UNIT 1: HTML, CSS & Client Side Scripting

World Wide Web (WWW) also called Web, the Internet's (worldwide computer network) leading information retrieval site. The Internet provides users with access to a vast array of documents that are linked to each other by hypertext or hypermedia links — i.e. hyperlinks, electronic connections that link similar pieces of information to allow an user to access them. Hypermedia files function links to images, sounds, animations, and movies. The Web operates inside the Internet's primary client-server format; servers are pc programs that save and transmit files to different computer systems on the network when asked to, whilst consumers are packages that request files from a server as the person asks for them. Browser software program permits users to view the retrieved documents.

- In Hypertext Mark-up Language (HTML), a hypertext document with its corresponding text and hyperlinks is written and assigned an online address called an Uniform Resource Locator (URL).
- Tim Berners-Lee and his colleagues at CERN, an international research organization, headquartered in Geneva, Switzerland, started creating the World Wide Web in 1989. They developed the Hypertext Transfer Protocol (HTTP) protocol which standardized server-client communication.
- In January 1992, their text-based Web browser was made available for publication.
- The World Wide Web rapidly gained recognition with the launch of a web browser named Mosaic, created in the United States by Marc Andreessen and others at the University of Illinois National Centre for Supercomputing Applications, and launched in September 1993.
- Mosaic allowed people the usage of the Web to use the identical type of "point-and-click" graphical manipulations that had been accessible in private computers for some years.
- In April 1994 Andreessen cofounded Netscape Communications Corporation, whose Netscape Navigator grew to become the dominant Web browser quickly after its launch in December 1994.
- The InternetWorks of BookLink Technologies, its first tabs browser in which a user would be able to visit another website without opening a completely new window, debuted the same year. The World Wide Web was having millions of active users by the mid-1990s.

Microsoft Corporation, the software giant, was interested in promoting internet applications on personal computers and in 1995 created its own web browser (based initially on Mosaic), Internet Explorer (IE), as an add-on to the Windows 95 OS. IE was incorporated into the Windows operating system in 1996 which decreased competition from other Internet browser manufacturers including Netscape. IE gradually became the default Web browser.

Apple's Safari was released on Macintosh personal computers as the default browser in 2003, and later on iPhones (2007) and iPads (2010). Safari 2.0 (2005) was the first privacy-mode browser, Private Browsing, where the client does not save Web sites in its history, save files in its cache, or enter personal information on Web pages.

The first major challenger to IE's dominance was Mozilla's Firefox, released in 2004 and designed to address problems surrounding IE with speed and security. Google launched Chrome in 2008, the first browser featuring isolated tabs, which meant that when one tab crashed, other tabs and the entire browser would still work. In 2013 Chrome had become the

dominant browser, popularly outperforming IE and Firefox. In 2015, Microsoft removed IE, replacing it with Edge.
Smartphone's became more computer-like in the early 21st century, and more modern services, such as Internet access, became possible. Web use on Smartphone's has gradually increased, and it accounted for over half of Internet browsing in 2016.

Internet vs. World Wide Web

The Internet is a comprehensive computer network and was conceptualized by the ARPA or Advanced Research Projects Agency during 1969. The World Wide Web is much newer than the Internet, and was introduced during the 1990s.
The World Wide Web is a series of web pages that follow the http protocol, accessible from any part of the world through the Internet. The http protocol is a type of language used on the Internet to transmit data and to communicate.
It is an application that is used on the Internet and all pages that are part of the World Wide Web start with http:/www, with www being a World Wide Web abbreviation. The World Wide Web is a framework for the knowledge exchange.
It represents a way of accessing information through the Internet. Understanding the differences between the Internet and the World Wide Web is key to understanding the true workings of search engines. Search engines search websites that are accessible on the World Wide Web and not other internet- based sites.
As Web 2.0 web applications seek to brand their domain names and make them easily pronounceable, the use of the www prefix is declining. Given the growing popularity of the mobile web, services such as Gmail.com, MySpace.com, Facebook.com and Twitter.com are most frequently mentioned without adding "www." (Or indeed ".com") to the domain.

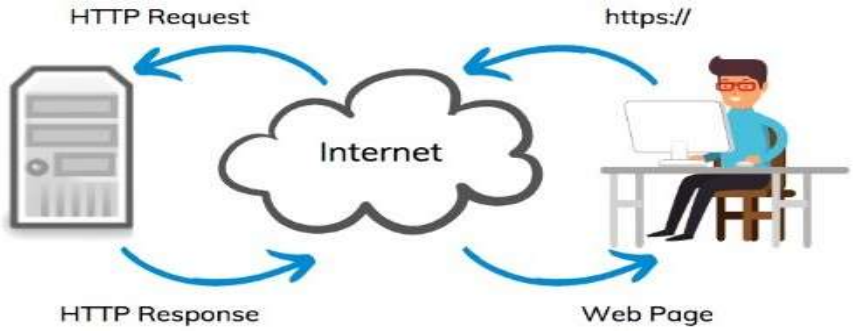
What Is the Web Made of?

The Web consists of:

- Your personal computer
- Web browser software to access the Web
- A connection to an Internet service provider (ISP)
- Servers to host the data
- Routers and switches to direct the flow of data

How the Web Works

Web pages are stored on web servers located around the globe. Entering the Uniform Resource Locator or URL of a web page in your web browser or clicking a link sends a request to the server that hosts the page. The server transmits the web page data to your computer and your web browser displays it on your screen.
A web page is an electronic document written in a computer language called HTML (Hypertext Mark-up Language).Web pages can contain text, graphics, audio, video, and animation, as well as interactive features, such as data entry forms and games. Each page has a unique address known as a URL (Uniform Resource Locator), which identifies its location on the server. Web pages usually contain hyperlinks to other web pages. Hyperlinks are text and images that reference the addresses of other web pages.



A website consists of one or more web pages that relate to a common theme, such as a person, business, organization, or a subject, such as news or sports. The first page is called the home page, which acts like an index, indicating the content on the site. From the home page, you can click links to access other pages on the site or other resources on the Web.

Basic components of the Web

The basic components of the Web are:

- *Web servers*, which are computers that carry distribution information over the Internet. For the example, one Web server may contain the online magazine what's On in Bath's text and images, and another server may contain details on which seats are available for a specific concert. The magazine will use the Web's own publishing language, HTML (Hypertext Mark-up Language), to format. The data on the seating applicable and their prices will be kept in a database with links to different forms published using HTML.
- *Servers* that can also be PCs, Macintosh systems or workstations from UNIX: it is the server software that makes them different, not the machine itself. The servers must be relatively up-to-market devices. Servers must always be left running, so that people can access information about them whenever they prefer. Another significant thing about servers: they are comparatively hard to set up. If you're a non- technical person who wants to be published on the Internet, it's best to rent some room on someone else's site.
- *Web clients* that can be PCs, Macintoshes and other internet enabled devices that can access information from Web servers. The machine at your desk is a Web client. Server applications can be run by PCs, Macintoshes, UNIX workstations and even single terminals. Different lient software for various platforms is marketed .Thus Mosaic has both an implementation of Macintosh and a PC.
- *HTTP protocol* used for transmission of files between servers and clients. When you click on a hypertext link or fill out a form in a Web document, the results must be sent as quickly as possible over the Internet and then understood by a server at the other end. Instructions like 'give me this file' or 'get me the picture' are carried by the HTTP web communication protocol. This protocol is the 'messenger' that gathers files from and to servers, and then delivers results to your computer whenever you click on a button. Among other Internet services, HTTP has its counterparts: FTP, file transfer protocol, and Gopher are protocols which obtain different kinds of information from across the Internet.
- *Browser* software which a Web client requires to view text, pictures, video clips, etc. This is given under the generic name 'browser,' which is perhaps the best-known example of Internet Explorer by Mosaic, Microsoft Corp. and Netscape

Communications Corp.'s Navigator and Communicator browsers. It gives the software ability to search Web server collected information, as you would browse through a book. It also provides you with facilities to save and print information accessed on the Internet.

Navigating the Web
There are three main ways to move between web pages or websites:

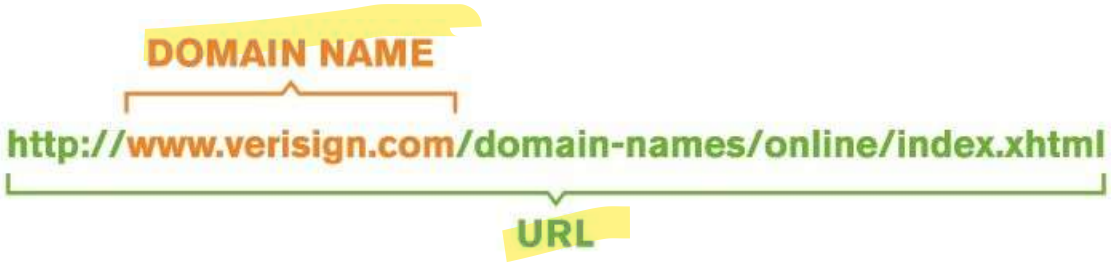
1. Clicking a text link.
2. Clicking a hyperlinked graphic, such as a button, photograph, or drawing.
3. Typing the URL of a web page in the location box (also known as the address field) of your web browser and then pressing the Enter or Return key.

Scheme specifiers

At the start of a web URI, the scheme specifiers http:/ and https:/ refer to the Hypertext Transfer Protocol or HTTP Stable, respectively. They specify the protocol of communication to be used for request and reply. The HTTP protocol is fundamental to the operation of the World Wide Web and when browsers send or retrieve sensitive data, such as passwords or banking information, the added encryption layer of HTTPS is vital. Web browsers normally prepend http:/ to user-entered URIs automatically, if omitted.

What is a URL?

URL is the abbreviation for Uniform Resource Locator and is known on the World Wide Web as the global address of documents and other resources. For instance, you'll go to the URL www.google.com to visit a website.
A domain name is part of a URL, which stands for Uniform Resource Locator.



Computers rely on a language consisting of numbers and letters called an IP address, so that computer networks and servers can "speak to each other." Each computer connecting to the Internet has a unique IP address, which looks like this

22.231.113.64 or 3ffe:1900:4545:3:200:f8ff:fe21:67cf

Typing in a long IP address is not ideal, or practical, for an online user to navigate quickly across the Network. That's why domain names have been developed-to cover IP addresses with something more memorable. The domain name may be called a "nickname" to the IP address. A URL, along with other basic information, includes the domain name to create a complete address (or "web address") to guide a visitor to a particular online website called a webpage. Essentially it's a series of directions and every web page has a special one.

Domain Name System

DNS stands for Domain Name System. DNS 'principal role is to convert domain names into IP addresses that computers can recognize. This also lists mail servers that accept Emails for each domain name. Internet machine is assigned a unique address, which is called an IP address. So looks a standard IP address: **199.123.456.7**
It's very hard to keep in mind the IP addresses of all the websites that we visit daily. Words are easier to remember than numeral strings. It's here that domain names get into the picture. What you need to know when you visit a website is their URL.
Computers remember numbers, and DNS helps to translate the URL to an IP address that the machine can use. If you type domain.com into the user, the user wants to get the www.domain.com IP address first. The user contacts a DNS server to ask where the web pages are located. It acts as an IP address directory service.

DNS Servers and IP Addresses

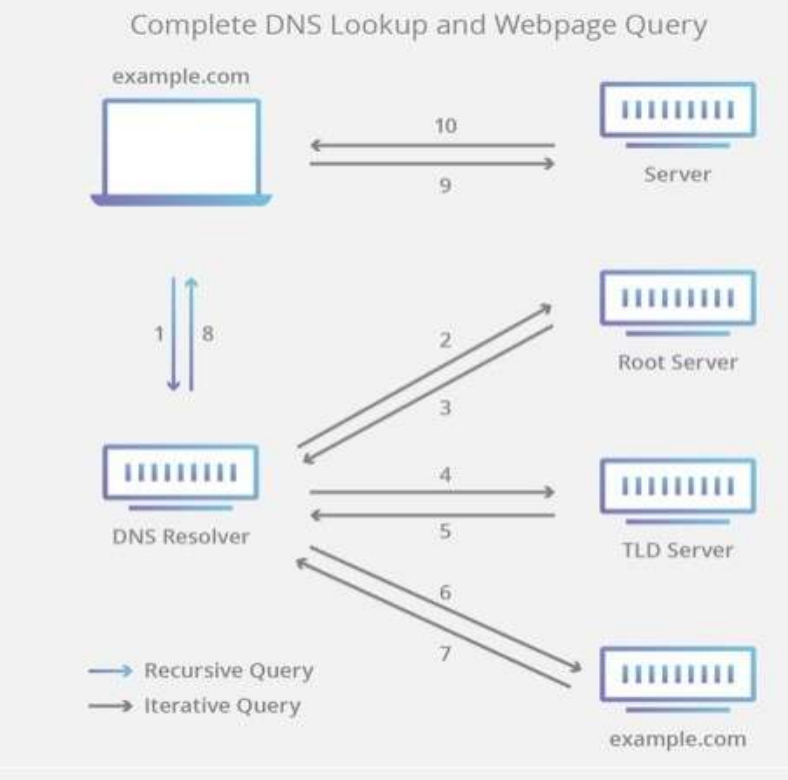
There are billions of IP addresses currently in use, and most machines have a human-readable name as well.DNS servers (cumulatively) are processing billions of requests across the internet at any given time. Millions of people are adding and changing domain names and IP addresses each day.
With so much to handle, DNS servers rely on network efficiency and internet protocols. Part of the IP's effectiveness is that each machine on a network has a unique IP address in both the IPV4 and IPV6 standards managed by the Internet Assigned Numbers Authority (IANA).
Here are some ways to recognize an IP address:

- An IP address in the IPV4 standard has four numbers separated by three decimals, as in: 70.74.251.42
- An IP address in the IPV6 standard has eight hexadecimal numbers (base-16) separated by colons, as in 2001:0cb8:85a3:0000:0000:8a2e:0370:7334. Because IPV6 is still a very new standard, we'll concentrate on the more common IPV4 for this article.
- Each number in an IPV4 number is called an "octet" because it's a base-10 equivalent of an 8-digit base-2 (binary) number used in routing network traffic. For example, the octet written as 42 stands for 00101010. Each digit in the binary number is the placeholder for a certain power of two from 2 to 27, reading from right to left. That means that in 00101010, you have one each of 21, 23 and 25. So, to get the base-10 equivalent, just add 21 + 23 + 25 = 2 + 8 + 32 = 42.
- There are only 256 possibilities for the value of each octet: the numbers 0 through 255.
- Certain addresses and ranges are designated by the IANA as reserved IP addresses, which mean they have a specific job in IP. For example, the IP address 127.0.0.1 is reserved to identify the computer you're currently using.

What are the steps in a DNS lookup?
DNS is concerned with a domain name being translated into the appropriate IP address. To learn how this process works, it helps to follow the path of a DNS lookup as it travels from a web browser, through the DNS lookup process, and back again. Let's take a look at the steps.
The 8 steps in a DNS lookup:

1. A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.

2. The resolver then queries a DNS root nameserver (.).
3. The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
4. The resolver then makes a request to the .com TLD.
5. The TLD server then responds with the IP address of the domain's name server, example.com.
6. Lastly, the recursive resolver sends a query to the domain's nameserver.
7. The IP address for example.com is then returned to the resolver from the nameserver.
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.
9. Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:
10. The browser makes a HTTP request to the IP address.
11. The server at that IP returns the webpage to be rendered in the browser (step 10).



Top Level Domain (TLD)

TLD refers to the last part of a domain name. For example, the .com in amazon.com is the Top Level Domain. The most common TLDs include .com,
.net, org, and .info. Country code TLDs represents specific geographic locations. For example: .in represents India. Here are some more examples:

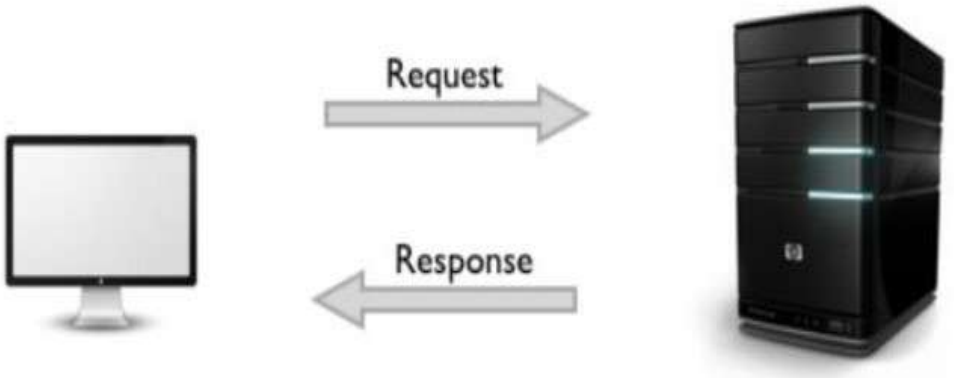
- COM — commercial websites, though open to everyone
- NET — network websites, though open to everyone
- ORG — non-profit organization websites, though open to everyone
- EDU — restricted to schools and educational organizations
- MIL — restricted to the U.S. military
- GOV — restricted to the U.S. government
- US, UK, RU and other two-letter country codes — each is assigned to a domain name authority in the respective country

In a domain name, each word and dot combination you add before a top-level domain indicates a level in the domain structure. Each level refers to a server or a group of servers that manage that domain level. An organization may have a hierarchy of sub-domains further organizing its internet presence, like "bbc.co.uk" which is the BBC's domain under CO, an additional level created by the domain name authority responsible for the U.K. country code.
The left-most word in the domain name, such as www or mail, is a host name. It specifies the name of a specific machine (with a specific IP address) in a domain, typically dedicated to a specific purpose. A given domain can potentially contain millions of host names as long as they're all unique to that domain. (The "http" part stands for Hypertext Transfer Protocol and is the protocol by which information is sent by the user to the website she is visiting. Nowadays, you're more likely to see "https" which is a sign the information is being sent by secure protocol where the information is encrypted.
A subdomain is a subdivision of a domain name, allowing you to put content in your URL before your namespace. For example, blog.companyname.com or shop.companyname.com would be a subdomain of the domain name companyname.com. For example: If a customer buys a domain with 123 Reg,
e.g. yourdomain.co.uk, they can set up subdomains, e.g. site1.yourdomain.co.uk or secure.yourdomain.co.uk. This is an excellent way of breaking up the website if you have different regions, products or even languages.

Protocol

Protocol is the standard means, rules and conventions which are accepted. It is the beginning and the core of the web. HTTP is the Hyper Text Transfer Protocol; it is a Web-based protocol. It is rules for providing to the web user,
and it is a standard delivered to the clients from all common web servers. When the web browser wants a user requesting documents, the server changes the document and the web browser converts this document to the appropriate type again. And it's getting sent to the user. It therefore serves to transmit files which contain web pages to users based on web protocols.
In other words, when communicate with Englishman, he can't recognize our phrases if talk Korean. Likewise, human beings made standards that can speak with all of webs. We called these standards a protocol, and when communicate on the web, it can communicate to suit internet protocols.

How Protocol Works?
The User uses his computer and connects Web server (Google, Yahoo, etc.) through Internet. Consider an example where user trying to access Google.



1. Enter http://www.google.com in the address bar using a web browser.

2. Web browser request information to the Google web server by the HTTP protocols.
3. Web server receives requests, and it sends the answer to the computer.

4. Web browser received the HTTP protocol information represented by texts and pictures.

What is HTTP (Hypertext Transfer Protocol)?
The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems that allows users to communicate data on the World Wide Web.
What is the purpose of HTTP?
HTTP was invented alongside HTML to create the first interactive, text-based web browser: the original World Wide Web. Today, the protocol remains one of the primary means of using the Internet.
How does HTTP work?
As a request-response protocol, HTTP gives users a way to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers. HTTP clients generally use Transmission Control Protocol (TCP) connections to communicate with servers.
HTTP utilizes specific request methods in order to perform various tasks:

- GET requests a specific resource in its entirety
- HEAD requests a specific resource without the body content
- POST adds content, messages, or data to a new page under an existing web resource
- PUT directly modifies an existing web resource or creates a new URI if need be
- DELETE gets rid of a specified resource
- TRACE shows users any changes or additions made to a web resource
- OPTIONS shows users which HTTP methods are available for a specific URL
- CONNECT converts the request connection to a transparent TCP/IP tunnel
- PATCH partially modifies a web resource

All HTTP servers use the GET and HEAD methods, but not all support the rest of these request methods.

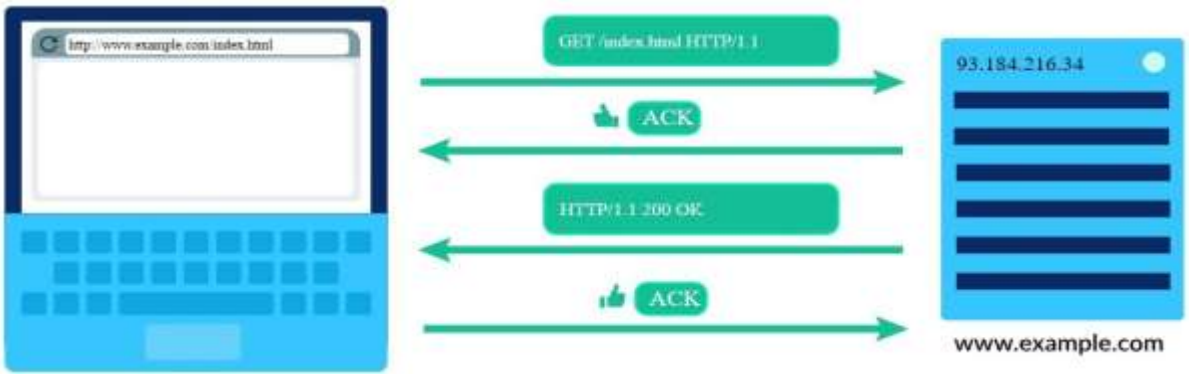
Basic aspects of HTTP

- HTTP is generally designed to be simple and human readable, even with the added complexity introduced in HTTP/2 by encapsulating HTTP messages into frames and reduced complexity for newcomers.
- Introduced in HTTP/1.0, HTTP headers make this protocol easy to extend and experiment with. New functionality can even be introduced by a simple agreement between a client and a server about a new header's semantics.
- HTTP is stateless: there is no link between two requests being successively carried out on the same connection. This immediately has the prospect of being problematic for users attempting to interact with certain pages coherently, for example, using e-commerce shopping baskets. But while the core of HTTP itself is stateless, HTTP cookies allow the use of stateful sessions. Using header extensibility, HTTP Cookies are added to the workflow, allowing session creation on each HTTP request to share the same context, or the same state.
- HTTP and connection is controlled at the transport layer, and therefore fundamentally out of scope for HTTP. Though HTTP doesn't require the underlying transport protocol to be connection-based; only requiring it to be reliable, or not lose messages (so at minimum presenting an error). Among the two most common transport protocols on the Internet, TCP is reliable and UDP isn't. HTTP therefore relies on the TCP standard, which is connection-based.
- Before a client and server can exchange an HTTP request/response pair, they must establish a TCP connection, a process which requires several round-trips. The default behaviour of HTTP/1.0 is to open a separate TCP connection for each HTTP request/response pair. This is less efficient than sharing a single TCP connection when multiple requests are sent in close succession.

HTTP and TCP/IP

HTTP is a protocol that's built on top of the TCP/IP protocols.

Each HTTP request is inside an IP packet, and each HTTP response is inside another IP packet--or more typically, multiple packets, since the response data can be quite large.



- Diagram with laptop on left and server on right. Laptop has browser window with

"http://www.example.com/index.html" in address bar. Server is labelled with "www.example.com" and its IP address "93.184.216.34". 4 arrows are shown:

- First arrow goes from laptop to server and displays packet with HTTP request inside.
- Second arrow goes from server to laptop and displays packet with "ACK" inside.
- Third arrow goes from server to laptop and displays packet with HTTP response inside.
- Fourth arrow goes from laptop to server and displays packet with "ACK" inside.

There are many other protocols built on top of TCP/IP, like protocols for sending email (SMTP, POP, and IMAP) and uploading files (FTP).
All of these protocols enable us to use the Internet to connect with other computers in useful ways, and to communicate and collaborate across wide distances.

HTTPS

HTTPS stands for Hypertext Transfer Protocol over Secure Socket Layer. Think of it as a secure version of HTTP. HTTPS is used primarily on web pages that ask you to provide personal or sensitive information (such as a password or your credit card details).
When you browse a web page using HTTPS, you are using SSL (Secure Sockets Layer). For a website to use HTTPS it needs to have an SSL certificate installed on the server. These are usually issued by a trusted 3rd party, referred to as a Certificate Authority (CA).
When you browse a web page using HTTPS, you can check the details of the SSL certificate. For example, you could check the validity of it. You could also check that the website does actually belong to the organization you think it does. You can usually do this by double clicking on the browser's padlock icon. The padlock icon only appears when you view a secure site.

Component of HTTP are

1. Transfer time
2. Computer IP
3. Web serverIP
4. Method (get/post)

5. HTTP protocol version (1.0/1.1)

6. File format (flash, file data, etc.)

7. Reference (Previous web page address)

8. Language (Language type)

9. Encoding (Encoding type of English)

10. Information of web browser (IE/Firefox/Chrome etc.)

11. Cookies (Cookie values stored on my computer)

12. Real transfer content (id = iboss/ password=1234, etc.)

What does a typical HTTP request look like?

An HTTP request is just a series of lines of text that follow the HTTP protocol. A GET request might look like this:

```
GET /hello.txt HTTP/1.1
User-Agent: curl/7.63.0 libcurl/7.63.0 OpenSSL/1.1.1 zlib/1.2.11
Host: www.example.com
Accept-Language: en
```

This section of text, generated by the user's browser, gets sent across the Internet. The problem is, it's sent just like this, in plaintext that anyone monitoring the connection can read. (Those who are unfamiliar with the HTTP protocol may find this text hard to understand, but anyone with baseline knowledge of the protocol's commands and syntax can read it easily.)

This is especially an issue when users submit sensitive data via a website or a web application. This could be a password, a credit card number, or any other data entered into a form, and in HTTP all this data is sent in plaintext for anyone to read. (When a user submits a form, the browser translates this into an HTTP POST request instead of an HTTP GET request.)

When an origin server receives an HTTP request, it sends an HTTP response, which is similar:

```
HTTP/1.1 200 OK
Date: Wed, 30 Jan 2019 12:14:39 GMT
Server: Apache
Last-Modified: Mon, 28 Jan 2019 11:17:01 GMT
Accept-Ranges: bytes
Content-Length: 12
Vary: Accept-Encoding
Content-Type: text/plain

Hello World!
```

If a website uses HTTP instead of HTTPS, all requests and responses can be read by anyone who is monitoring the session. Essentially, a malicious actor can just read the text in the request or the response and know exactly what information someone is asking for, sending, or receiving.
When consider with HTTPS, HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses, so in the example above, instead of the text, an attacker would see a bunch of seemingly random characters.
Instead of:

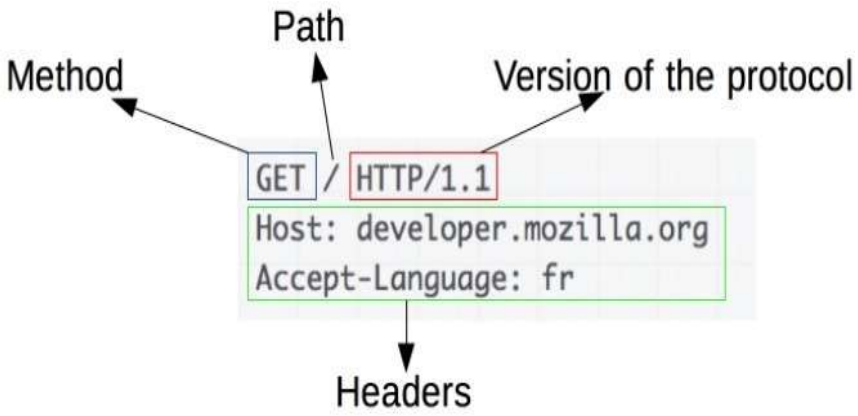
```
GET /hello.txt HTTP/1.1
User-Agent: curl/7.63.0 libcurl/7.63.0 OpenSSL/1.1.1 zlib/1.2.11
Host: www.example.com
Accept-Language: en
```

The attacker sees something like:

```
t8Fw6T8UV81pQfyhDkhebbz7+oiwldr1j2gHBB3L3RFTRsQCpaSnSBZ78Vme+DpDVJPvZdZUZHpzbbcqmSW1+3xXGsERHg9Y
```

HTTP Messages
HTTP messages, as defined in HTTP/1.1 and earlier, are human-readable. In HTTP/2, these messages are embedded into a binary structure, a frame, allowing optimizations like compression of headers and multiplexing. Even if only part of the original HTTP message is sent in this version of HTTP, the semantics of each message is unchanged and the client reconstitutes (virtually) the original HTTP/1.1 request. It is therefore useful to comprehend HTTP/2 messages in the HTTP/1.1 format.
There are two types of HTTP messages, requests and responses, each with its own format.

Requests
An example HTTP request:



Requests consist of the following elements:

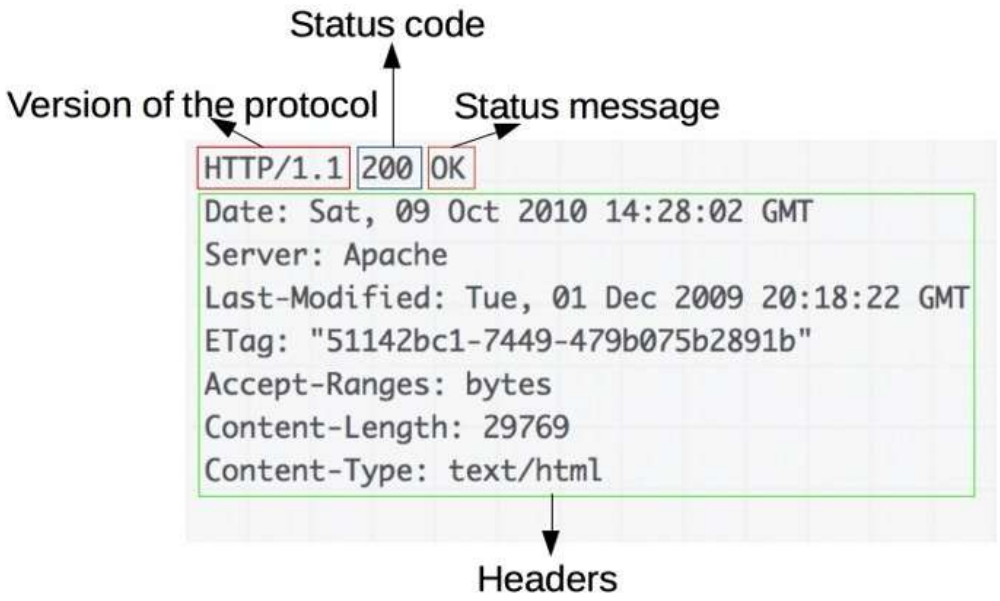
- An HTTP method, usually a verb like GET, POST or a noun like OPTIONS or HEAD that defines the operation the client wants to perform. Typically, a client wants to

fetch a resource (using GET) or post the value of an HTML form (using POST), though more operations may be needed in other cases.

- The path of the resource to fetch; the URL of the resource stripped from elements that are obvious from the context, for example without the protocol (http://), the domain (here, google.com), or the TCP port (here, 80).
- The version of the HTTP protocol.
- Optional headers that convey additional information for the servers.
- Or a body, for some methods like POST, similar to those in responses, which contain the resource sent.

Responses

An example response:



Responses consist of the following elements:

- The version of the HTTP protocol they follow.
- A status code, indicating if the request was successful, or not, and why.
- A status message, a non-authoritative short description of the status code.
- HTTP headers like those for requests.
- Optionally, a body containing the fetched resource.

HTTP can use both nonpersistent connections and persistent connections. A nonpersistent connection is the one that is closed after the server sends the requested object to the client. In other words, the connection is used exactly for one request and one response.

With persistent connections, the server leaves the TCP connection open after sending responses and hence the subsequent requests and responses between the same client and server can be sent. The server closes the connection only when it is not used for a certain configurable amount of time. With persistent connections, the performance is improved by 20%.

A persistent connection takes 2 RTT for the connection and then transfers as many objects, as wanted, over this single connection.

Nonpersistent connections are the default mode for HTTP/1.0 and persistent connections are the default mode for HTTP/1.1.
The non-persistent connection takes the connection time of 2RTT + file transmission time. It takes the first RTT (round-trip time) to establish the connection between the server and the client. The second RTT is taken to request and return the object. This case stands for a single object transmission.

HTTP Status Codes

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

1. Informational responses (100–199),
2. Successful responses (200–299),
3. . Redirects (300–399),
4. Client errors (400–499),
5. And Server errors (500–599).

1. Information responses

100 Continue : This interim response indicates that everything so far is OK and that the client should continue the request, or ignore the response if the request is already finished.

2. Successful responses

200 OK: The request has succeeded. The meaning of the success depends on the HTTP method:

- GET: The resource has been fetched and is transmitted in the message body.
- HEAD: The entity headers are in the message body.
- PUT or POST: The resource describing the result of the action is transmitted in the message body.
- TRACE: The message body contains the request message as received by the serve

3. Redirection messages

301 Moved Permanently: The URL of the requested resource has been changed permanently. The new URL is given in the response.

4. Client error responses

400 Bad Request: The server could not understand the request due to invalid syntax.

5. Server error responses

500 Internal Server Error: The server has encountered a situation it doesn't know how to handle.