



Power Up

Introdução

ao

JavaScript



BYTES4FUTURE



FUNDAÇÃO AGA KHAN



comunidade de aprendizagem



LISBOA
2020 Capital Europeia
da Cultura



ÁREA
METROPOLITANA
DE LISBOA



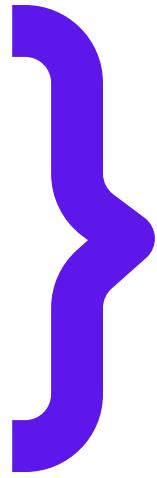
PRR
PLANO DE RECUPERAÇÃO
E RESILIÉNCIA



REPÚBLICA
PORTUGUESA



Financiado pela
União Europeia
NextGenerationEU



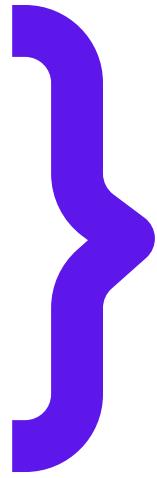
Array

- Os arrays são usados para **armazenar múltiplos valores** e acessá-los por meio de índices.

1. Declarar um array:

Podes criar um array simplesmente utilizando parêntesis retos/colchetes [] e inserir os valores separados por vírgulas.

```
let frutas = ["maçã", "banana", "laranja"];
```

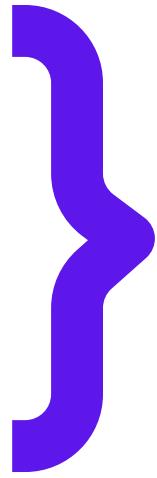


Array

2. Aceder aos elementos de um array:

Os elementos de um array são acessados por **índices**, começando do índice 0 para o primeiro elemento.

```
let primeiraFruta = frutas[0]; // "maçã"
let segundaFruta = frutas[1]; // "banana"
```



Array

3. Tamanho do array:

Podes obter o tamanho de um array utilizando a propriedade length.

```
let tamanho = frutas.length; // 3
```

4. Modificar elementos de um array:

Podes atribuir novos valores a elementos de um array usando o índice.

```
frutas[2] = "uva"; // Altera "laranja" para "uva"
```

Array

5. Adicionar elementos a um array:

Podes adicionar elementos a um array usando o método push()

```
frutas.push("pera"); // Adiciona "pera" ao final do array
```

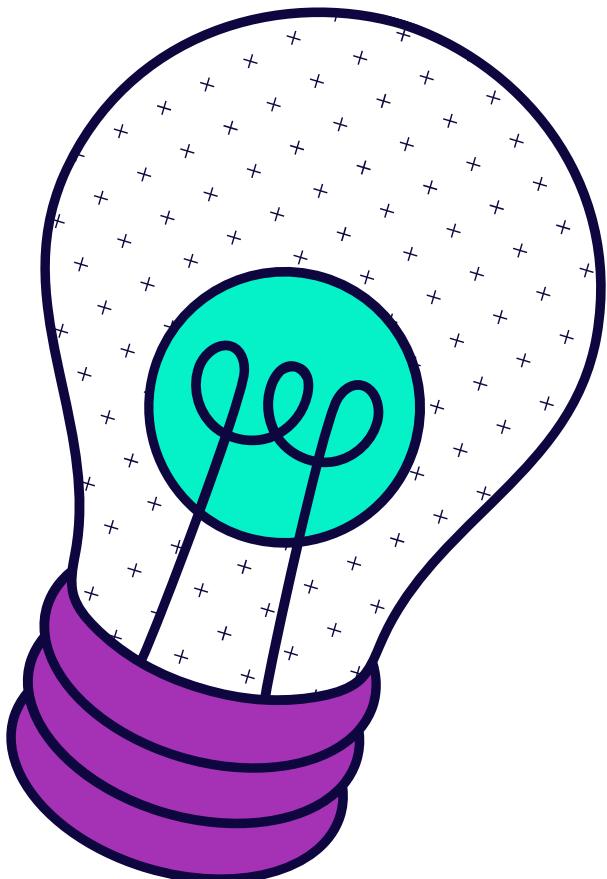
6. Remover elementos de um array:

Podes remover elementos de um array usando o método pop() para remover o último elemento.

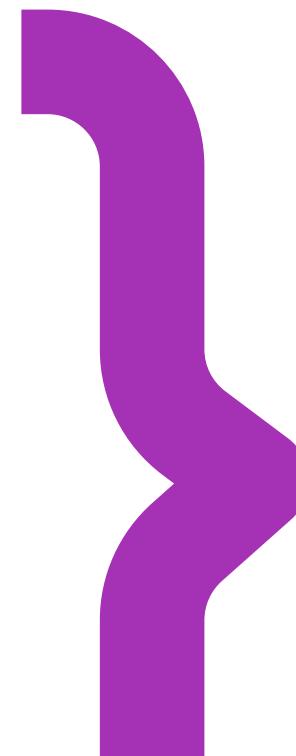
```
frutas.pop(); // Remove "pera" do final do array
```

search

QUIZ



Vamos praticar



Exercício:

1. Cria um array chamado **frutas** com três nomes de frutas à tua escolha.
2. **Exibe** o conteúdo do array na terminal.
3. **Adiciona** uma nova fruta ao array.
4. Exibe o conteúdo do array novamente para verificar que a nova fruta foi adicionada.

Exercício:

Considera um array inicial chamado “numeros” com os seguintes elementos:
1, 2, 3, 4 e 5.

Utilizando apenas as operações de array **pop**, **push**, **shift**, **unshift** e manipulação de índices, realiza as seguintes tarefas:

1. Remove o último elemento do array.
2. Adiciona o número 6 ao final do array.
3. Remove o primeiro elemento do array.
4. Adiciona o número 0 no início do array.
5. Substitui o número no índice 2 por 10.
6. Imprime cada elemento do array numa linha.

Exercício:

Considera um array inicial chamado “cores” com os seguintes elementos:
"Vermelho", "Verde", "Azul" e "Amarelo".

Utilizando apenas as operações de array **pop**, **push**, **shift**, **unshift** e manipulação de índices, realize as seguintes tarefas:

1. Remove o último elemento do array.
2. Adiciona a cor "**Roxo**" ao final do array.
3. Remove o primeiro elemento do array.
4. Adiciona a cor "**Laranja**" no início do array.
5. Substitui a cor no índice 1 por "**Preto**".
6. Imprime cada elemento do array numa linha.

Objetos

- Objetos permitem armazenar informações relacionadas em pares de **chave-valor**.
 - É uma maneira eficaz de representar entidades do mundo real com atributos e propriedades.

1. Declarar um objeto:

Neste exemplo, person é um **objeto com três propriedades: name, age, e job**. Cada propriedade tem um nome (a chave) e um valor associado.

```
const person = {  
  name: "Alice",  
  age: 30,  
  job: "teacher"  
};
```

Objetos

2. Aceder às propriedades de um objeto:

Podes aceder às propriedades de um objeto usando a notação de ponto (dot notation) ou parênteses retos.

```
console.log(person.name); // Output: "Alice"
```

```
console.log(person["age"]); // Output: 30
```

Objetos

3. Modificar Propriedades:

Podes modificar as propriedades de um objeto atribuindo novos valores.

```
person.age = 31;  
person.job = "developer";
```

4. AdicionarPropriedades:

Também é possível adicionar novas propriedades a um objeto a qualquer momento.

```
person.city = "New York";
```

Objetos

5. Remover Propriedades:

Para remover uma propriedade de um objeto, podes usar o operador **delete**.

```
delete person.job;
```

{

Objeto

Recapitulando...

```
// Criar um objeto person
const person = {
  name: "Alice",
  age: 30,
  job: "teacher"
};

// Modificar propriedades
person.age = 31;
person.job = "developer";

// Adicionar uma nova propriedade
person.city = "New York";

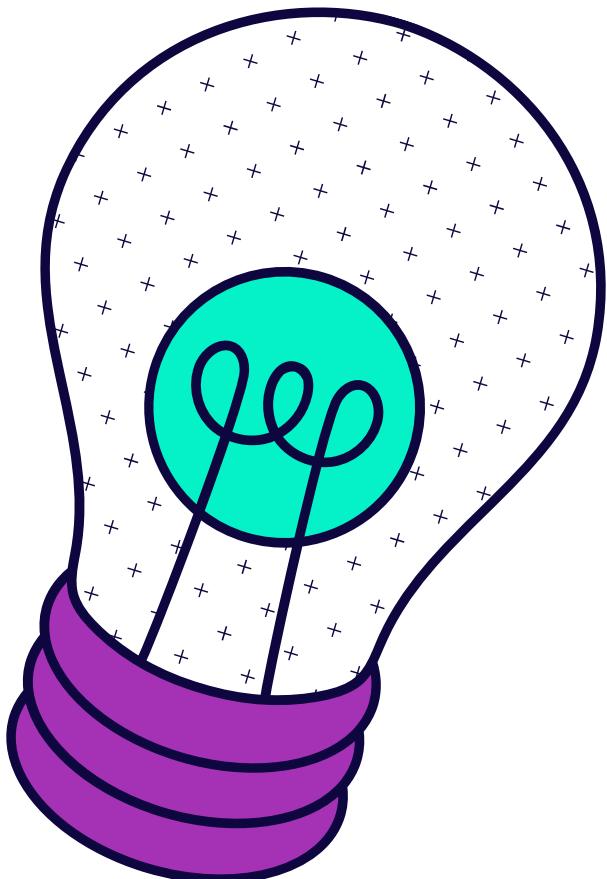
// Remover uma propriedade
delete person.job;
```

Objeto final:

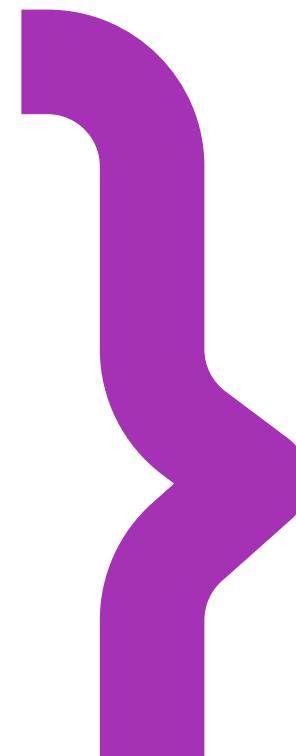
```
{
  name: "Alice",
  age: 31,
  city: "New York"
}
```

search

QUIZ



Vamos praticar



Exercício:

1. Cria um objeto chamado **pessoa** com duas propriedades:
 - a. nome com o valor "João".
 - b. idade com o valor 25.
2. Exibe as informações da pessoa na terminal, incluindo o nome e a idade.

Exercício:

1. Cria um objeto chamado **carro** com as seguintes propriedades:
 - a. **marca** (marca do carro).
 - b. **modelo** (modelo do carro).
 - c. **ano** (ano de fabrico do carro).
 - d. **cor** (cor do carro).
2. **Exibe** as informações do carro na terminal, incluindo a marca, o modelo, o ano e a cor.
3. **Altera** a cor do carro para "azul" e exibe novamente as informações atualizadas na terminal.
4. **Adiciona** uma nova propriedade chamada **quilometragem** com um valor à tua escolha.
5. **Exibe** todas as propriedades e os seus valores na terminal.



Power Up

Introdução

ao

JavaScript



BYTES4FUTURE



FUNDAÇÃO AGA KHAN



comunidade de prática



LISBOA
Câmara Municipal



área metropolitana
de lisboa



PRR
projeto de recuperacão e reabilitaçao

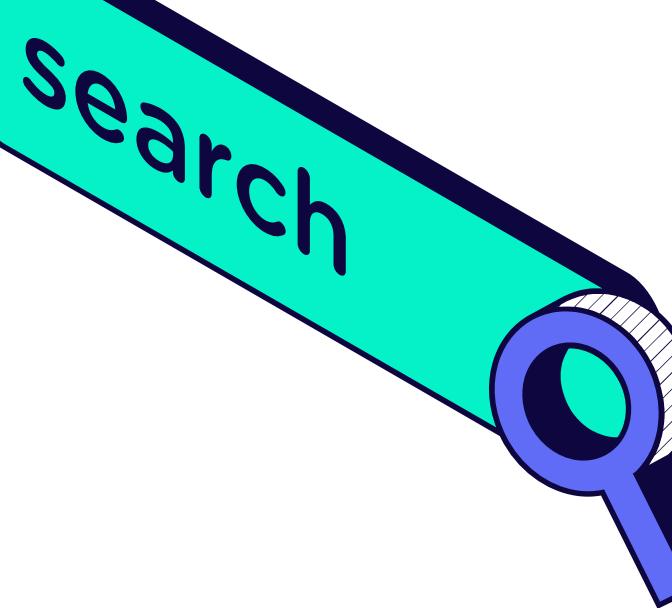
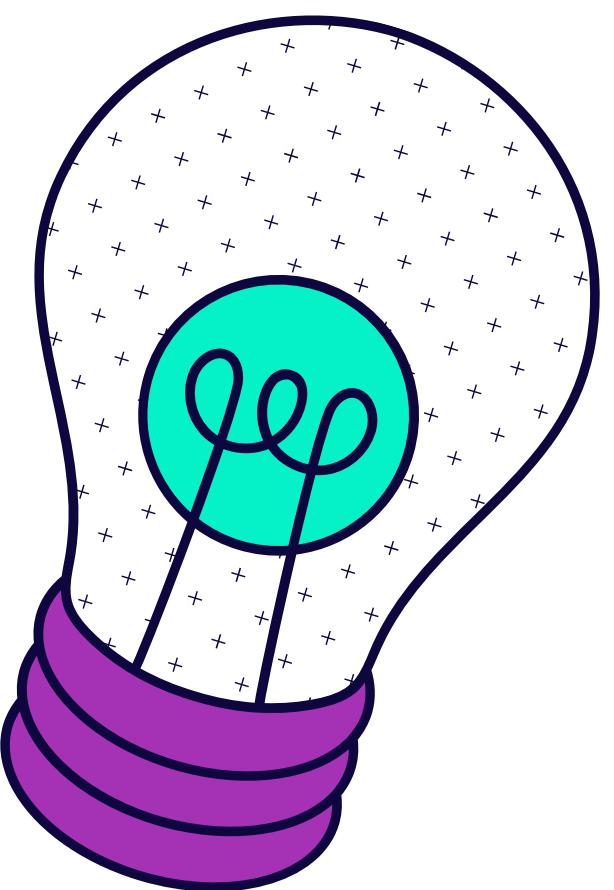
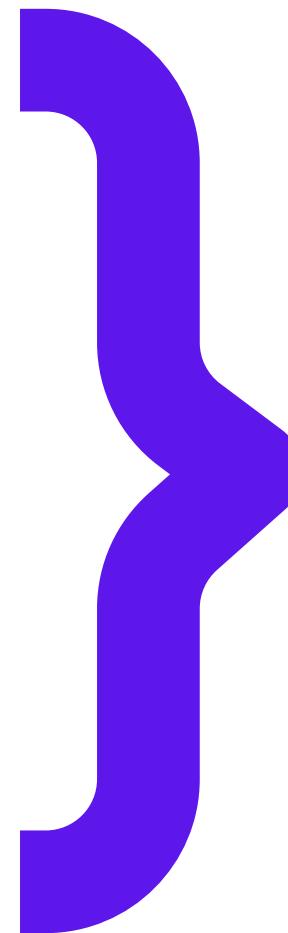


REPÚBLICA
PORTUGUESA



Financiado pela
União Europeia
NextGenerationEU

CONTROLO DE FLUXO

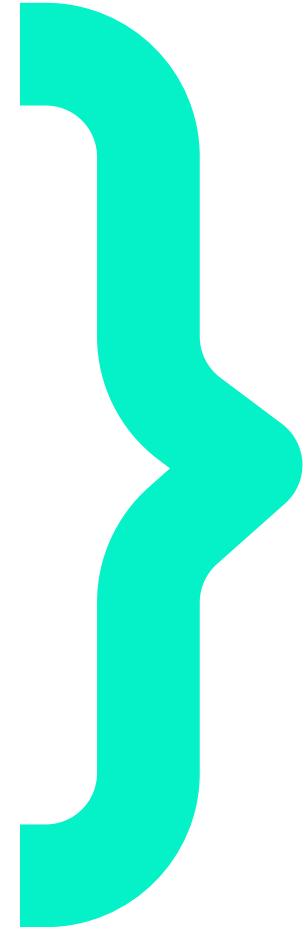


Controlo de Fluxo

Com o que aprendeste até aqui consegue escrever um programa completamente sequencial e previsível, o que sinceramente torna-se muito aborrecido, muito rápido.

Existem duas formas principais de **controlar o fluxo** de um programa em JavaScript:

1. Executar partes do código apenas **se** uma determinada condição se verificar;
2. Executar partes do código **repetidamente, em ciclo**, até atingir a condição em que o programa sai do ciclo.



ESTRUTURAS CONDICIONAIS

{

if ... else

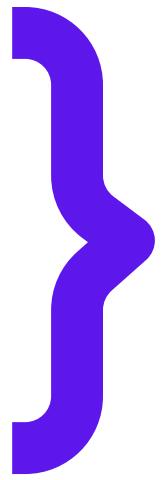
Uma das formas que permite controlar o fluxo de uma aplicação é utilizando o if, que pode ser opcionalmente seguido de um else.

A sintaxe de um if em JavaScript é:

```
if (condicao) {  
    // Código a ser executado se a condição for verdadeira  
} else {  
    // Código a ser executado se a condição for falsa  
}
```



If ... Else



if ... else

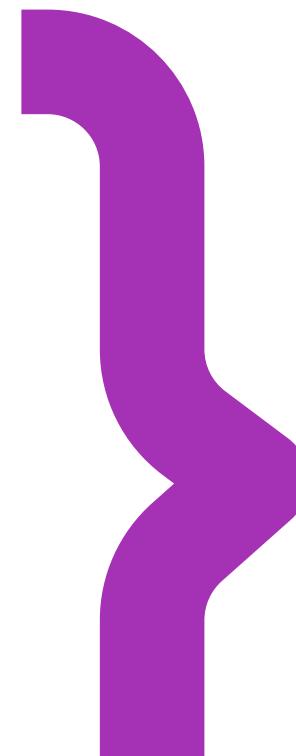
Um exemplo muito simples, se quiseres imprimir no terminal:

- par quando um número é par
- ímpar nos outros casos
- 7, se for o número 7
- Erro, se não for um número

console.log(...) permite-nos imprimir no terminal.

```
if (n === 7) {  
    console.log(n);  
} else if (n % 2 === 0) {  
    console.log('par');  
} else if (n % 2 === 1) {  
    console.log('ímpar');  
} else {  
    console.log('Erro')  
}
```

Vamos praticar



Exercício:

Verificar se é maior de idade:

- Crie um código que verifica a idade de alguém, e imprime na terminal se a pessoa é maior de idade ou menor de idade.

Exercício:

Verificação de Número Par ou Ímpar

- Cria um programa que imprime na terminal, se um número é par ou ímpar.

Exercício:

Existem duas equipas de ginástica: Dolphins e Koalas. Elas competem entre si 3 vezes. A equipa vencedora com a pontuação média mais alta ganha um troféu!

As tuas tarefas:

1. Calcular a pontuação média para cada equipa, utilizando os dados de teste incluídos abaixo. A pontuação média dos Dolphins deve ser atribuída à variável `scoreDolphins`, e a pontuação média dos Koalas deve ser atribuída à variável `scoreKoalas`.
2. Comparar as pontuações médias das equipas para determinar o vencedor da competição e imprimir na terminal.
3. "*Os Dolphins ganham o troféu*" se os Dolphins vencerem, ou
4. "*Os Koalas ganham o troféu*" se os Koalas vencerem, ou
5. "*Ambos ganham o troféu*" se as pontuações médias forem iguais.

DADOS DE TESTE: Os **Dolphins** marcaram 96, 108 e 89. Os **Koalas** marcaram 88, 91 e 110.

Exercício:

Escreve um código que atribua notas aos estudantes com base nas suas pontuações:

80-100 -> A

70-89 -> B

60-69 -> C

50-59 -> D

0-49 ->F

search

QUIZ

