



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital

Módulo 4: Introdução à programação em javascript

# Aula 01

## Introdução ao Javascript

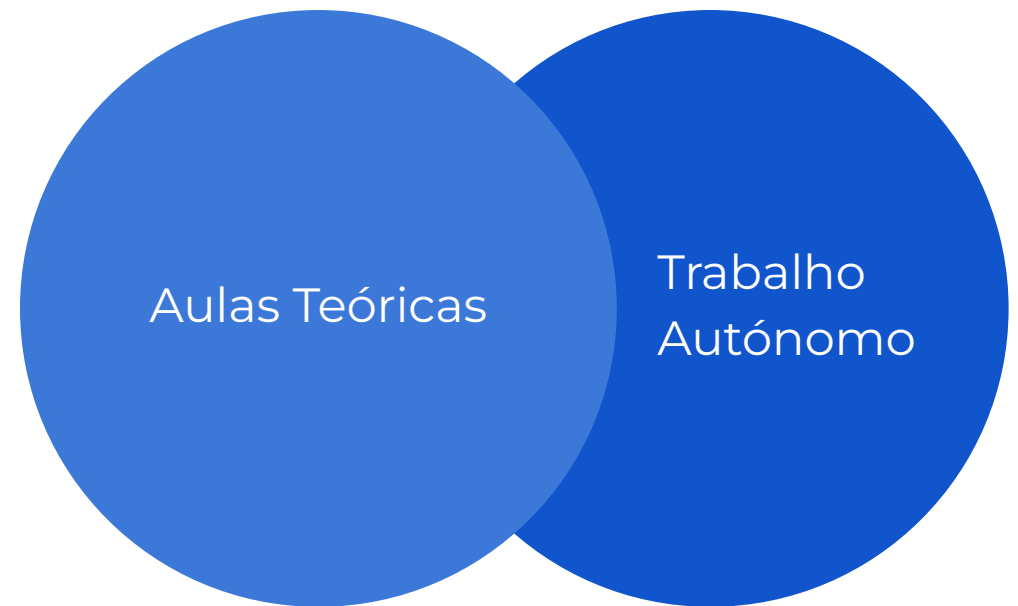


# Programa do módulo

- Introdução à linguagem JavaScript;
- Tipos de dados e operadores;
- Atribuição e manipulação de variáveis;
- Funções e encadeamento;
- Estruturas de controlo;
- Strings e operações respetivas;
- Estruturas de dados e apontadores;
- Ciclos;
- Callbacks, forEach, map, filter, find;
- **Projeto de grupo**

# Método de trabalho

- A cada 3,5h de aula teórica correspondem 3,5h de trabalho autónomo.
- Durante o trabalho autónomo, devem colocar dúvidas através do canal de discussão criado no [Discord](#).

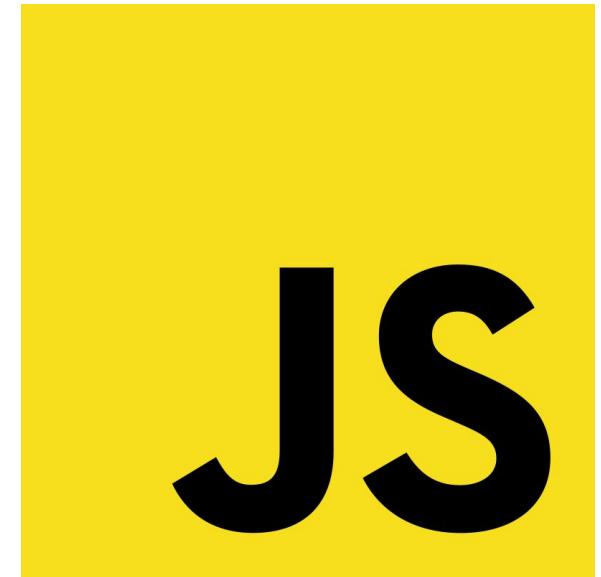


# Avaliação

Trabalho autónomo Submissão obrigatória no moodle	<b>20%</b>
Teste	<b>20%</b>
Projeto final	<b>50%</b>
Avaliação contínua (qualidade e organização do trabalho, autonomia, sentido de responsabilidade, assiduidade, etc)	<b>10%</b>

# O que é o JavaScript?

- JavaScript ou JS é uma **linguagem de programação**, no coração das tecnologias da Web, em conjunto com o HTML e o CSS;
- Foi criada para utilização na Web como mecanismo de **manipulação, validação e interação** dos documentos;
- Tem tido uma utilização crescente em ambientes de servidor, suportando plataformas como a Netflix e Uber, etc.



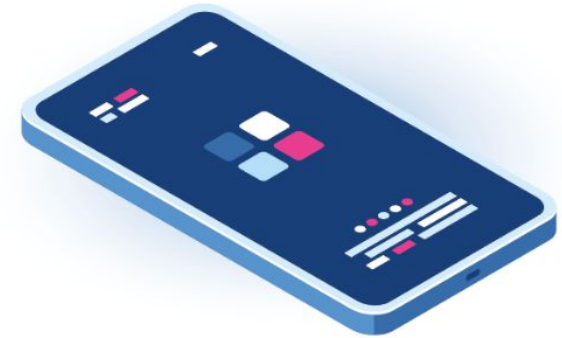
# O que é o JavaScript?



Cliente / Browser



Servidor

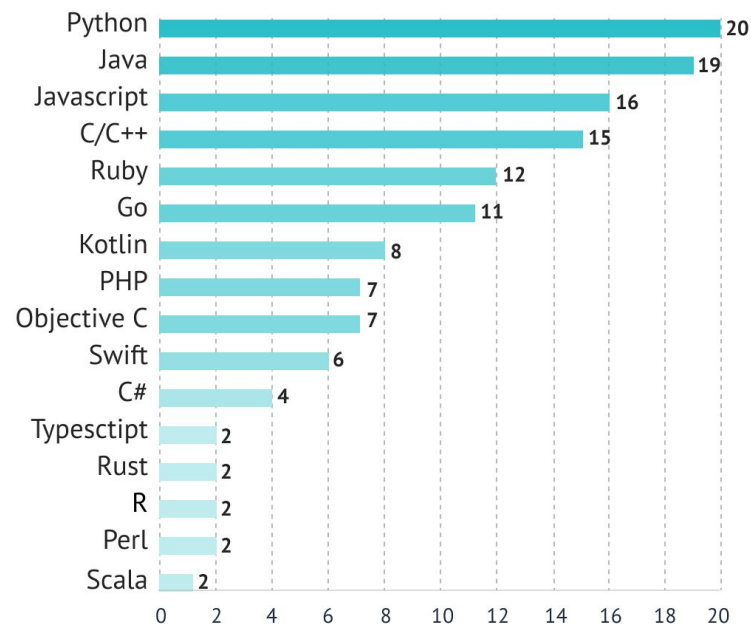


Cliente / Browser



# Linguagens de Programação mais usadas

- 16 das 25 maiores empresas mundiais de tecnologia usam JavaScript, e praticamente todos os sites usam JavaScript (>97%)



Linguagens de programação **mais usadas pelas empresas de tecnologia do top 25 mundial.**

Dados de 2019, recolhidos com informação extraída de anúncios de emprego.

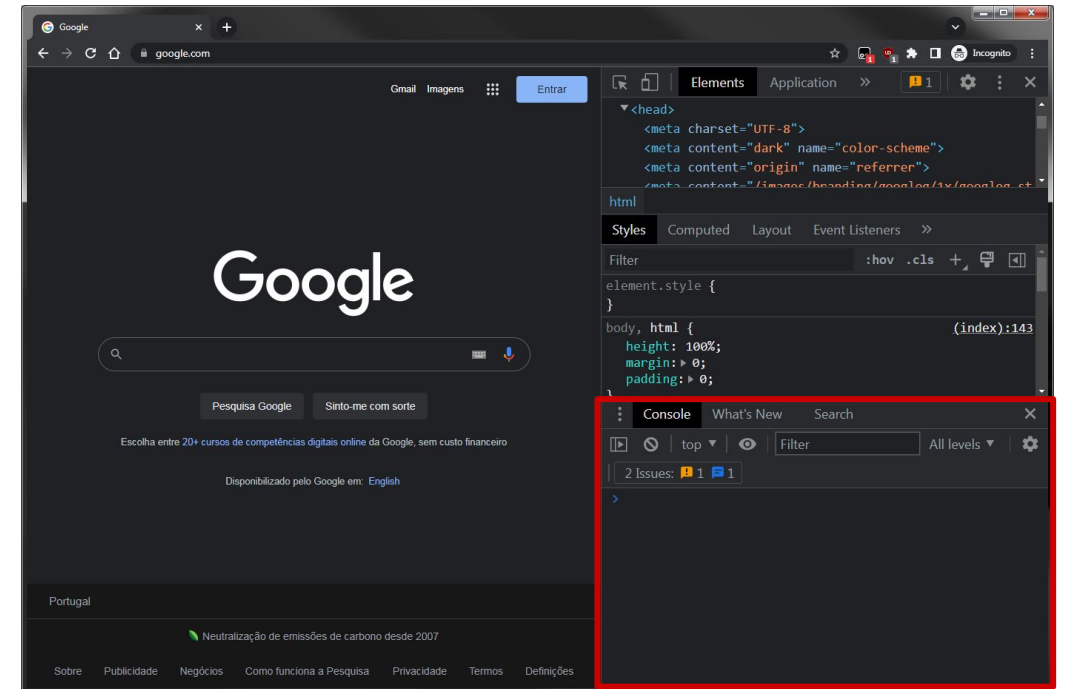
# Exemplo de algoritmo JavaScript

```
let ano_atual = 2022;  
let ano_nascimento = 1996;  
  
let idade = ano_atual - ano_nascimento;  
  
console.log(idade);
```



# Consola (DevTools)

- Podemos encontrar uma instância desta ferramenta em **todas as páginas** dos nossos browsers (chrome, edge, firefox, etc);
- Permite inspecionar, depurar e **executar código JavaScript** diretamente sem ter que correr um script à parte;



# Consola (DevTools) Standalone

- Para facilitar o processo de aprendizagem e criar a ponte de familiarização entre o JavaScript e a web, utilizaremos uma versão “*standalone*” da consola;
- Para executar código, introduzimos a linha que queremos executar e clicamos no **Enter**. Se quisermos introduzir várias linhas, utilizamos o Shift + Enter.

# Consola - Instalação

- **[Windows]** Para instalar, basta transferir e correr o script para a instalação e utilização da consola. Podem encontrar o script no canal de Conteúdos no discord ou nos recursos na página da disciplina no moodle:

<https://moodle.upskill.appx.pt/mod/resource/view.php?id=65>

- Em alternativa:

1. Instalar node.js
2. Abrir linha de comando (Terminal)
3. Instalar o NDB:
4. Executar o NDB:

```
npm install -g ndb
```

```
ndb .
```

<https://nodejs.org/dist/v12.9.1/node-v12.9.1-x64.msi>

ou para MacOS:

<https://nodejs.org/dist/v12.9.1/node-v12.9.1.pkg>

# Tipos de Dados

- Definem as **operações** que podem ser feitas nos dados, o **significado** dos dados e a maneira como os valores desse tipo são **armazenados**.

```
let numero = 8;
let texto = "olá";
let booleano = true;
let lista = [1, 2, 3];
let objeto = {
  nome: "João",
  idade: 30
};
let funcao = () => {
  //código
};
//etc..
```

# Tipos de Dados - Numérico

- Valores numéricos inteiros, decimais, positivos ou negativos

```
let numero = 8;  
let numero = 8.5;  
let numero = -2;
```

# Tipos de Dados - Texto

- Também conhecidos por “string”, guardam uma sequência de caracteres. Devem ser delimitadas por aspas, plicas, ou acentos graves

```
let texto = "João";  
let texto = 'Maria';  
let texto = `José`;
```

# Tipos de Dados - Booleano

- Pode ter dois valores: verdadeiro ou falso.

```
let booleano = true;  
let booleano = false;
```

# Tipos de Dados - Lista

- Permite agrupar vários valores numa determinada ordem. A lista pode ter elementos de qualquer um dos tipos existentes.

```
let lista = [1, 2, 3];  
let lista = ["João", "Maria", "José"];
```



# Tipos de Dados - Objeto

- Permite agrupar vários valores, acessíveis através de uma chave/key. Embora a chave tenha de começar com uma letra, o valor que está à direita dos dois pontos pode ser de qualquer tipo existente.

```
let objeto = {  
  nome: "João", // Texto  
  idade: 30      // Número  
};
```

# Tipos de Dados - Função

- A estudar mais à frente, funções permitem-nos reutilizar segmentos de código

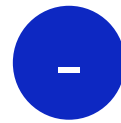
```
let funcao_somar = (a, b) => {  
  return a + b;  
}
```

# Operadores Aritméticos

- Os operadores permitem-nos indicar ao computador o cálculo que queremos que faça entre dois valores.
- Alguns dos operadores aritméticos:



**soma / adição**



**subtração**



**multiplicação**



**divisão**



**resto da divisão inteira** (eg:  $21\%7 = 0$  ou  $22\%7 = 1$ )

# Operadores Aritméticos - Exemplo

- $2 + 3 \rightarrow 5$
- $5 - 3 \rightarrow 2$
- $5 * 3 \rightarrow 15$
- $15 / 3 \rightarrow 5$
- $10 \% 3 \rightarrow 1$

# Operadores Relacionais

- Se no entanto pretendemos saber como dois valores se comparam, devemos utilizar operadores relacionais:



é igual?



é diferente?



é menor?



é maior?



é menor ou igual?



é maior ou igual?

# Operadores Relacionais - Exemplo

- $2 === 2 \rightarrow \text{true}$
- $2 === 3 \rightarrow \text{false}$
- $2 !== 3 \rightarrow \text{true}$
- $2 !== 2 \rightarrow \text{false}$
- $2 < 3 \rightarrow \text{true}$
- $2 < 2 \rightarrow \text{false}$
- $2 > 3 \rightarrow \text{false}$
- $2 > 2 \rightarrow \text{false}$
- $2 \leq 3 \rightarrow \text{true}$
- $2 \leq 2 \rightarrow \text{true}$
- $2 \geq 3 \rightarrow \text{false}$
- $2 \geq 2 \rightarrow \text{true}$

# Operadores Lógicos

- Podemos também aplicar operações lógicas sobre dois valores.



conjunção (e / and)



disjunção (ou / or)



negação

# Operadores Lógicos - Exemplo

`true && true → true`

`true && false → false`

`false && true → false`

`false && false → false`

`true || true → true`

`true || false → true`

`false || true → true`

`false || false → false`

`!true → false`

`!false → true`

`!!true → true`

`!!false → false`



# Operadores - Questão

- Tendo em conta os valores de a, b, c e d, qual será o resultado da operação da última linha?

```
a = 1
```

```
b = 2
```

```
c = 3
```

```
d = 4
```

```
a > b || c < d
```

# O futuro profissional começa aqui

iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



UP**skill**