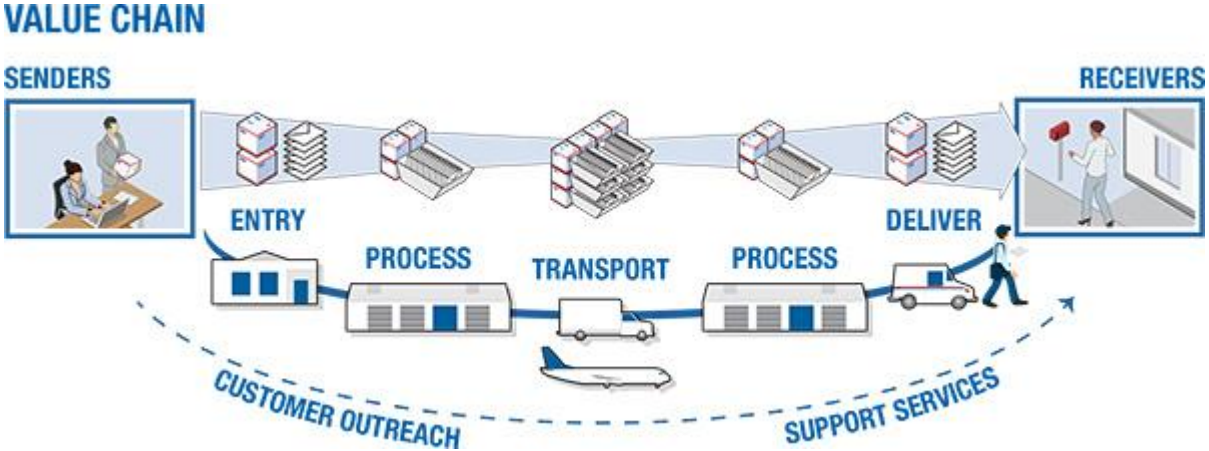
Internet and WWW

SF Coding Bootcamp

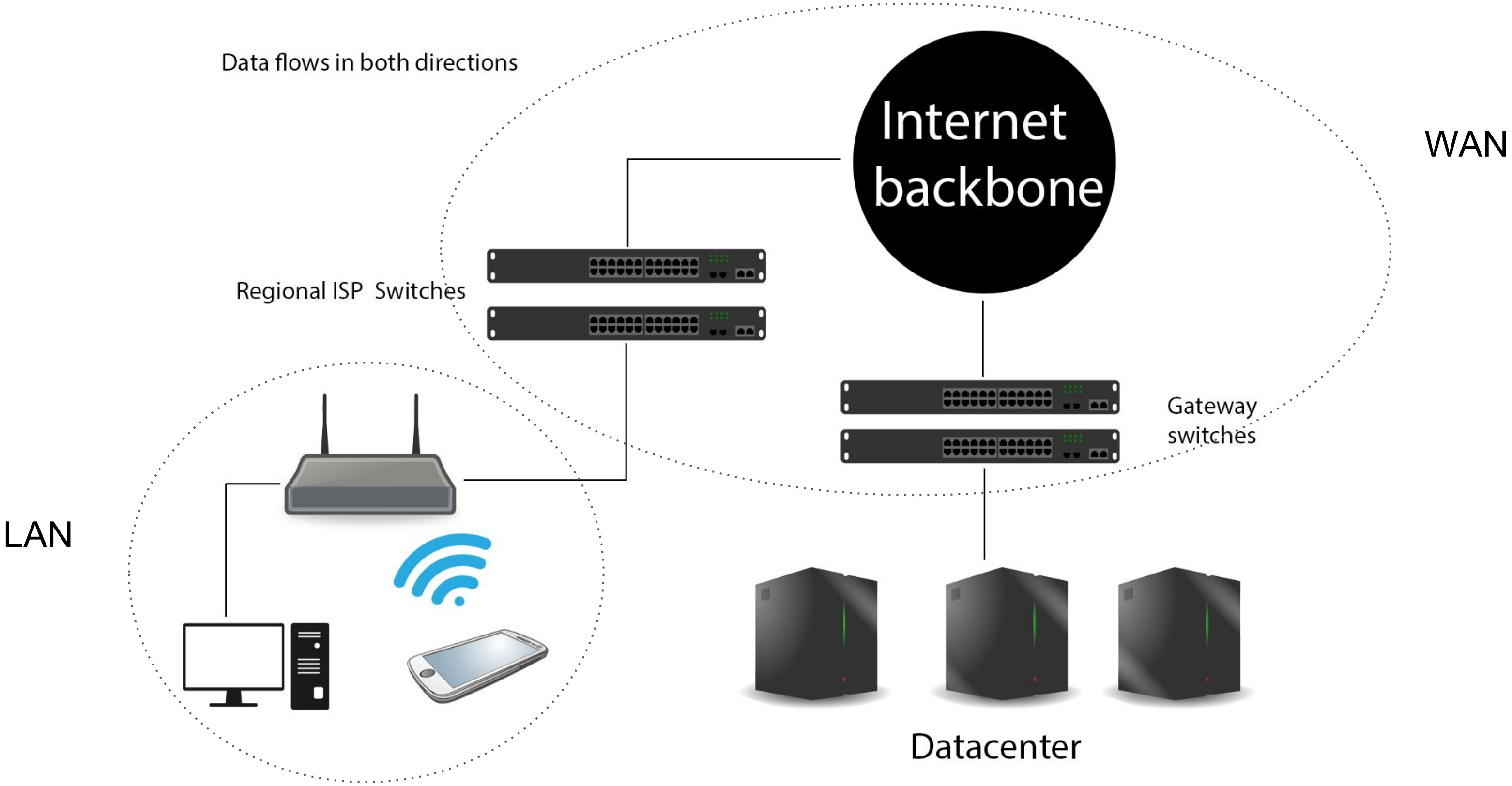
Internet is a modern postal system

- Internet is a world wide **network of computers**. If a device is connected to the internet, it can communicate with a billion of other devices.
- In many ways, this network is similar to the US Postal Service Network.
- For postal service to work, every connected home/office/place must have an address. When your device connects to the internet, it too gets an unique address.
- When you use the postal service, the packet you send go throughs many stops on the way before it reaches its destination. On the internet, when you browse a website or send an email, **data packets** are being sent from one device to another, and these data packets are go through many intermediaries.

Internet vs Post : Postal delivery



Internet vs Post: Internet delivery

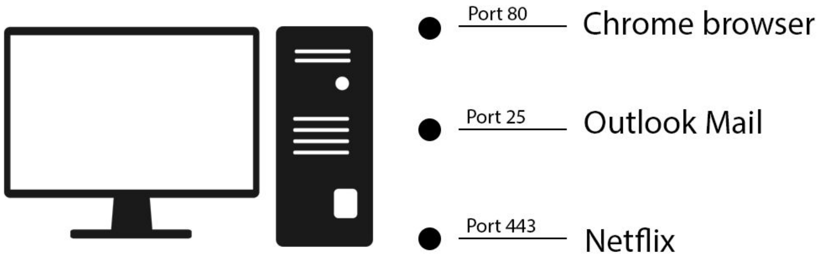


Internet Addresses

- Every time a device connects to the internet, the internet service provider (ISP) gives that device a unique address.
- This address is called the **IP address**. IP stands for **Internet Protocol**.
- For example, as I am connected to the internet right now, my IP address is “108.83.16.138”. You can see your current IP address at https://whatismyipaddress.com/
- The IP address (IP version 4), is a string of 4 numbers (each upto 255) joined together by dots. This system can represent up to about 4 Billion unique addresses.
- That is not enough in the current state of the technology world, so most of the devices now understand a new addressing system defined in **IP version 6**, which can represent approximately 3 * 10^38 unique addresses.

Internet Addresses: Ports

- Each device usually uses the internet for many different applications simultaneously.
- Thus, just the device IP address is not enough for one service to communicate to the device.
- IP Address:Port combination uniquely defines a specific communication endpoint.

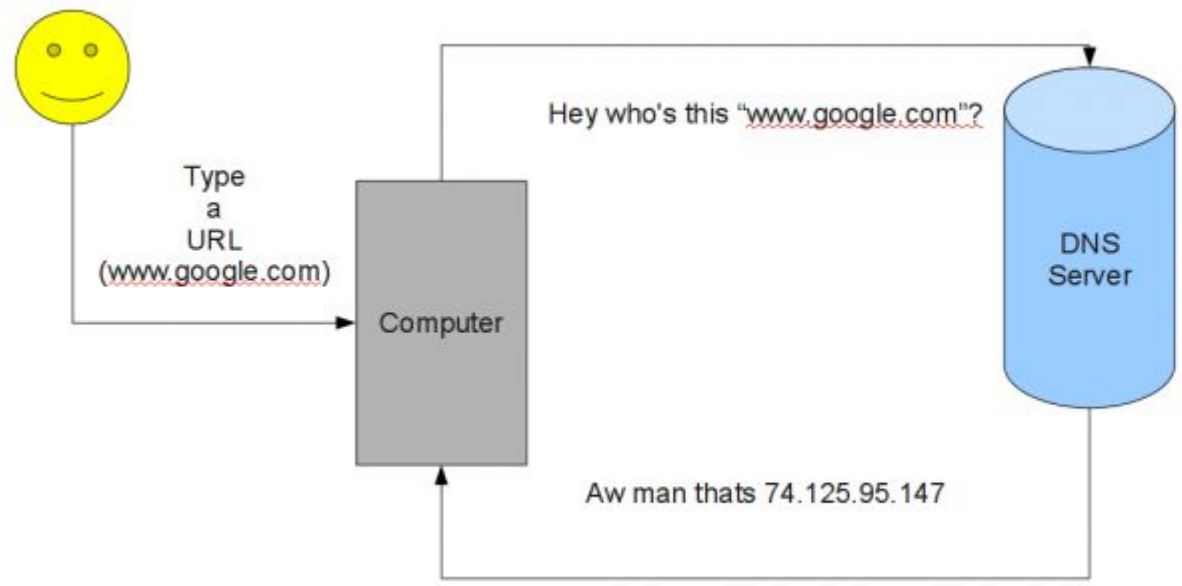


IP address: 5.10.15.255

Domain Name Service

- When you connect to a website, like www.wikipedia.org, you don't tell your browser the IP address for wikipedia.
- How does your computer know where to send your request data packet?
- Your ISP provides a **Domain Name Service (DNS),** which translates easy to remember names to the IP addresses.
- It's like referring to your phone directory to find out where Molly's house is.
- If a name → IP address is mapping is not present (or is deliberately hidden) in the DNS server, then it will appear as if that website is not working for you.

DNS - how it works



Type
a
URL
(www.google.com)

Computer

Hey who's this “www.google.com”?

DNS
Server

Aw man thats 74.125.95.147

Internet works through protocols

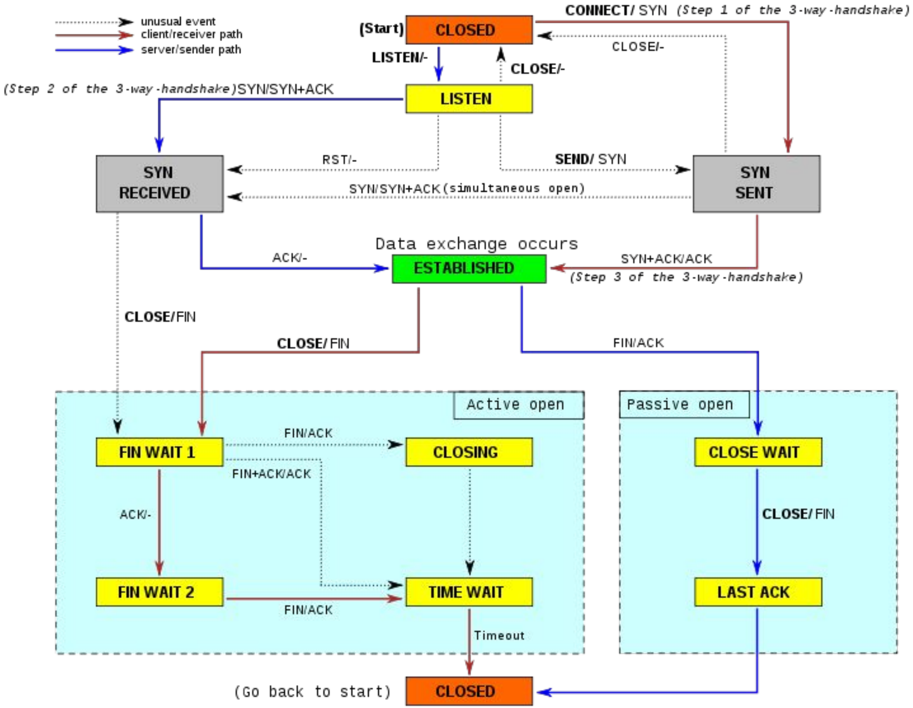
- On the internet, machines communicate among each other. Since machines are dumb, we have designed many **protocols** for machines to understand each other.
- There are protocols to define processes in real life. For example, when the British royals enter a room, there is a protocol for the order in which they should enter (essentially, they enter in the order of who’s next in line to become the monarch).
- Similarly, one protocol specification says that a data packet on the internet must present it’s data attributes in a pre-decided order - the version of IP is the first element, length of IP header is next, an so on. This is defined in the Internet Protocol (IP).

Common protocols on the internet - TCP

TCP (Transmission Control Protocol) - This protocol defines how two machines should exchange data, such that data is not lost in the way.

Almost every data transfer on the internet uses this protocol.

It is pretty complicated, as can be seen in the diagram on the right.

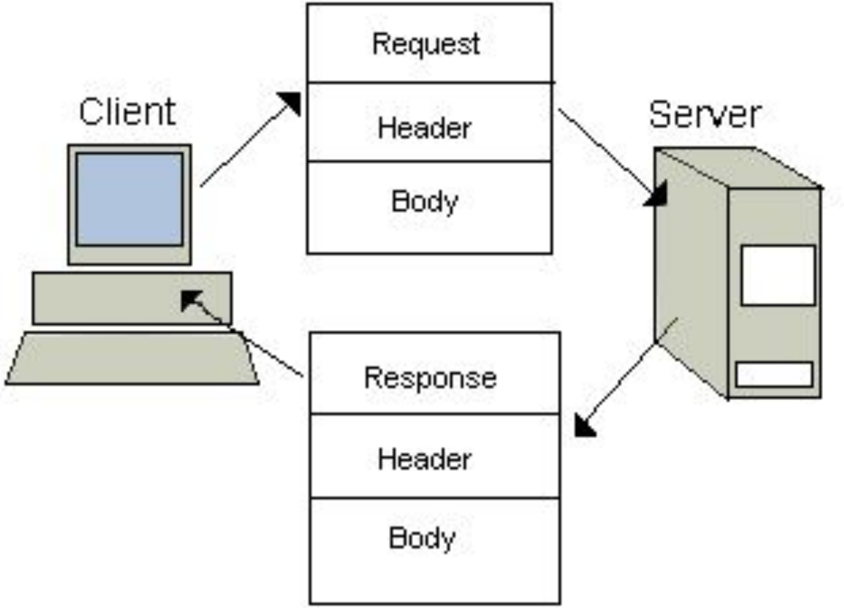


Common protocols on the Internet - HTTP

HyperText Transfer Protocol (HTTP) - this is the protocol which is used by websites to deliver to content to browsers.

Whenever you go to a web address in your browser (client), your browser sends a HTTP request to the website (server) using this protocol.

The website (server) responds using this same protocol, which your browser (client) understands and then it shows you the content.



HTTP - more context

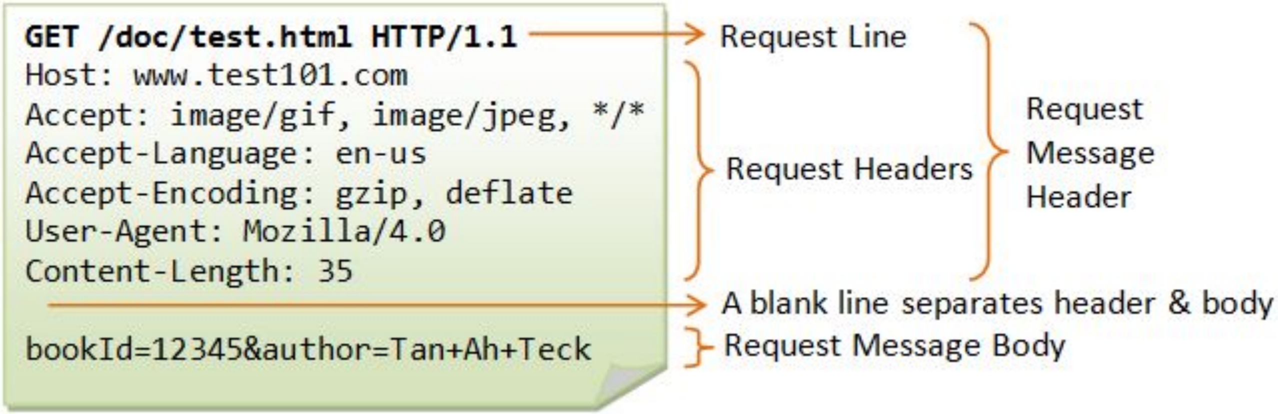
- This is the protocol using which the whole **World Wide Web (WWW)** works.
- Note that the term **internet** doesn't mean the same thing as the term **World Wide Web.**
    - Internet refers to the all the hardware and devices which connects billions of devices worldwide.
    - The term WWW refers to one *application* which uses the internet. It if by far the most popular application, but it's not the only one.
    - For example, when an email is sent (when you click send on gmail, or outlook mail), it internally uses another protocol to make that delivery (that protocol is called Simple Mail Transfer Protocol or SMTP).
- Sir Tim Berners-Lee headed the team which developed the HTTP Protocol. He is frequently called the inventor of the World Wide Web.

Making your own website

- To make your own website, you have to develop a program that acts as a **HTTP server.** Let's break it down:
- A **server** is a program that runs continuously. It accepts ***requests*** of various kinds (requests are sent by other computers), and sends back an appropriate ***response*** to every request.
- HTTP defines the format in which requests and response must be formulated using which the server and client (the user's browser) must communicate.
- As you might expect, there are a lot of code libraries which help make it easier to write an HTTP server.
- Even so, we need to understand some elements of HTTP to be able to build a website.

HTTP GET Request

- In the HTTP, the client can make many *types* of requests to the server.
- The most common one is the GET request.
- As the name suggests, the GET request is used when client want to receive content from the server.



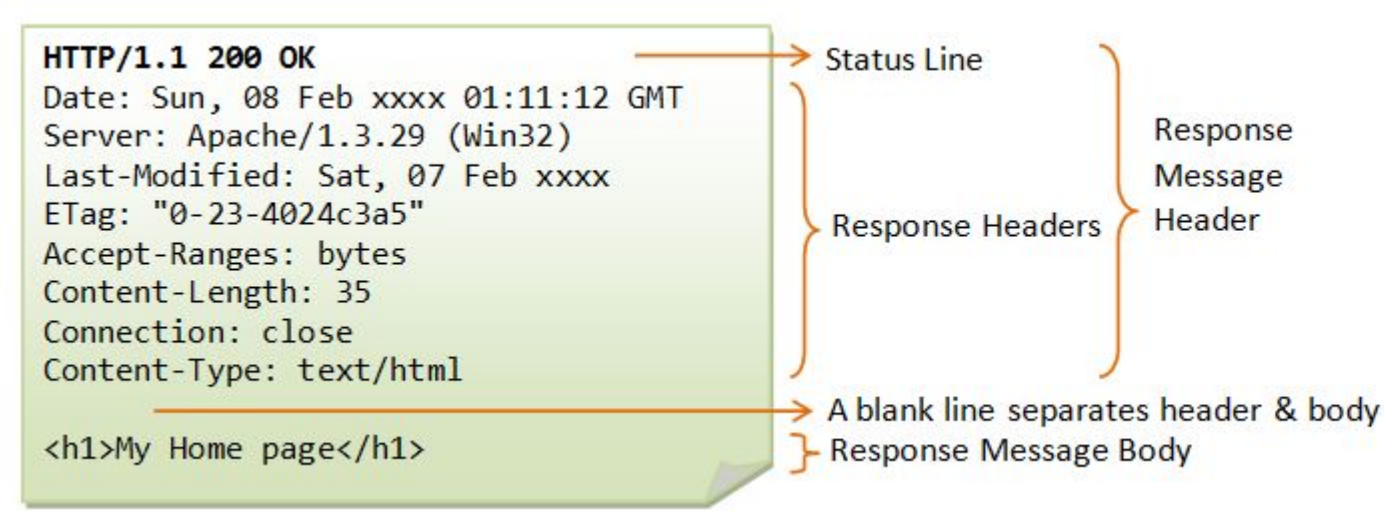
```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

HTTP Request path and query

- On the previous slide, the exact GET request was **GET /doc/test.html.**
- The string **/doc/test.html** is called the **path**.
- This is how a HTTP server identifies what content to serve. For example, Dropbox website uses **/home** when you are looking at your own files, **/share** when you are looking at shared folders, etc.
- Often, the path is augmented with a query: On Dropbox, when viewing a file, the path becomes: **/home?preview=filename.ext**. The part **?preview=filename.ext** gives the website additional details on what the client is requesting.
- When writing our own website, we will also define certain paths and then send back an appropriate response.

HTTP Response

- The HTTP server sends a response according to the protocol.
- If you are browsing the internet, then the response would contain the HTML code to render the page.



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

HTTP Response Status codes

- The most important part of an HTTP response is the **status code**.
- It is mainly used to communicate if there was an issue in processing the request by the server.
- The most common status codes are:
  - 200 - Status OK. Everything is fine!
  - 400 - Bad Request. Client sent a malformed request.
  - 404 - Not Found. The requested path is not available.
  - 500 - Internal Server Error. The server experienced an unexpected error while serving the request.

Browsing developer style

- Now that you understand the HTTP, you can look for different elements we discussed when you are browsing the internet.
- You can look at the address bar in your browser to look at the URL, which will be of the format: scheme:[//authority]path[?query][#fragment]
  - **scheme** will be http or https. **authority** will be the website name.
- Browsers have a **developer tool**. In both Chrome and Firefox, you can use the shortcut Ctrl/Cmd + Shift + C to get the developer tool.
- The developer tool has a **network** section, and you can find the complete list of HTTP Requests and Responses there.

What is HTML?

- Hypertext Markup Language (HTML) is a language that is used to define the *structure* of a web page.
- Every web page on the internet is written in the HTML. All browsers understand HTML and then *render* the page on the screen using it.
- To look at the HTML code of any page on the internet, you can right-click in your browser and select “View Page Source” option in the menu.
- If one has to build a website, then one needs to create the HTML structure of the pages they want to show to the user.

What is Javascript?

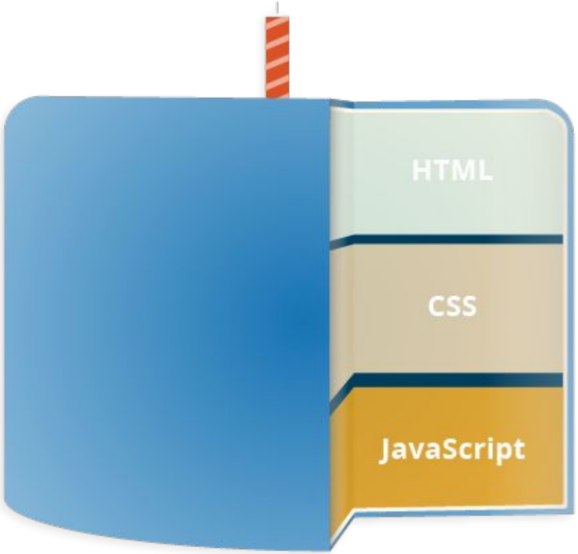
- Javascript is a programming language. It’s main use is to add interactive behaviour in the website (for e.g., read an online book using keyboard, add animations to a website, etc.)
- Javascript can fiddle with the HTML structure of the page. Thus, javascript can be used to completely change the look of the page, without asking the HTTP server for another web page.
- Javascript can also be used to send and receive HTTP requests in the background, without interfering with the display and behavior of the existing page (e.g. when you click on any email in Gmail, the content of the email opens up without refreshing the whole page). This technique is called **AJAX.**

What is CSS?

- Cascading Style Sheets (CSS) is a language which is used to describe the *presentation* of an HTML page.
- Essentially, it defines how a web page looks - the size and font of the text, how much spacing is there, what color the background is, etc.
- It’s also used to create websites which have ***responsive design***.
- It works by defining *classes* of rules (for presentation), and then the appropriate class is applied to any section of the page (as described in the HTML structure).
- This is the standard way of designing the look of web pages.

Web Development cake

- HTML, Javascript and CSS are best buddies and practically every web page you visit on the internet uses a combination of these three technologies to build the website.



Client side vs Server side

- When someone build a website and addition of functionality is discussed, it’s common to use the terms **client side** and **server side**.
- Client side refers to the code that *runs on the user’s device*. Using javascript to add interactivity is an example of that.
- Server side refers to the code that *runs on the HTTP server.* A server written in Python or Java is an example of that.
- Sometimes, when a new feature is required on the website - there can be a choice between client-side vs server-side.
  - E.g. If there is a language translation website, it can implement the actual translation code in javascript and make it so that it works on the device (client-side). Or it can send a request to the HTTP server for every translation required which does the translation and responds with the output (server-side).
- Because the technologies are very different, most developers specialize in either the client-side development or the server-side development.

What about mobile app development?

- If seen from a bird’s eye view, then a website is not very different from a mobile app. Both need development on the client side and the server side.
- The main differences are:
  - **Client-side**: The HTML, Javascript, CSS cake doesn’t work to implement client-side behavior for apps. The *client app* is written using the proprietary technologies of the mobile platform (like Android SDK on Android, or Swift on iOS).
  - **Server-side**: The essential components of server side remain the same. The main difference is that there is no longer a need for the server to create web pages in HTML (because the client doesn’t use HTML). The server can respond with data (like the contents of an email when using the Gmail app) in any format the developer finds convenient.
- Since the client-side technologies of web development are also very different from mobile app development, developers working on the client side usually choose to specialize in one of them.