**WWW and Web Development**

* WWW stands for World Wide Web and is commonly known as the Web.  WWW is defined as the collection of different websites around the world, containing different information shared via local servers (or computers). It is an information system on the internet which allows documents to be connected to other documents by hypertext links, enabling the user to search for information by moving from one document to another. The Internet refers to the entire global network of interconnected computers and devices. The World Wide Web specifically refers to information resources. These are accessed via the Internet using URLs and browsers.
* When building web pages, understanding the basic structure of an HTML document is the first step. The structure not only provides a framework for the document but also enables browsers to render content correctly. [HTML (Hypertext Markup Language)](https://www.geeksforgeeks.org/html-introduction/), structures web pages. Unlike programming languages, it isn’t compiled or interpreted. Browsers render HTML directly, displaying content without typical errors. It’s a markup language and not programming, making execution smooth without encountering compilation or interpretation issues.
* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
* Example: (<html>, <body>, <h1> and <p>)
* Front end development is an essential part of web development, and it refers to the creation of the visual and interactive elements of a website. HTML, CSS, and JavaScript are the three main technologies used in front end development, and they each play a unique role in creating a website that is attractive, functional, and user-friendly.
* There are two basic methods of web design: static and dynamic web pages. Users access static web pages, which present the same content every time they are viewed. On the other hand, dynamic webpages create content instantly in response to user input and present customized or updated information. Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc. In dynamic web pages, the Content of pages is different for different visitors. It takes more time to load than the static web page. [Dynamic web pages](https://www.geeksforgeeks.org/dynamic-websites/) are used where the information is changed frequently, for example, stock prices, weather information, etc.

**Web Applications**

* Web Application: Web application is a piece of software that can be accessed by the browser. A Browser is an application that is used to browse the internet. Web application needs authentication. The web application uses a combination of server-side scripts and client-side scripts to present information. It requires a server to manage requests from the users. Example: Google Apps, Amazon, YouTube

**Website:** Website is a collection of related web pages that contains images, text, audio, video, etc. It can consist of one page, two pages, and n number of pages. A website provides visual and text content that users can view and read. To view a website requires a browser (Chrome, Firefox). There are many types of websites like Archive website, Blog, Community website, etc.

* The main types of web applications include Content Management Systems (CMS), Progressive Web Applications (PWA), and Single-Page Applications (SPA). A CMS allows users to create, edit, and manage digital content without coding.  A PWA combines website and mobile app features, offering offline access and fast loading. An SPA loads a single page and updates content dynamically without refreshing, improving user experience.
* Web applications have almost replaced the old desktop applications. They are more convenient to use, they are easy to update, and they can be used in multiple devices. In today’s time, there are two types of web applications we can create: Single Page Application (SPA) and Multi Page Application (MPA). SPAs are considered more modern than MPAs. SPAs are all about serving an outstanding User Experience with no page reloading and no extra time waiting. A single page application works inside a browser and it is just one web page that you visit which then loads all other content using JavaScript. The most common SPAs are: Gmail, Google Maps, Facebook, GitHub and many more.

**Web Client and Web Server**

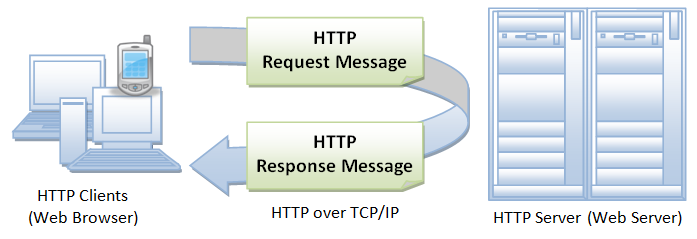
* A web client is a software application that accesses and displays content from the World Wide Web, typically through a web browser. It acts as an intermediary between users and web servers, enabling users to interact with websites, view multimedia content, and access various online services. At its core, a web client is responsible for fetching, interpreting, and rendering web content, enabling users to view web pages, interact with web applications, and consume multimedia content seamlessly.
* A web server is a software or hardware system that stores, processes, and delivers web content to users when requested. It acts as a mediator between your web browser and the website’s files, facilitating the transfer of data across the internet.
* While client-side languages, such as HTML, CSS, and JavaScript, determine how a website looks and behaves in a browser, server-side languages determine how a website operates and interacts with a server.

**Client-Server Communication**

* Client (You) makes a request and the server responds, that happens every single time you enter a link and press enter on your browser. Many protocols are used like DNS, FTP, HTTP, SMTP (simple mail transfer protocol), SSL (Secure sockets layer).
* The modern web is a complex mesh of interconnected services, with RESTful APIs acting as the bridges that allow different software systems to communicate with each other. RESTful APIs provide a framework for the interactions between client-side and server-side applications, offering a set of guidelines that facilitate this complex communication in a standardized way. REST stands for Representational State Transfer. It’s not a protocol or a set of standards, but rather an architectural style for networked applications. REST is resource-focused, leveraging the precepts of the web’s existing infrastructure, primarily HTTP.
* A RESTful API relies on stateless client-server communication, connecting the web-based client and server through HTTP protocol. When clients want to perform an action or retrieve data, they make a request to the server. The server then processes the request, performs the necessary action, and sends back a response. This process typically involves the following HTTP methods, which correspond to create, read, update, and delete (CRUD) operations:
  + **GET:** Requests a representation of the specified resource. Requests using GET should only retrieve data and have no other effect.
  + **POST:**Submits an entity to the specified resource, often causing a change in state or side effects on the server.
  + **PUT:**Replaces all current representations of the target resource with the request payload.
  + **DELETE:** Removes the specified resource.
  + **PATCH:**Partially updates a resource.
* In the context of [web development](https://www.baeldung.com/cs/front-end-vs-back-end-development), a session refers to a way of maintaining state information about a user’s interactions with a website or [web application](https://en.wikipedia.org/wiki/Web_application)**.** When a user visits a website, the server can create a session for that user. Additionally, a session allows the server to keep track of information such as the user’s login status, preferences, and any data entered into forms. The server typically initiates a session when a user logs in to a website. Furthermore, we can identify a session by a unique session ID**.** Generally, we pass the session IDs as a parameter in URLs or store them in the [cookies](https://www.baeldung.com/cs/browser-cookie-domains). The session ID allows the server to associate the user’s requests with their specific session. Additionally, it also helps to retrieve and update the session data as needed.

**HTTP and HTTP Methods**

* HTTP (Hypertext Transfer Protocol) is a fundamental protocol of the Internet, enabling the transfer of data between a client and a server. It is the foundation of data communication for the World Wide Web. HTTP provides a standard between a web browser and a web server to establish communication. HTTP provides a standard between a web browser and a web server to establish communication. It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web. Whenever a web user opens their web browser, the user indirectly uses HTTP. It is an application protocol that is used for distributed, collaborative, hypermedia information systems.
* **HTTP (Hypertext Transfer Protocol)** specifies a collection of request methods to specify what action is to be performed on a particular resource. The most commonly used HTTP request methods are **GET, POST, PUT, PATCH, and DELETE**. These are equivalent to the **CRUD operations (create, read, update, and delete)**.
* **GET:**GET request is used to read/retrieve data from a web server. GET returns an HTTP status code of **200 (OK)** if the data is successfully retrieved from the server.
* **POST:**POST request is used to send data (file, form data, etc.) to the server. On successful creation, it returns an HTTP status code of **201**.
* **PUT:**A PUT request is used to modify the data on the server. It replaces the entire content at a particular location with data that is passed in the body payload. If there are no resources that match the request, it will generate one.
* **PATCH:**PATCH is similar to PUT request, but the only difference is, it modifies a part of the data. It will only replace the content that you want to update.
* **DELETE:**ADELETE request is used to delete the data on the server at a specified location.
* A typical representation of HTTP request-response cycle:



**HTTPS vs HTTP**

* HTTP stands for Hypertext Transfer Protocol. Hypertext is the type of text that is specially coded with the help of some standard coding language called [Hypertext Markup Language (HTML)](https://www.geeksforgeeks.org/html-introduction/). HTTP provides a standard between a web browser and a web server to establish communication. It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web. Whenever a web user opens their web browser, the user indirectly uses [HTTP](https://www.geeksforgeeks.org/http-full-form/). It is an application protocol that is used for distributed, collaborative, hypermedia information systems.
* [HTTPS](https://www.geeksforgeeks.org/https-full-form/) stands for Hyper Text Transfer Protocol Secure. HTTP Secure (HTTPS), could be a combination of the Hypertext Transfer Protocol with the SSL/[TLS](https://www.geeksforgeeks.org/transport-layer-security-tls/) convention to supply encrypted communication and secure distinguishing proof of an arranged web server. HTTPS is more secure than HTTP because HTTPS is certified by the [SSL (Secure Socket Layer)](https://www.geeksforgeeks.org/secure-socket-layer-ssl/). Whatever website you are visiting on the internet, if its URL is HTTP, then that website is not secure.
* HTTPS encrypts all message substance, including the HTTP headers and the request/response data. The verification perspective of HTTPS requires a trusted third party to sign server-side [digital certificates](https://www.geeksforgeeks.org/what-are-digital-certificates/).
* HTTPS is presently utilized more frequently by web clients than the first non-secure HTTP, fundamentally to ensure page genuineness on all sorts of websites, secure accounts and to keep client communications.

**HTTP Status Codes**

* An HTTP status code is a server response to a browser’s request. When you visit a website, your browser sends a request to the site’s server, and the server then responds to the browser’s request with a three-digit code: the HTTP status code, which indicates that the server is communicating the status of the request. These status codes are the Internet equivalent of a conversation between your browser and the server. They communicate whether things between the two are A-okay, touch-and-go, or whether something is wrong. Understanding status codes and how to use them will help you to diagnose site errors quickly to minimize downtime on your site.
* Following is some HTTP status codes and its meanings:

**200 (success/OK):** The HTTP status code 200 represents success which means the page you have requested has been fetched. The action made has been accepted and has been delivered to the client by delivering the requested page.

GET: entity in reference to the requested source sent to the response

POST: entity describing the response of action made

HEAD: an entity-header field similar to the requested source

TRACE: a request made by the client is taken care of by the server

**301 (Permanent Redirect):** The HTTP status code 301 means that the page you have requested has moved to a new URL and which is permanent. In the future, whenever the user requests the same website, it will be redirected to the new URL. The modified permanent URL is given by the location filed in response.

**404 (Not Found):** 404 HTTP Status code appears when you request a URL and then the server has not found anything. This happens when the server doesn’t wish to describe the reason why the request has been refused. Also, the server is not able to find a representation for the target resource.

**500 (Internal Server Error):** 500 HTTP status code means requesting a URL is not fulfilled because the server encounters an unexpected condition. It gives information about the request made if it is successful, and throws an error. When there’s an error during a connection to the server, and the requested page cannot be accessed then this message is displayed.