Front-end 101

Foundation of frontend development

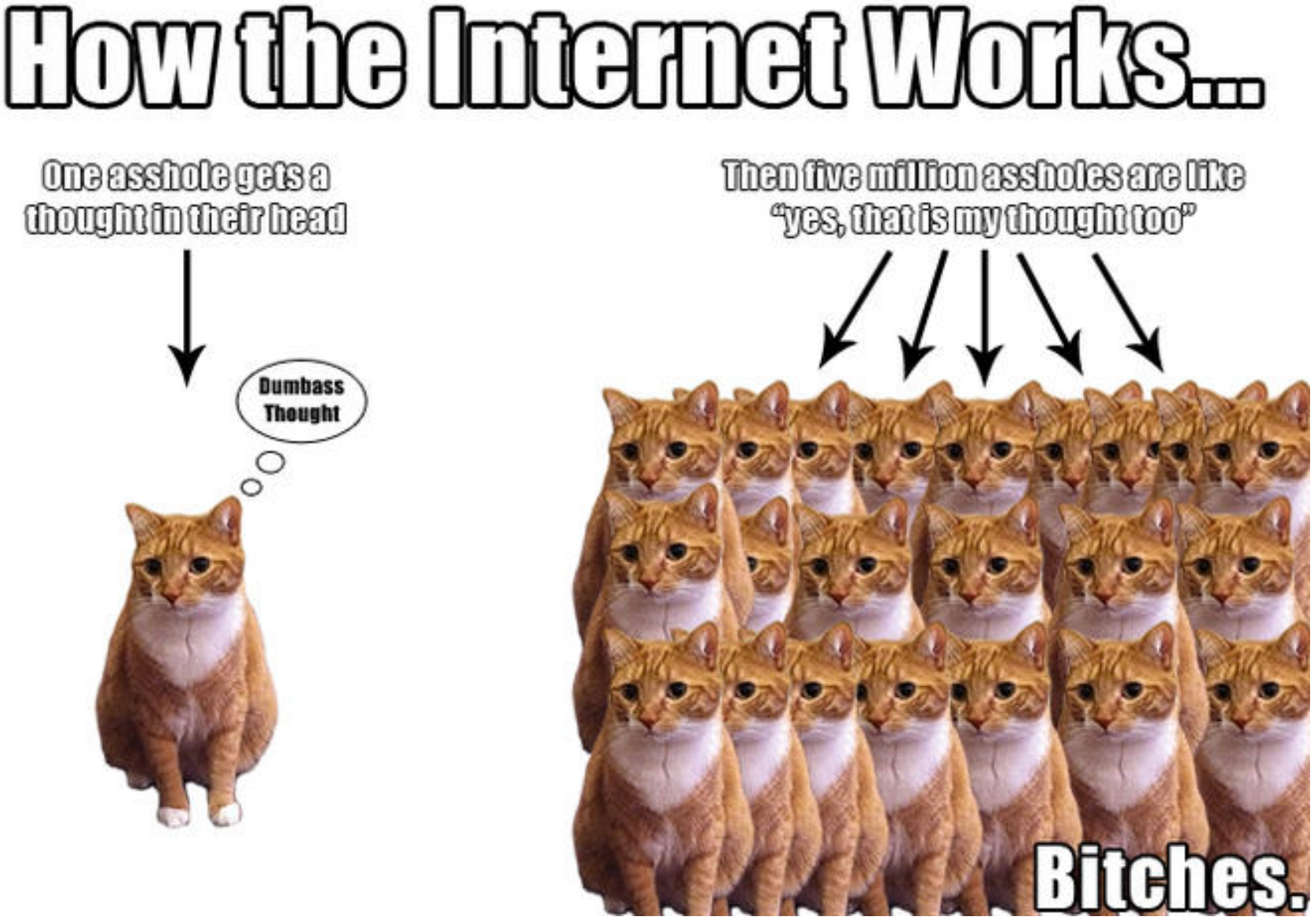
Jean-Loïc De Jaeger

Senior Data Engineer at Galeries Lafayette

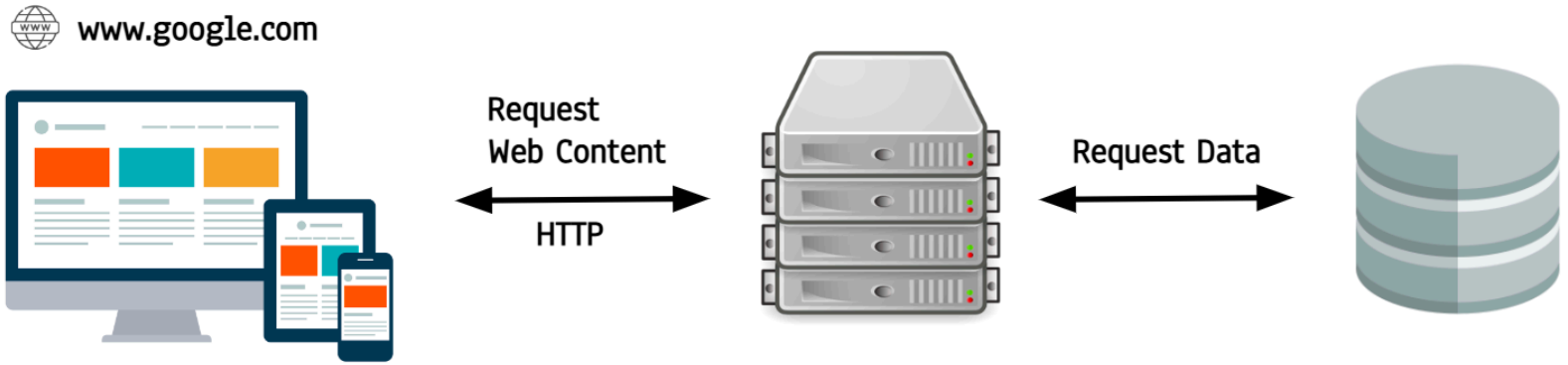
Table of contents

Web 101

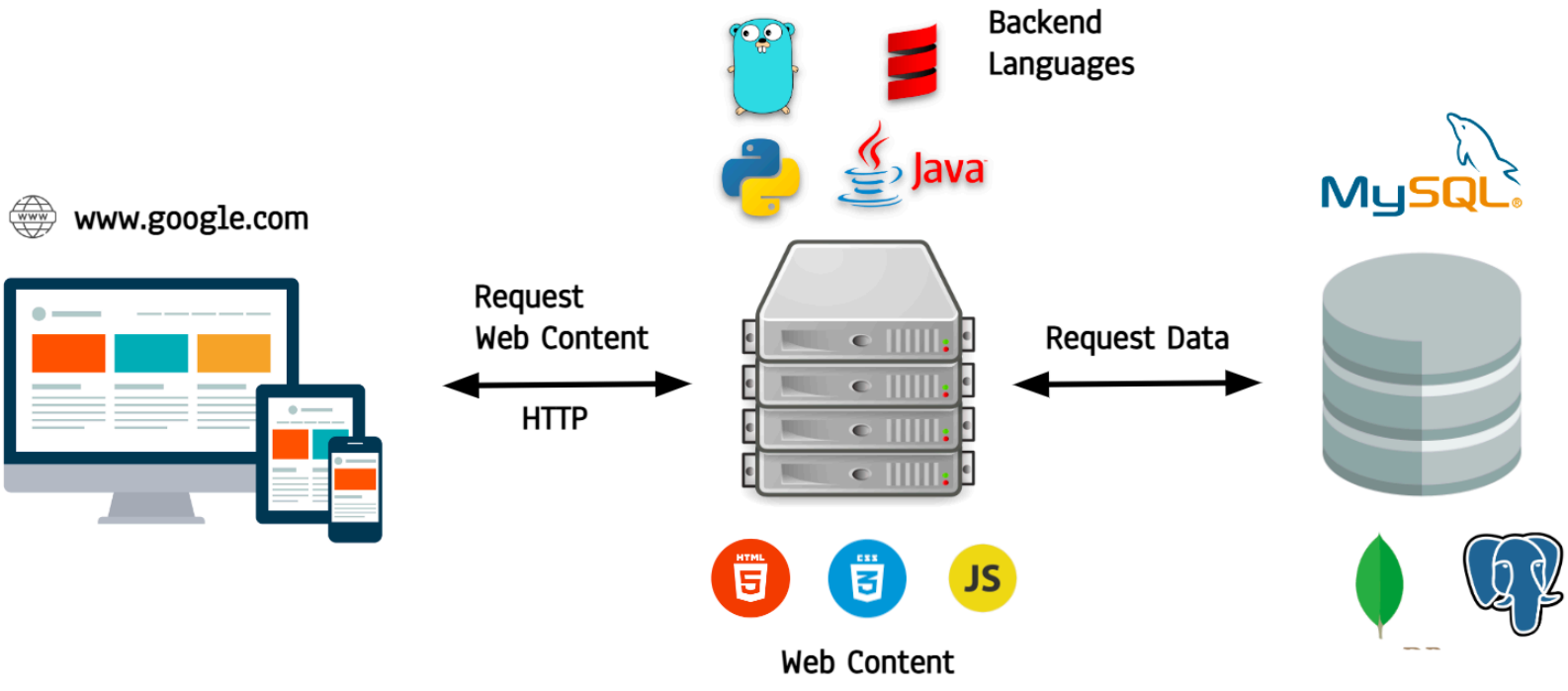
How does the web work?



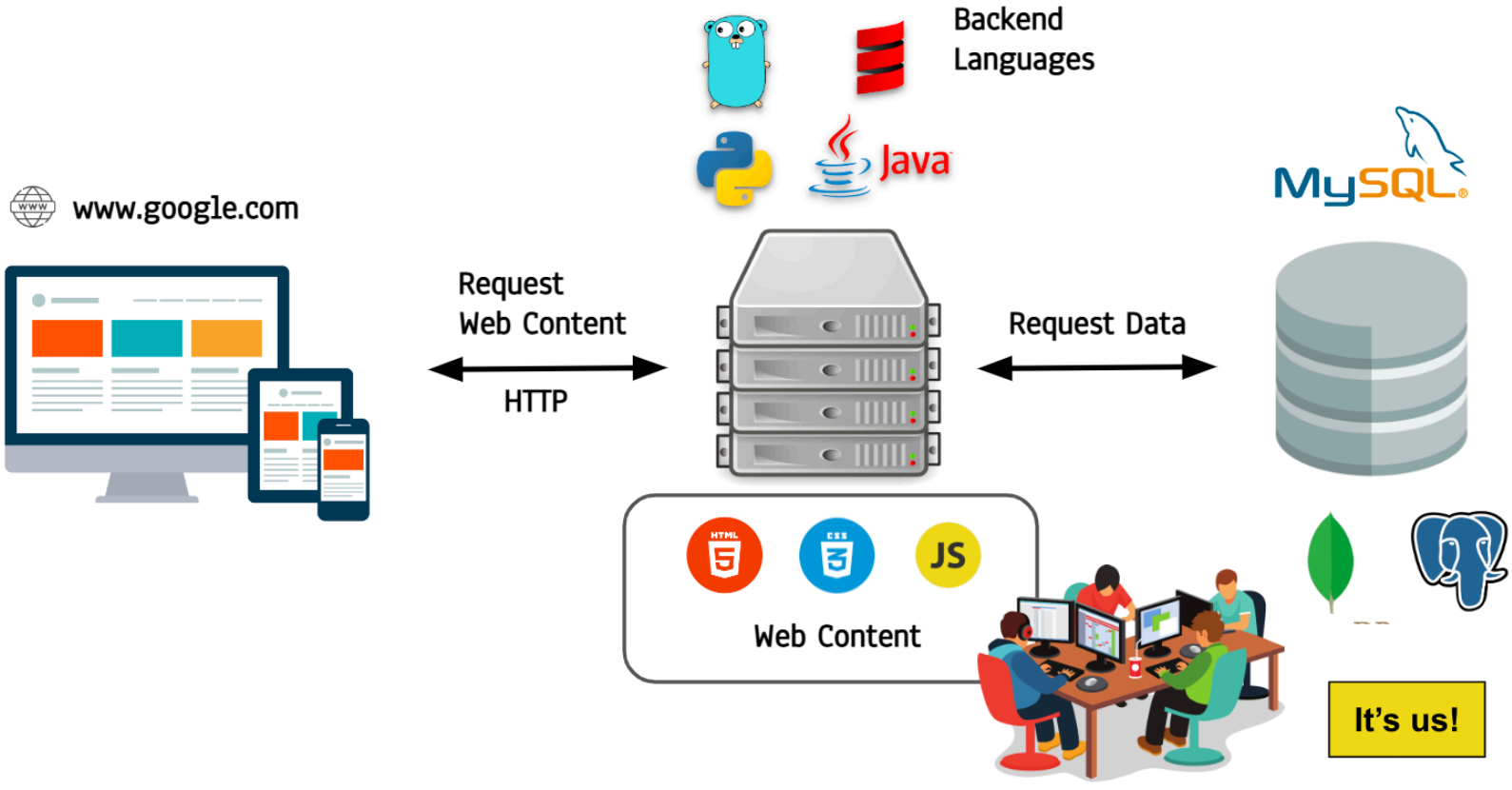
How does the web work?



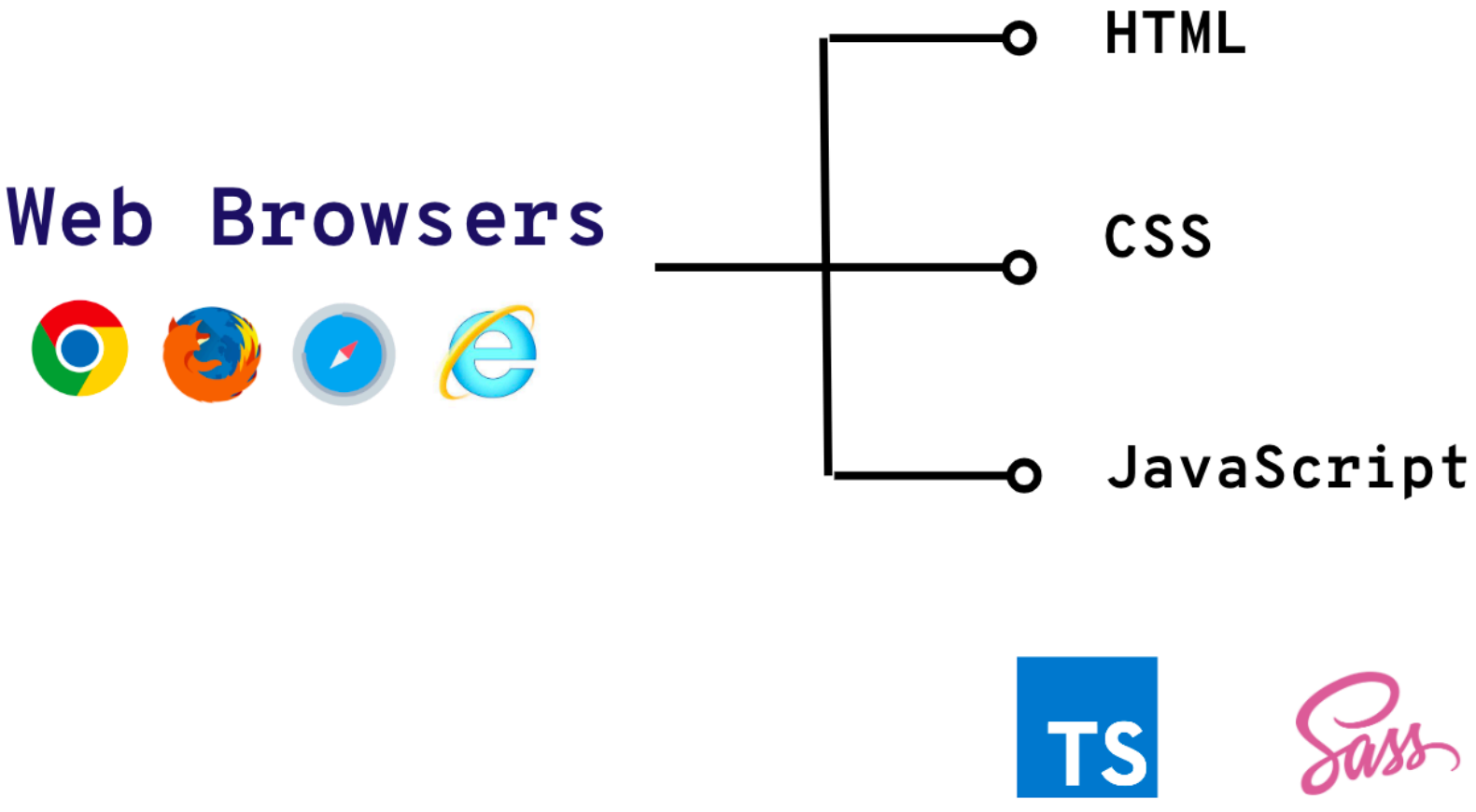
How does the web work?



How does the web work?



What can browser understand?



HTML Basics

HTML: HyperText Markup Language

- HTML is the standard markup language for Web pages.
- Not a programming language
- Structure the document

HTML 101

HTML easy example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

HTML 101 - Most common tags

```
<div></div>
<p>This is a paragraph.</p>

<img src="www.some-url.com" />

<span>Span 1</span>
<label>Label 1</label>

<ol>
    <li>Item 1</li>
    <li>Item 2/li>
</ol>

<ul>
    <li>Item 1</li>
    <li>Item 2/li>
</ul>
```

HTML 101 - HTML tables

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
</table>
```

- tr: table row
- th: table header
- td: table data

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

HTML 101 - HTML Forms

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

```
<form action="/some-webpage.html" method="post">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" required><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" required><br>
  <label for="lname">Email:</label><br>
  <input type="email" id="email" name="email" required><br><br>
  <label for="lname">Last name:</label>
  <input type="checkbox" id="active" name="active" checked>
  <br><br>
  <input type="submit" value="Submit">
</form>
```

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

My beautiful form

- [ ] I have a bike
- [ ] I have a car
- [ ] I have a boat

Submit

HTML 101 - Element selection

HTML class

The HTML class attribute is used to specify a class for an HTML element. Multiple HTML elements can share the same class.

```
<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>
```

ID class

The HTML id attribute is used to specify a unique id for an HTML element. Only one element with the same id in an HTML document.

```
<h1 id="myHeader1">My Header</h1>
```

CSS Basics

CSS: Cascading Style Sheets

- CSS describes how HTML elements should be displayed
- Not a programming language

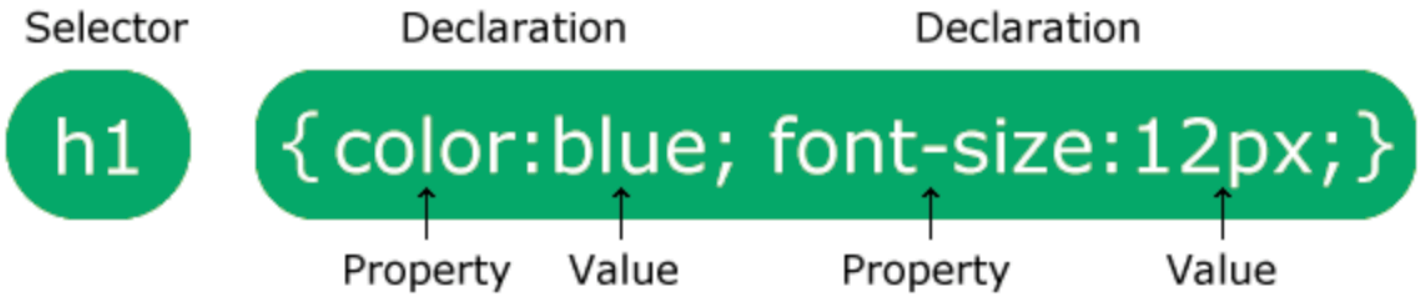
HTML easy example

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS 101 - Selectors



class selector

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

id selector

```
#para1 {
    text-align: center;
    color: red;
 }
```

CSS 101 - Three ways to add CSS

External CSS

index.html

```
<!DOCTYPE html>
 <html>
 <head>
 <link rel="stylesheet"
 href="style.css">
 </head>
 <body>

 <h1>This is a heading</h1>
 <p>This is a paragraph.</p>

 </body>
 </html>
```

style.css

```
body {
    background-color: lightblue;
 }

 h1 {
    color: navy;
    margin-left: 20px;
 }
```

CSS 101 - Three ways to add CSS

Internal CSS

```
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>

<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS 101 - Three ways to add CSS

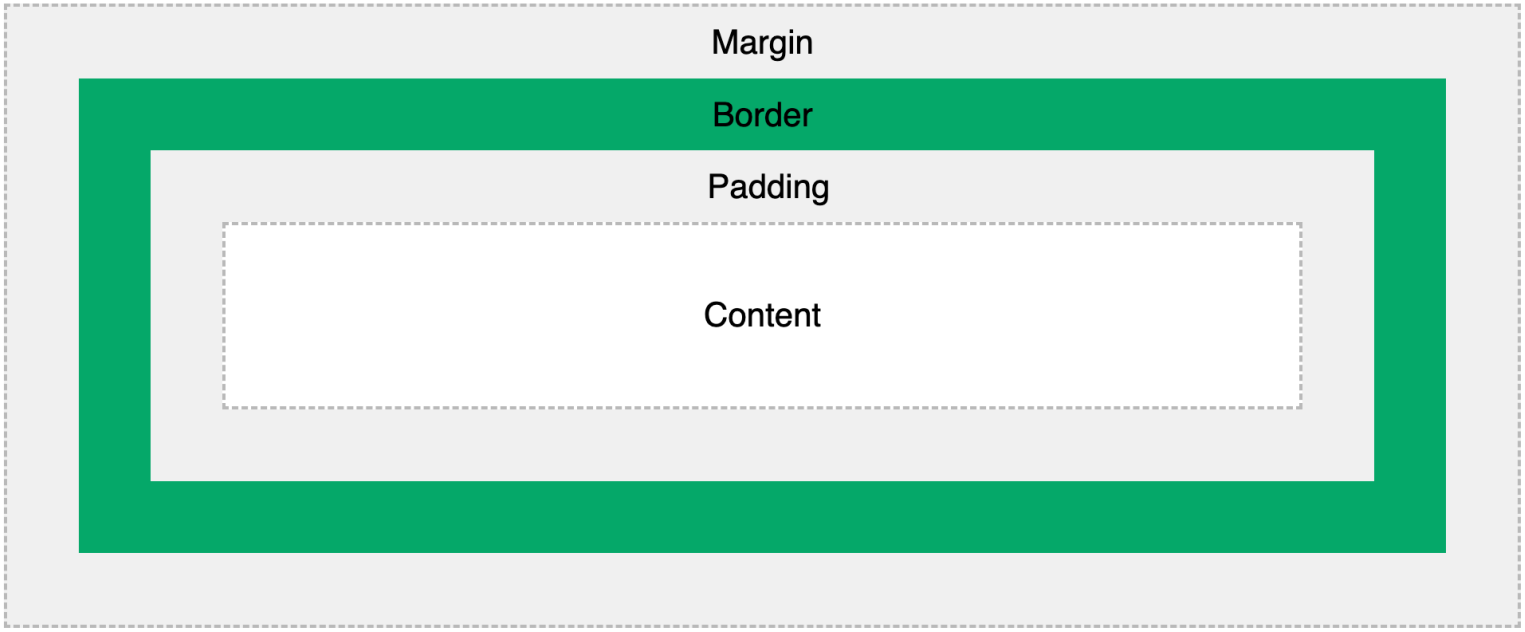
Inline CSS

```
<html>
<head>
<style>
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

CSS 101 - Box model

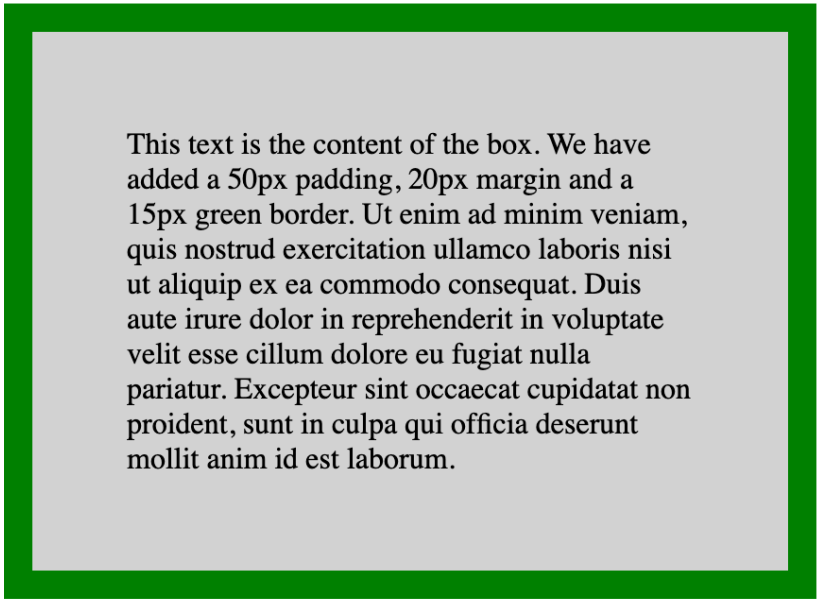


Explanation of the different parts

- **content**: The content of the box, where text and images appear
- **padding**: Clears an area around the content. The padding is transparent
- **border**: A border that goes around the padding and content
- **margin**: Clears an area outside the border. The margin is transparent

CSS 101 - Box model example

```
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
```



CSS: Cascading Style Sheets

- CSS describes how HTML elements should be displayed
- Not a programming language

JavaScript Introduction

Why JavaScript?

One word! Dynamic !

Why JavaScript?

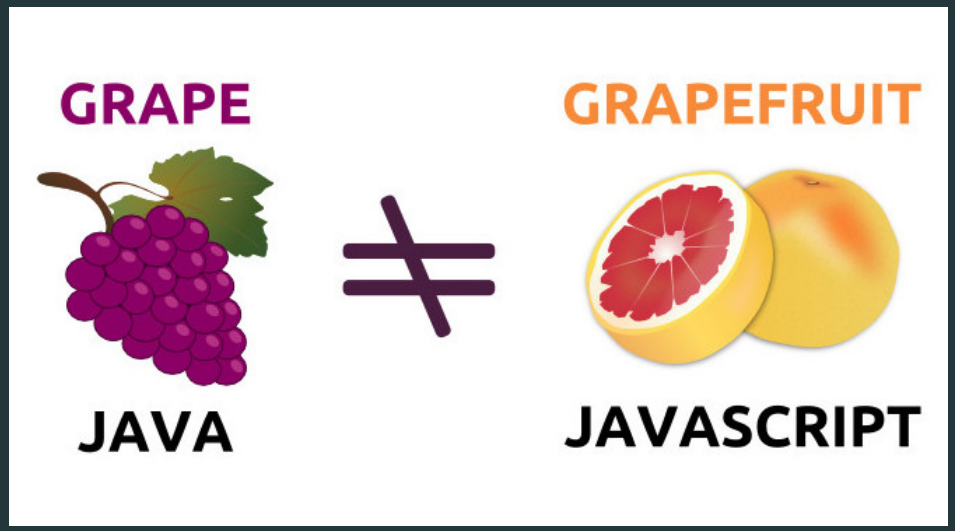
Client-side scripting (JavaScript) benefits:

- usability: can modify a page without having to post back to the server (faster UI)
- efficiency: can make small, quick changes to page without waiting for server
- event-driven: can respond to user actions like clicks and key presses

JavaScript quick summary

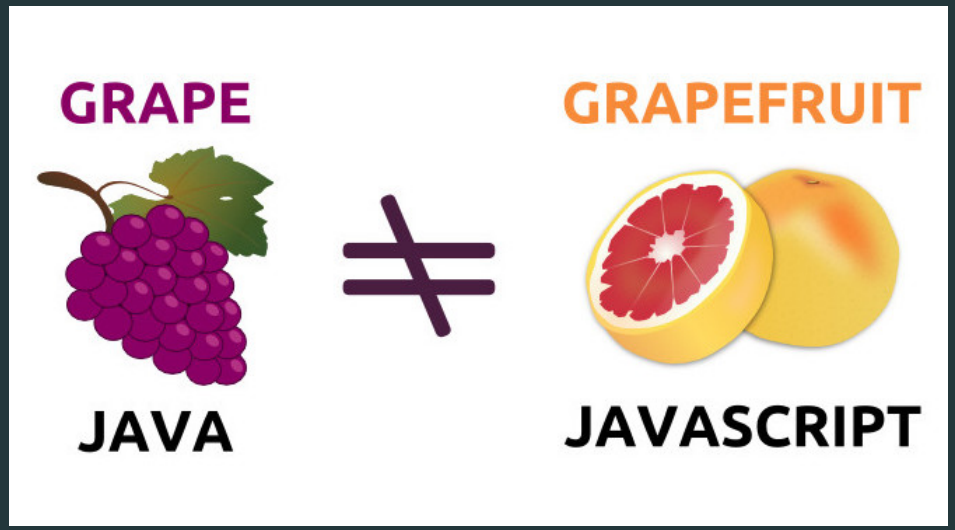
- a lightweight programming language (”scripting language”)
- used to make web pages interactive
- insert dynamic text into HTML (ex: user name)
- react to events (ex: page load user click)
- get information about a user’s computer (ex: browser type)
- perform calculations on user’s computer (ex: form validation)
- a web standard (but not supported identically by all browsers)

Do not forget something about JavaScript



Not related to Java in any way!

Do not forget something about JavaScript



Not related to Java in any way!
Alright, there are some similarities in the language syntax but that’s about it.

Where to place JavaScript

Add JavaScript to your HTML files

JavaScript scripts are added to your HTML files within the <script></script> tags.

```
<script type="text/javascript">
    var num1 = 10;
    var str1 = "Hello World";
    console.log(str1);
</script>
```

JavaScript basic features

Variables

The var keyword is used in all JavaScript code from 1995 to 2015.

```
var x = 5;
var y = 6;
var z = x + y;
```

The let and const keywords were added to JavaScript in 2015.

```
const x = 5;
let y = 6;
```

When to use const or var

Always declare variables with const if the value will not change

If you think the value of the variable can change, use let.

Variables - Mutability

Let variables cannot be redeclared

```
let x = "John Doe";
let x = 0;
// SyntaxError: 'x'
// has already been declared
```

Const variables cannot be reassigned

```
const PI = 3.141592653589793;
PI = 3.14; // This will give an error
PI = PI + 10; // This will also give an error
```

Variables - Scope

Variables declared inside a block cannot be accessed from outside the block

```
{
  let x = 2;
}
// x can NOT be used here
```

Variables declared with the var keyword can NOT have block scope

```
{
  var x = 2;
}
// x CAN be used here
```

Data Type

Main types found in JavaScript

```
let length = 16;                // Number
let lastName = "Johnson"; // String
let x = {
    firstName:"John",
    lastName:"Doe"
};                              // Object
let y = null                    // Null object
```

JavaScript typing is very dynamic

```
let x;              // Now x is undefined
x = 5;              // Now x is a Number
x = "John";         // Now x is a String
```

Functions

JavaScript functions are very similar to the Python functions

```
function name(parameter1, parameter2) {
    // code to be executed
}
```

Functions invokation

```
function myFunction(a, b) {
  return a * b;
}
// Anonymous function (equivalent)
const myFunction = (a, b) => a * b;

// Calling the function
let x = myFunction(4, 3);
```

For Loops

Regular for loop

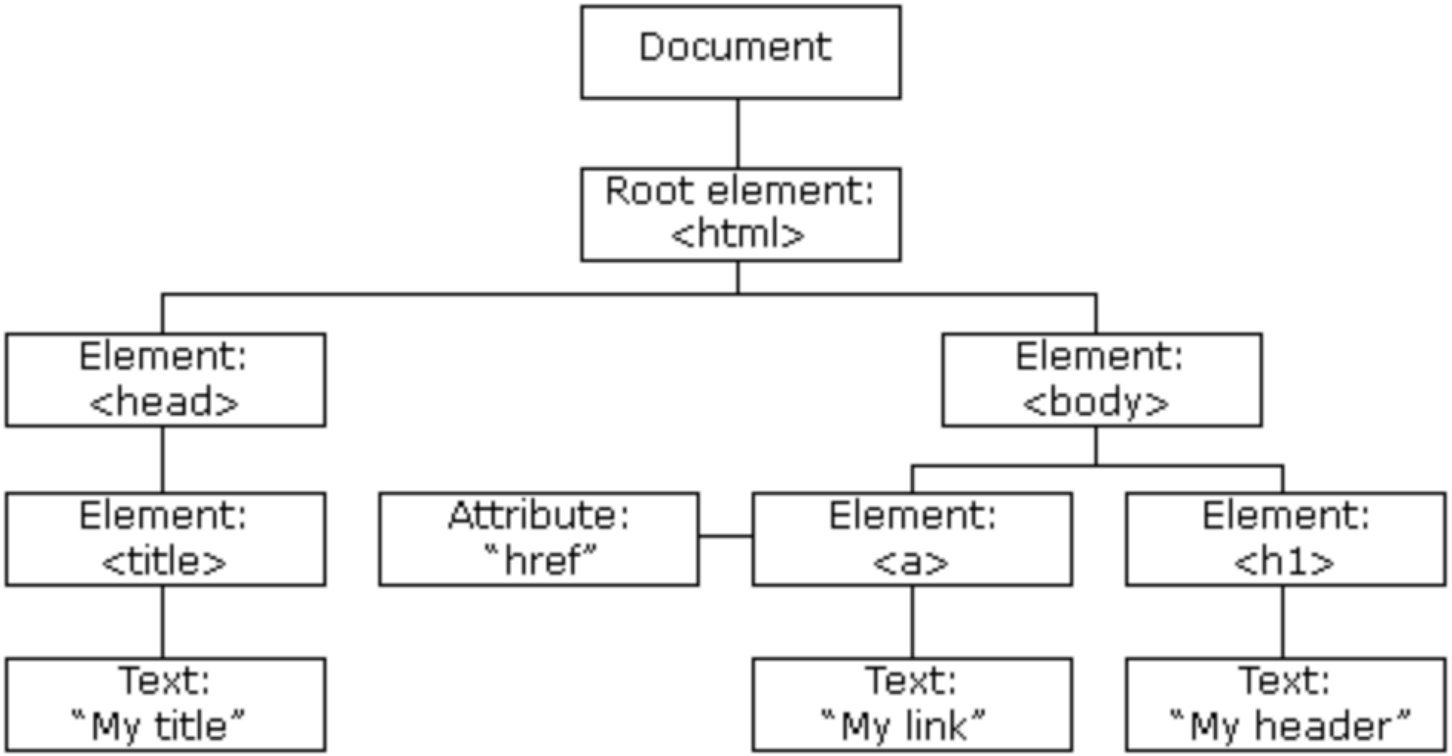
```
const cars = ["BWM", "Mercedes", "Aston Martin"];
for (let i = 0; i < cars.length; i++) {
   console.log(cars[i]);
 }
```

For in loop

```
const cars = ["BWM", "Mercedes", "Aston Martin"];
for (let car in cars) {
  console.log(car));
}
```

DOM Manipulation in JavaScript

DOM visual representation



DOM Manipulation - JavaScript power

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

DOM Manipulation - JavaScript power

Select elements

HTML can be selected in JavaScript by:

- id
- tag name
- class name
- CSS selectors
- HTML object collections

```
let element = document.getElementById(id);
let element = document.getElementsByTagName(name);
let element = document.getElementsClassName(id);
let element = document.querySelectorAll(cssSelector);
```

HTML Events

An HTML event can be something the browser does, or something a user does. Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

```
<button onclick="document.getElementById('demo')
.innerHTML = Date()">The time is?</button>
```

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

HTML Events

```
<button onclick="document.getElementById('demo')
.innerHTML = Date()">The time is?</button>
```

Clean Code tips

A cleaner code can be obtained by using a function

```
<body>
     <button onclick=setCurrentDate>The time is?</button>
 </body>
<script>
     function setCurrentDate() {
          document.getElementById('demo')
.innerHTML = Date();
     }
</scrip>
```

Happy Coding!