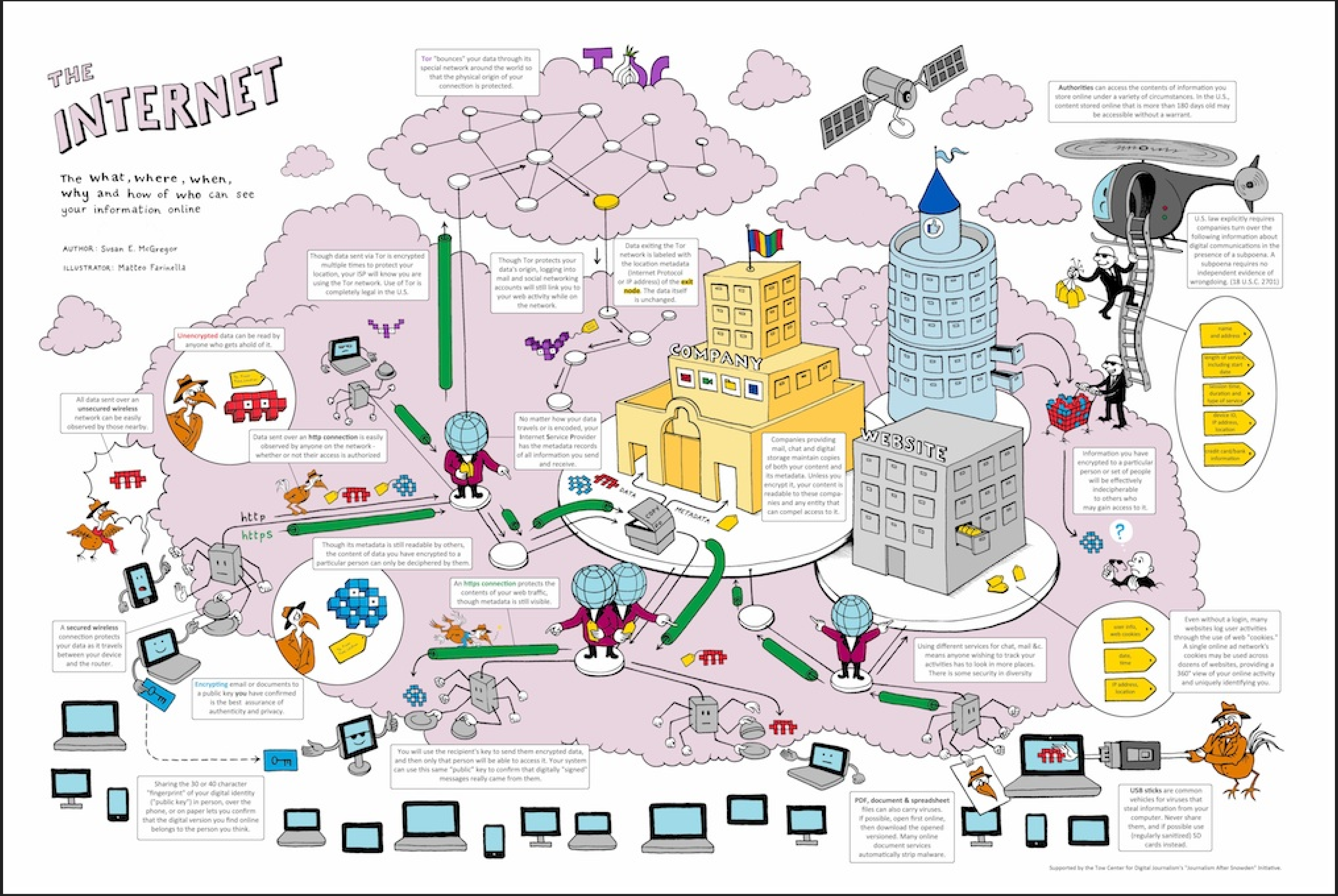
2. BASIC CONCEPTS


THE INTERNET
The what, where, when, why and how of who can see your information online
AUTHOR: Susan E. McGregor
ILLUSTRATOR: Matteo Farinella
Tor
http
https
COMPANY
WEBSITE
DATA
METADATA
name and address
length of service including start date
session time, duration and type of service
device ID, IP address, location
credit card/bank information
user info, web cookies
date, time
IP address, location

Tor "bounces" your data through its special network around the world so that the physical origin of your connection is protected.

Though data sent via Tor is encrypted multiple times to protect your location, your ISP will know you are using the Tor network. Use of Tor is completely legal in the U.S.

Though Tor protects your data's origin, logging into mail and social networking accounts will still link you to your web activity while on the network.

Data exiting the Tor network is labeled with the location metadata (Internet Protocol or IP address) of the **exit node**. The data itself is unchanged.

Authorities can access the contents of information you store online under a variety of circumstances. In the U.S., content stored online that is more than 180 days old may be accessible without a warrant.

U.S. law explicitly requires companies turn over the following information about digital communications in the presence of a subpoena. A subpoena requires no independent evidence of wrongdoing. (18 U.S.C. 2701)

Unencrypted data can be read by anyone who gets ahold of it.

All data sent over an **unsecured wireless** network can be easily observed by those nearby.

Data sent over an **http connection** is easily observed by anyone on the network - whether or not their access is authorized

No matter how your data travels or is encoded, your Internet Service Provider has the metadata records of all information you send and receive.

Companies providing mail, chat and digital storage maintain copies of both your content and its metadata. Unless you encrypt it, your content is readable to these companies and any entity that can compel access to it.

Information you have encrypted to a particular person or set of people will be effectively indecipherable to others who may gain access to it.

Though its metadata is still readable by others, the content of data you have encrypted to a particular person can only be deciphered by them.

An https connection protects the contents of your web traffic, though metadata is still visible.

A **secured wireless** connection protects your data as it travels between your device and the router.

Encrypting email or documents to a public key **you** have confirmed is the best assurance of authenticity and privacy.

Using different services for chat, mail &c. means anyone wishing to track your activities has to look in more places. There is some security in diversity

Even without a login, many websites log user activities through the use of web "cookies." A single online ad network's cookies may be used across dozens of websites, providing a 360° view of your online activity and uniquely identifying you.

You will use the recipient's key to send them encrypted data, and then only that person will be able to access it. Your system can use this same "public" key to confirm that digitally "signed" messages really came from them.

Sharing the 30 or 40 character "fingerprint" of your digital identity ("public key") in person, over the phone, or on paper lets you confirm that the digital version you find online belongs to the person you think.

PDF, document & spreadsheet files can also carry viruses. If possible, open first online, then download the opened versioned. Many online document services automatically strip malware.

USB sticks are common vehicles for viruses that steal information from your computer. Never share them, and if possible use (regularly sanitized) SD cards instead.

Supported by the Tow Center for Digital Journalism's "Journalism After Snowden" Initiative.

2.1 OSI LAYER MODEL

Suppose we dinner tonight with the England Queen. "Wait until the queen extends her hand to take it", they indicate us, "It's the social protocol".

NETWORK PROTOCOLS

Social protocols are rules that enable individuals and communities to express social capabilities like discuss important things with the English royalty.

In computer networks are a very similar concept.

NETWORK PROTOCOLS

For example, suppose we are designing a very simple room chat. We establish that for each message the data must be separated with semicolon (;) and structured as follows:

```
receiver;"message";priority
```

If anyone send the data:

```
Tania;"Hello dear",HIGH
```

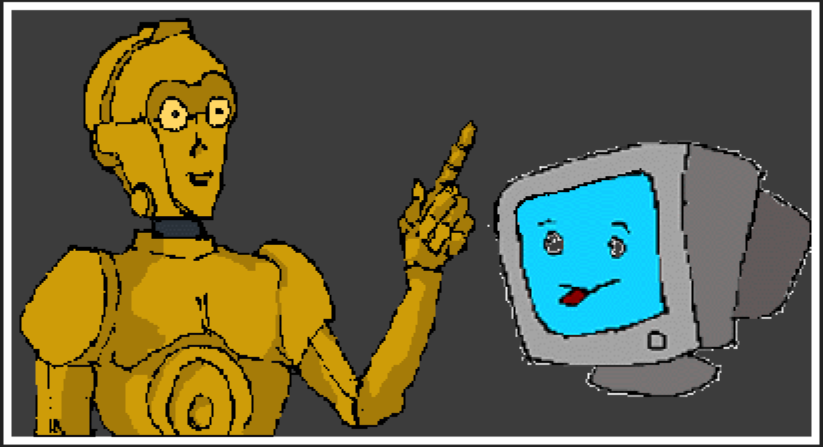
our chat does not know who deliver it.

NETWORK PROTOCOLS

At the beginning of the computer networks (1980's) all companies defined different ways to communicate their computers.
As the network become bigger and bigger, the world realized how serious will be the incompatibility, and then, the concept of network protocol was born.

NETWORK PROTOCOLS

A network protocol is a list of rules that indicates how two or more computers must transmit information.
These rules enable one computer to understand messages sent from another and then act on those messages, defining: syntax and semantics.

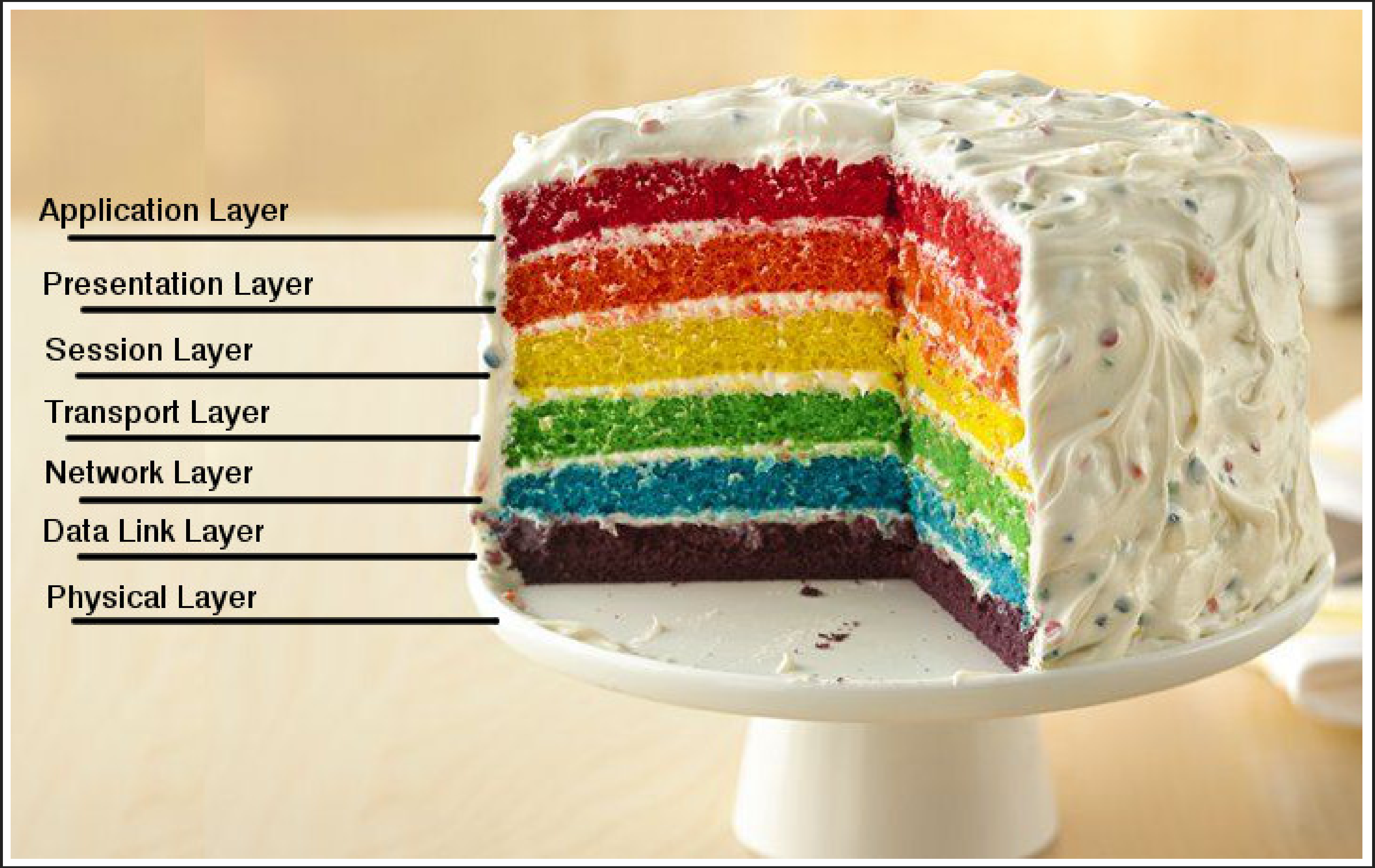


http://www.library.louisville.edu/tilt/glossary/protocol2.htm

OSI LAYERS

The OSI (Open System Interconnection) Model is the standard conceptual model of the today computer communications.
This model break down the network data in layers. In each layer have different network protocols.
So we have different protocols for each layer.

OSI LAYERS


Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

OSI LAYERS

- **Physical Layer.** Transmit bits (0, 1) without any structure through a physical media. *Protocols: Ethernet, WiFi, Optical Fiber, etc*

- **Data Link Layer.** Binary data form packages called frames. This layer controls the flow and delimits the frames. *Protocols: MAC, LLC*

- **Network Layer.** Routes the message, and encode into frames. *Protocols: IP*

- **Transport Layer.** Stablish point-to-point connections between two different computer processes. Concept of port is introduces. *Protocols: TCP, UDP*

- **Session Layer.** Put rules about sessions if are necessary.

- **Presentation Layer.** Handle data compression and data encode.

- **Application Layer.** Entry point for all the network services. Acts like a window for user purposes: transfer files *(FTP)*, send mails *(SMTP)*, transmitir datos web *(HTTP)*.

2.2 TCP / UDP PROTOCOL

Of all of previous protocols, these are two that you must should know.
Both are included in the Transport Layer and have the concept of port.

MAIN DIFFERENCES

TCP UDP

TCP	UDP
It is a connection-oriented protocol.	It is a message-oriented protocol.
Bidirectional	Unidirectional
Based on sockets	Based on datagrams
Reliable (always receives)	Unreliable
Guaranteed order	Packages could be delivered in different order than send.

2.3 HTTP PROTOCOL

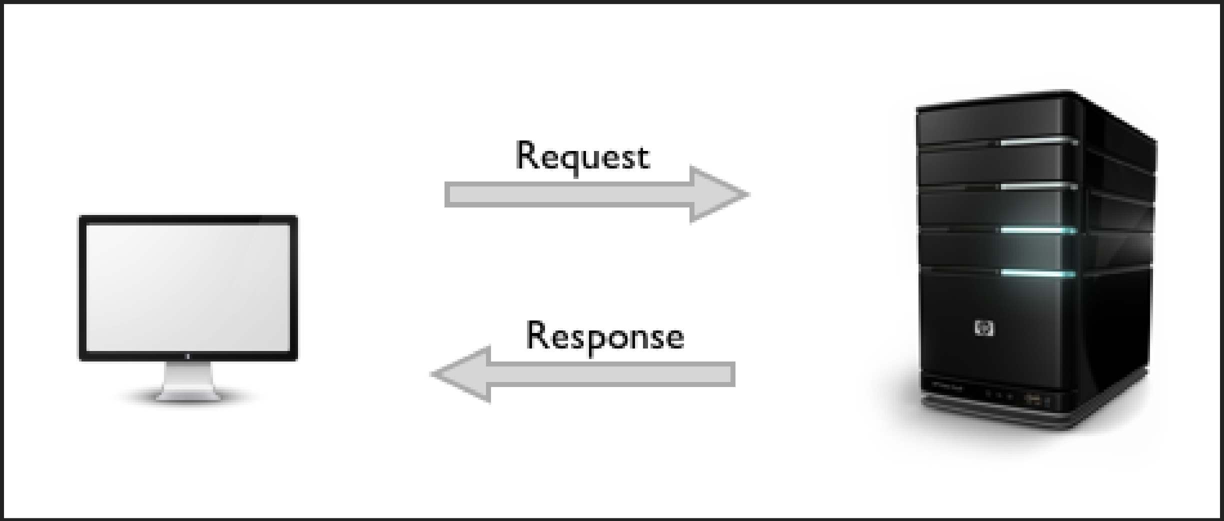
HTTP stands for HyperText Transfer Protocol. It's an application layer protocol, and is the FOUNDATION OF THE MODERN WEB.
As a web developer, we all must have a strong understanding of this protocol.

HTTP

- HTTP it is based on a host-client architecture.
- All communication between a host and a client occurs, via a **request/response pair.**
- Is a stateless protocol.
- Uses TCP/IP protocols and the default port is 80.
- The current version of the protocol is HTTP/1.1.
- Has a security version: HTTPS

REQUESTS

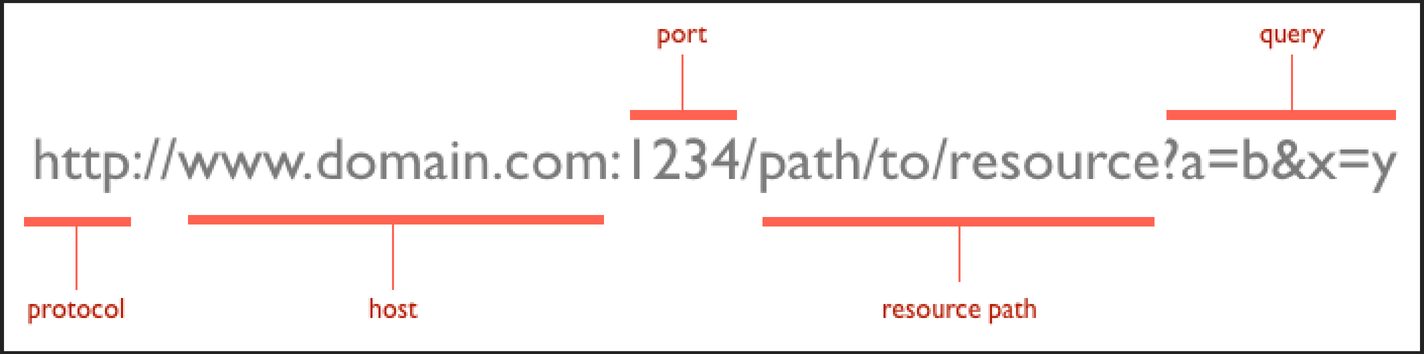
The communication in HTTP **always begin** in the client side with a HTTP request message, which is serviced through a HTTP response message in return for the host.
A host is called a server in that sense.



REQUESTS - URL

A HTTP request message have two parts: an URL and a HTTP verb.

The URL (Uniform Resource Locator) tells from who and what resource the client its wanting.



REQUESTS - VERBS

HTTP verbs indicate to the host the action that should be performed with the resource specified in the URL. The basic HTTP verbs are:

1. **`GET`.** Fetch an existing resource.

2. **`POST`.** Create a new resource, extra payload usually specifies the data.

3. **`PUT`.** Update an existing resource. The payload may contain the updated data for the resource.

4. **`DELETE`.** Delete an existing resource.

`PUT` and `DELETE` are sometimes considered specialized versions of the `POST` verb.

RESPONSES

Once the send the HTTP request, the host (server) process the request and send the response.
A response message from the server has two parts: the status code and the payload message.
The HTTP spec defines certain number ranges for the status codes:

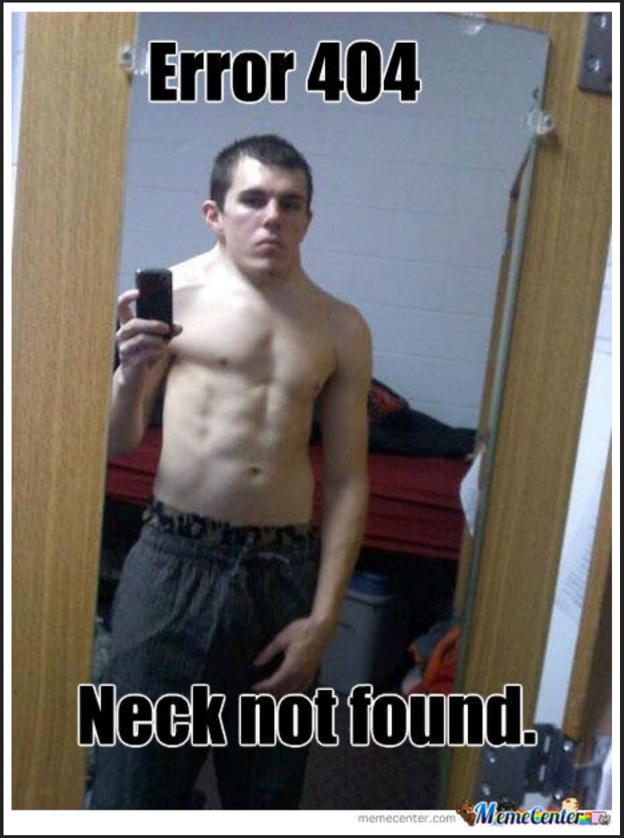
- **2xx Successful.** This tells the client that the request was successfully processed. The most common code is 200 OK.

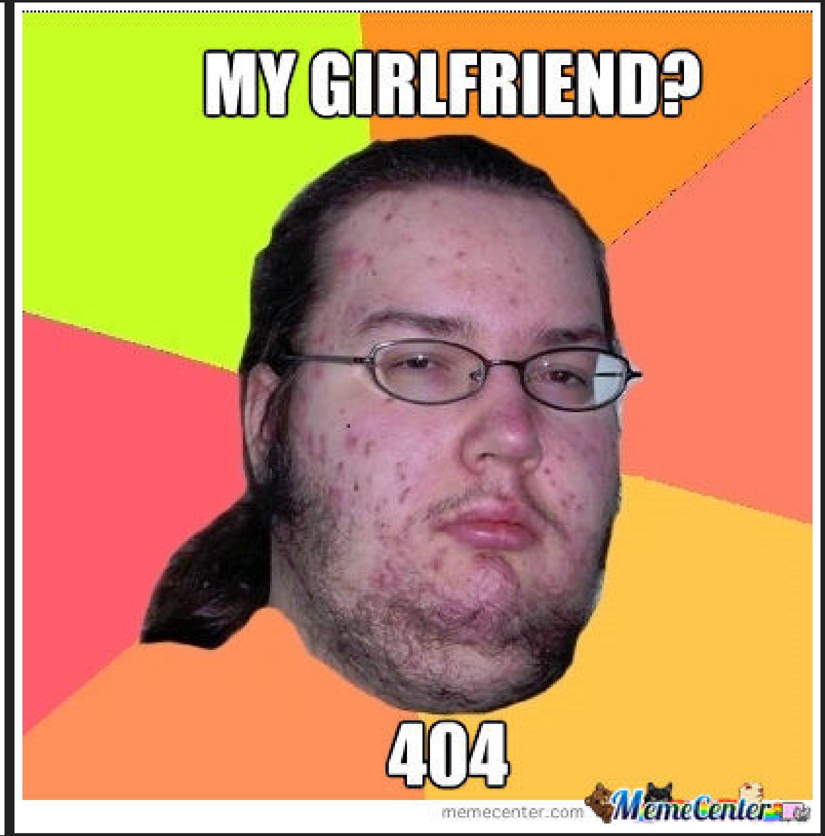
- **3xx Redirection.**This requires the client to take additional action.

- **4xx Client Error.**These codes are used when the server thinks that the client is at fault. The most popular code in this class is 404 Not Found.

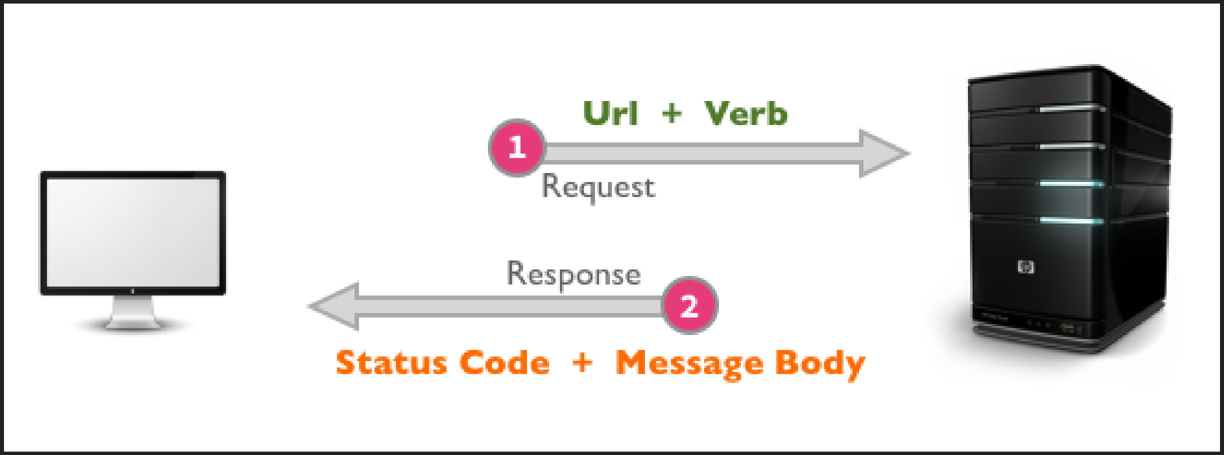
- **5xx Server Error** This class of codes are used to indicate a server failure while processing the request. The most commonly error code is 500 Internal Server Error.

RESPONSES


Error 404
Neck not found.
memecenter.com
MemeCenter


MY GIRLFRIEND?
404
memecenter.com
MemeCenter

SUMMARY

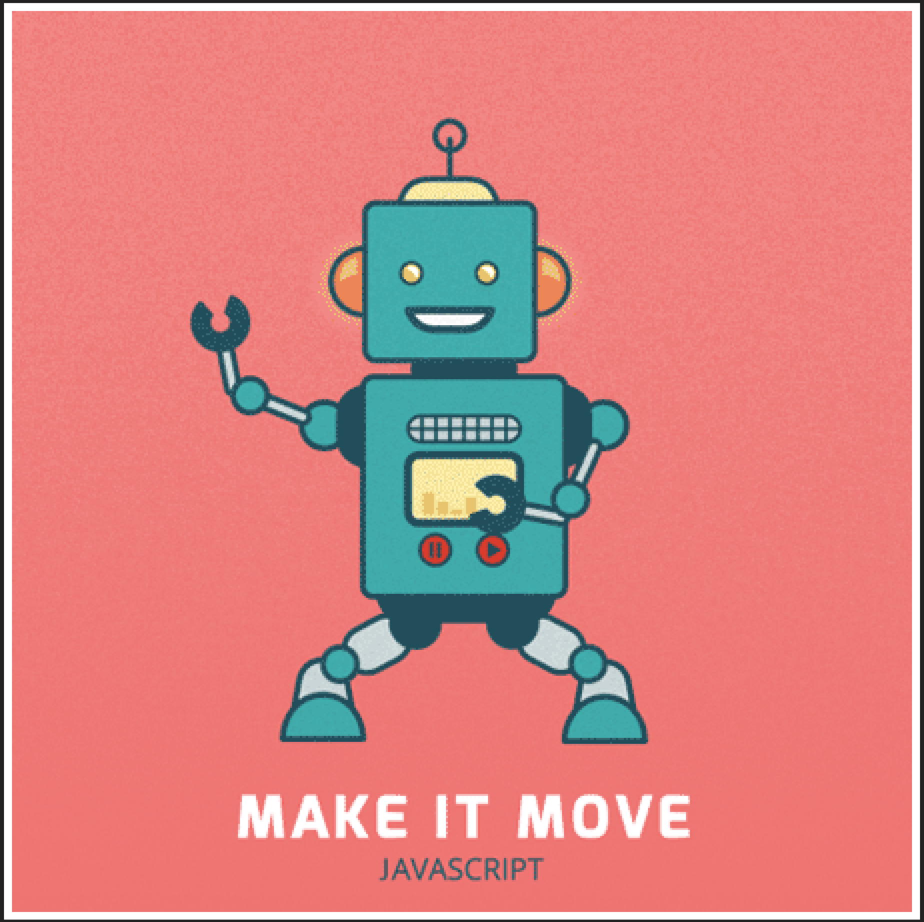


2.4 HTML

With CSS and JavaScript it forms a triad of cornerstone technologies for the WorldWideWeb.

ANATOMY OF A WEBSITE

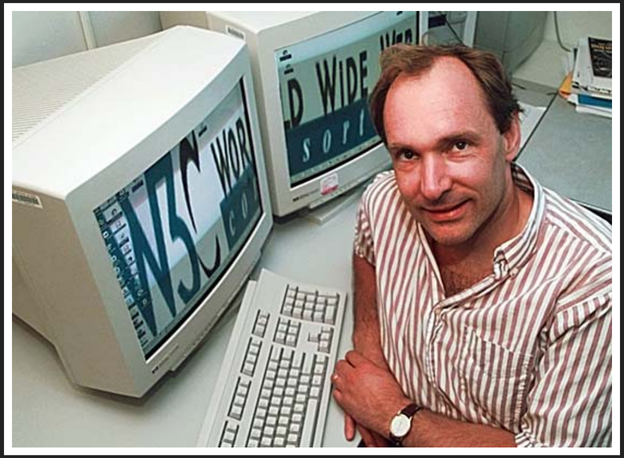
HTML (structure) + CSS (style) + JavaScript (logic and behaviour)



HTML

HTML is the most elemental building block to program any web site.
All web sites that you can see through a web browser is written with HTML.

It was created at the starts of 1990's by Tim Berners Lee, and currently mantained by the W3C.



HTML

HTML stands for HyperText Markup Language.
It means that is a language made of hypertext and marks. Both are used to indicates a browser the structure of a web page.

HTML only gives structure (like the bones) to the web page, not style, nor logic or behaviour.

HYPERTEXT

The Hypertext allow embed external items (fonts, style, images, video, scripts) indirectly through a reference of each item. It is responsability of the render get the items with an extra effort and combine all stuff to get the final view.

TAGS AND ELEMENTS

HTML documents has a structure of nested HTML elements (hypertexted content). These are indicated in the document by marks, hereinafter referred to as tags.

A <u>tag</u> is a code element (text) enclosed by angle brackets: < >. Generally comes in pairs: "open tag" and "close tag", like:

```
<h1></h1>
```

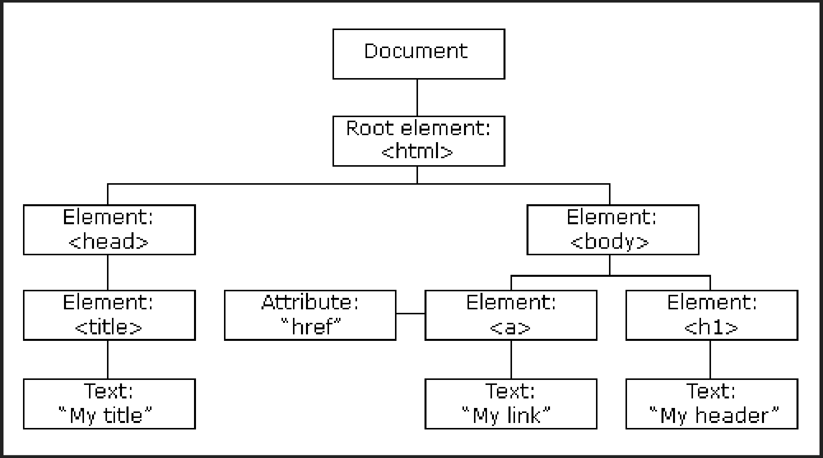
An HTML element has two properties: attributes and content.

```
<tag attribute1="value1" attribute2="value2">content</tag>
```

HIERARCHY

There are tags that may enclose no content, like `<img>`.
The tags that enclose content defines a hierarchy, where the outside tag has more hierarchy than the deeper.

When the renderer, parse the HTML document text, create an object in memory called Document Object Model (DOM).

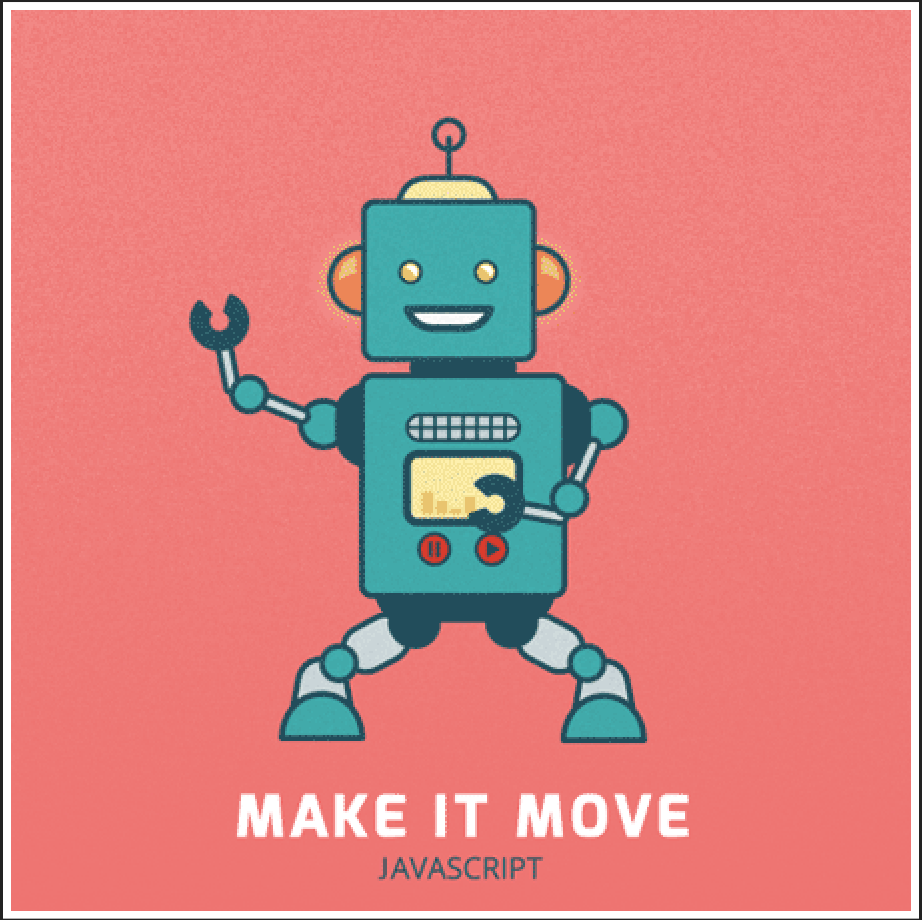


2.5 CSS

The Cascading Style Sheets (CSS) are files used for give the style to a website.

ANATOMY OF A WEBSITE

HTML (structure) + CSS (style) + JavaScript (logic and behaviour)



STYLE SHEETS

A style sheet consists of "style rules".

Each style rule consistsof a "selector" and "declarations" of property-value pairs:

```
selector {
  property: value;
  property: value;
}
```

```
body {
  color: yellow;
  background-color: black;
}
```

CSS IN HTML

There are three main ways to add CSS to an HTML:

1. Style tag.

2. Link tag.

3. Inline styles (apply directly - VERY NASTY).

SELECTOR

The selector is used to select which elements in the HTML page will be given the styles inside the curly braces.

Types:

1. By element (`img, div, body`)
2. By id (#)
3. By class (`.`)
4. By property (`div[name="value"]`)

NAMING CONVENTIONS

Some rules to follow when making IDs and class names:

1. Describe the content, not the presentation ("warning", not "redbox").
2. Use all lowercase, and hyphens when needed for readability ("header-info", not "headerInfo").
3. Use hyphens to show that a class or ID is part of something else. (e.g. "footer", "footer-copyright", and "footer-logo").