

Welcome to Javascript

WEB APPLICATION

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:

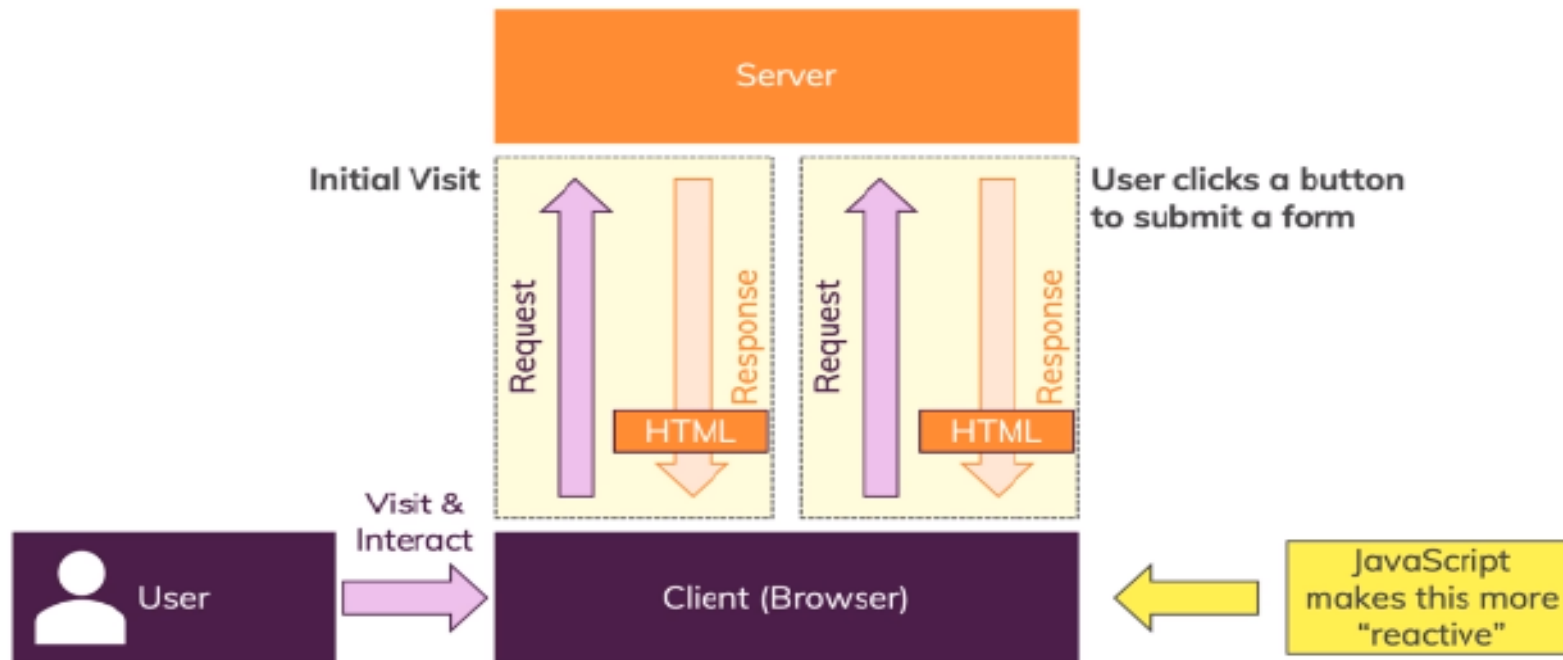
- Backend is like the brain.
- It has everything to do with the logical.
- It also takes care of data storage and management by connecting to the database.
- It can combine various services to produce the desired results.

3. Database:

- Used to store data.

HOW DO WEB PAGES WORK?

How do Web Pages Work?



HISTORY OF JAVASCRIPT

Brief Overview of the JavaScript History

1995	Netscape introduces "LiveScript" / "JavaScript"
1996	Microsoft releases its own version for IE
Late 1996	JavaScript submitted to ECMA International to start standardization
1997 - 2005	Standardization efforts, Microsoft didn't really join and support the standardized JS version though
2006 - 2011	Huge progress in JavaScript ecosystem, Microsoft eventually joined forces

What is JavaScript?

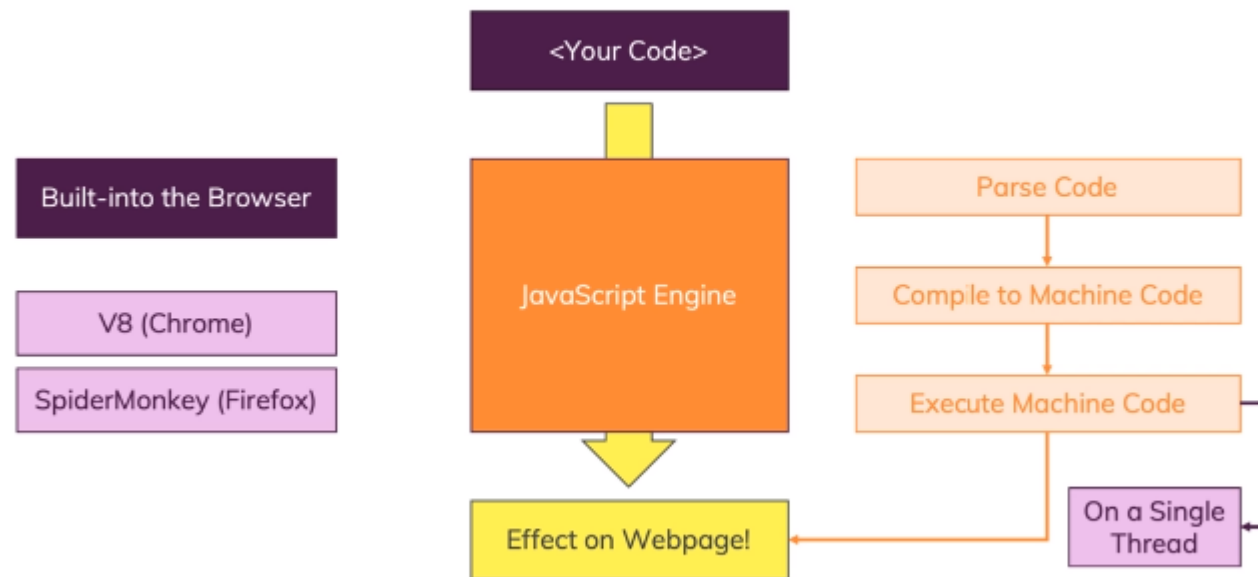
- JavaScript is a client side scripting language (interpreted programming language)
- JavaScript make web pages interactive
- Open source and cross-platform
- Case sensitive
- Most commonly used as a part of web pages
- JS was created to make web pages more Dynamic (Change content on a page directly from inside the browser)
- Supported by all major browsers and enabled by default

WHAT IS JAVASCRIPT

- JavaScript is a lightweight, cross-platform, object-oriented computer programming language
- JavaScript is one of the three core technologies of web development
- Today, JavaScript can be used in different places:
 - **Client-side: JavaScript was traditionally only used in the browser**
 - Server-side: Thanks to node.js, we can use JavaScript on the server as well
- Javascript is what made modern web development possible:
 - Dynamic effects and interactivity
 - Modern web applications that we can interact with
- **Frameworks/libraries like React and Angular are 100% based on JavaScript: you need to master JavaScript in order to use them!**

HOW JAVASCRIPT WORKS

How is JavaScript Executing?



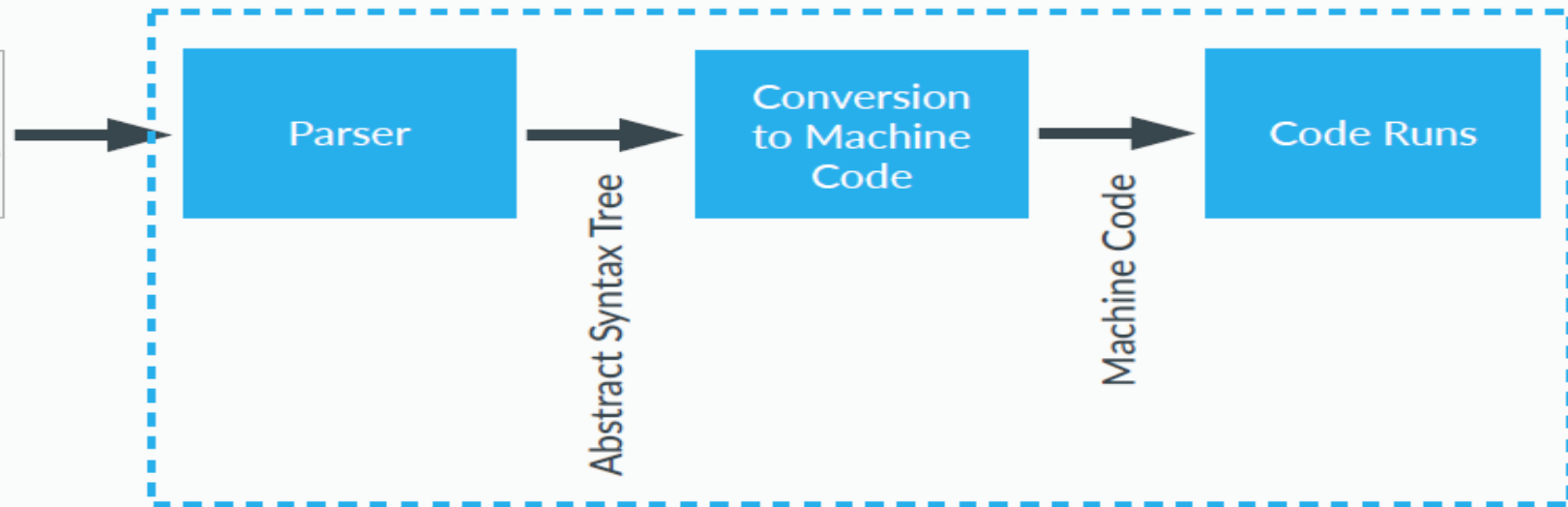
JAVASCRIPT ENGINE

WHAT HAPPENS TO OUR CODE?

OUR CODE

```
function calculateAge(yearOfBirth) {  
  return 2016 - yearOfBirth;  
}  
  
var johnAge = calculateAge(1990);  
  
function yearsUntilRetirement(name, yearOfBirth) {  
  var age = calculateAge(yearOfBirth);  
  var retirement = 65 - age;  
  if (retirement >= 0) {  
    console.log(name + ' retires in ' + retirement + ' years.');  } else {  
    console.log(name + ' is already retired.');  }  
}  
  
yearsUntilRetirement('John', 1990);
```

JAVASCRIPT ENGINE



WHAT JAVASCRIPT CAN DO?

- Add new HTML to the page, change the existing content, modify styles. (**DOM Manipulation**).
- React to user actions, Execute on mouse clicks, pointer movements, key presses. (Events)
- Send requests over the network to remote servers, read and write files (Ajax).
- Get and set cookies, ask questions to the visitor, show messages.
- Remember the data on the client-side ("local storage").

Advantages of JavaScript

- Executed on the client side.
- Instant response to the visitors.
- Rich interfaces.
- Speed.
- Less server interaction.

Disadvantages of JavaScript

- Code Always Visible.
 - **Stop Render** : JavaScript single error can stop rendering with the entire site. However browsers are extremely tolerant of JavaScript errors.
 - Less Security.
-

Why JavaScript?

- JavaScript adds behavior to web pages
- Show or hide more information with the click of a button
- Change the color of a button when the mouse hovers over it
- Less server interaction
- Immediate feedback to the visitors



**Less Server
Interaction**



**Immediate
feedback to the
visitors**



**Increased
Interactivity**



Richer Interfaces

What's the Difference?



HTML

Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



CSS

Cascading Style Sheet

Stylize the website

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- The fundamental building block of any web page



Javascript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

FEATURES



UNDERSTANDING

HTML



CONTENT

NOUNS

```
<p></p>
```

means "paragraph"

CSS



PRESENTATION

ADJECTIVES

```
p {color: red;}
```

means "the paragraph
text is red"

JS

DYNAMIC EFFECTS/
PROGRAMMING

VERBS

```
p.hide();
```

means "hide the
paragraph"

JS in Browser side vs Server side

Browser-side

JavaScript was invented to create more dynamic websites by executing in the browser!

JavaScript can manipulate the HTML code, CSS, send background Http requests & much more

JavaScript CAN'T access the local filesystem, interact with the operating system etc

"Other" (e.g. Server-side)

Google's JavaScript Engine (V8) was extracted to run JavaScript anywhere (called "Node.js")

Separate Module!

Node.js can be executed on any machine and is therefore often used to build web backends (server-side JavaScript)

Node.js CAN access the local filesystem, interact with the operating system etc. It CAN'T manipulate HTML or CSS

How to Add JavaScript

- ❖ Internal JS - Internal JavaScript code is code that's placed anywhere within the web page between the HTML tags

```
<script>  
  alert("Happy Learning");  
</script>
```

- ❖ External JS

- JavaScript code placed in a file separate from the HTML code is called external Javascript.
- External JavaScript code is written and used in the same way as internal Javascript.
- The file should have the ".js" extension.

```
<script src="myScript.js"></script>
```