTML <a> anchor tag.
- The href attribute specifies the destination address
- The **link text** is the visible part

- **HTML Links - The target Attribute**
- The target attribute specifies where to open the linked document.

- **_blank** - Opens the linked document in a new window or tab
- **_self** - Opens the linked document in the same window/tab as it was clicked (this is default)
- **_parent** - Opens the linked document in the parent frame
- **_top** - Opens the linked document in the full body of the window
- **framename** - Opens the linked document in a named frame


_blank
Tab 1
Tab 2
Frame 3
_top
_top
_parent
Frame 2
_blank
_blank
Frame 1
_blank
_self

- Imagine a webpage containing 3 nested <iframe> aka "frame"/"frameset". So:

- the outermost webpage/browser is the starting context
- the outermost webpage is the parent of frame 3
- frame 3 is the parent of frame 2
- frame 2 is the parent of frame 1
- frame 1 is the innermost frame

- Then target attributes have these effects:
- If frame 1 has a link with target="_self", the link targets frame 1 (i.e. the link targets the frame containing the link (i.e. targets itself))
- If frame 1 has a link with target="_parent", the link targets frame 2 (i.e. the link targets the parent frame)
- If frame 1 has a link with target="_top", the link targets the initial webpage (i.e. the link targets the topmost/outermost frame; (in this case; the link skips past the grandparent frame 3))
- If frame 2 has a link with target="_top", the link also targets the initial webpage (i.e. again, the link targets the topmost/outermost frame)
- If any of these frames has a link with target="_blank", the link targets an auxiliary browsing context, aka a "new window"/"new tab"
- This applies to frame 3, frame 2, frame 1, and the outermost webpage. Be careful of "tabnabbing" in case of target="_blank"; use the rel="noopener" attribute

- **Image as Link**

```
<a href="default.asp">
        <img src="smiley.gif" alt="HTML tutorial"
style="width:42px;height:42px;border:0;">
</a>
```

- **Link Titles**
- specifies extra information about an element.

Example 1

```
<html lang="en-US">
    <body>
        <a href="https://www.marwadiuniversity.ac.in/">Visit our
        HTML tutorial</a>
        <a href=https://www.marwadiuniversity.ac.in/“
        target="_blank">Visit W3Schools!</a>
        <a href="default.asp">
          <img src="smiley.gif" alt="HTML tutorial"
        style="width:42px;height:42px;border:0;">
        </a>
        <a href="https://www.marwadiuniversity.ac.in/” title="Go
        to W3Schools HTML section">Visit our HTML Tutorial</a>
    </body>
</html>
```

Example 2

```
<!DOCTYPE html>
<html>
        <head>
                <title>anchor tag</title>
        </head>
        <body>
        <a href="https://www.google.com/" target="_blank">Click here</a><br>
        <a href="https://www.google.com/" target="_self">Click here</a><br>
        <a href="https://www.google.com/" target="_top">Click here</a><br>
        <a href="https://www.google.com/" target="_parent">Click here</a><br>
        <a href="https://www.google.com/" target="framename">Click here</a>
        </body>
</html>
```

HTML IMAGES

- Images can improve the design and the appearance of a web page.
- In HTML, images are defined with the <img> tag.
- <img src="*url*">
- The <img> tag is empty, it contains attributes only, and does not have a closing tag.
- The src attribute specifies the URL (web address) of the image
- **alt Attribute**
- The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

- If a browser cannot find an image, it will display the value of the alt attribute
- **Image Size - Width and Height**
- You can use the style attribute to specify the width and height of an image.
- **Images in Another Folder**
- If not specified, the browser expects to find the image in the same folder as the web page.
- However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

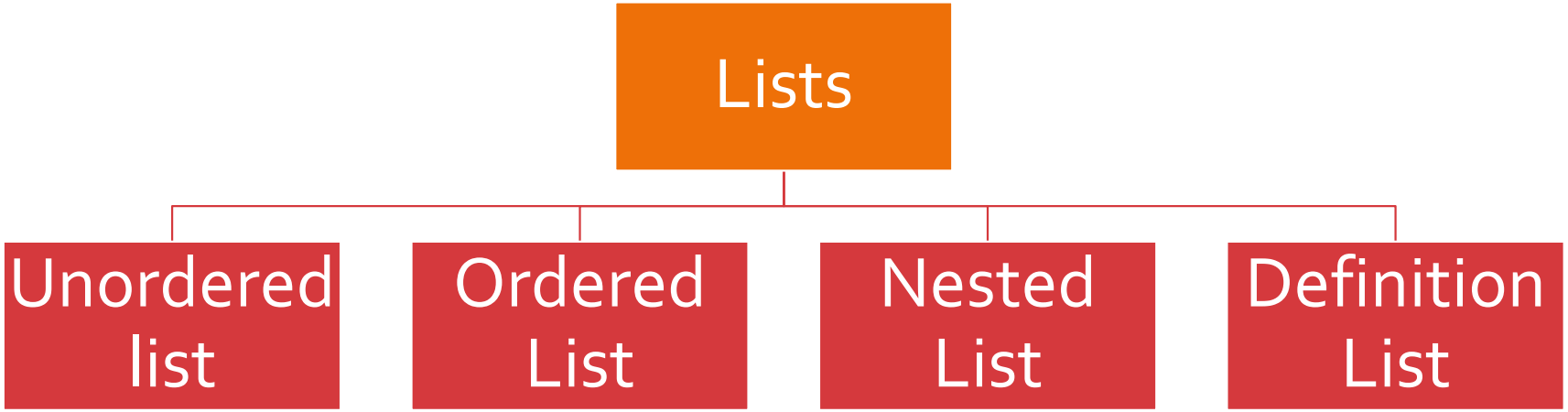
FILE PATH

Path	Description
<img src="picture.jpg">	picture.jpg is located in the same folder as the current page
<img src="images/picture.jpg">	picture.jpg is located in the images folder in the current folder
<img src="/images/picture.jpg">	picture.jpg is located in the images folder at the root of the current web
<img src="../picture.jpg">	picture.jpg is located in the folder one level up from the current folder

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

                        <img src="../images/rose.jpg" alt="rose"
                        width="500" height="333"><br><br>
                        <img src="images/rose.jpg" alt="rose"
                        width="500" height="333"><br><br>
                        <img src="/images/rose.jpg" alt="rose"
                        width="100" height="333"><br><br>
                        <img src="" alt="rose" width="500"
                        height="333">


        </body>
</html>
```

List

- Using HTML we can form Lists, which can be numbered or just bulleted.



UNORDERED List

- An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- The list items will be marked with bullets (small black circles) by default.
- The CSS list-style-type property is used to define the style of the list item marker.

```
<!DOCTYPE html>
<html>
        <body>
                <h2>An unordered HTML list</h2>

                <ul>
                 <li>Coffee</li>
                 <li>Tea</li>
                 <li>Milk</li>
                </ul>

        </body>
</html>
```

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

```
<!DOCTYPE html>
<html>
	<body>
		<h2>An unordered HTML list</h2>

		<ul style="list-style-type:disc;">
		 <li>Coffee</li>
		 <li>Tea</li>
		 <li>Milk</li>
		</ul>

	</body>
</html>
```

Ordered HTML List

- An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.
- The list items will be marked with numbers by default.The type attribute of the <ol> tag, defines the type of the list item marker.

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

```
<!DOCTYPE html>
<html>
        <body>
                <h2>An ordered HTML list</h2>
                    <ol>
                     <li>Coffee</li>
                     <li>Tea</li>
                     <li>Milk</li>
                    </ol>
        </body>
</html>
```

```
<!DOCTYPE html>
<html>
        <body>
                <h2>An ordered HTML list</h2>
                    <ol type="1">
                     <li>Coffee</li>
                     <li>Tea</li>
                     <li>Milk</li>
                    </ol>
        </body>
</html>
```

Nested HTML Lists

- List can be nested means we can nest list within another list.
- So we can make an unordered list inside a numbered one list.
- **Note:** List items can contain new list, and other HTML elements, like images and links, etc.

```
<!DOCTYPE html>
<html>
        <body>
                <ol>
                        <li>Coffee</li>
                        <li>Tea
                          <ul>
                                <li>Black tea</li>
                                <li>Green tea</li>
                        </ul>
                        </li>
                        <li>Milk</li>
                </ol>

        </body>
</html>
```

HTML Definition Lists

- HTML also supports **Definition** lists.
- A **Definition** list is a list of terms, with a description of each term.
- The <dl> tag defines the description list,
- the <dt> tag defines the term (name), and
- the <dd> tag describes each term.

```
<html>
        <body>
                <dl>
                        <dt>Coffee</dt>
                        <dd>- black hot drink</dd>
                        <dt>Milk</dt>
                        <dd>- white cold drink</dd>
                </dl>
        </body>
</html>
```

HTML TABLES

- To display data in tabular form, that is area is divided into rows and columns to show association.
- An HTML table is defined with the **<table>** tag.
- Each table row is defined with the **<tr>** tag.
- A table header is defined with the **<th>** tag. By default, table headings are bold and centered.
- A table data/cell is defined with the **<td>** tag.
- **Note:** The <td> elements are the data containers of the table.
- They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

```
<html>
        <body>
                <h2>Basic HTML Table</h2>
                <table style="width:100%" border="1">
                        <tr>
                                <th>Firstname</th>
                                <th>Lastname</th>
                               <th>Age</th>
                        </tr>
                        <tr>
                                 <td>jalpa</td>
                                <td>sonagar</td>
                                <td>50</td>
                        </tr>
```

```
                <tr>
                        <td>Eva</td>
                        <td>Jack</td>
                        <td>94</td>
                    </tr>
                    <tr>
                        <td>nency</td>
                        <td>divakar</td>
                        <td>80</td>
                    </tr>
                </table>

        </body>
</html>
```

Irregular Tables

- **Rowspan:** To make a cell span more than one row, use the rowpan attribute.
- **Colspan:** To make a cell span more than one column, use the colspan attribute.

Colspan

```
<html>
        <body>
                <table style="width:100%“ border=“1”> >
                        <tr>
                                <th>Name</th>
                                <th colspan="2">Telephone</th>
                        </tr>
                        <tr>
                                <td>Bill Gates</td>
                                <td>55577854</td>
                                <td>55577855</td>
                        </tr>
                </table>
        </body>
</html>
```

Rowspan

```
<html>
        <body>
                <table style="width:100%“ border=“1”>
                        <tr>
                                <th>Name</th>
                                <td>Bill Gates</td>
                        </tr>
                        <tr>
                                <th rowspan="2">Telephone</th>
                                <td>55577854</td>
                        </tr>
                        <tr>
                                <td>55577855</td>
                        </tr>
                </table>
        </body>
</html>
```

	Bus station			
	raiya chowk	nanavati chowk	indira circle	west zone
rmts bus	4 min	3 min	10 min	6 min
brts bus	5 min	8 min	11 min	9 min

Table Property

- **Adding a Caption**
- To add a caption to a table, use the <caption> tag right after table tag.
- **<caption>caption for table</caption>**
- **Adding table border:**
- If you do not specify a border for the table, it will be displayed without borders.
- A border is set using the CSS border property
- **<table border="1px">**
- **Collapsed Borders**
- If you want the borders to collapse into one border, add the CSS border-collapse property.
- **<table border="1px" style="border-collapse:collapse">**
- **Adding Cell Padding**
- Cell padding specifies the space between the cell content and its borders.
- To set the padding, use the CSS **padding** property or we can specify it in table tag **cellpadding** property.

- **Adding Cell spacing(Border Spacing)**
- The cell spacing is used to define the space between the cells.
- To set the spacing, use the CSS **border-spacing** property or we can specify it in table tag using **cellspacing**.
- Note: If the table has collapsed borders, border-spacing has no effect.
- <table style="width:100%" cellspacing="8px" cellpadding="15px">
- **Align Headings:**
- By default, table headings are bold and centered.
- To left-align the table headings, use the CSS text-align property
- **ID attribute:**
- To define a special style for a special table, add an id attribute to the table.
- **<table id="t01">**

HTML 5 Layout and syntax,

- HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces(API) and Document Object Model(DOM).
- **Features:**
- It has introduced new multimedia features which supports audio and video controls by using <audio> and <video> tags.
- There are new graphics elements including vector graphics and tags.
- Enrich semantic content by including <header> <footer>, <article>, <section> and <figure> are added.
- Drag and Drop- The user can grab an object and drag it further dropping it on a new location.

- Geo-location services- It helps to locate the geographical location of a client.
- Web storage facility which provides web application methods to store data on web browser.
- Uses SQL database to store data offline.
- Allows to draw various shapes like triangle, rectangle, circle, etc.
- Capable of handling incorrect syntax.
- Easy DOCTYPE declaration i.e. <!doctype html>
- Easy character encoding i.e. <meta charset=”UTF-8”>

New Added Elements in HTML 5:

- **<article>:** The <article> tag is used to represent an article. More specifically, the content within the <article> tag is independent from the other content of the site (even though it can be related).
- **<aside>:** The <aside> tag is used to describe the main object of the web page in a shorter way like a highlighter. It basically identifies the content that is related to the primary content of the web page but does not constitute the main intent of the primary page. The <aside> tag contains mainly author information, links, related content and so on.
- **<figcaption>:** The <figurecaption> tag in HTML is used to set a caption to the figure element in a document.

- **<figure>:** The <figure> tag in HTML is used to add self-contained content like illustrations, diagrams, photos or codes listing in a document. It is related to main flow but it can be used in any position of a document and the figure goes with the flow of the document and if remove it then it should not affect the flow of the document.
- **<header>:** It contains the section heading as well as other content, such as a navigation links, table of contents, etc.
- **<footer>:** The <footer> tag in HTML is used to define a footer of HTML document. This section contains the footer information (author information, copyright information, carriers etc). The footer tag are used within body tag. The <footer> tag is new in the HTML 5. The footer elements require a start tag as well as an end tag.
- **<main>:** Delineates the main content of the body of a document or web app.

- **<mark>:** The <mark> tag in HTML is used to define the marked text. It is used to highlight the part of the text in the paragraph.
- **<nav>:** The <nav> tag is used to declaring the navigational section in HTML documents. Websites typically have sections dedicated to navigational links, which enables user to navigate the site. These links can be placed inside a nav tag.
- **<section>:** It demarcates a thematic grouping of content.
- **<details>:** The <details> tag is used for the content/information which is initially hidden but could be displayed if the user wishes to see it. This tag is used to create interactive widget which user can open or close it. The content of details tag is visible when open the set attributes.

- **<summary>:** The <summary> tag in HTML is used to define a summary for the <details> element. The <summary> element is used along with the <details> element and provides a summary visible to the user. When the summary is clicked by the user, the content placed inside the <details> element becomes visible which was previously hidden. The <summary> tag was added in HTMl 5. The <summary> tag requires both starting and ending tag.

- **<output>:** The <output> tag in HTML is used to represent the result of a calculation performed by the client-side script such as JavaScript.
- **<progress>:** It is used to represent the progress of a task. It is also define that how much work is done and how much is left to download a things. It is not used to represent the disk space or relevant query.
- **<svg>:** It is the Scalable Vector Graphics.
- **<canvas>:** The <canvas> tag in HTML is used to draw graphics on web page using JavaScript. It can be used to draw paths, boxes, texts, gradient and adding images. By default it does not contains border and text.
- **<audio>:** It defines the music or audio content.

- **<embed>:** Defines containers for external applications (usually a video player).
- **<source>:** It defines the sources for <video> and <audio>.
- **<track>:** It defines the tracks for <video> and <audio>.
- **<video>:** It defines the video content.
- **<time>:** The <time> tag is used to display the human-readable data/time. It can also be used to encode dates and times in a machine-readable form. The main advantage for users is that they can offer to add birthday reminders or scheduled events in their calender's and search engines can produce smarter search results.

- **<wbr>:** The <wbr> tag in HTML stands for word break opportunity and is used to define the position within the text which is treated as a line break by the browser. It is mostly used when the used word is too long and there are chances that the browser may break lines at the wrong place for fitting the text.
- **<datalist>:** The <datalist> tag is used to provide autocomplete feature in the HTML files. It can be used with input tag, so that users can easily fill the data in the forms using select the data.

```
<!DOCTYPE html>
<html>
    <head>
        <title>html5 </title>
        <style>
            section{
                border:1px solid pink;
                padding:15px;
                margin:10px;
                }
        </style>
    </head>
<body>
        <article>
            <h2>Narendra Modi</h2>
            <i>(Naam to suna hi hoga) </i>
            <p>Narendra DamodarDas Modi is the 15th and current
Prime Minister of India,
            Modi, a leader of the Bharatiya Janata Party (BJP),
previously served as the Chief Minister
            of Gujarat state from 2001 to 2014. He is currently the
Member of Parliament (MP) from Varanasi. </p>
        </article>
```

<aside>,<canvas>

- <aside>
- <h3>New Delhi</h3>
- New Delhi is the capital of India.
  </aside>
- <canvas id="myCanvas1" width="300" height="100" style="border:2px solid;">Your browser does not support the HTML5 canvas tag.
- </canvas>

<data>,<datalist>

```
<ul>
        <li><data value="101">Java Tutorial</data></li>
        <li><data value="111">SQL tutorial</data></li>
        <li><data value="121">HTML tutorial</data></li>
</ul>
<label>
        Enter your favorite cricket player: Press any character<br />
        <input type="text" id="favCktPlayer" list="CktPlayers">
                <datalist id="CktPlayers">
                        <option value="Sachin Tendulkar">
                        <option value="Brian Lara">
                        <option value="Jacques Kallis">
                        <option value="Ricky Ponting">
                        <option value="Rahul Dravid">
                        <option value="Shane Warne">
                </datalist>
</label>
```

<details>,<summary>

- <details>
- <summary>Copyright 2011-2014.</summary>
- - by JavaTpoint. All Rights Reserved.
- All content and graphics on this web site are the property of the javatpoint.com
- </details>
- Note: The details tag is currently only supported in Opera, Chrome, and in Safari 6.
-

<figure>, <figurecaption>

- The Taj Mahal is widely recognized as "the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage". It is regarded by many as the finest example of Mughal architecture, a style that combines elements from Islamic, Persian, Ottoman, Turkish and Indian architectural styles.

<figure>

- <imgsrc="https://static.javatpoint.com/htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>
- <figcaption>Fig.1.1 - A front view of the great Taj Mahal in Agra.</figcaption>

</figure>

<header>, <nav>

```
<header>
    <h2>ABCOnline.com</h2>
    <p> World's no.1 shopping website</p>
    <nav>
        <a href="#">Home</a> |
        <a href="#">Courses</a> |
        <a href="#">About-us</a> |
        <a href="#">Contact-us</a> |
    </nav>
</header>
```

<section>

- <h2> Indian Leader</h2>
- <section>
- <h3> Jawaharlal Nehru </h3>
- Jawaharlal Nehru was the first Prime Minister of India and a central figure in Indian politics for much of the 20th century. He emerged as the paramount leader of the Indian independence movement under the tutelage of Mahatma Gandhi. -Source Wikipedia
- </section>
- <section>
- <h3>Subhas Chandra Bose </h3>
- Subhas Chandra Bose was an Indian nationalist whose attempt during World War II to rid India of British rule with the help of Nazi Germany and Japan left a troubled legacy. The honorific Netaji (Hindustani language: "Respected Leader"), first applied to Bose in Germany, by the Indian soldiers of the Indische Legion and by the German and Indian officials in the Special Bureau for India in Berlin, in early 1942, is now used widely throughout India. -source Wikipedia
- </section>

<progress>, <menu>

- Downloading progress:
- <progress value="43" max="100"></progress>
- <h2>Example of Menu Tag</h2>
- <menu>
- Home
- Registration
- Contact-us
- About-us
- </menu>

<footer>,<address>,

- <footer>
- Posted by: Sonoo Jaiswal
-
- <address> JavaTpoint, plot no. 6, near MMX Mall,Mohan Nagar, Ghaziabad Pin no. 201007
- </address>
-
- Contact information:
- <ahref="mailto:sonoojaiswal1987@gmail.com">sonoojaiswal1987@gmail.com.
-
- </footer>

<svg>,<video> <source>,<circle>

- <svg width="100" height="100">
- <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />
- </svg>
- <video controls>
- <source src="movie.mp4" type="video/mp4">
- Your browser does not support the html video tag.
- </video>

<audio>

```
<audio controls>
        <source src="koyal.mp3" type="audio/mpeg">
        Your browser does not support the html audio tag.
</audio>
</body>
</html>
```

HTML Event Attributes

- When a browser reacts on user action, then it is called as an event. For example, when you click on the submit button, then if the browser displays an information box.
- In HTML5 there are lots of event attributes available which can be activated using a programming language such as JavaScript.
- Following is a table of event attributes, using these attributes you can perform several events.
- **Windows Event Attributes**
- Windows events are related for the window object, and it can only be applied with <body> tag

Windows Event Attributes

Attribute	Description
onafterprint	Executed the script after the document is printed.
onbeforeprint	Executed the script before the document is printed.
onbeforeunload	Executed the script before a document being unloaded.
onerror	Executed the script when an error occurs.
onhashchange	Executed the script when the anchor part in URL of the webpage is changed.
onload	Executed the script when the webpage is entirely loaded.
onmessage	Executed the script when a message event occurs.
onoffline	Executed the script when the network connection is disconnected, and browser started working offline.

Windows Event Attributes

ononline	Executed the script when the browser started working online
onpagehide	Executed the script when the current webpage is hidden such as if the user has moved away from the current webpage.
onpageshow	Executed the script when the current webpage is focused.
onpopstate	Executed the script when the window's active history is changed.
onresize	Executed the script when the window is resized.
onstorage	Executed the script when web storage is updated.
onunload	Executed the script when the current webpage is unloaded, or window is closed.

Example of Window event

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

<img src="w3html.gif" onload="loadImage()" width="100" height="132">

<script>
function loadImage() {
  alert("Image is loaded");
}
</script>

</body>
</html>
```

Form Event Attributes

- **Form Event Attributes**
- Form event occurs when the user performs some action within the form such as submitting the form, selecting input field, etc.
- The form events can be used with any element, but these are mainly used with HTML form elements.
- Following is the list of all Form Event attributes:

Attribute	Description
onblur	Executed the script when form element loses the focus.
onchange	Executed the script when the value of the element is changed.
onfocus	Trigger an event when the element gets focused.
oninput	Executed the script when the user enters input to the element.
oninvalid	Executed the script when the element does not satisfy its predefined constraints.
onreset	Triggers the event when user reset the form element values.
onsearch	Triggers the event when a search field receives some input.
onselect	Triggers the event when the user has selected some text.
onsubmit	Triggers the event when a form is submitted.

Form event Example

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

<p>When you reset the form, a function is triggered which alerts some text.</p>

<form onreset="myFunction()">
  Enter name: <input type="text">
  <input type="reset">
</form>

<script>
function myFunction() {
  alert("The form was reset");
}
</script>

</body>
</html>
```

- More example on: https://www.w3schools.com/tags/ref_eventattributes.asp

Keyboard Event Attributes

- **Keyboard Event Attributes**
- Keyboard event occurs when a user interacts with the keyboard. Following is a list of the Keyboard event.

Attribute	Description
onkeydown	Triggers the event when the user presses down a key on the keyboard.
onkeypress	Trigger the event when the user presses the key which displays some character.
onkeyup	Trigger the event when the user releases the currently pressed key.

Keyboard Example

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

<p>A function is triggered when the user is pressing a key in the input field.</p>

<input type="text" onkeypress="myFunction()">

<script>
function myFunction() {
  alert("You pressed a key inside the input field");
}
</script>

</body>
</html>
```

Mouse Event Attributes

Attribute	Description
onclick	Trigger the event when the mouse clicks on the element.
ondblclick	Trigger the event when mouse double-click occurs on the element.
onmousedown	Trigger the event when the mouse button is pressed on the element.
onmousemove	Trigger the event when the mouse pointer moves over the element.
onmouseout	Trigger the event when the mouse moves outside the element.
onmouseover	Trigger the event when the mouse moves onto the element.
onmouseup	Trigger the event when the mouse button is released.
onmousewheel	Deprecated. Use the onwheel attribute.
onwheel	Trigger the event when the mouse wheel rolls up or down on the element

Mouse Event Example

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

<button ondblclick="myFunction()">Double-click me</button>

<p id="demo"></p>

<p>A function is triggered when the button is double-clicked. The function outputs some text in a p element with id="demo".</p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello World";
}
</script>

</body>
</html>
```

Drag and Drop Event

Event Attribute	Description
ondrag	It triggers when an element is dragged.
ondragenter	It triggers when an element has been dragged to a valid drop target.
ondragend	It triggers at the end of the drag operation.
ondragover	It triggers when an element is being dragged over a valid drop target.
ondragleave	It triggers when an element leaves a valid drop target.
ondragstart	It triggers at the start of a drag operation.
ondrop	It triggers when dragged element is being dropped.

HTML VIDEO

The HTML <video> element is used to show a video on a web page.

How it Works?

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

```
<!DOCTYPE html>
<html>
        <head>
                <title>video</title>
        </head>
        <body>
                <video width="200px" height="250px" controls>
                <source src="movie.mp4  “type="video/mp4">
                </video>
        </body>
</html>
```

HTML <video> Autoplay

To start a video automatically use the autoplay attribute:

```
<!DOCTYPE html>
<html>
	<head>
		<title>video</title>
	</head>
	<body>
		<video width="200px" height="250px" autoplay>
		<source src="movie.mp4" type="video/mp4">
		</video>
		<p>Your browser does not support the autoplay attribute.</p>
	</body>
</html>
```

HTML Video - Methods, Properties, and Events

- The HTML DOM defines methods, properties, and events for the <video> element.
- This allows you to load, play, and pause videos, as well as setting duration and volume.
- There are also DOM events that can notify you when a video begins to play, is paused, etc.

```
<!DOCTYPE html>
<html>
        <head>
                <title>video</title>
        </head>
        <body>
                <div style="text-align:center">
                        <button
onclick="playpause()">Play/Pause</button>
                        <button onclick="big()">Big</button>
                        <button onclick="small()">small</button>
                        <button onclick="normal()">Normal</button>
```

```
<br><br>
<video id="v1" height="250px" width="250px">
    <source src="mov_bbb.mp4" type="video/mp4">
</video>
</div>
        <p>Your browser does not support the autoplay attribute.</p>
        <script>
            var myvd = document.getElementById("v1");
                function playpause(){
                    if(myvd.paused)
                        myvd.play();
                    else
                        myvd.pause();
                }
```

```
function big(){
        myvd.width= 400;
    }
function small(){
        myvd.width= 100;
    }
function normal(){
        myvd.width= 250;
        }
                </script>
        </body>
</html>
```

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML Video Tags

Tag	Description
<video>	Defines a video or movie
<source>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

Audio

- The HTML <audio> element is used to play an audio file on a web page.
- How It Works?
- The controls attribute adds audio controls, like play, pause, and volume.

- The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

- The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.
- HTML Audio Formats

- There are three supported audio formats: MP3, WAV, and OGG.

Common Audio Formats

- MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music.
- If your website is about recorded music, MP3 is the choice.

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

- <!DOCTYPE html>
- <html>
- <body>

- <audio controls>
- <source src="04. Kun Faya Kun.mp3" type="audio/mpeg">
- Your browser does not support the audio element.
- </audio>

- </body>
- </html>

HTML AUDIO

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML Audio - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the <audio> element. This allows you to load, play, and pause audios, as well as set duration and volume.

HTML Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>

HTML YouTube Videos

- The easiest way to play videos in HTML, is to use YouTube.
- Playing a YouTube Video in HTML

- To play your video on a web page, do the following:

- Upload the video to YouTube
- Take a note of the video id
- Define an <iframe> element in your web page
- Let the src attribute point to the video URL
- Use the width and height attributes to specify the dimension of the player

```
<!DOCTYPE html>
<html>
<body>
        <iframe width="420" height="345"
        src="https://www.youtube.com/embed/tgbNymZ7v
qY">
        </iframe>

</body>
</html>
```

HTML SVG

- The **HTML SVG** is an acronym which stands for Scalable Vector Graphics.
- HTML SVG is a modularized language which is used to describe graphics in XML. It describe two-dimensional vector and mixed vector/raster graphics in XML. It is a W3C recommendation. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with text editor, but generally drawing programs like inkspace are preferred to create it.
- SVG is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X,Y coordinate system etc.
- The <svg> element specifies the root of a SVG fragment. You can animate every element and every attribute in SVG files.

HTML SVG Circle Example

```
<!DOCTYPE html>
<html>
<body>
 <svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />
 </svg>
</body>
</html>
```

- **r**: It defines the radius of the circle.
- **cx**: It defines the x coordinates of the center of the circle.
- **cy**: It defines the y coordinates of the center of the circle.

HTML SVG Rectangle Example

```
<!DOCTYPE html>
<html>
<body>
<svg width="200" height="100">
 <rect width="200" height="100" stroke="yellow" stroke-width="4" fill="red" />
</svg>
</body>
</html>
```

- **x** : It define the position of the top left corner of the rectangle.
- **y** : It define the top position of the rectangle.
- **width** : It defines the width of the rectangle.
- **height** : It defines the height of the rectangle.
- **fill-opacity** : It is used to define the opacity of the fill color. Its range can be 0 to 1.
- **stroke-opacity** : It defines the opacity of the stroke color. Its range can be 0 to 1.

Ellipse

- Ellipse is a more general form of the circle element. You can scale the x and y radius of the circle separately.
- Attribute used to draw an ellipse:
- rx: It defines the horizontal radius.
- ry: It defines the vertical radius.
- cx: It defines the x coordinate of the center of the ellipse.
- cy: It defines the y coordinate of the center of the ellipse.

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

<svg height="500" width="500">
  <ellipse cx="250" cy="100" rx="120" ry="70" style=
"fill:red;stroke:black;stroke-width:3" />
</svg>

</body>
</html>
```

Line

- The line attribute is used to draw a straight line on the screen. It has two points which specify the start and end point of the line.
- **x1**: It defines the start of the line on the x-axis.
- **y1**: It defines start of the line on the y-axis.
- **x2**: It defines the end of the line on the x-axis.
- **y2**: It defines the end of the line on the y-axis.

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

<svg height="500" width="450">
 <line x1="5" y1="5" x2="150" y2="150" style="stroke:rgb(0,0,255);stroke-width:3" />
</svg>

</body>
</html>
```

HTML SVG polygon Example

```
<!DOCTYPE html>
<html>
<body>
 <svg height="210" width="500">
 <polygon points="100,10 40,198 190,78 10,78 160,198"
 style="fill:red;stroke:yellow;stroke-width:5;fill-rule:nonzero;" />
</svg>
</body>
</html>
```

Why SVG is preferred over other image formats?

- SVG images can be saved as the smallest size possible. Unlike bitmap image formats like JPG or PNG, it does not contain a fixed set of dots. So it is also easy to print with high quality at any resolution.
- SVG images can be zoomed to a certain level without degradation of the picture quality.
- SVG images and their behaviors are defined in XML text files, so they can be created and edited with any text editor.

SVG Text

- To draw text, <text> element is used.
- **x**: It defines the position of the top left corner of the text.
- **y**: It defines the top position of the text.
- **width**: It defines the width.
- **height**: It defines the height.
- **fill**: fill attribute is used to define the fill color.

```
<!DOCTYPE html>
<html>
  <title>SVG Text</title>
  <body>

    <h1>SVG Text</h1>

    <svg width="950" height="950">
      <g>
        <text x="40" y="23" >Text: </text>
        <text x="40" y="40" fill="rgb(121,0,121)">WWW.JavaTPoint
.COM</text>
      </g>
    </svg>

  </body>
</html>
```

SVG Transformation

- SVG provides four transformation functions:
- translate()
- rotate()
- scale()
- skew()
- **Translate**
- The translate() function moves a shape. You pass the x and y value to the translate() function inside the parameters.

Translate

- **<!DOCTYPE html>**
- **<html>**
- **<body>**
- **<svg height="200" width="300">**
- **<rect x="20" y="20" width="50" height="50" style="fill: #cc3333"/>**
- **<rect x="20" y="20" width="50" height="50" style="fill: #3333cc" transform="translate(75,95)" />**
- **</svg>**
- **</body>**
- **</html>**
- This example moves a shape 75 units along the x-axis and 95 units along the y-axis.

Rotate

- The rotate() function rotates a shape around the point 0,0. Here is an example showing a rectangle (outline), and an equal rectangle (filled) after a rotation of 15 degrees:
- <!DOCTYPE html>
- **<html>**
- **<body>**
- **<svg height="200" width="300">**
- **<rect x="20" y="20" width="50" height="50" style="fill: #cc3333"/>**
- **<rect x="20" y="20" width="50" height="50" style="fill: #3333cc" transform="rotate(20)" />**
- **</svg>**
- **</body>**
- **</html>**
- All rotation is clock-wise with a number of degrees going from 0 to 360. If you want to rotate counter-clock-wise, pass a negative number of degrees to the rotate() function.

Scale

- The scale() function scales a shape up or down in size. The scale() function scales both the shapes dimensions and its position coordinates. Thus, a rectangle positioned at 10,10 with a width of 20 and a height of 30, scaled by a factor of 2 will appear at 20,20 with a width of 40 and a height of 60.
- The scale() function also scales the stroke width of a shape.
- Here is an example that shows a rectangle (blue) positioned at 10,0 with a width of 20 and a height of 20, and an equal rectangle (black) which is scaled by a factor of 2:

Scale

- **<!DOCTYPE html>**
- **<html>**
- **<body>**
- **<svg height="200" width="300">**
- **<rect x="20" y="20" width="50" height="50" style="fill: #cc3333"/>**
- **<rect x="20" y="20" width="50" height="50" style="fill: #3333cc" transform="scale(2,3)" />**
- **</svg>**
- **</body>**
- **</html>**
- This example would scale the shape by a factor of 2 along the x-axis, and a factor of 3 along the y-axis.

Skew

- The skewX() and skewY() functions skew the x-axis and y-axis. Actually, the functions skew the given axis according to a certain angle specified in degrees.
- As you can see, the skewX() function makes the vertical lines look like they were rotated by the given angle.

```
<!DOCTYPE html>
<html>
    <body>
        <svg height="200" width="300">
        <rect x="20" y="20" width="50" height="50" style="fill: #cc3333"/>
        <rect x="20" y="20" width="50" height="50" style="fill: #3333cc" transform="skewX(30)" />
<rect x="20" y="20" width="50" height="50" style="fill: #3333cc" transform="skewY(30)" />

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

Web forms and validations

- To collect information from user, there is need to generate form.
- **The <form> Element**
- HTML <form> element defines a form that is used to collect user input.
- <form>
  - Form elements
- </form>
- An HTML form contains form elements.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

HTML form Elements and Attributes

- An HTML form contains form elements.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.
- **The <input> Element:**
- Collects information from user.
- Enter Your email address<input type="text" name="email" size=35>
- Text- what is expected from user
- tag- name attribute
- Size- size of characters for entry box

HTML form Elements and Attributes

- **Text Input**
- **<input type="text">** defines a one-line input field for text input.
- **Action Attribute**
- The action attribute defines the action to be performed when the form is submitted.
- Normally, the form data is sent to a web page on the server when the user clicks on the submit button.
- <form **action="/action_page.php**">
- **The Name Attribute**
- Each input field must have a name attribute to be submitted.
- If the name attribute is omitted, the data of that input field will not be sent at all.
- **<input type="text" name="last-name" value="nobita">.**

HTML form Elements and Attributes

- **Target Attribute**
- The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.
- The default value is **"_self"** which means the form will be submitted in the current window.
- To make the form result open in a new browser tab, use the value **"_blank“.**
- **<form action="/action_page.php" target="_blank">**

HTML form Elements and Attributes

- **The Method Attribute**
- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.
- **<form action="/action_page.php" method="get">**
- **<form action="/action_page.php" method="post">**
- **GET METHOD:**
- The default method when submitting form data is GET.
- However, when GET is used, the submitted form data will be visible in the page address field.
- /action_page.php?firstname=Mickey&lastname=Mouse .
- **POST METHOD:**
- if the form data contains sensitive or personal information.
- The POST method does not display the submitted form data in the page address field.

HTML form Elements and Attributes

- **Notes on POST**:
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked.
- **Fieldset:**
- The <fieldset> element is used to group related data in a form.
- The <legend> element defines a caption for the <fieldset> element.

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

```
<html>
    <body>
        <fieldset>
            <legend>Personal information:</legend>
                <form action="/action_page.php“ method=“get” 
target=“_blank”>
                    First name:<br>
                    <input type="text" value="doremon“><br>
                    Last name:<br>
                    <input type="text" name="lastname" value="nobita">
                    <br><br>
                    </form>
        </fieldset>
    </body>
</html>
```

HTML form Elements and Attributes

- **The <select> Element**
- The <select> element defines a drop-down list.
- The <option> elements defines an option that can be selected.
- By default, the first item in the drop-down list is selected.

```
<select name="cars">
        <option value="volvo">Volvo</option>
        <option value="toyato"> toyato </option>
        <option value="fiat">Fiat</option>
        <option value="audi">Audi</option>
</select>
```

- To define a pre-selected option, add the selected attribute to the option:
- **<option value="fiat" selected>Fiat</option>**

HTML form Elements and Attributes

- **<select>**
- **Visible Values:**
- Use the size attribute to specify the number of visible values.
- <select name="cars" **size="3"**>
- **Allow Multiple Selections:**
- Use the multiple attribute to allow the user to select more than one value.
- <select name="cars" size="4" **multiple**>
- Use ctrl+arrow to select multiple elements

<optgroup>

The <optgroup> is used to group related options in a drop-down list.

```
<select>
        <optgroup label="Swedish Cars">
                <option value="volvo">Volvo</option>
                <option value="saab">Saab</option>
        </optgroup>
        <optgroup label="German Cars">
                <option value="mercedes">Mercedes</option>
                <option value="audi">Audi</option>
        </optgroup>
</select>
```

HTML form Elements and Attributes

- **The <textarea> Element:**
- The <textarea> element defines a multi-line input field (a text area).
- <textarea name="message" rows="10" cols="30">
  The cat was playing in the garden.
  </textarea>
- The rows attribute specifies the visible number of lines in a text area.
- The cols attribute specifies the visible width of a text area.

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">

HTML form Elements and Attributes

- **Input Type Password**
- **<input type="password">** defines a password field.
- **Input Type Submit**
- **<input type="submit">** defines a button for submitting form data to a form-handler.
- The form-handler is typically a server page with a script for processing input data.
- The form-handler is specified in the form's action attribute.
- If you omit the submit button's value attribute, the button will get a default text.

HTML form Elements and Attributes

- **Input Type Reset**
- **<input type="reset">** defines a reset button that will reset all form values to their default values.
- If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.
- **Input Type Radio**
- **<input type="radio">** defines a radio button.
- Radio buttons let a user select ONLY ONE of a limited number of choices.

HTML form Elements and Attributes

```
<form>
        <input type="radio" name="gender" value="male" checked> Male<br>
        <input type="radio" name="gender" value="female"> Female<br>
        <input type="radio" name="gender“ value="other">Other
</form>
```

- **Input Type Checkbox**
- <input type="checkbox"> defines a checkbox.
- Checkboxes let a user select ZERO or MORE options of a limited number of choices.

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

HTML form Elements and Attributes

- **Input Type Button**
- <input type="button"> defines a button.
- **Input Type File**
- The **<input type="file">** defines a file-select field and a "Browse" button for file uploads.
- To upload a file through form we must use enctype.
- The enctype attribute specifies how the form-data should be encoded when submitting it to the server.
- Note: The enctype attribute can be used only if method="post".
- Syntax
- **<form enctype="value">**
- **<form enctype="multipar/form-data">**

HTML form Elements and Attributes

Attribute Values

Value	Description
application/x-www-form-urlencoded	Default. All characters are encoded before sent (spaces are converted to "+" symbols, and special characters are converted to ASCII HEX values)
multipart/form-data	This value is necessary if the user will upload a file through the form
text/plain	Sends data without any encoding at all. Not recommended

The value Attribute

The value attribute specifies the initial value for an input field.

```
<input type="text" name="firstname" value="John">
```

The readonly Attribute

The readonly attribute specifies that the input field is read only (cannot be changed).

```
<input type="text" name="firstname" value="John" readonly>
```

The disabled Attribute

The disabled attribute specifies that the input field is disabled.

A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form.

```
<input type="text" name="firstname" value="John" disabled>
```

HTML form Elements and Attributes

- **The size Attribute**
- The size attribute specifies the size (in characters) for the input field.
- **<input type="text" name="firstname" value="John" size="40">**
- **The maxlength Attribute**
- The maxlength attribute specifies the maximum allowed length for the input field.
- **<input type="text" name="firstname" maxlength="10">**
- With a maxlength attribute, the input field will not accept more than the allowed number of characters.
- The maxlength attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code

Log in now

Please Enter Your Name

Your Password is

Choices!

Favourite colour

my favourite fruit is papaya

I live in Flat House bedsit carvan

My upload file is: Browse... No file selected.

Leaving!

You can leave a message for me here:

Go start again

HTML 5 Input Types

- HTML5 added several new input types:
- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

```
<!DOCTYPE html>
<html>
    <head>
        <title>html 5 form</title>
    </head>
    <body>
        <form action = "success.html" method = "post">
          <label for = "firstname">first name: </label>
          <input type = "text" id = "firstname"><br />

          <label for = "lastname">last name: </label>
          <input type = "text" id = "lastname"><br />

          <label for = "email">email: </label>
          <input type = “email" id = "email"><br>

        <input type = "radio" name = “gen" value = "male">
        Male<br>
          <input type = "radio" name = “gen" value =
        "female"> Female<br>

          <label for = "datetime"> Date and Time :  </label>
         <input type = "datetime" name = "newinput" /><br>

          <label for = "datetime"> Local Date and Time : </label>
                <input type = "datetime-local" name = "dtlocal" /><br>

                Date : <input type = "date" name = "date" /><br>
                Month : <input type = "month" name = "month" /><br>
                Week : <input type = "week" name = "week" /><br>
                Time : <input type = "time" name = "time" /><br>
               Select Number : <input type = "number" min = "0" max
               = "10" step "1“ value = "5" name = "num" /><br>
                Select Range : <input type = "range" min = "0" max =
               "10" step "1" value = "5" name = "ran" /><br>
                Enter URL : <input type = "url" name = "url" /><br>
                <input type = "submit" value = "send"> <input type =
               "reset">
               </form>
        </body>
</html>
```

HTML 5 form Validation

- Before submitting data to the server, it is important to ensure all required form controls are filled out, in the correct format. This is called client-side form validation, and helps ensure data submitted matches the requirements set forth in the various form controls. This article leads you through basic concepts and examples of client-side form validation.
- Go to any popular site with a registration form, and you will notice that they provide feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

- "This field is required" (You can't leave this field blank).
- "Please enter your phone number in the format xxx-xxxx" (A specific data format is required for it to be considered valid).
- "Please enter a valid email address" (the data you entered is not in the right format).
- "Your password needs to be between 8 and 30 characters long and contain one uppercase letter, one symbol, and a number." (A very specific data format is required for your data).

- This is called form validation. When you enter data, the browser and/or the web server will check to see that the data is in the correct format and within the constraints set by the application. Validation done in the browser is called client-side validation, while validation done on the server is called server-side validation. In this chapter we are focusing on client-side validation.
- We want to make filling out web forms as easy as possible. So why do we insist on validating our forms? There are three main reasons:

- We want to get the right data, in the right format. Our applications won't work properly if our users' data is stored in the wrong format, is incorrect, or is omitted altogether.
- We want to protect our users' data. Forcing our users to enter secure passwords makes it easier to protect their account information.
- We want to protect ourselves. There are many ways that malicious users can misuse unprotected forms to damage the application.

Different types of client-side validation

- There are two different types of client-side validation that you'll encounter on the web:
- Built-in form validation uses HTML form validation features, which we've discussed in many places throughout this module. This validation generally doesn't require much JavaScript. Built-in form validation has better performance than JavaScript, but it is not as customizable as JavaScript validation.
- JavaScript validation is coded using JavaScript. This validation is completely customizable, but you need to create it all (or use a library).

Using built-in form validation

- One of the most significant features of modern form controls is the ability to validate most user data without relying on JavaScript. This is done by using validation attributes on form elements. We've seen many of these earlier in the course, but to recap:

- **required:** Specifies whether a form field needs to be filled in before the form can be submitted.
- **minlength and maxlength:** Specifies the minimum and maximum length of textual data (strings).
- **min and max:** Specifies the minimum and maximum values of numerical input types.
- **type:** Specifies whether the data needs to be a number, an email address, or some other specific preset type.
- **pattern:** Specifies a regular expression that defines a pattern the entered data needs to follow.

- If the data entered in a form field follows all of the rules specified by the above attributes, it is considered valid. If not, it is considered invalid.

The required attribute

- The simplest HTML validation feature is the required attribute. To make an input mandatory, add this attribute to the element. When this attribute is set, the element matches the :required and the form won't submit, displaying an error message on submission when the input is empty. While empty, the input will also be considered invalid, matching the :invalid
- <form>
- <label for="choose">Would you prefer a banana or cherry? (required)</label>
- <input id="choose" name="i-like" required />
- <button>Submit</button>
- </form>

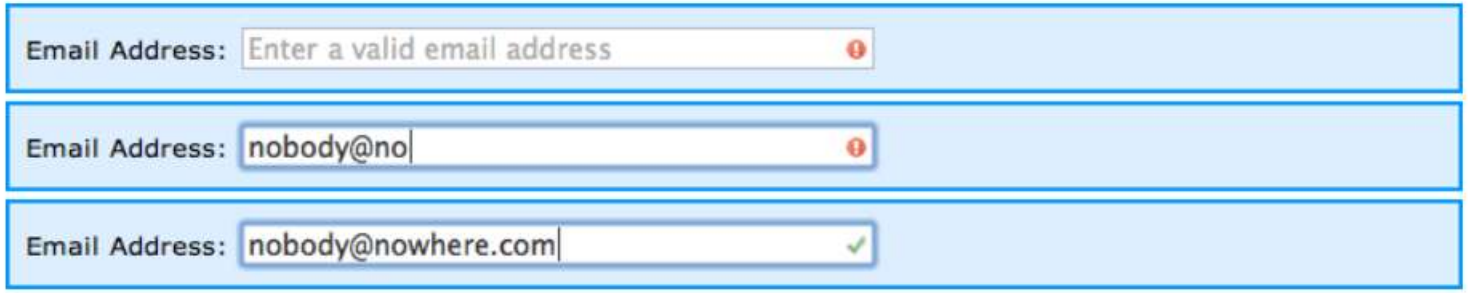
New text INPUT types

- This is where HTML5 really gets interesting and more useful. Along with the text input type, there are now a host of other options, including email, url, number, tel, date and many others.
- INPUT type="email"

- By changing the input type to email while also using the required attribute, the browser can be used to validate (in a limited fashion) email addresses:
- **Email Address: <input type="email" name="email" required placeholder="Enter a valid email address">**

- Note that for this example we've made use of another HTML5 attribute placeholder which lets us display a prompt or instructions inside the field - something that previously had to be implemented using messy onfocus and onblur JavaScript events.

- Again, different browsers implement this differently. In Opera it's sufficient to enter just *@* for the input to be accepted. In Safari, Chrome and Firefox you need to enter at least *@-.-.
- Obviously neither example is very limiting, but it will prevent people from entering completely wrong values, such as phone number, strings with multiple '@'s or spaces.



- **INPUT type="url"**
- In a similar fashion to the email input type above, this one is designed to accept only properly-formatted URLs. Of course it currently does nothing of the kind, but later you will see how to improve it's behavior using the pattern attribute.
- **Website: <input type="url" name="website" required>**

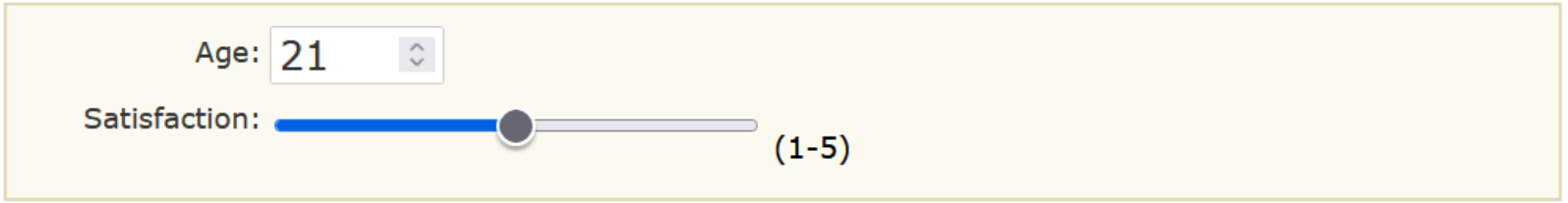
- As mentioned above, we can improve on this by making use of the pattern attribute which accepts a JavaScript regular expression. So the code above becomes:
- Website: <input type="url" name="website" required pattern="https?://.+">

- Now our input box will only accept text starting with http:// or https:// and at least one additional character:

INPUT type="number" and type="range"

- **INPUT type="number" and type="range"**
- The number and range input types also accept parameters for min, max and step. In most cases you can leave out step as it defaults to 1.
- Here you see an example including both a number input, typically displayed as a 'roller' and a range input displayed as a 'slider':
- **Age: <input type="number" size="6" name="age" min="18" max="99" value="21">
**
- **Satisfaction: <input type="range" size="2" name="satisfaction" min="1" max="5" value="3">**
- As with other HTML5 input types, browsers that don't recognise the new options will default to simple text inputs. For that reason it's a good idea to include a size for the input box.

INPUT type="password"



- **INPUT type="password"**
- We have a separate article with details on validating passwords using HTML5, including JavaScript code for customizing the browser generated alert messages.
- **Minlength and Maxlength:**
- You can constrain the character length of all text fields created by <input> or <textarea> by using the minlength and maxlength attributes. A field is invalid if it has a value and that value has fewer characters than the minlength value or more than the maxlength value.
- **<input type="text" name="i-like" required minlength="6" maxlength="6" />**

Validating against a regular expression

- Another useful validation feature is the pattern attribute, which expects a Regular Expression as its value. A regular expression (regexp) is a pattern that can be used to match character combinations in text strings, so regexps are ideal for form validation and serve a variety of other uses in JavaScript.

- Regexps are quite complex, and we don't intend to teach you them exhaustively in this article. Below are some examples to give you a basic idea of how they work.

- **a — Matches one character that is a (not b, not aa, and so on).**
- **abc — Matches a, followed by b, followed by c.**
- **ab?c — Matches a, optionally followed by a single b, followed by c. (ac or abc)**
- **ab*c — Matches a, optionally followed by any number of bs, followed by c. (ac, abc, abbbbbc, and so on).**
- **a|b — Matches one character that is a or b.**
- **abc|xyz — Matches exactly abc or exactly xyz (but not abcxyz or a or y, and so on).**

- Validating against a regular expression

- **URL input pattern:**
- <input type="url" pattern="https?://.+“>
- **IPv4 Address input pattern:**
- <nput type="text" pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}“>
- **Date input pattern (dd/mm/yyyy or mm/dd/yyyy):**
- <input type="text" pattern="\d{1,2}/\d{1,2}/\d{4}“>
- **Price input pattern:**
- <input type="text" pattern="\d+(\.\d{2})?“>
- **Latitude/Longitude input pattern:**
- <input type="text" pattern="-?\d{1,3}\.\d+“>

- To match a double-quote (") inside a pattern you can use the hex encoding \x22.

THANK YOU..!

Happy Learning...