

Javascript

Revisões – Parte 1

Integração do JavaScript numa página web

a) Código dentro do HTML (interno)

Podemos escrever JavaScript diretamente no ficheiro HTML usando a tag `<script>`:

```
<html>
<head>
  <title>Exemplo</title>
</head>
<body>

  <script>
    alert("Olá!");
  </script>

</body>
</html>
```

b) Código num ficheiro externo

```
<script src="script.js"></script>
```

Sintaxe geral do JavaScript

- As instruções podem terminar com ponto e vírgula ;.
- Um bloco de código é normalmente definido entre chavetas { }.
- O JavaScript distingue maiúsculas de minúsculas, ou seja, é case sensitive.
- Strings são escritas entre aspas (" ") ou entre plicas (' ').
- Os comentários servem para explicar o código e não são executados. Comentários numa linha, indicam-se por //
Comentários em várias linhas iniciam-se por /* e terminam com */

Comentários em JavaScript

Comentário de uma linha:

```
// Isto é um comentário  
let x = 10;
```

Comentário de várias linhas:

```
/*Isto é um comentário  
com várias linhas*/  
let y = 20;
```

Variáveis

Em Javascript as variáveis são declaradas com as instruções `let` ou `var`.

```
let nome = "Ana";  
let idade = 15;
```

Sempre que o valor de uma variável nunca é alterado no programa, esta pode (e deve) ser declarada como `const`.

```
const PI = 3.14;
```

Tipos de dados básicos

String `let cidade = "Almada";`

Número `let preco = 12.5;`
 `let quantidade = 4;`

Boolean `let ligado = true;`

Operadores

Operadores aritméticos: +, -, *, /

```
console.log(10 / 2); // 2
```

Operador de atribuição: =, +=, -=, *=, /=

```
let x = 7;  
x += 2;    //é o mesmo que x = x+2;
```

Incremento e decremento: ++, --

```
x++; //é o mesmo que x = x+1;
--y; //é o mesmo que y = y-1;
```

Operadores de comparação: ==, !=, >, >=, <=, <

```
console.log(10 > 5); // true
```

Entrada e saída de dados

Saída de dados

Mostrar informação ao utilizador:

```
alert("Olá!");  
console.log("Olá!");  
document.write("Olá!");
```

Entrada de dados

Podemos pedir um valor ao utilizador:

```
let nome = prompt("Qual é o teu nome?");  
alert("Olá, " + nome);
```

O `prompt()` mostra uma caixa onde o utilizador pode escrever.

`prompt()` retorna uma string. Podemos converter a string para int usando a função `parseInt()` ou para float usando a função `parseFloat()`.

```
let valor = prompt("Indica um valor");  
let dobro = parseInt(valor) * 2;  
alert("O dobro é: " + dobro);
```

Estruturas condicionais

As estruturas condicionais permitem tomar decisões.

Exemplo:

```
let idade = 18;

if (idade >= 18) {
    console.log("Maior de idade");
} else {
    console.log("Menor de idade");
}
```

Estruturas condicionais – switch

O `switch` é uma estrutura que permite **escolher entre várias opções**, com base no valor de uma variável.

É uma alternativa ao uso de vários `if...else if`.

Estrutura do `switch`

```
switch (variavel) {  
    case valor1:  
        // código  
        break;  
  
    case valor2:  
        // código  
        break;  
  
    default:  
        // código se nenhum caso for verdadeiro  
}  

```

```
let dia = prompt("Escolhe uma opção: 1 ou 2");  
  
switch (dia) {  
    case 1:  
        console.log("Segunda-feira");  
        break;  
  
    case 2:  
        console.log("Terça-feira");  
        break;  
  
    default:  
        console.log("Outro dia");  
}
```

Ciclos

Ciclo for

Estrutura:

```
for (inicialização; condição; incremento) {  
    // código a repetir  
}
```

Exemplo

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

Resultado

1 2 3 4 5

Como funciona?

let i = 1 → começa em 1

i <= 5 → repete enquanto for verdadeiro

i++ → aumenta 1 em cada repetição

Ciclo while

Estrutura:

```
while (condição) {  
    // código  
}
```

Exemplo

```
let i = 1;  
  
while (i <= 5) {  
    console.log(i);  
    i++;  
}
```

Característica

O ciclo pode **não executar nenhuma vez**, se a condição for falsa logo no início.

Temos que atualizar a variável de controlo senão o ciclo por nunca terminar (ciclo infinito)

Ciclo do...while

Estrutura

```
do {  
    // código  
} while (condição);
```

Exemplo

```
let i = 1;  
  
do {  
    console.log(i);  
    i++;  
} while (i <= 5);
```

Diferença importante para o ciclo while:

O do...while executa **sempre pelo menos uma vez**, mesmo que a condição seja falsa.