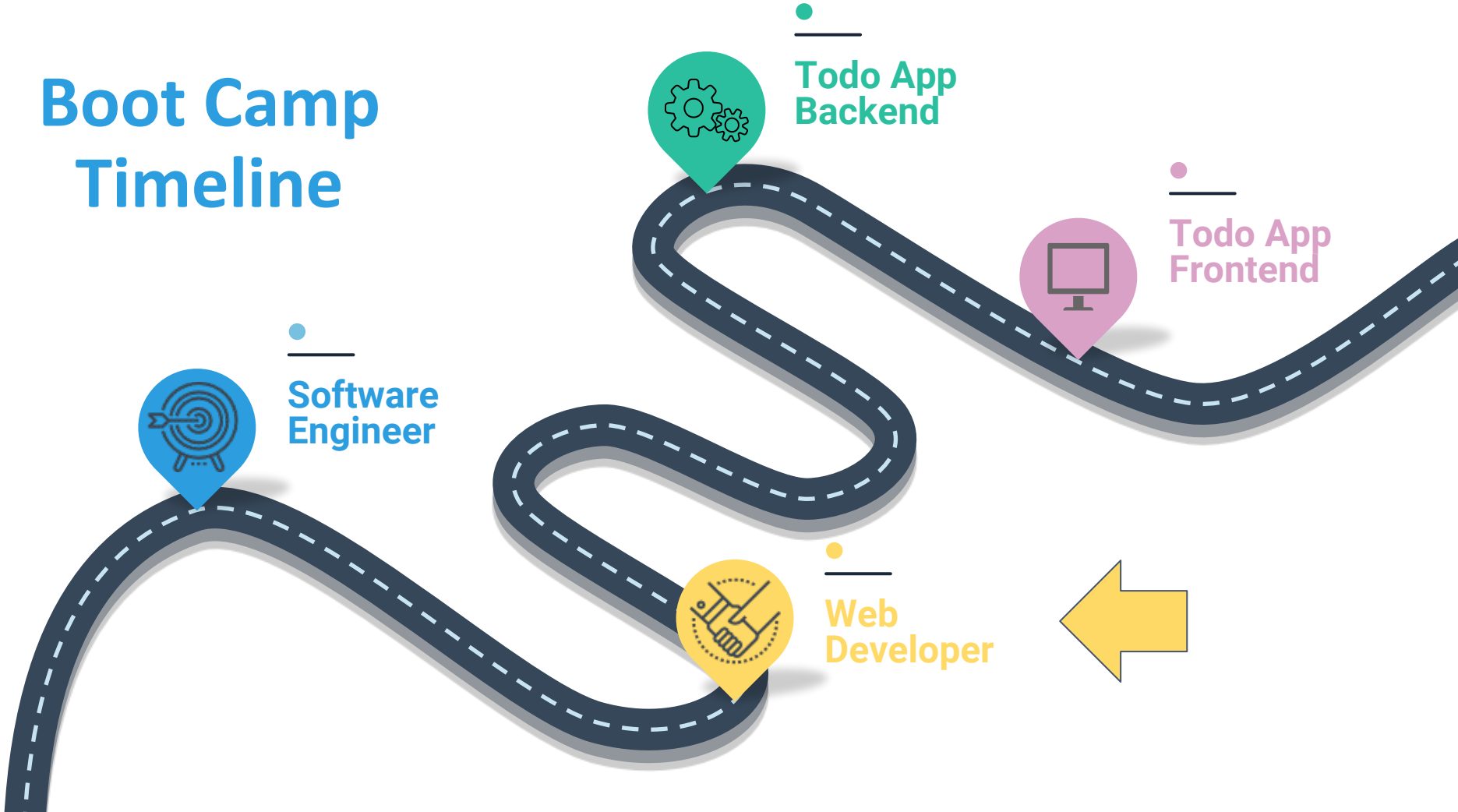
Software Engineering

Boot Camp Timeline



Plan

- Road Map
- Backend
  - Servers
  - Frameworks
  - Databases
  - Middleware
  - Languages
- Frontend
  - HTML5, CSS3, JavaScript, jQuery
  - Frameworks
  - UI Libraries
  - Tools (Gulp, ...)
  - JS & CSS preprocessors

1 Road Map

Required for any path

- Git - Version Control
- SSH
- HTTP/HTTPS and APIs
- Basic Terminal Usage
- Learn to Research
- Data Structures & Algorithms
- Character Encodings
- Design Patterns
- GitHub

Create a profile. Explore relevant open source projects. Make a habit of looking under the hood of projects you like. Create and contribute to open source projects.

Web Developer in 2018

Choose your path

- Front-end
- Back-end
- DevOps

Legends

- Personal Recommendation!
- Possibilities
- Pick any!
- Now build something

Frontend Vs Backend

Frontend

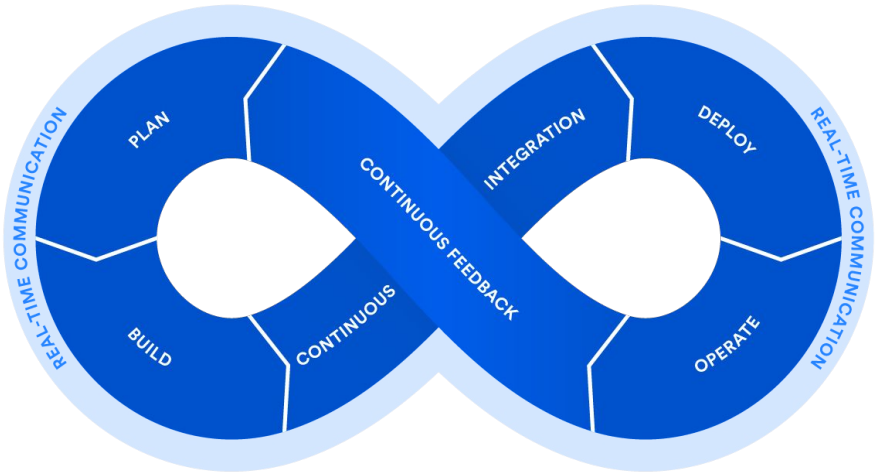
- Front end development manages everything that users visually see first in their browser or application.
- Front end developers are responsible for the look and feel of a site.

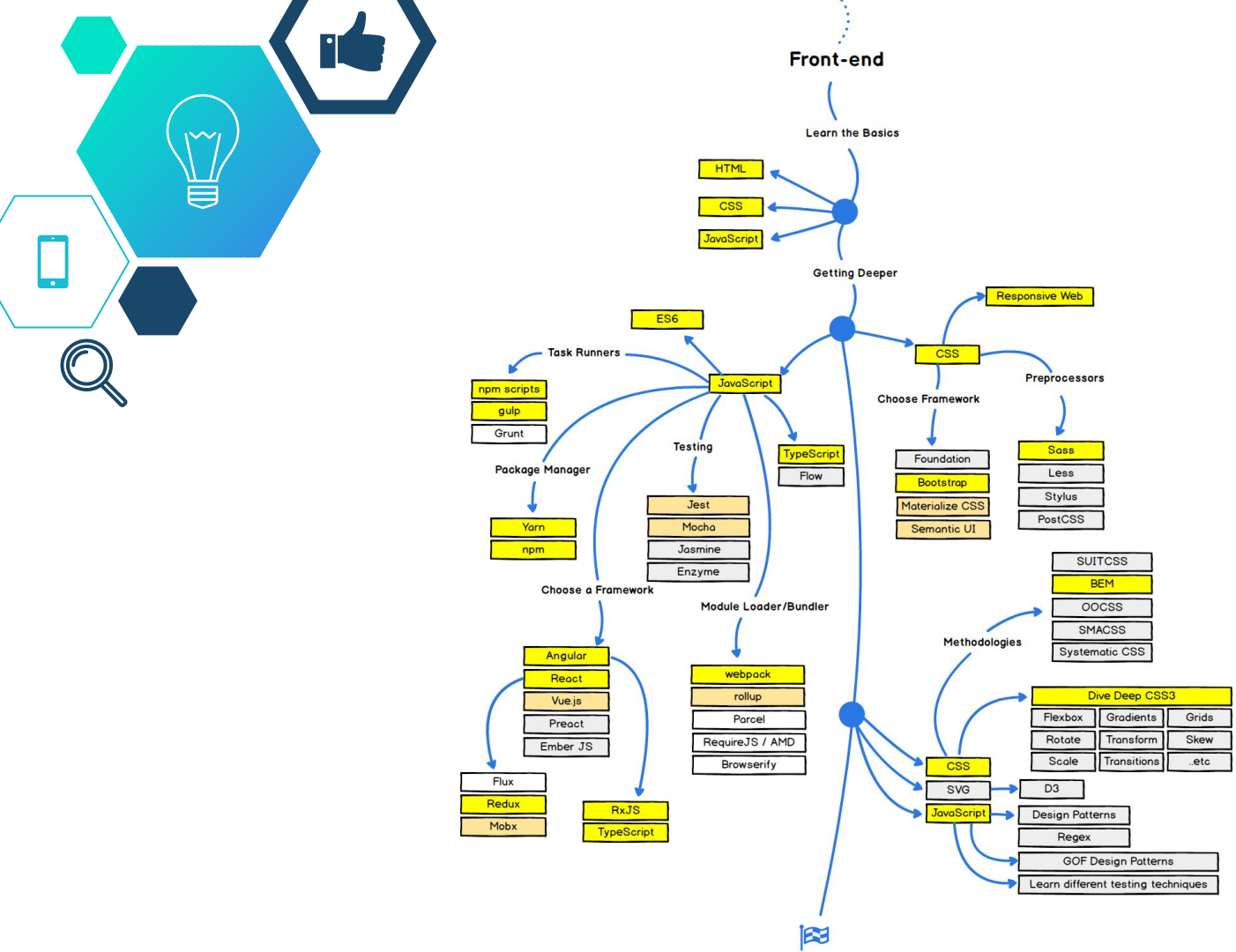
Backend

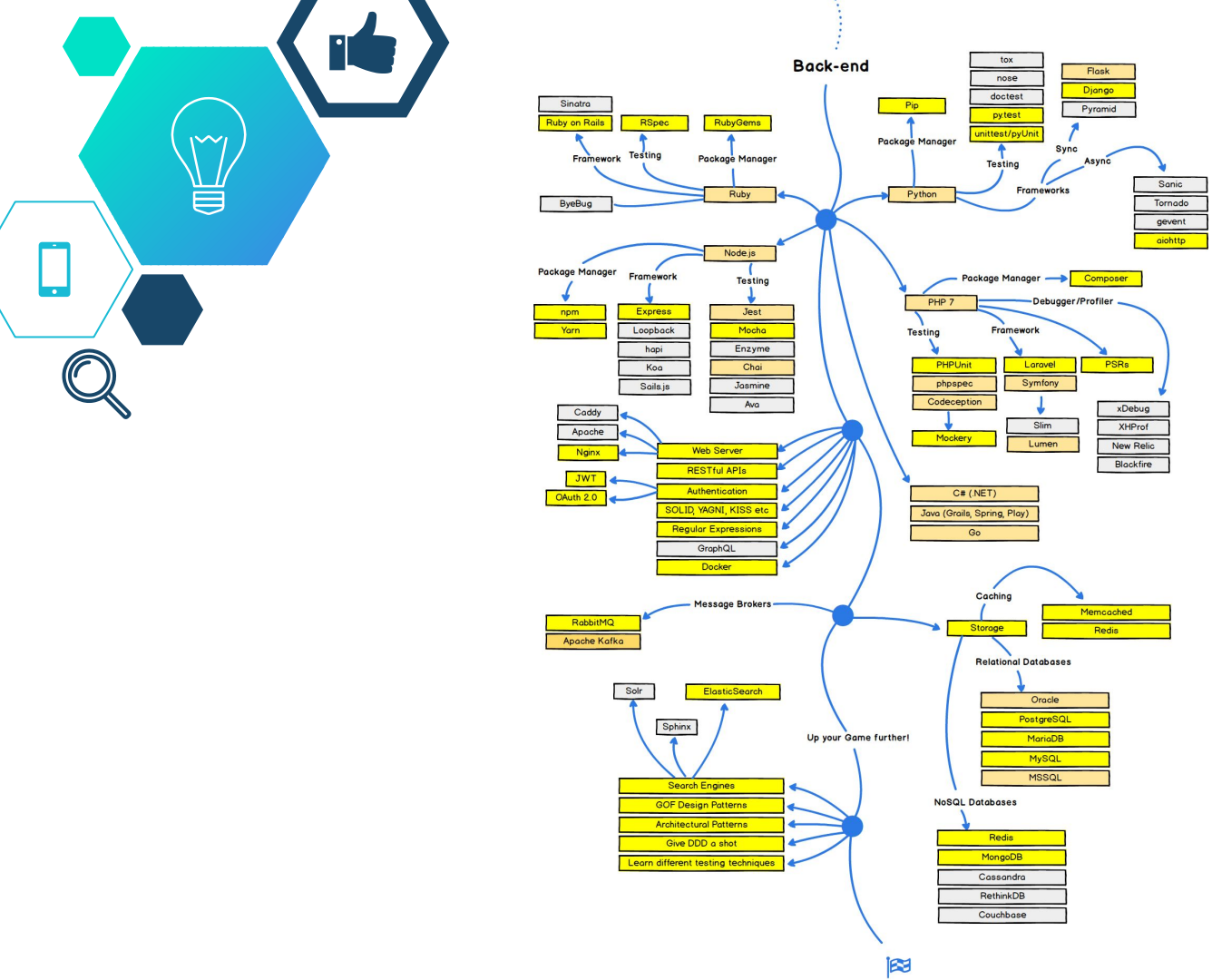
- Back end development refers to the server side of an application.
- It powers what's happening
- Handles the communication between the database and the browser.

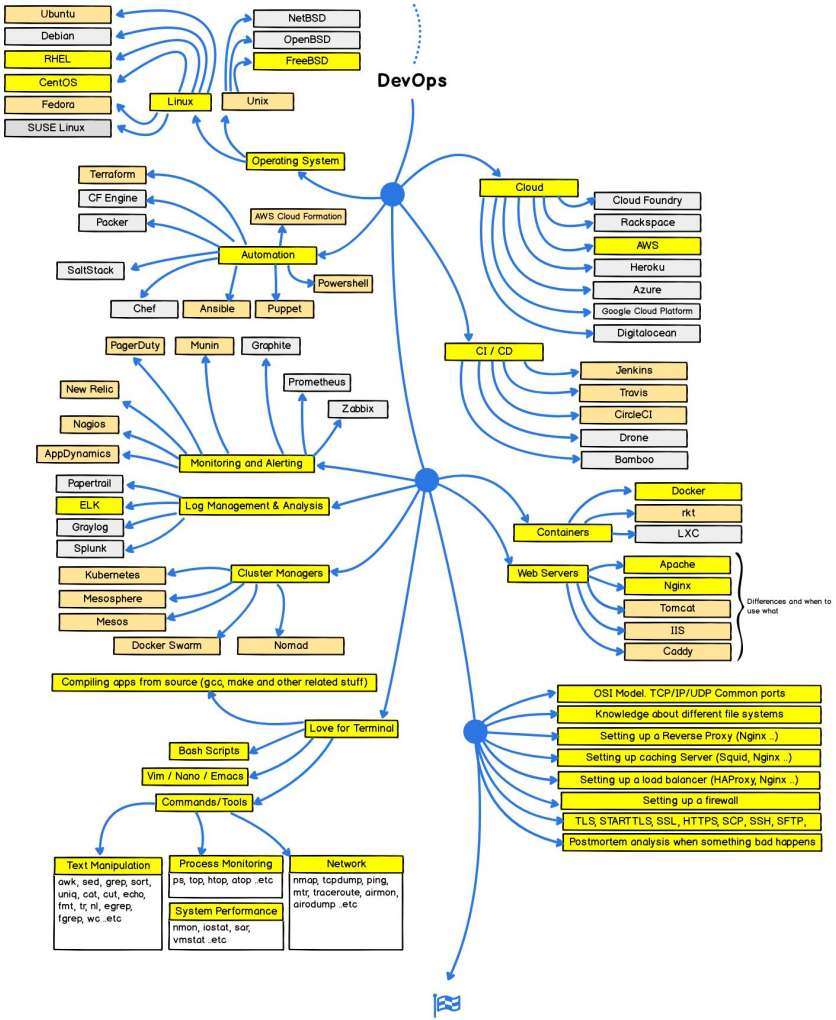
DevOps

- Automates the processes between software development and IT teams.
- To build, test, and release software faster and more




Front-end
Learn the Basics
HTML
CSS
JavaScript
Getting Deeper
ES6
Task Runners
npm scripts
gulp
Grunt
JavaScript
Package Manager
Yarn
npm
Testing
Jest
Mocha
Jasmine
Enzyme
TypeScript
Flow
Choose a Framework
Angular
React
Vue.js
Preact
Ember JS
Flux
Redux
Mobx
RxJS
TypeScript
Module Loader/Bundler
webpack
rollup
Parcel
RequireJS / AMD
Browserify
CSS
Responsive Web
Choose Framework
Foundation
Bootstrap
Materialize CSS
Semantic UI
Preprocessors
Sass
Less
Stylus
PostCSS
Methodologies
SUITCSS
BEM
OOCSS
SMACSS
Systematic CSS
CSS
Dive Deep CSS3
Flexbox
Gradients
Grids
Rotate
Transform
Skew
Scale
Transitions
..etc
SVG
D3
JavaScript
Design Patterns
Regex
GOF Design Patterns
Learn different testing techniques

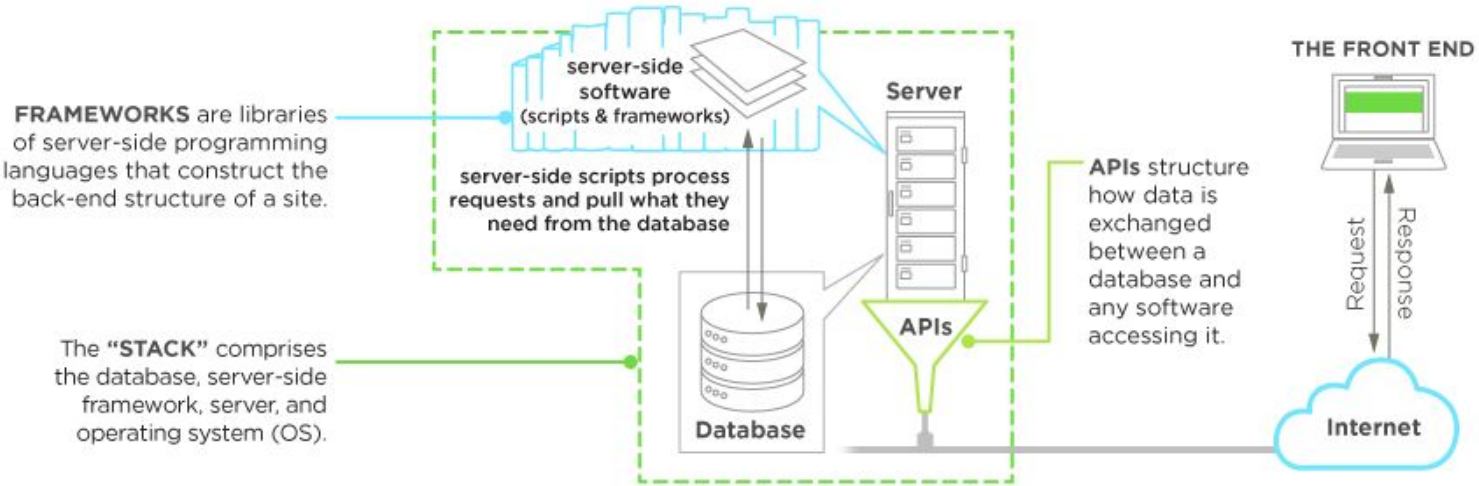

Back-end
Sinatra
Ruby on Rails
RSpec
RubyGems
Framework
Testing
Package Manager
ByeBug
Ruby
Pip
Package Manager
tox
nose
doctest
pytest
unittest/pyUnit
Testing
Flask
Django
Pyramid
Sync
Async
Frameworks
Python
Sanic
Tornado
gevent
aiohttp
Node.js
Package Manager
Framework
Testing
npm
Yarn
Express
Loopback
hapi
Koa
Sails.js
Jest
Mocha
Enzyme
Chai
Jasmine
Ava
Package Manager
Composer
PHP 7
Debugger/Profiler
Testing
Framework
PHPUnit
phpspec
Codeception
Mockery
Laravel
Symfony
Slim
Lumen
PSRs
xDebug
XHProf
New Relic
Blackfire
Caddy
Apache
Nginx
Web Server
RESTful APIs
JWT
OAuth 2.0
Authentication
SOLID, YAGNI, KISS etc
Regular Expressions
GraphQL
Docker
C# (.NET)
Java (Grails, Spring, Play)
Go
Message Brokers
RabbitMQ
Apache Kafka
Caching
Storage
Memcached
Redis
Relational Databases
Oracle
PostgreSQL
MariaDB
MySQL
MSSQL
NoSQL Databases
Redis
MongoDB
Cassandra
RethinkDB
Couchbase
Solr
ElasticSearch
Sphinx
Search Engines
Up your Game further!
GOF Design Patterns
Architectural Patterns
Give DDD a shot
Learn different testing techniques


DevOps
Ubuntu
Debian
RHEL
CentOS
Fedora
SUSE Linux
Linux
NetBSD
OpenBSD
FreeBSD
Unix
Operating System
Terraform
CF Engine
Packer
AWS Cloud Formation
Automation
SaltStack
Powershell
Chef
Ansible
Puppet
Cloud
Cloud Foundry
Rackspace
AWS
Heroku
Azure
Google Cloud Platform
Digitalocean
CI / CD
Jenkins
Travis
CircleCI
Drone
Bamboo
PagerDuty
Munin
Graphite
Prometheus
Zabbix
New Relic
Nagios
AppDynamics
Monitoring and Alerting
Papertrail
ELK
Graylog
Splunk
Log Management & Analysis
Containers
Docker
rkt
LXC
Kubernetes
Mesosphere
Mesos
Cluster Managers
Docker Swarm
Nomad
Web Servers
Apache
Nginx
Tomcat
IIS
Caddy
Differences and when to use what
Compiling apps from source (gcc, make and other related stuff)
Love for Terminal
Bash Scripts
Vim / Nano / Emacs
Commands/Tools
Text Manipulation
awk, sed, grep, sort, uniq, cat, cut, echo, fmt, tr, nl, egrep, fgrep, wc ..etc
Process Monitoring
ps, top, htop, atop ..etc
System Performance
nmon, iostat, sar, vmstat ..etc
Network
nmap, tcpdump, ping, mtr, traceroute, airmon, airodump ..etc
OSI Model. TCP/IP/UDP Common ports
Knowledge about different file systems
Setting up a Reverse Proxy (Nginx ..)
Setting up caching Server (Squid, Nginx ..)
Setting up a load balancer (HAProxy, Nginx ..)
Setting up a firewall
TLS, STARTTLS, SSL, HTTPS, SCP, SSH, SFTP,
Postmortem analysis when something bad happens

2 Backend

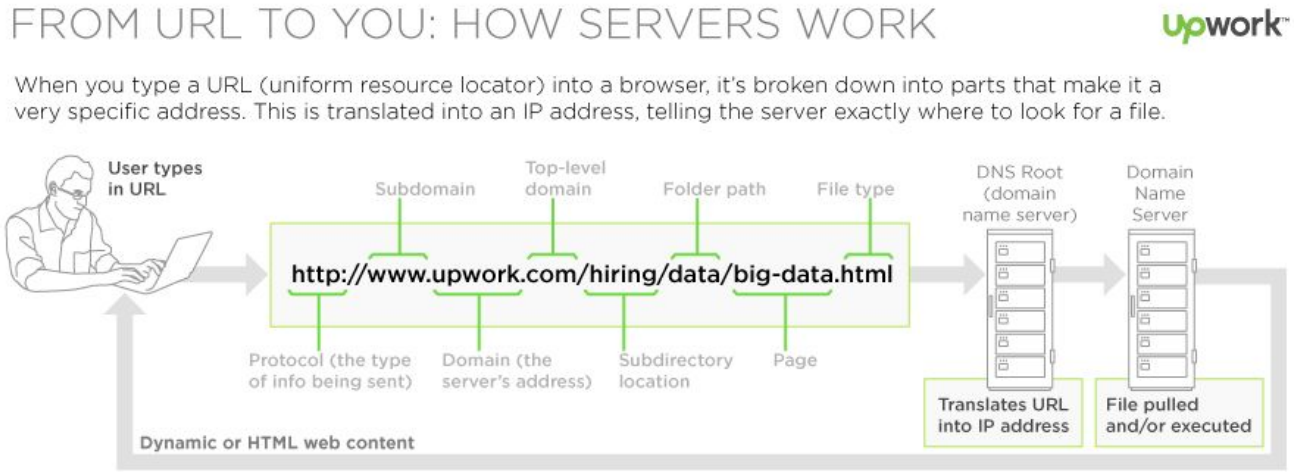
BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE

Upwork™



2.1 Servers

Servers



2.2 Languages

Backend Languages

JavaScript

Node.js

Node.js is a **JavaScript** runtime built on Chrome's V8 JavaScript engine.

Brings JavaScript to the server-side.

PHP

PHP is a server-side scripting language designed for web development.

Ruby

A dynamic, open source programming language with a focus on simplicity and productivity.

Python

Python is an interpreted high-level programming language for general-purpose programming.

2.3 Frameworks

Backend Frameworks

- Node.js
  - Express
- PHP
  - Laravel
- Ruby
  - Ruby on Rails
- Python
  - Django

Frameworks Benefits

Frameworks boost performance, extend capabilities, and offer libraries of coding shortcuts so developers aren't hand-coding web applications from the ground up.

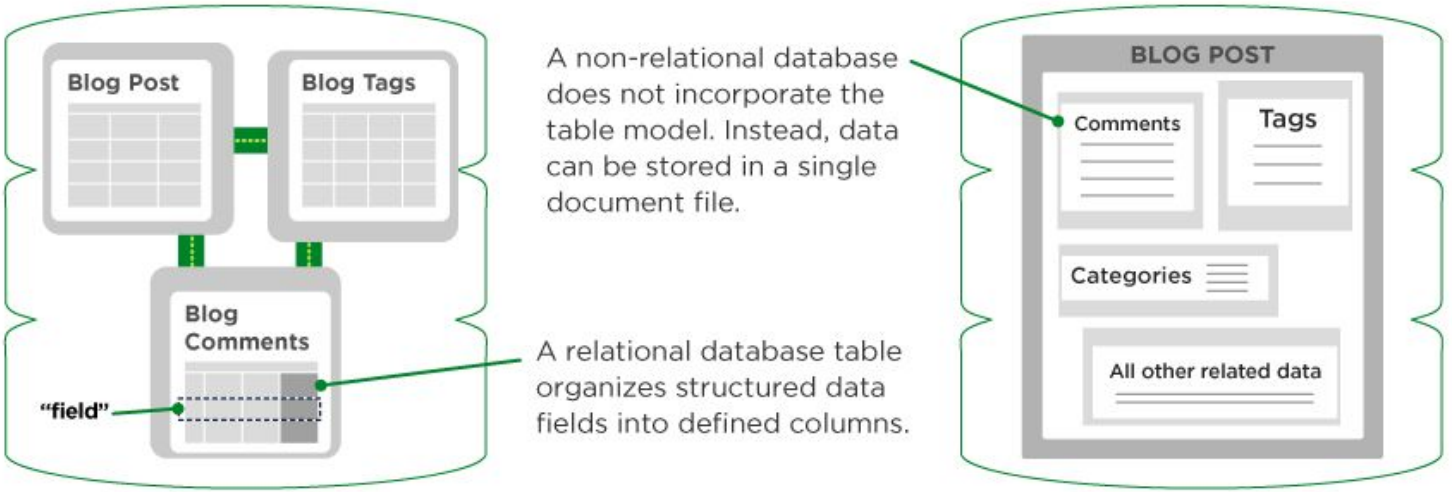
- **Libraries**: shareable, reusable bits of low-level code.
- **APIs**: facilitate access to the database back end.
- **Scaffolding**: a technique some MVC frameworks employ that strengthens how a database can be accessed.
- **Caching**, which cuts back on server workload
- **Security**, via authentication and authorization frameworks
- **Compilers**, or Just-in-Time compilers

2.4 Databases

Types of Databases

- Non-relational/NoSQL
  - MongoDB
  - Redis
  - Couch DB
- Relational Databases
  - MySQL, PostgreSQL
  - Oracle database
  - IBM
  - MariaDB

SQL Vs NoSQL

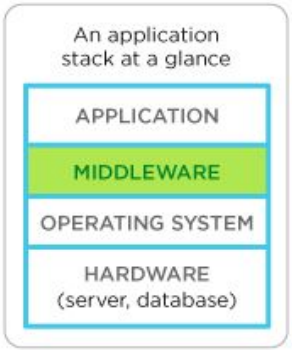


2.5 Middleware

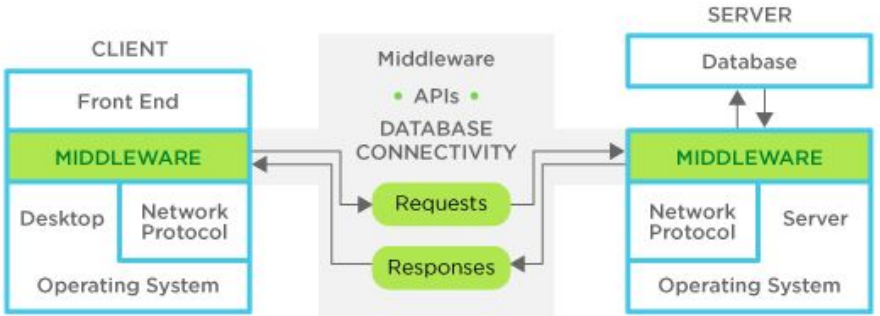
Middleware

WHAT IS MIDDLEWARE?

Upwork™



MiddleWare is any software that acts like "glue" between an application and its network. It controls the flow of information between an application and the server, database, and operating system.



3 Frontend

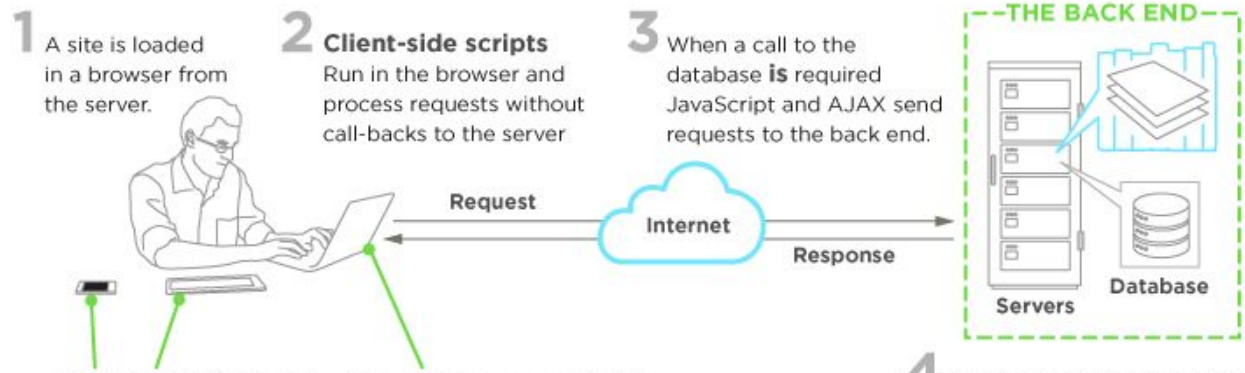
FRONT-END DEVELOPMENT

Upwork

1 A site is loaded in a browser from the server.

2 **Client-side scripts** Run in the browser and process requests without call-backs to the server

3 When a call to the database **is** required JavaScript and AJAX send requests to the back end.



5 **Server-side scripts** process the data, then update the site—populating drop-down menus, loading products to a page, updating a user profile, and more.

Responsive front-end design allows a site to adapt to a user's device.

Everything a user sees in the browser is a mix of **HTML**, **CSS**, and **JavaScript**.

4 The **back-end server-side scripts** process the request, pull what they need from the database then send it back.

3.1

HTML5, CSS3, JavaScript, jQuery

“HyperText Markup Language is what provides the structure of a website so that web browsers know what to show.”

HTML Documents

- <!DOCTYPE html> : defines the document type
- HTML document is enclosed inside <html></html> tags
- The **visible** part of the HTML document is between <body> and </body>.

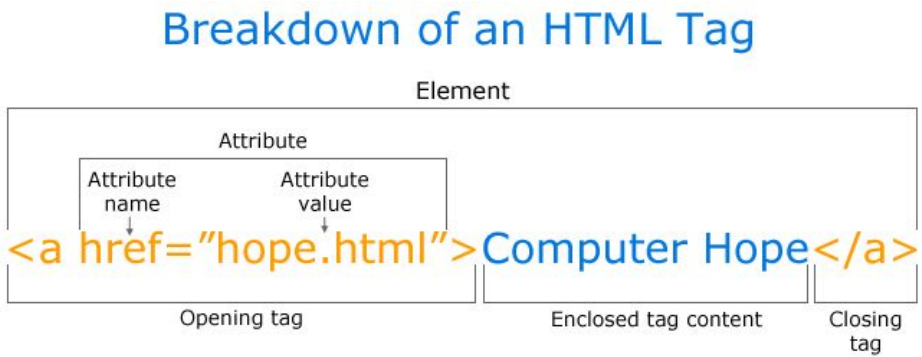
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

HTML Documents

- Elements:
  - Opening & closing tags (content goes in between)
  - Single tag (e.g. input)
- Attributes:
  - name = “value”

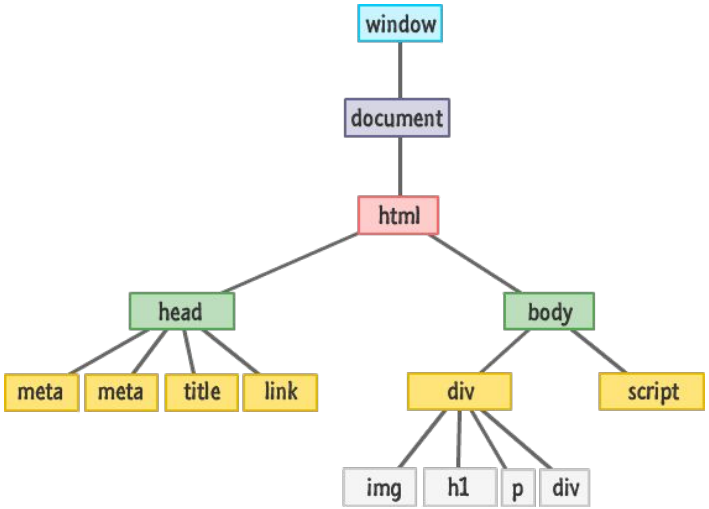


HTML Basic Tags

Tag	*Description*
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Defines information about the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph
	Defines an unordered list
	Defines a list item
	Defines a section
<img src="">	Defines an image
<input>	Defines an input (text, checkbox, select, password, file,..etc)

Document Object Model (DOM)

- On page load, browser creates a DOM.
- Object-oriented representation of the web page in a tree structure.
- Further modified by Javascript.

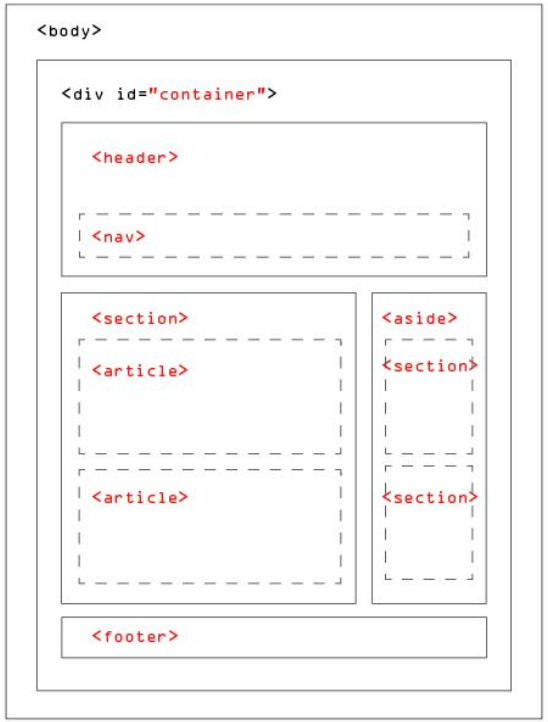


HTML5

- Latest evolutionary version of HTML
- Semantics
- Multimedia
- Offline Storage
- Graphics (Canvas, SVG and 3D)

HTML5 (Semantics)

- Describing page content precisely
- More readability, Screen readers support
- Common tags:
  - <header>, <footer>
  - <section>, <article>
  - <nav>, <aside>
  - <main>



HTML5 (Multimedia)

- Manipulation of multimedia content.
- Common tags:
  - <audio>
  - <video>

3.2 CSS

“Cascading Style Sheet is used to style and layout web pages — for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features.”

HTML with/without CSS

My First Website

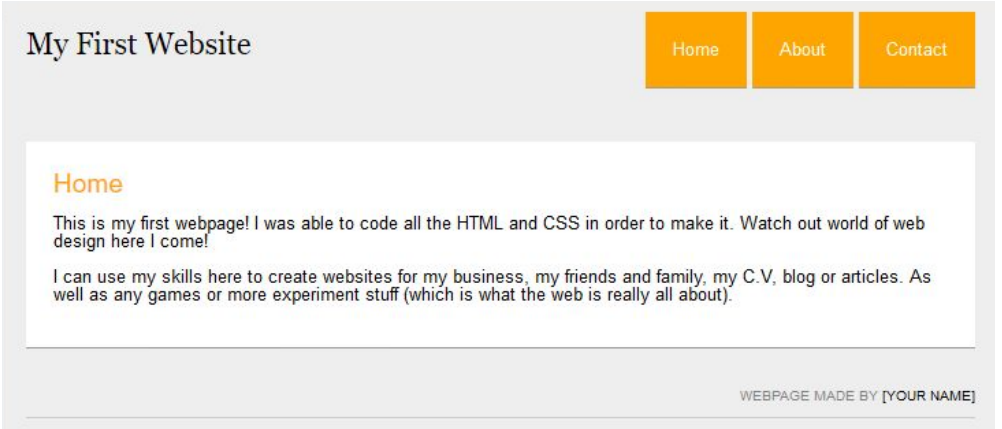
- Home
- About
- Contact

Home

This is my first webpage! I was able to code all the HTML and CSS in order to make it. Watch out world of web design here I come!

I can use my skills here to create websites for my business, my friends and family, my C.V, blog or articles. As well as any games or more experiment stuff (which is what the web is really all about).

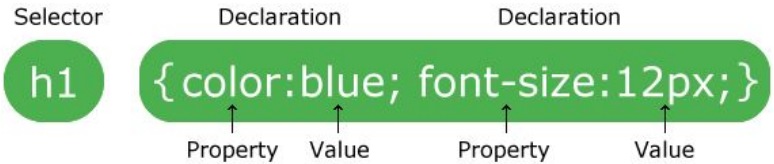
Webpage made by [your name]



CSS rule-set

A CSS rule-set consists of a selector and a declaration block enclosed by curly brackets

- The selector points to the HTML element you want to style.
- Declarations are property-value pairs separated by semicolons.




```
p {
    color: red;
    text-align: center;
}
```

Selectors

HTML elements can be selected through different ways by:

- **Tag**: selecting an element **directly**
- **Class (.)**: selecting **multiple** elements
- **Id (#)**: selecting single **specific** element

```html
<body>
    <section id="todos-section">
        <h1>Todos Section</h1>
        <button class="custom-button">Button 1</button>
        <button class="custom-button">Button 2</button>
        <button class="custom-button">Button 3</button>
    </section>
</body>
```

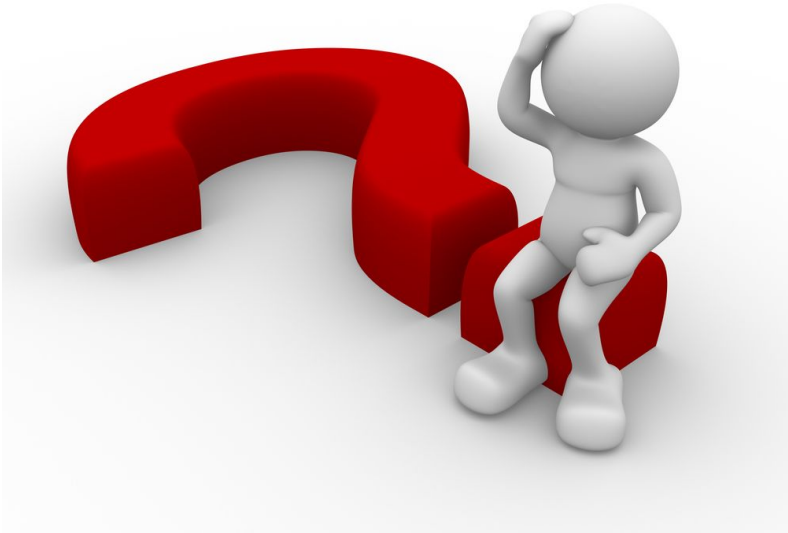
```css
/* tag selector */
h1{
    margin: 0;
}

/* class selector */
.custom-button{
    background: #f8f8f8
}

/* id selector */
#todos-section{
    width: 800px;
    height: 600px;
}
```

Where can we write CSS?

- Inline
- Style tag
- External

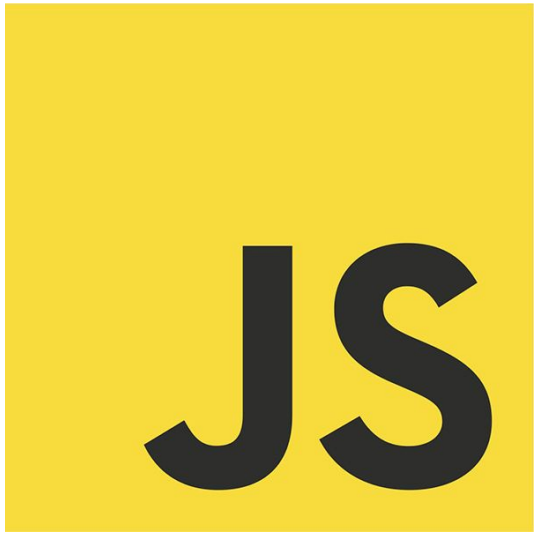


JavaScript

“JavaScript is what makes webpages alive.”

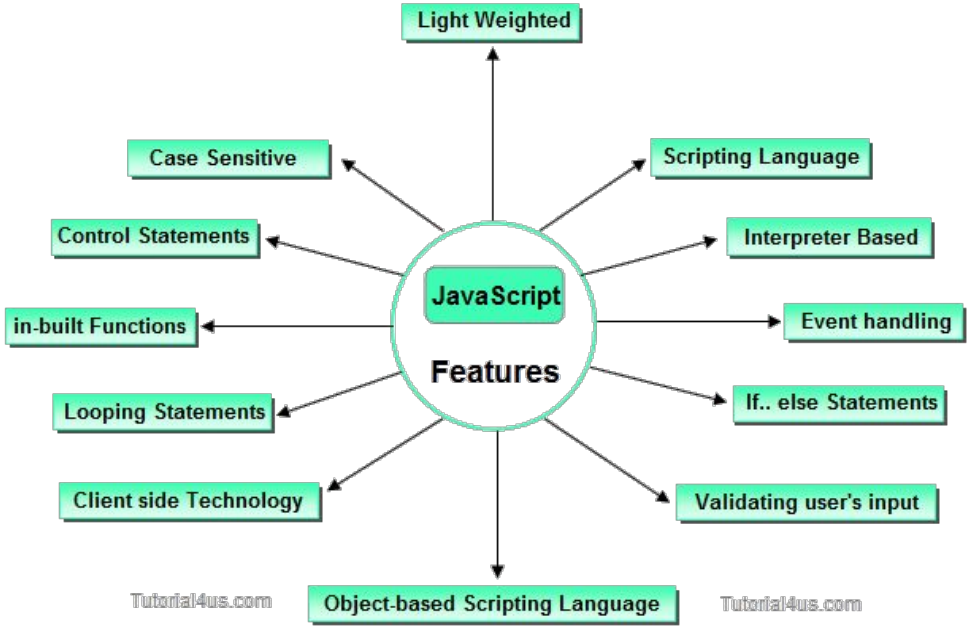
What's JavaScript?

- High-level, interpreted scripting language.
- Released in 1995 by NetScape (aka Mozilla nowadays)
- Developed by **Brendan Eich** in 10 days.



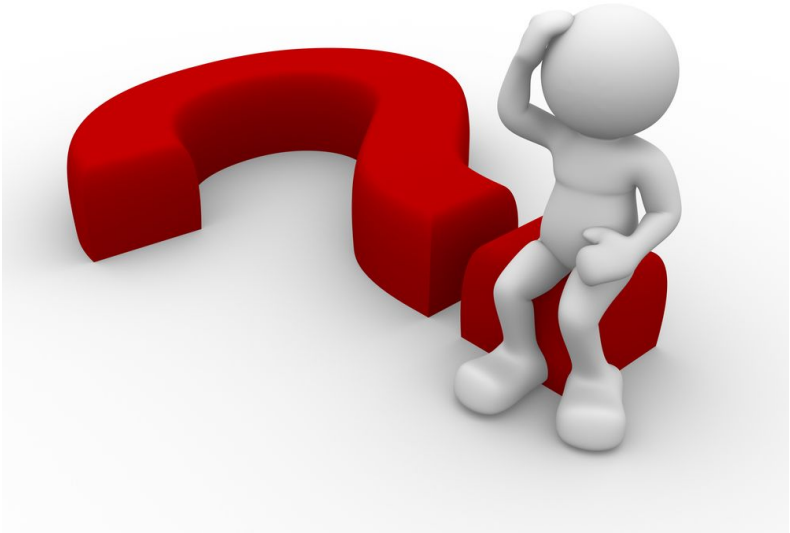
JavaScript Features

- Event-driven:
  - User actions
  - Page loading
  - Timeout
- DOM manipulation
- AJAX requests
- Browser access (e.g. local storage)



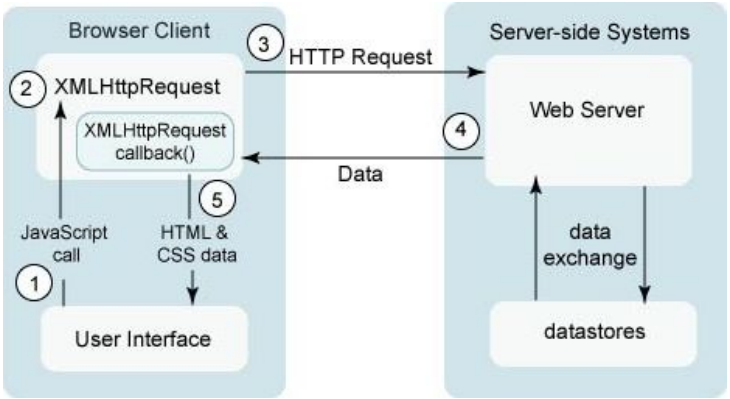
Where can we write Javascript?

- Internal
- External



AJAX Requests

- Stands for **Asynchronous Javascript and XML**
- Async way of refreshing certain parts after page load.
- Different formats (e.g JSON, HTML, txt)



JQuery

- Javascript **library**
- Released in 2005
- Makes event handling, DOM manipulation and AJAX much easier.
- Cross-platform support across different browsers.

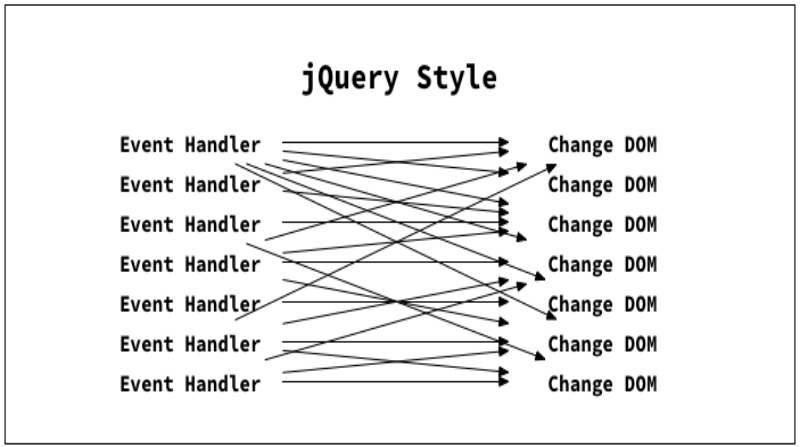
ECMAScript

- A **specification/ standard** that Javascript follows and implements.
- Every update makes life much easier
- The most supported version in all modern browsers is ES5 (2009)

3.4 Frameworks

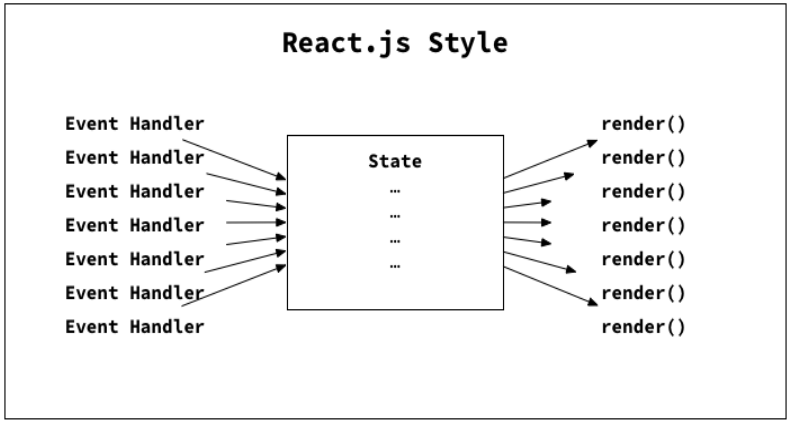
Spaghetti Code

- Write Everything Twice (WET)
- Tightly coupled
- Doing too many things at once
- Massive block of code
- Afraid to touch
- Testing isn't an option
- No standard way of coding



Rise of Frameworks

- Don't Repeat Yourself (DRY): using **components**
- Loosely Coupled: changes according to **state**
- Separation of concerns
- **Maintainable** code
- Safe to modify
- Testing is a lot easier
- Offers a **unified standard** or a direction of thinking to follow.
- Routing



Popular Frameworks

Vue.js

React

ember

BACKBONE.JS

ANGULAR

3.5 UI Frameworks

Popular UI Frameworks

- Bootstrap.
- Semantic-UI.
- Foundation.
- Materialize.
- Material UI.

3.6 Tools

Task Runners

- **Automation**. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc,
- Popular ones:
  - Grunt
  - Gulp

JS & CSS preprocessors

JS Preprocessors

- **Supersets** of Javascript.
- **Compiles** to Javascript running in the browser.
- Popular ones:
  - Babel
  - TypeScript

CSS Preprocessors

- Makes it easier to write CSS
- Nesting (e.g. style li inside ul)
- Variables (e.g. storing global styles)
- Common ones:
  - SASS
  - LESS

Thanks!

Any questions?

You can find me at:

- Github: https://github.com/MoAgamia/SE-Boot-Camp

References

- https://github.com/kamranahmedse/developer-roadmap
- https://www.upwork.com/hiring/development/a-beginners-guide-to-back-end-development/
- https://www.upwork.com/hiring/development/understanding-software-frameworks/
- https://www.atlassian.com/devops