



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital

Módulo 4: Introdução à programação em javascript

Aula 4

Strings e Objetos



Texto ou “Strings”

- No JavaScript o texto é delimitado por aspas duplas (“ ”), simples (‘ ’) ou acentos graves (` `):

```
let texto1 = "João";  
let texto2 = 'Maria';  
let texto3 = `José`;
```

Strings - Tamanho

- As strings **têm um tamanho** que é possível obter através da propriedade **.length**

```
let texto1 = "João";  
console.log(texto1.length);  
>> 4
```

```
let texto2 = "";  
console.log(texto2.length);  
>> 0
```

Strings - Caracteres

- Podemos olhar para uma String como uma **sequência de caracteres**, e isso permite-nos obter o caractere específico dado uma posição no texto.
- Na posição **2** teremos o terceiro caractere “m” (mais detalhes nos slides sobre as listas):

```
let texto1 = "Bom dia!";  
console.log(texto1[2]);  
>> 'm'
```

Strings - Comparação

- É possível comparar Strings com os operadores relacionais:

```
let texto1 = "bom dia";  
let texto2 = "BOM DIA";
```

```
texto1 === "bom dia";  
>> true
```

```
texto1 === texto2;  
>> false
```

comparação é
Case Sensitive!



```
"Susana" > "Soraia";  
>> true
```

Strings - Concatenação

- Para “juntar” duas strings numa, utilizamos um processo chamado concatenação. Existem duas formas comuns de o fazer; com o método **.concat** ou simplesmente com o operador de adição (+):

```
let texto1 = "hello ";  
let texto2 = "world!";
```

```
texto1.concat(texto2);  
>> "hello world!"
```

```
texto1 + "everyone!";  
>> "hello everyone!"
```

Strings - Interpolação

- Parecido à concatenação, é possível através da interpolação criar “***String templates***” que permite **substituir marcadores de variáveis**. Para este processo, só podemos usar strings com acentos graves (``):

```
let nome = "João Silva";  
let idade = 30;  
let template = `O meu nome é ${nome}, tenho ${idade} anos`;  
>> "O meu nome é João Silva, tenho 30 anos"
```

Exercício 1 - Strings

- Criar uma função que recebe o nome e idade de uma pessoa e devolve uma String construída com uma mensagem que diz quando nasceu:

```
function mensagem(nome, idade) {  
    ...  
}  
  
>> "Olá José, nasceste em 1993"
```


Exercício 2 - Strings

- Desenvolver uma função que capitalize a primeira letra de um nome. Pista: Experimentem a ver o que fazem os métodos `.toUpperCase()` e `.slice(2)` numa String.

```
function capitalizarPalavra(palavra) {  
    ...  
}  
  
>> capitalizarPalavra("teste")  
>> "Teste"
```

Javascript - Tipos de Dados

- Vimos na aula passada que existem tipos diferentes de dados, com significados e operações diferentes.
- Vamos analisar com mais atenção os tipos “lista” e “objeto”.

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

Estruturas

- Estruturas como as **listas** e **objetos** permitem-nos guardar e gerir várias informações sem termos de criar uma variável para cada uma.
- É habitual não se saber à partida que informações existem numa aplicação, por isso precisamos de uma forma dinâmica de guardar os dados.

```
let lista = [  
  "João",  
  "Maria",  
  "Teresa"  
];  
  
let objeto = {  
  nome: "Francisco",  
  apelido: "Santos",  
  idade: 25,  
  casado: false  
}
```

Objetos

- Os objetos (Object) permitem-nos guardar diferentes informações de forma estruturada.
- Os valores são associados a uma determinada chave/key e podem ser acedidos da mesma forma.

```
let objeto = {  
  nome: "Francisco",  
  apelido: "Santos",  
  idade: 25,  
  casado: false  
}
```

Objetos - Obter uma propriedade

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  apelido: "Santos",
  idade: 25,
  casado: false
}

// Para obter o valor do campo "nome"
console.log(objeto.nome);
>> "Francisco"
```

Objetos - Obter uma propriedade

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  apelido: "Santos",
  idade: 25,
  casado: false
}

// Para obter o valor do campo "idade"
console.log(objeto.idade);
>> 25
```

Objetos - Definir valores

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  apelido: "Santos",
  idade: 25,
  casado: false
}

// Para adicionar um valor
objeto.altura = 172;

// Para alterar um valor
objeto.idade = 26;
```

Objetos - String como chave

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  apelido: "Santos",
  idade: 25,
  casado: false
}

// adicionar um valor, usando uma String como chave
objeto['altura'] = 172;

// usando o valor de uma variável como chave
let chave_altura = 'altura';
objeto[chave_altura] = 26;
```


Objetos - Remover propriedades

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  idade: 25,
  casado: false
}

// Para remover um valor
delete objeto.casado;
```

Objetos - Obter chaves

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  idade: 25,
  casado: false
}

// Para obter todas as chaves
console.log(Object.keys(objeto));
>> ["nome", "idade", "casado"]
```

Objetos - Obter valores

```
// Para criar um objeto
let objeto = {
  nome: "Francisco",
  idade: 25,
  casado: false
}

// Para obter todos os valores
console.log(Object.values(objeto));
>> ["Francisco", 25, false]
```

Exercício 3 - Objetos

- Criar uma função que recebe um objeto com as propriedades de **nome** e **anoNascimento** e devolve um objeto com o **nome** e a **idade**:

```
function converterIdade(pessoa) {  
    ...  
}
```

```
>> {  
  nome: "Francisco",  
  idade: 24  
}
```

```
let pessoa = {  
  nome: "Francisco",  
  anoNascimento: 1998  
}  
  
converterIdade(pessoa);
```

O futuro profissional começa aqui

iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA

**emprego
digital**

UPskill