Web application - 3 tier Architecture

1. Front-end:

- Anything that a user faces is a part of Frontend.
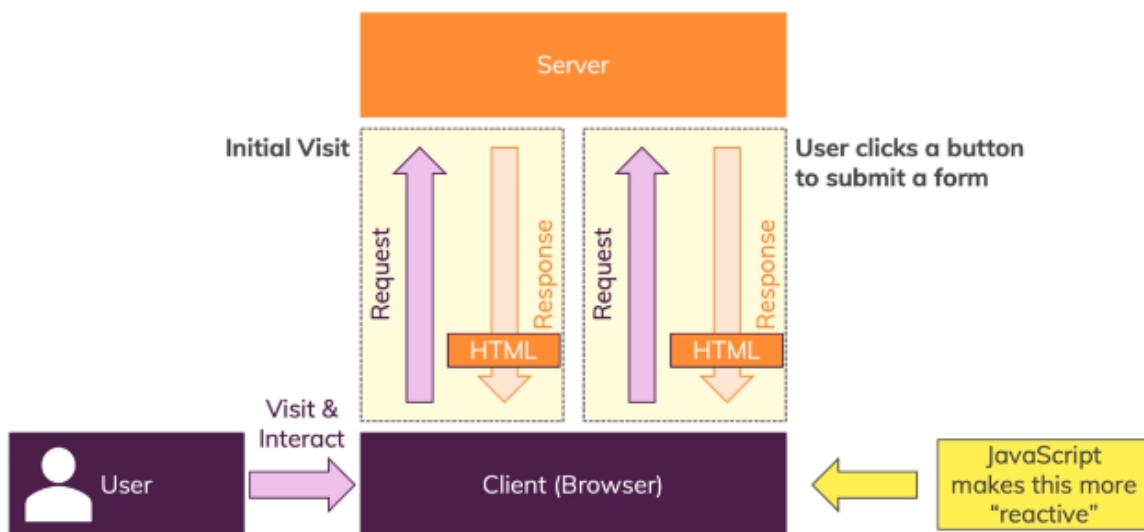- Frontend development has everything to do from design to dynamism of a web application.

2. Back-end:

- Backend is like the brain.
- It has everything to do with the logical.
- It also takes care of data storage and management by connecting to the database.
- It can combine various services to produce the desired results.
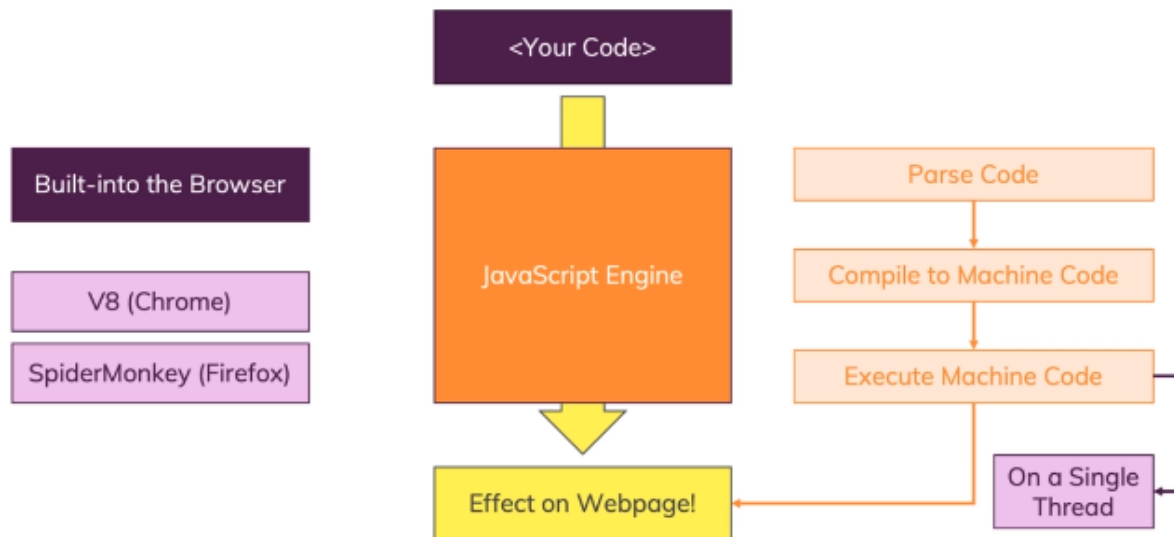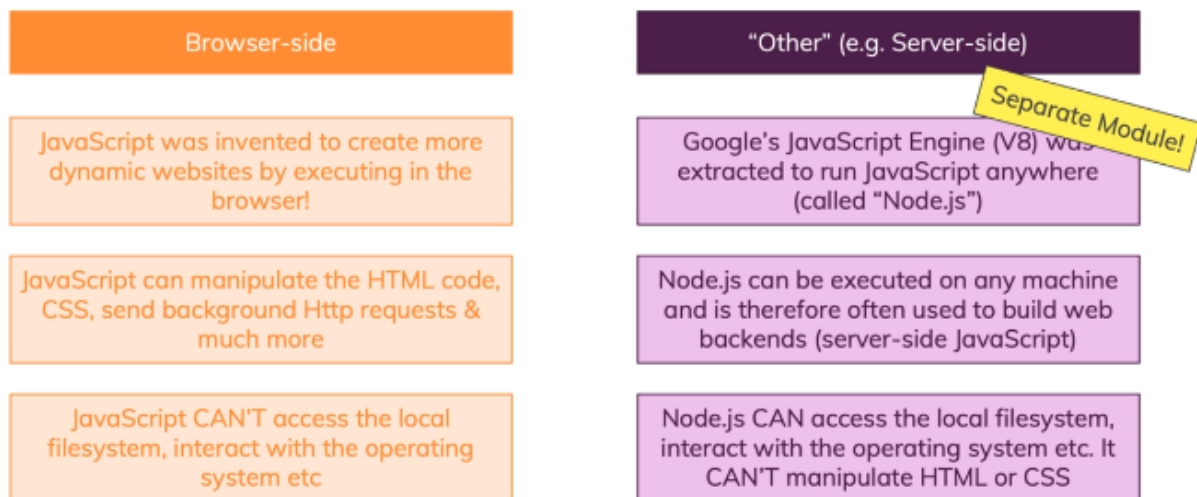
3. Database:

- Used to store data.

How do Web Pages Work?

How is JavaScript Executing?


```
<Your Code>
```

Built-into the Browser

V8 (Chrome)

SpiderMonkey (Firefox)

JavaScript Engine

Effect on Webpage!

Parse Code

Compile to Machine Code

Execute Machine Code

On a Single Thread

JS in Browser side vs Server side

Browser-side	"Other" (e.g. Server-side)
JavaScript was invented to create more dynamic websites by executing in the browser!	Google's JavaScript Engine (V8) was extracted to run JavaScript anywhere (called "Node.js") *Separate Module!*
JavaScript can manipulate the HTML code, CSS, send background Http requests & much more	Node.js can be executed on any machine and is therefore often used to build web backends (server-side JavaScript)
JavaScript CAN'T access the local filesystem, interact with the operating system etc	Node.js CAN access the local filesystem, interact with the operating system etc. It CAN'T manipulate HTML or CSS

Basically, when Javascript compiles all of your code, all variable declarations using **var** are hoisted/lifted to the top of their functional/local scope (if declared inside a function) or to the top of their global scope (if declared outside of a function) regardless of where the actual declaration has been made. This is what we mean by "**hoisting**".

Functions declarations are also hoisted, but these go to the very top, so will sit above all of the variable declarations.

Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope (to the top of the current script or the current function).

Note: Variables and constants declared with let or const are not hoisted

Window Object

The window object represents a window containing a DOM document.

window.location.href sets or gets absolute url of the current page

window.location.hostname returns the domain name of the web host

window.location.pathname returns the path and filename of the current page

window.location.protocol returns the web protocol used (http: or https:)

window.location.port property returns the host port number

window.history.back() - goes to one step back

window.history.forward() - goes to one step forward

Client / Web Storage

There are 2 different kind of Storages are available in HTML 5.

1. **Local Storage:**

   Permanent Storage.

No expire date unless you remove it.

Domain specific.

Syntax:

window.localStorage.setItem("key", "value");

window.localStorage.getItem("key");

window.localStorage.removeItem("key");

window.localStorage.clear();

2. **Session Storage**:

Temporary Storage.

Once you closed the browser or inactive of particular period.

Page specific.

Syntax:

window.sessionStorage.setItem("key", "value");

window.sessionStorage.getItem("key");

window.sessionStorage.removeItem("key");

window.sessionStorage.clear(); // logout

JSON

- JSON stands for JavaScript Object Notation.
- JSON used for store and exchange data between servers or remote servers.

- Its lightweight and self describing.
- JSON language independent.
- JSON object starts with '{' JSON array starts with '['.
- Easy to parse.
- JSON is purely a data format — it contains only properties, no methods.

Rules:

- Key must be a string which is denoted by "".
- JSON supports the following data types,

I. string
Ii. number
Iii. Array.
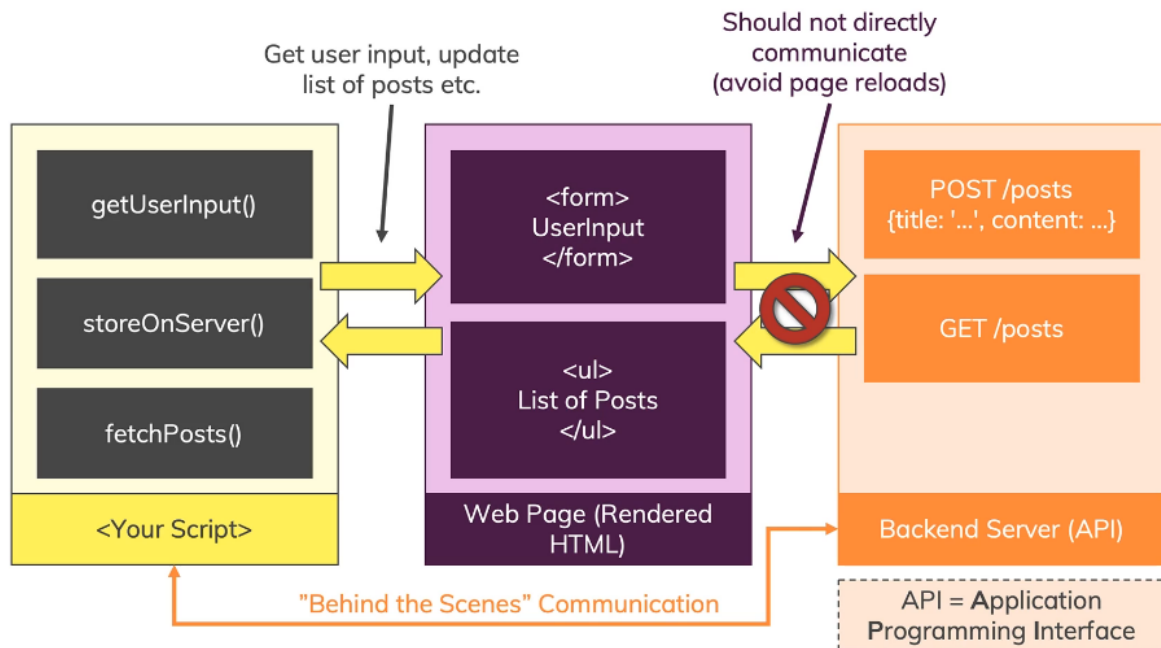Iv. Object
V. Boolean
Vi. null

JSON doesn't support

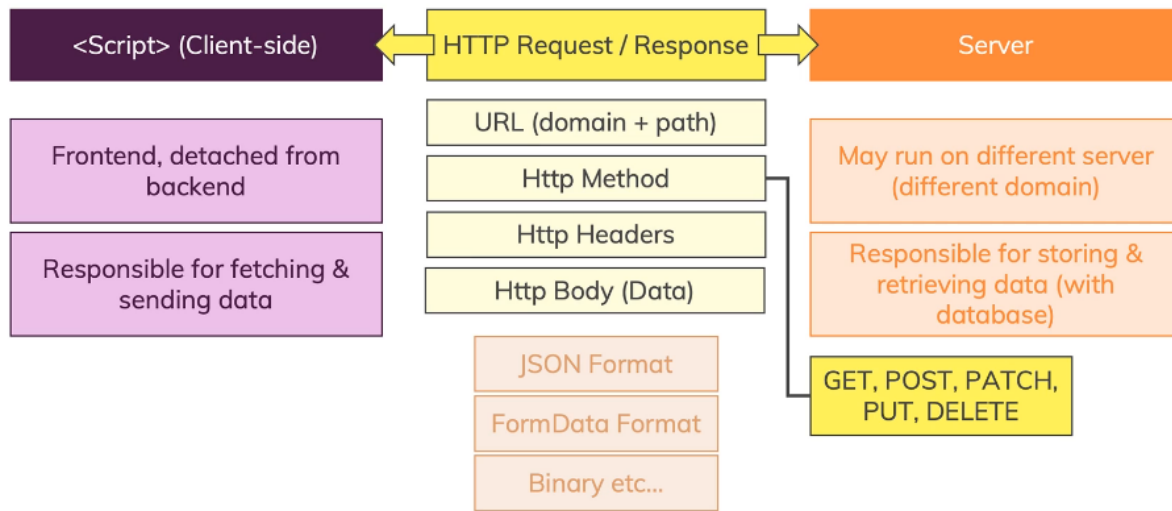I. Function and inbuilt methods.

Ii. Undefined

Example:

```
var data = {"city":{"id":1851632,"name":"Shuzenji"},
"coord":{"lon":138.933334,"lat":34.966671},
"country":"JP",
"cod":"200",
"message":0.0045,
"cnt":38,
"list":[{
        "dt":1406106000,
        "main":{
            "grnd_level":1005.93,
            "humidity":87,
            "temp_kf":0.26},
        "weather":[{"id":804,"main":"Clouds","description":"overcast
clouds","icon":"04d"}],
        "clouds":{"all":88},
        "wind":{"speed":5.71,"deg":229.501},
        "sys":{"pod":"d"},
```

```
        "dt_txt":"2014-07-23 09:00:00"}
        ]}
```

Working with Http Requests

HTTP Overview

<Script> (Client-side)	HTTP Request / Response	Server

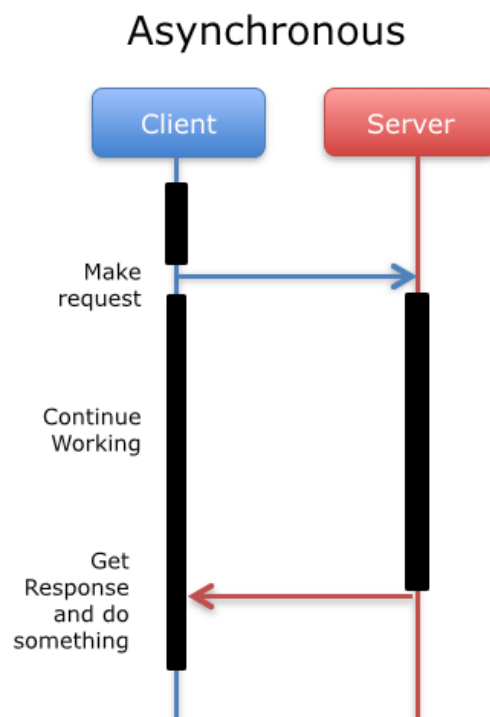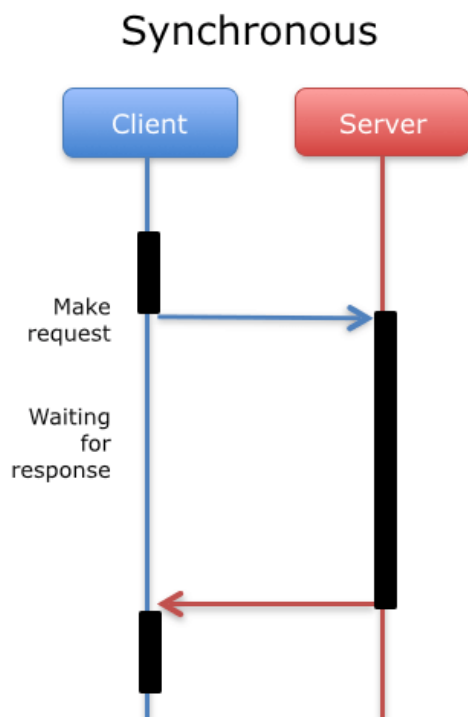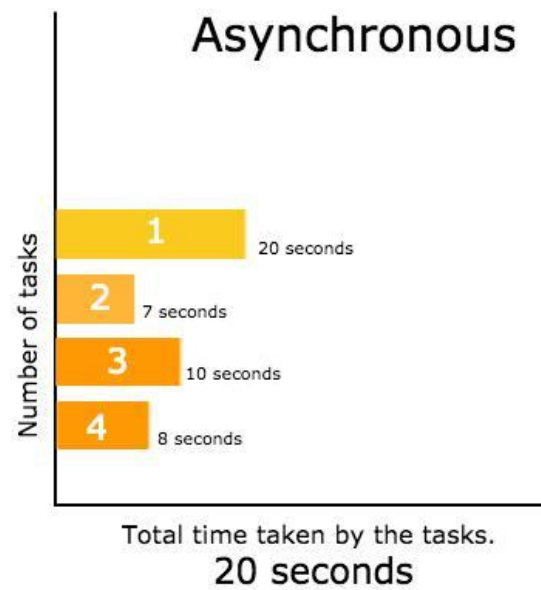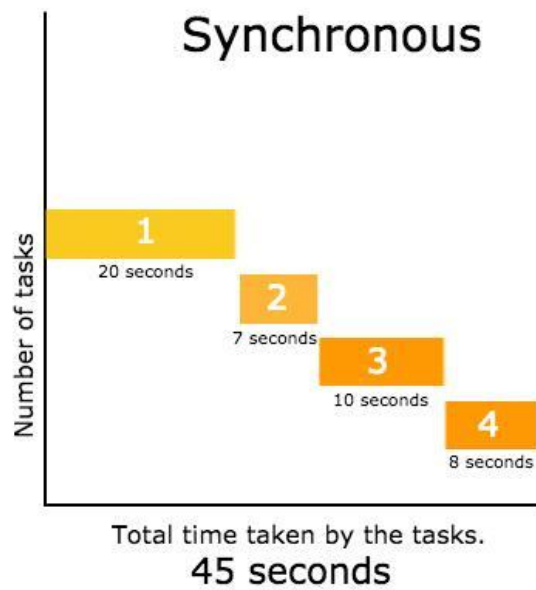| Frontend, detached from backend | URL (domain + path) | May run on different server (different domain) |
Responsible for fetching & sending data	Http Method	Responsible for storing & retrieving data (with database)
	Http Headers	
	Http Body (Data)	
	JSON Format	GET, POST, PATCH, PUT, DELETE
	FormData Format	
	Binary etc...	

AJAX

AJAX stands for Asynchronous JavaScript And XML.

Ajax used for get / post data from/to server **without reloading** the page.

Ajax uses, Browser's inbuilt Object **XMLHttpRequest**.

- The XMLHttpRequest object is used to exchange data with a server.
- Use XMLHttpRequest (XHR) objects to interact with servers.
- You can retrieve data from a URL without having to do a full page refresh.
- This enables a Web page to update just part of a page without disrupting what the user is doing.

Ajax send / receive data to / from the server by asynchronously.

Synchronous

Number of tasks

| 1 |
20 seconds

| 2 |
7 seconds

| 3 |
10 seconds

| 4 |
8 seconds

Total time taken by the tasks.
45 seconds

Asynchronous

Number of tasks

| 1 | 20 seconds

| 2 | 7 seconds

| 3 | 10 seconds

| 4 | 8 seconds

Total time taken by the tasks.
20 seconds

Synchronous

| Client | | Server |

Make
request

Waiting
for
response

Asynchronous

| Client | | Server |

Make
request

Continue
Working

Get
Response
and do
something

SPA is the inspiration of Ajax.

Ajax has 5 states,

0: request yet not initialized

1: server connection established

2: request received

3: processing request

4: request completed and response is ready

Movie Api:

http://www.omdbapi.com/?t=theri&apikey=c429066e

http://api.icndb.com/jokes/random