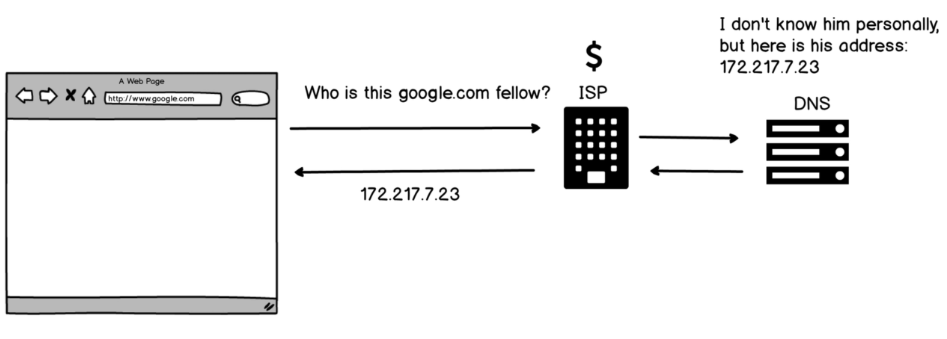
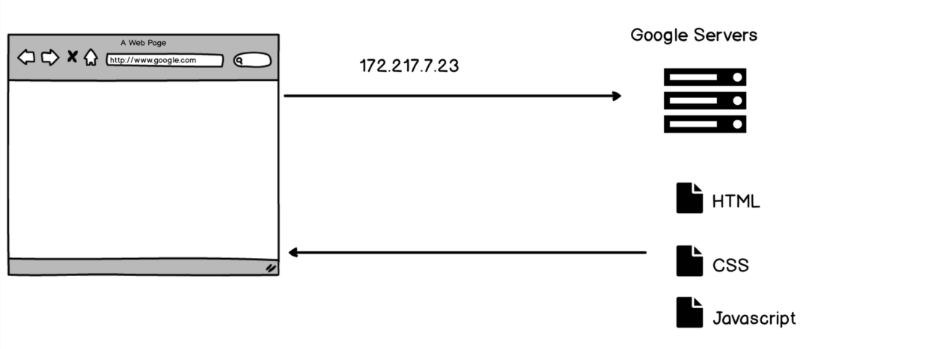
**BROWSING THE WEB**

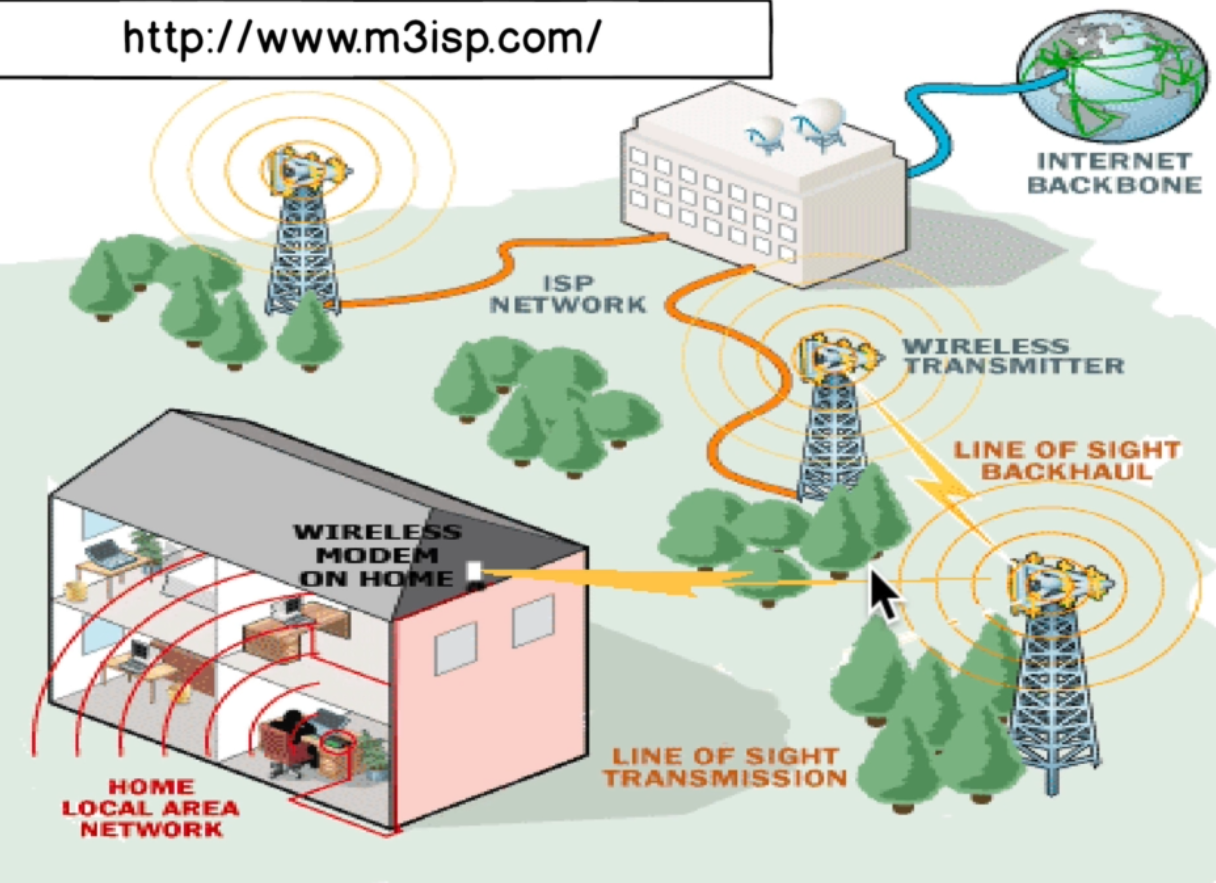
What happens when we type [www.google.com](http://www.google.com) to the browser?



* We ask ISP “Who is this google.com fellow?”.
* ISP ask this request to DNS.
* DNS sends google.com’s address.



* Browser sends another request to address of google.com which is 172.217.7.23
* Then google.com gives us HTML, CSS, Javascript.
* Then we have google.com working.



tracert google.com in Windows cmd shows the route through the google.com

**HTML, CSS, JAVASCRIPT**

HTML (Hypertext Markup Language)

HTML is a way for us to just write text on a website. These texts on websites can also have hyperlinks that can link to other parts of the website.

Tim Berners Lee invented in 1989.

<!DOCTYPE html>

<html>

<head>

<title>My first website!</title>

</head>

<body>

<h1>Helllooooo.</h1>

<h2>Helllooooo.</h2>

…

<h6>Helllooooo.</h6>

This is not a paragraph.

This is also not a paragraph.

<p>This is a <b>paragraph.</b></p>

<p>This is also a <i>paragraph.</i></p>

</body>

</html>

Some mobile phones can’t bold with b tag and can’t italicize with i tag. So we use strong for bold and em for italicize.

head tag contains information about what the page needs. You can like a CSS file to your HTML document with:

* <link rel=”stylesheet” type = “text/css” href = “style.css”>
  + rel 🡪 specifies what relationship this link will have
  + type 🡪 media type
  + href 🡪 linking file

CSS (Cascading Style Sheets)

To make website pretty.

Selector {

Property: value;

}

Cascading means: it always takes the selector that is at the end.

You can add CSS features to your page in browser. This doesn’t affect your original document.

You can also change style in HTML document too:

* <header style=”background-color: green; color: red”;>

OR

* <style>

li {

background-color: purple;

}

</style>

**CHECK 1**

JAVASCRIPT (ECMASCRIPT)

Created in 1995 by the Netscape Web Browser.

Adds functionality. You can use Console in browser to add functions.

JS has 7 types:

1. Number
2. String
3. Boolean
4. Undefined
5. Null
6. Symbol (new in ECMAScript 6)
7. Object

**Number**

You can write “3+4” kinda things in console in browser.

🡪 clear() will clear the console.

**String**

“Bob” or ‘Bob’ 🡪 this is string. You can simply type this to console in browser.

You can do 🡪 “Hello” + “There!”

🡪 ‘This isn\’t very nice’

🡪 10 + “34” 🡪 “1034”

🡪 10 - “3” 🡪 7

🡪 “hello” - “bye” 🡪 NaN (Not a number, it is number type.)

**Boolean**

true / false

3 > 2 🡪 true

3 === 3 🡪 true

3 !== 3 🡪 false

🡪 <script type="text/javascript" src="script.js"></script>

* If you put this in header, JS works before anything else so this delay what gets seen by the user . HTML runs JS when it sees above line.
* So put script tags at the bottom.

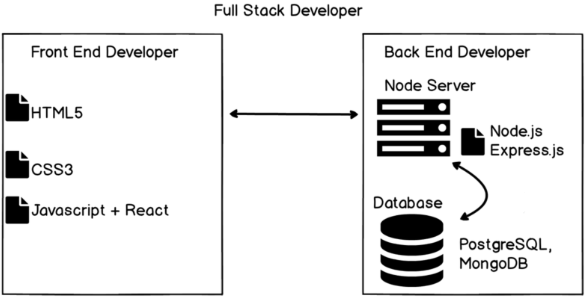
🡪 You can add JS directly to the HTML document:

<script type="text/javascript">

alert("hello");

</script>

**CHECK 2**



**HTTP / HTTPS**

Tim Berners Lee invented HTTP (Hypertext Transfer Protocol). It is the protocol that we use over the internet wires.

HTTP is a protocol which allows the fetching of resources such as HTML documents.

HTTP request words

* GET
  + Means you want to get sth.
  + e.g., getting Twitter feed with all the tweets from today
* POST
  + Means you want to post sth.
  + e.g., creating new user and add that user to the Twitter servers
* PUT
  + Means “I am gonna send you some data and I want you to update some data that already exists in Google servers.”
  + e.g., editing the posted tweet
* DELETE
  + Delete piece of data on the database.
  + e.g., deleting user account or tweet etc.

Over the years HTTP was extended to not only worry about text. Also images, videos, etc.

HTTP response

Server sends you 2 main things:

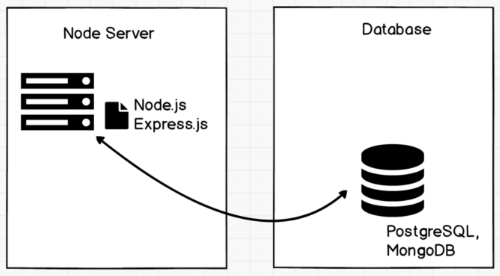
* HTTP message
  + Number code that comes with every response that tells you the response of the server.
* Data
  + e.g., HTML etc.

What if I wanna send some information (e.g., forms) to the server other than some URL saying I want this information?

2 ways:

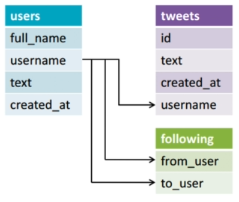
* query string
* through the body of the request

**DATABASE**



Relational Databases

Consists of 2 or more tables with columns and rows.

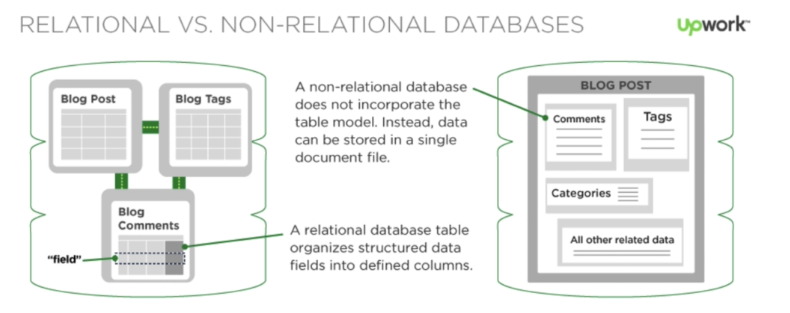


All relational databases use SQL. SQL allows us to communicate with the database.

NoSQL / Non Relational Database

This one is used when if your data requirements aren’t clear at the outset of your project and maybe you have a massive amount of unstructured data then you may not have the luxury of developing a relational database with a clearly defined schema.

This database is more like folders, just assembling related information of all types.



**PHP**

PHP can be embedded into HTML like JAVASCRIPT.

<!DOCTYPE html>

<html>

<head>

<title>Example</title>

</head>

<body>

<?php>

echo “Hi, I’m a PHP script!”;

?>

</body>

</html>

JS is used for client side while PHP is used for server side. Client can’t view code that is being run by server if everything is set properly and not misconfigured.

Code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there’s really no way that users can tell what you have up your sleeve.