

JavaScript for Beginners

le wagan 



Rui Freitas

Teacher @ Le Wagon

 linkedin.com/in/freitasrui



github.com/rodloboz



le wagon

Change your life,
Learn to code



le wagon

Bring technical skills
to creative people

#1

Coding Bootcamp

3710+

Graduates

1010+

Web Apps Developed



**30
Cities**

Let's talk about JavaScript

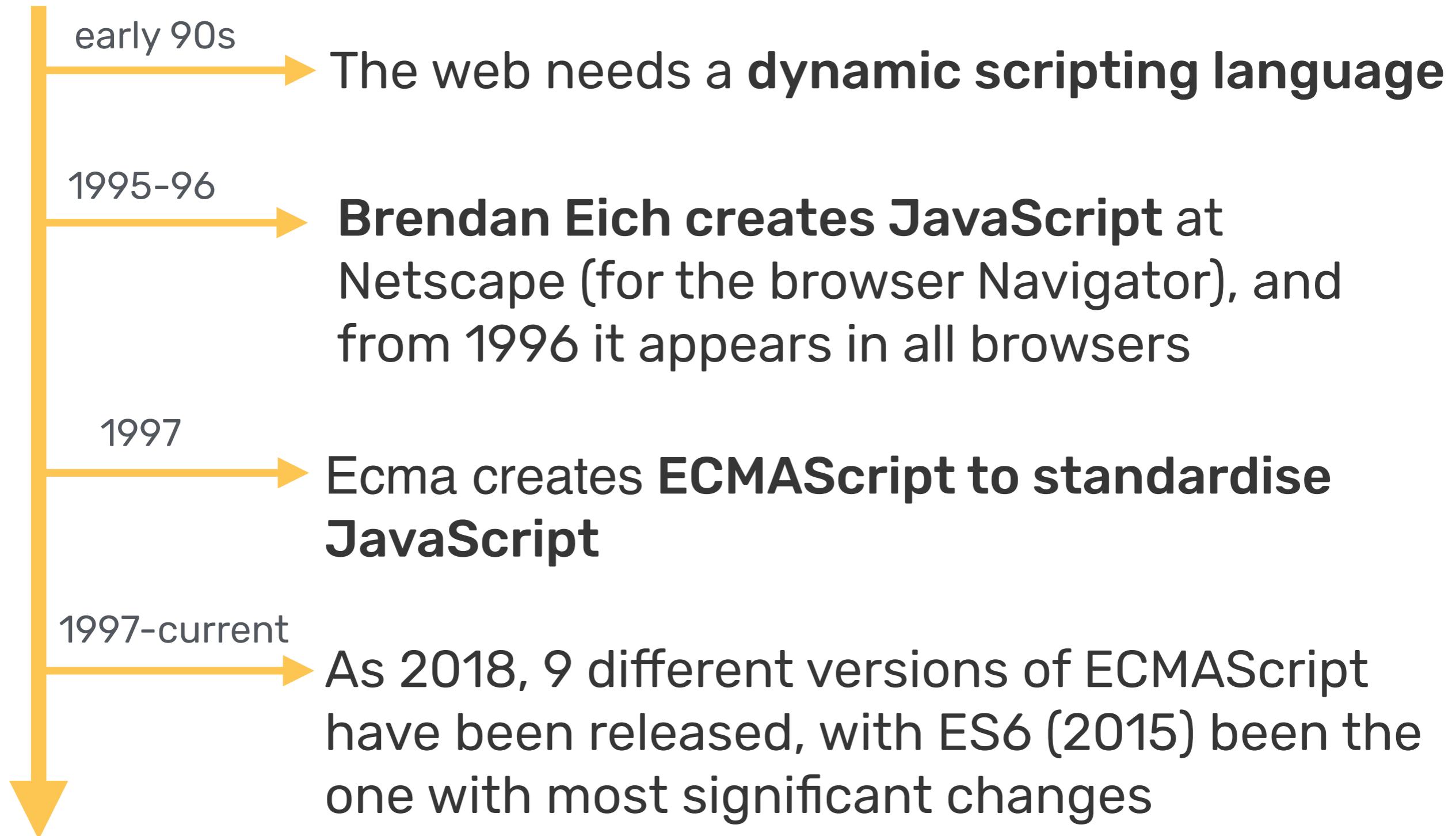
Today's Goals:

<https://github.com/rodloboz/js-beginners>

At the end of this workshop, you should be able to:

1. Understand the concept of **variables**
2. Be familiar with basic **data types**
3. Use **conditions** to control the program
4. Use **loops** to repeat instructions
5. Define and call **functions**
6. Perform basic **DOM** manipulation

The in-browser programming language



Coding Bootcamp Le Wagon | ✅

Gabriele

Secure | https://www.lewagon.com

CITIES | PROGRAM | FAQ | BLOG

DEMODY | ALUMNI | BUSINESS

Apply Now

Le Wagon - Coding Bootcamp

Change your life: learn to code

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels ▾ Group similar

```
> alert("Hello!");
< undefined
> const applyButton = document.querySelector(".navbar-wagon-link-cta");
< undefined
> applyButton.style.background = "#CE373D";
< "#CE373D"
> applyButton.style.color = "white";
< "white"
>
```

Development

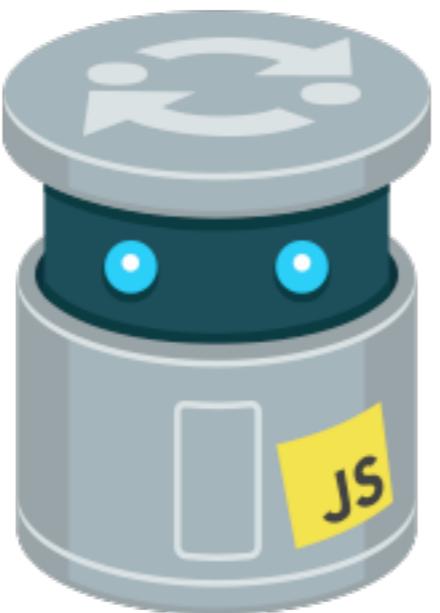


Sublime Text



Node.js

Demo



JS Bin

Data types

typeof

String

"John"
'John Lennon'
`à`
'''

Numbers

-100

0

1

3.14

42

Boolean

true

false

Array

Indexes 0 1 2 3
↑ ↑ ↑ ↑

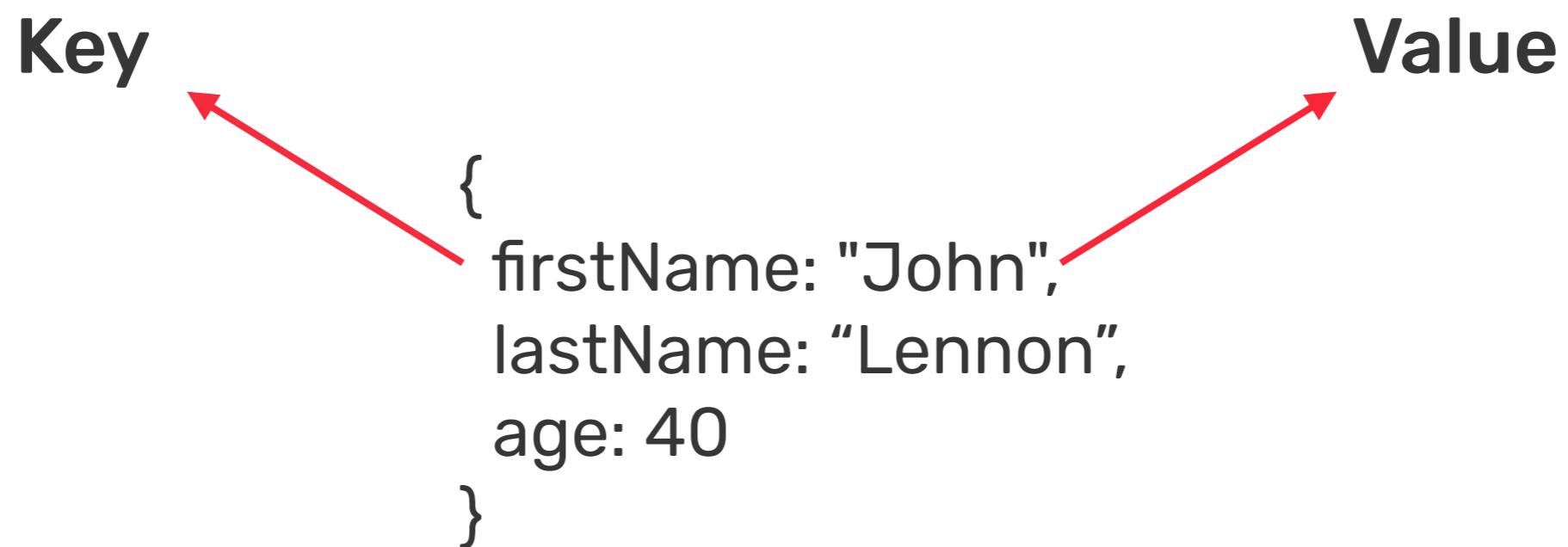
["John", "Paul", "Ringo", "George"]

[1, 2, 3, 4]

[1, "John", 2, "Paul"]

[1, true, "John", false, 2]

Object



null
(assignment value)

undefined
(unassigned value)

Variables

`var` (old ES versions)

`const, let` (ES6+)

The variable **won't be reassigned**

The value **can change**

during the execution of the code

const firstName;



```
const firstName = "Beatrix";
```



```
console.log(firstName);
```



String Concatenation

(joining two strings together)

```
// Old syntax:  
‘Hello’ + ‘world’;
```

ES6

```
// New syntax: Template literals  
`Hello world`;
```

String Interpolation

(embedding expressions in strings)

```
const name = "Beatrix";
```

```
// Old syntax:  
var greetings = "Hello" + name;
```

ES6

```
const greetings = `Hello ${name}`;
```

```
console.log(greetings);
```

Exercise 1

Concatenate two strings
Interpolate two strings

Conditions

if, else if, else



```
if (weather === "rainy") {  
    console.log("Take an umbrella!");  
}
```



```
if (weather === "rainy") {  
    console.log("Take an umbrella!");  
} else {  
    console.log("Just go out!");  
}
```



```
if (weather === "rainy") {  
    console.log("Take an umbrella");  
} else if (weather === "stormy") {  
    console.log("Stay at home!");  
} else {  
    console.log("Just go out!");  
}
```

Boolean algebra

`&&, ||, !`

Logical and

```
const result = a && b;
```

a	b	result
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

```
let rain = true;  
let wind = true;  
  
if (rain && wind) {  
    console.log("Really stay at home!");  
}
```

Logical or

```
const result = a || b;
```

a	b	result
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

```
let day = "Saturday";  
  
if (day === "Saturday" || day === "Sunday") {  
    console.log("It's the week-end!");  
}
```

Logical not

```
const result = !a;
```

a	result
TRUE	FALSE
FALSE	TRUE

```
let sunny = true;  
  
if (!sunny) {  
  console.log("Cancel the beach");  
}  
  
```

```
let weather = "raining";  
  
if (weather != "sunny") {  
    console.log("Cancel the beach");  
}
```

Arrays

Used for lists of things



Example

```
const beatles = ["John", "Paul", "George", "Ringo"];
```

```
beatles.length // => 4
```

beatles[0]	// => "John"
beatles[1]	// => "Paul"
beatles[2]	// => "George"
beatles[3]	// => "Ringo"

Loops

for and forEach (ES6+)
while

for and forEach

```
for (let i = 0; i < beatles.length; i += 1) {  
    console.log(beatles[i]);  
}
```

ES6

```
beatles.forEach((beatle) => {  
    console.log(beatle);  
})
```

Exercise 2

Filter names starting with a "B"

Exercise

Filter names starting with a "B"

```
const names = ["Ben", "Charlie", "Beatrix"];
let bNames = [];

names.forEach((name) => {
  if (name.charAt(0) === 'B') {
    bNames.push(name);
  }
})

console.log(bNames);
```

While

```
const names = ["Ben", "Charlie", "Beatrix"];
let i = 0;
while (i < names.length) {
    console.log(names[i]);
    i += 1;
}
```

Functions

function

+

arrow function (ES6+)

Anonymous functions

```
function (parameters) {  
    body  
}
```

Named functions

```
function name(parameters) {  
    body  
}
```

Arrow functions

(ES6+ only)

ES6

```
(parameters) => {  
  body  
}
```

Reusing arrow functions

(ES6+ only)

ES6

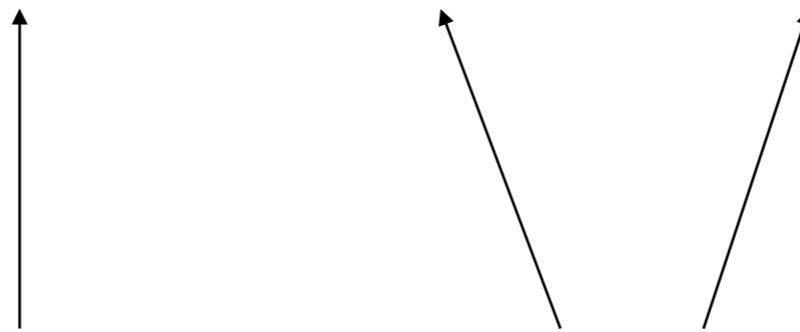
```
const name = (parameters) => {  
  body  
}
```

Example: Full Name

```
const fullName = (firstName, lastName) => {  
  return `${firstName} ${lastName}`;  
};  
console.log(fullName('Beatrix', 'Kiddo'));
```

Calling a function

```
fullName('Beatrix', 'Kiddo');
```



Function name Arguments
(or variable name in
arrow functions)

Exercise 3

**Filter names starting with "B"
starting with "C"**

...

**How can we dynamically do it
with a function?**

Exercise

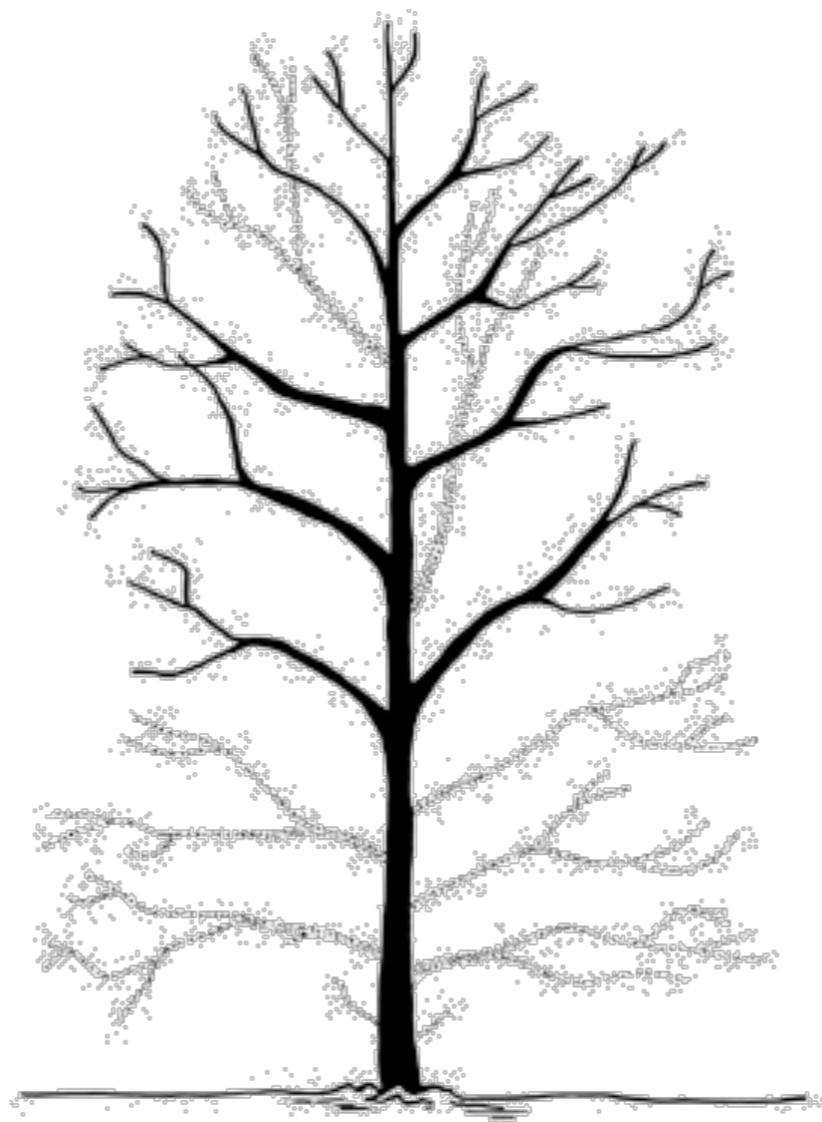
```
const filterNames = (names, letter) => {
  const filteredNames = [];

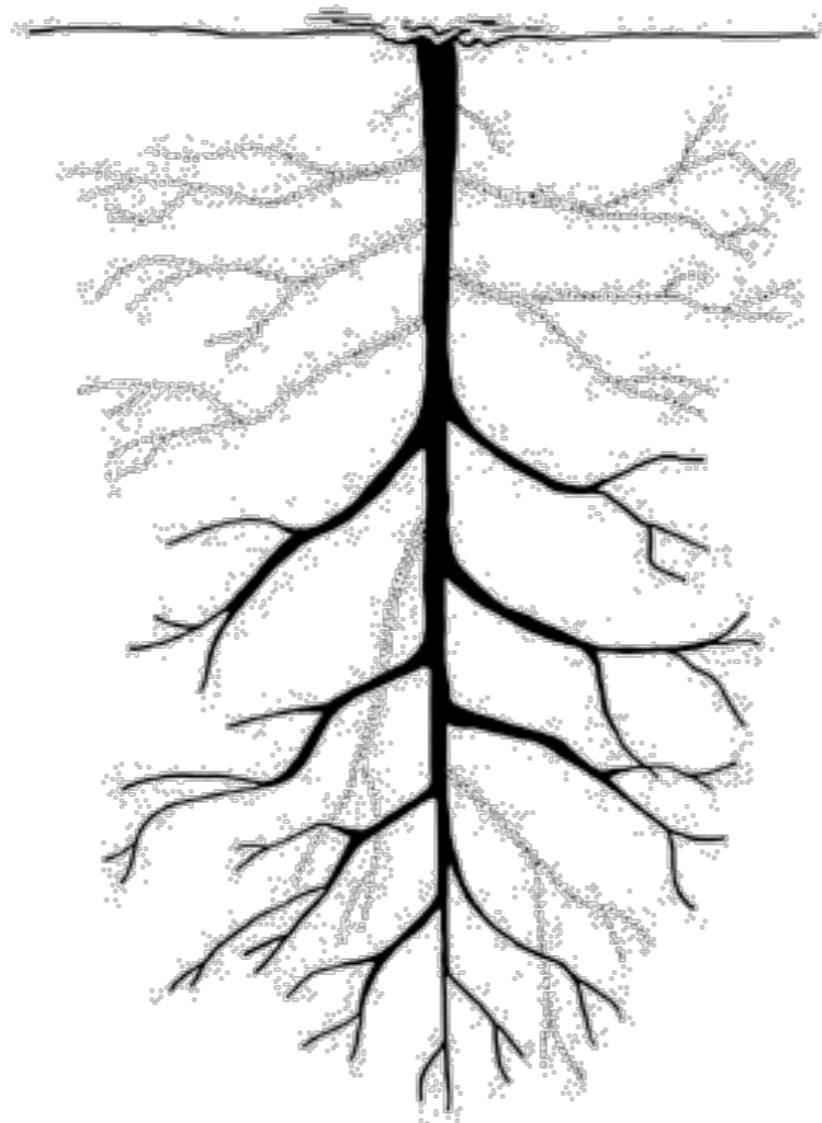
  names.forEach((name) => {
    if (name.charAt(0) === letter) {
      filteredNames.push(name);
    }
  });

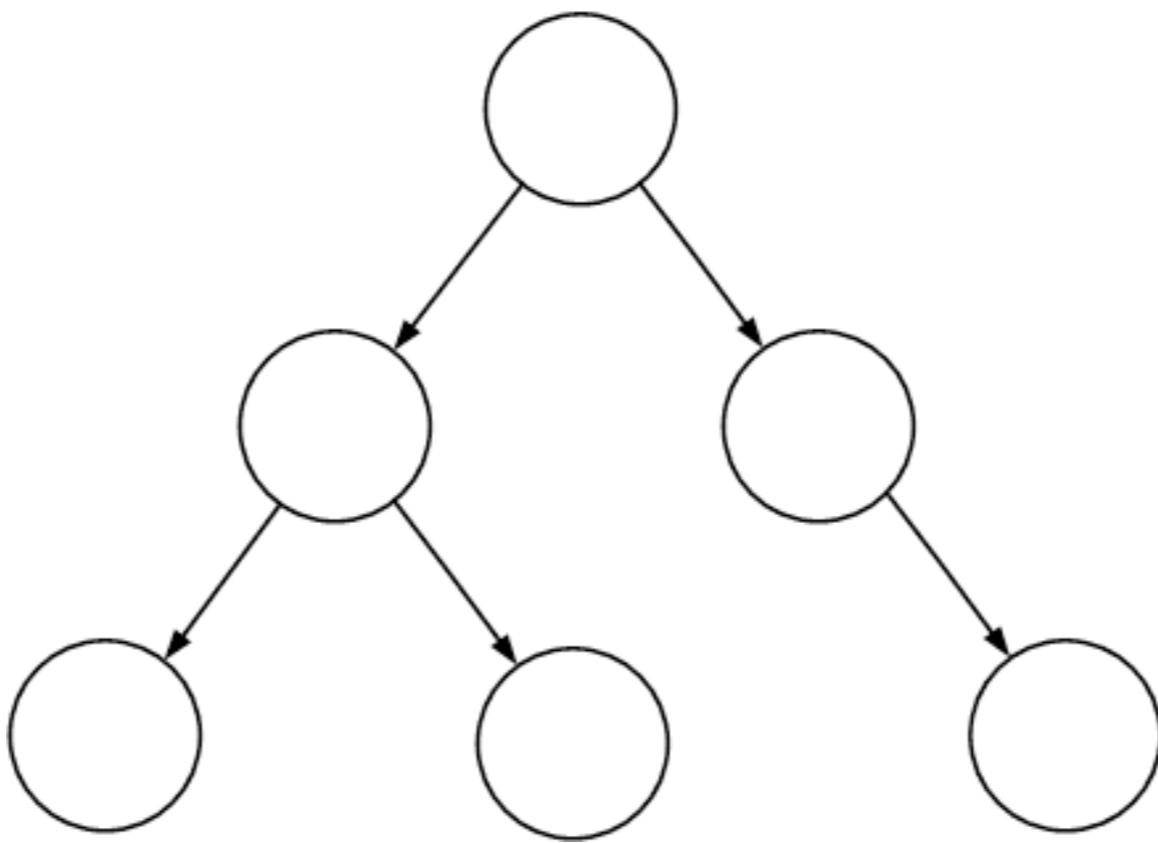
  return filteredNames;
};
```

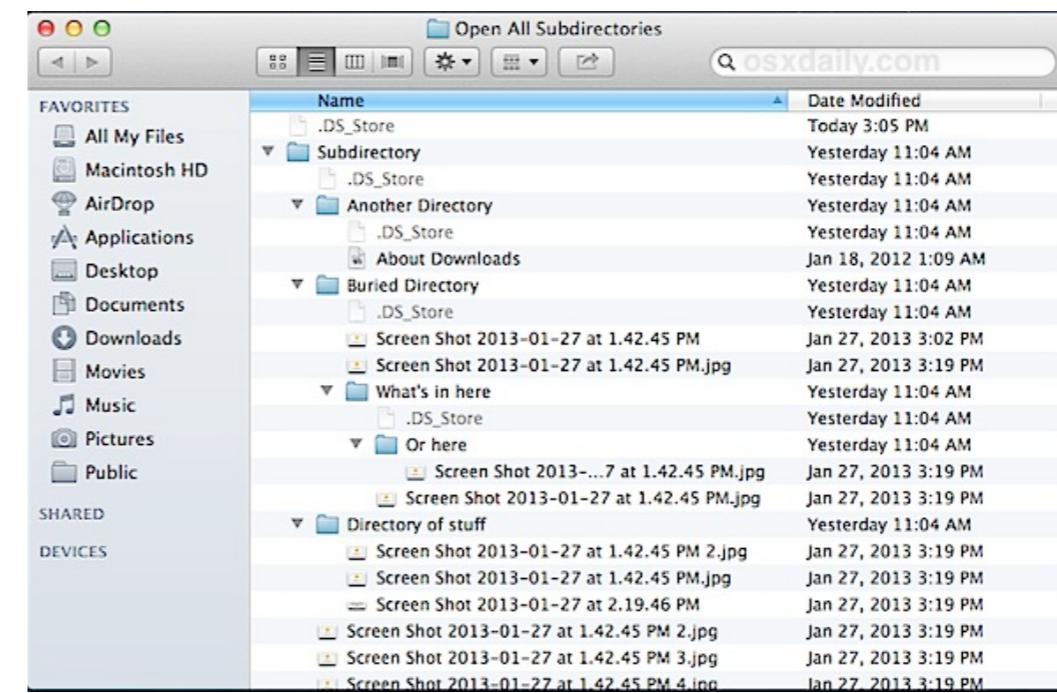
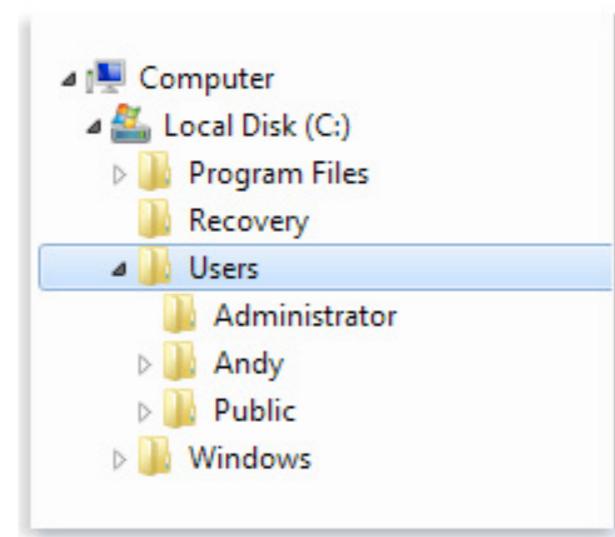
DOM

Document Object Model





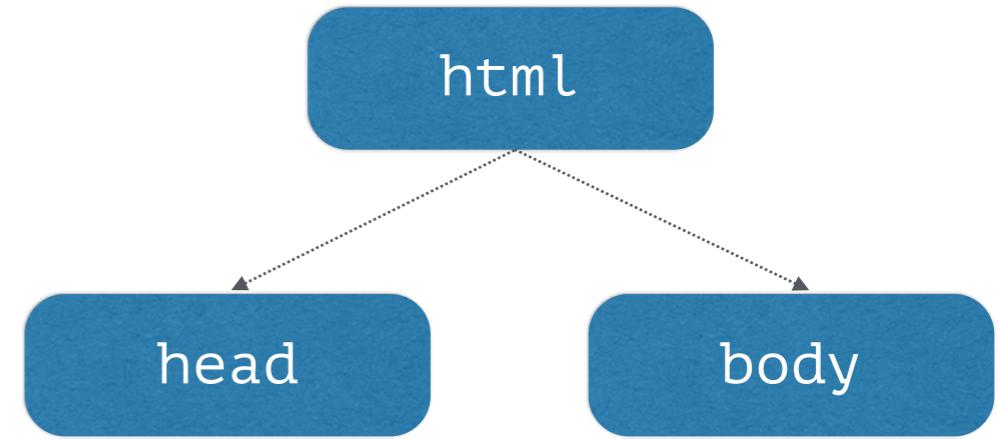




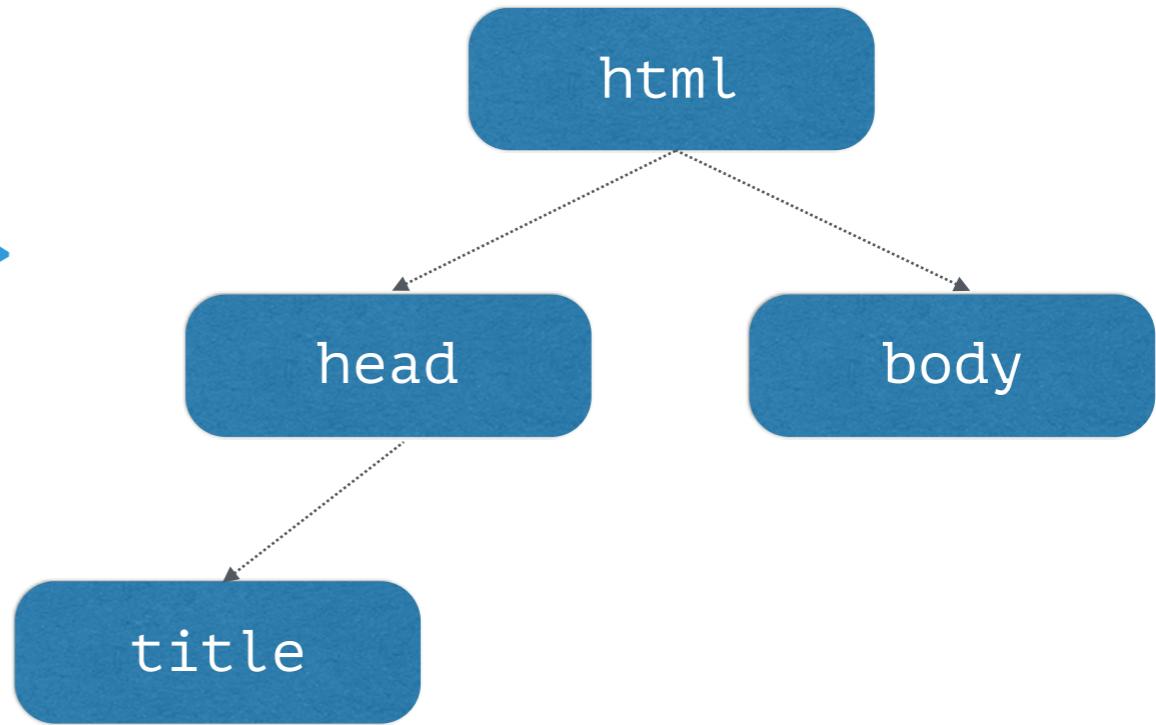
html

```
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <h1>
      Hello
    </h1>
    <p>
      Lorem Ipsum..
    </p>
  </body>
</html>
```

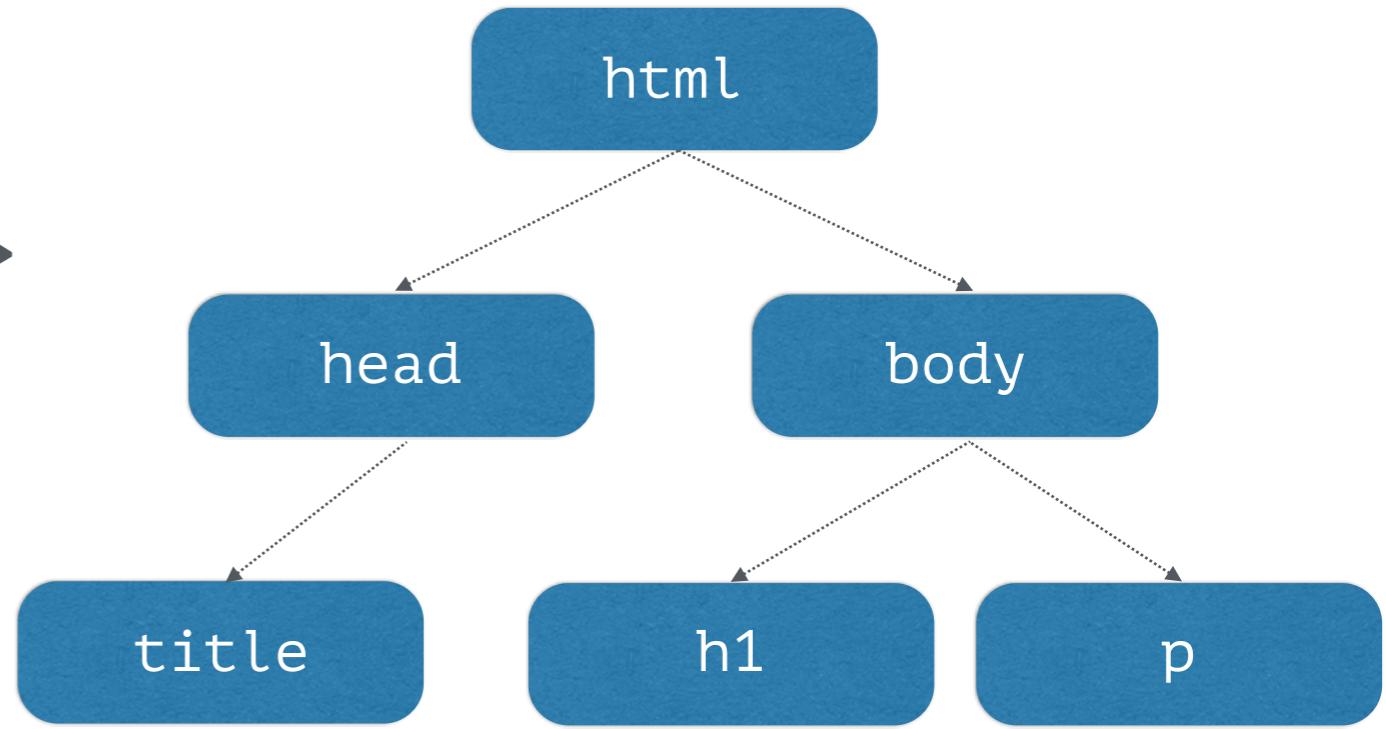
```
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <h1>
      Hello
    </h1>
    <p>
      Lorem Ipsum..
    </p>
  </body>
</html>
```



```
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <h1>
      Hello
    </h1>
    <p>
      Lorem Ipsum..
    </p>
  </body>
</html>
```



```
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <h1>
      Hello
    </h1>
    <p>
      Lorem Ipsum..
    </p>
  </body>
</html>
```



1 - DOM Manipulation

Id selector

```
<h1>  
  Hello  
</h1>  
<div id="myElement">  
  [...]  
</div>  
<div>  
  [...]  
</div>
```

#myElement

```
const section = document.getElementById('myElement');
```

Query selector

```
<h1>  
  Hello  
</h1>  
<p>  
  Lorem Ipsum...  
</p>  
<p class="red">  
  Autrum Ipsum...  
</p>
```

p

```
const paragraph = document.querySelector('p');
```

.red

```
const redParagraph = document.querySelector('.red');
```

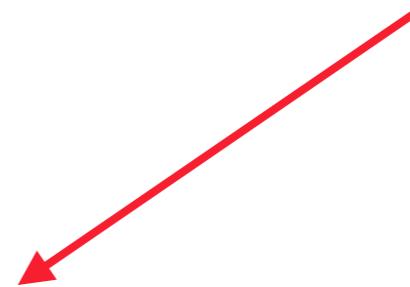
Same selector used in CSS

Multiple query selector

```
<div class="green">  
  Hello  
</h1>  
<div class="red">  
  [...]  
</div>  
<div class="red">  
  [...]  
</div>
```

.red

```
const redParagraphs = document.querySelectorAll(".red");
```



Returns an array of HTML objects

Manipulating DOM objects

```
<div id="car" class="red">  
  [...]  
</div>
```

Manipulating DOM objects

```
<div id="car" class="red">  
  [...]  
</div>
```



```
const car = document.getElementById("car"); // storing the object
```

```
car.classList.remove("red"); // removing the red class
```

```
car.classList.add("bold"); // adding the bold class
```

Manipulating DOM objects

```
<div id="car" class="red">  
  [...]  
</div>
```



```
const car = document.getElementById("car"); // storing the object
```

```
car.classList.remove("red"); // removing the red class
```

```
car.classList.add("bold"); // adding the bold class
```



```
<div id="car" class="bold">  
  [...]  
</div>
```

Manipulating multiple DOM objects

```
<p class="red">Hello!</p>
<p>This paragraph has no classes applied</p>
<p class="red">This paragraph has a class applied</p>
```

Manipulating multiple DOM objects

```
<p class="red">Hello!</p>
<p>This paragraph has no classes applied</p>
<p class="red">This paragraph has a class applied</p>
```



```
const redParagraphs = document.querySelectorAll(".red");
// selecting all the paragraphs with the class red (an array of paragraphs)

redParagraphs.forEach((paragraph) => { // looping through the array
  paragraph.classList.remove("red");
  paragraph.classList.add("blue");
})
```

Manipulating multiple DOM objects

```
<p class="red">Hello!</p>
<p>This paragraph has no classes applied</p>
<p class="red">This paragraph has a class applied</p>
```



```
const redParagraphs = document.querySelectorAll(".red");
// selecting all the paragraphs with the class red (an array of paragraphs)
```

```
redParagraphs.forEach((paragraph) => { // looping through the array
  paragraph.classList.remove("red");
  paragraph.classList.add("blue");
})
```



```
<p class="blue">Hello!</p>
<p>This paragraph has no classes applied</p>
<p class="blue">This paragraph has a class applied</p>
```

2 - Event Handling

Click and Mouseover Events

```
const button = document.getElementById('click-me');

button.addEventListener('click', () => {
  alert('You have clicked the button!');
});

button.addEventListener('mouseover', (event) => {
  event.currentTarget.classList.add('btn-primary');
});
```

3 - AJAX

Asynchronous JavaScript And XML

(To be continued...)

Exercise 4

Animate a TODO app

(add an item)
(delete an item)

Todos

Get things done, one item at a time.

Add new todo

Learn JavaScript

ADD ITEM

Code a todo app



Resources

JavaScript Tutorial: Learn Java X Gabriele

Secure | https://www.codecademy.com/learn/introduction-to-javascript

codecademy Community Catalog Log in Sign up

The screenshot shows the Codecademy website with a focus on the 'Introduction To JavaScript' section. At the top, there's a navigation bar with links for 'Community' and 'Catalog'. On the right, there are 'Log in' and 'Sign up' buttons. The main content area features a large teal background image with a white 'JS' logo. The title 'Introduction To JavaScript' is prominently displayed, followed by a subtitle 'Continue your learning by starting with Introduction To JavaScript'. A large yellow 'START' button is centered below the subtitle. Below this section, there's a dark blue banner with the word 'INTENSIVE' in large white letters. It describes the program as teaching technologies used by Netflix and Airbnb over an 8-week period. A green 'LEARN FASTER' button is located on the right side of the banner. At the bottom, there are links for 'Overview' and 'Syllabus'. The 'Syllabus' link is underlined, indicating it's the active page. The main content area has a light gray background with a subtle hexagonal grid pattern.

INTENSIVE

Learn the same technologies that enable Netflix and Airbnb to create powerful interactive web apps in the 8-week **Build Front-End Apps** program.

LEARN FASTER

Overview Syllabus

1 Learn JavaScript: Introduction

› </> Introduction to JavaScript Interactive Lesson

Intro to JavaScript | Udacity Gabriele

Secure | https://eu.udacity.com/course/intro-to-javascript--ud803

UDACITY Nanodegrees All Courses For Business Blog My Classroom

FREE COURSE

Intro to JavaScript

Fundamentals of the JavaScript Syntax

START FREE COURSE

NANODEGREE PROGRAM

Become a Professional Full Stack Developer

by  Amazon Web Services

Accelerate your career with the credential that fast-tracks you to job success. ➤

About this Course

COURSE COST TIMELINE SKILL LEVEL

JavaScript - Learn web develop x Gabriele

Secure | https://developer.mozilla.org/en-US/docs/Learn/JavaScript

MDN web docs Technologies References & Guides Feedback Sign in  

JavaScript

Languages  

Learn web development > JavaScript

Related Topics

[Complete beginners start here!](#)

▶ Getting started with the Web

[HTML — Structuring the Web](#)

▶ Introduction to HTML

▶ Multimedia and embedding

▶ HTML tables

▶ HTML forms

[CSS — Styling the Web](#)

▶ Introduction to CSS

▶ Styling text

▶ Styling boxes

▶ CSS layout

[JavaScript — Dynamic client-side scripting](#)

▶ [Getting started with the Web](#)

▶ [Introduction to HTML](#)

JavaScript is a programming language that allows you to implement complex things on web pages. Every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, or interactive maps, or animated 2D/3D graphics, or scrolling video jukeboxes, and so on — you can bet that JavaScript is probably involved.

Learning pathway

JavaScript is arguably more difficult to learn than related technologies such as [HTML](#) and [CSS](#). Before attempting to learn JavaScript, you are strongly advised to get familiar with at least these two technologies first, and perhaps others as well. Start by working through the following modules:

- [Getting started with the Web](#)
- [Introduction to HTML](#)

Fullstack Program



le wagon

Fullstack Program

1 - Ruby and coding basics	(1 week)
2 - Object-Oriented Programming	(1 week)
3 - Databases & SQL	(1 week)
4 - HTML, CSS and JavaScript	(2 weeks)
5 - Ruby on Rails	(2 weeks)
6 - Final Project.	(2 weeks)

Admissions: Emily



emily@lewagon.org



+351 918 882 20

Thank you!