



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital

Módulo 4: Introdução à programação em javascript

# Aula 11

## Funcionalidades avançadas



# Introdução

- Para além da sintaxe e funções que aprendemos nas últimas aulas, o JavaScript apresenta características e oferece um conjunto de outras ferramentas que podem nos poderão ser úteis.

# setTimeout

- Permite atrasar a execução de um determinado algoritmo durante um determinado tempo (em milisegundos). Esse código é executado de forma assíncrona, ie: o que estiver definido depois do timeout é executado imediatamente.

```
setTimeout(() => {  
    console.log("Passou 1 segundo");  
}, 1000);  
  
console.log("Começar contagem");
```

# setInterval

- Permite repetir a execução de um determinado algoritmo com um determinado intervalo (em milisegundos). A primeira execução ocorre após o intervalo definido.

```
setInterval(() => {  
    console.log("Intervalo completo");  
}, 1000);
```

# Exercício 1

- Implementa um algoritmo que faz uma contagem decrescente na consola de 5 a 0 utilizando o setTimeout.

Bónus: Como poderíamos obter o mesmo resultado com o setInterval?  
(pista: para que serve a função clearInterval?)

# Operador “Spread”

- A sintaxe mais recente do Javascript oferece-nos esta ferramenta muito útil, que permite poupar muitas linhas quando estamos a trabalhar com listas e objetos.
  - Clonar listas sem necessitar de implementar um ciclo
  - Passar a uma função um número variável de argumentos (ex `Math.max(...numeros)` vs `Math.max(numeros[0], numeros[1], etc)`)
  - Parâmetro de resto - `juntarNomes(primeiro_nome, ...resto_nome)`;
  - União de listas, adicionar elementos a listas
  - Clonar objetos, possibilidade de adicionar propriedades a esse clone -> `{...obj, ano_nascimento: 1990}`

## Exercício 2

- Implementa uma função que recebe dois objetos A e B, e devolve o objeto C, utilizando o operador spread (sem definir novas variáveis dentro da função):

```
A = {  
  primeiro_nome: "Manuel",  
  idade: 25  
};
```

```
B = {  
  ultimo_nome: "João",  
  idade: 15,  
};
```

```
C = {  
  ultimo_nome: 'João',  
  idade: 25,  
  primeiro_nome: 'Manuel',  
  ano_nascimento: 1997  
}
```

# Operador Ternário

- É a única operação condicional que aceita 3 operandos: Uma condição, seguida de um ponto de interrogação (?), uma expressão caso a condição seja verdadeira, dois pontos (:) e uma expressão caso a condição seja falsa.
- Permite substituir o comportamento de um “if .. else” em apenas uma linha.

```
if(condição) {  
    expressão1;  
} else {  
    expressão2;  
}
```



```
condição ? expressão1 : expressão2;
```



# Operador de Coalescencia


- Utilizado para definir valores padrão ou valores de fallback. Caso um determinado valor não esteja definido, é possível assumir um novo valor.

Qual é o resultado de cada um destes blocos de código?

```
let utilizador = {};  
let idadeUtilizador = utilizador.idade ?? 15;  
console.log(idadeUtilizador);
```

```
let utilizador = {idade: 0};  
let idadeUtilizador = utilizador.idade ?? 15;  
console.log(idadeUtilizador);
```

```
let utilizador = {idade: 0};  
let idadeUtilizador = utilizador.idade || 15;  
console.log(idadeUtilizador);
```



# Encadeamento opcional - ?.

- Tem em consideração os seguintes objetos.

```
let A = {  
  nome: {  
    primeiro: "Manuel",  
    segundo: "João"  
  },  
  idade: 25  
}  
  
console.log(A.nome.primeiro);
```

```
let B = {  
  primeiro_nome: "Manuel",  
  segundo_nome: "João",  
  idade: 25  
}  
  
console.log(B.nome.primeiro);
```

↑  
Erro

# Encadeamento opcional - ?.

- O operador de encadeamento opcional quebra um encadeamento caso um valor não seja definido ou se for nulo. Assim, podemos evitar o erro no objeto B com este operador. Utilizado em conjunção com o operador de coalescência podemos aceder à propriedade 'primeiro\_nome' em último recurso.

```
let B = {  
  primeiro_nome: "Manuel",  
  segundo_nome: "João",  
  idade: 25  
}  
  
console.log(B.nome?.primeiro ?? B.primeiro_nome);
```

# O futuro profissional começa aqui

iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



UPskill