

## Folha de Exercícios - JavaScript

- 1>** Faça uma análise aos trechos de código apresentados em cada uma das alíneas e antecipe qual será o resultado, devendo ter em consideração que poderão existir erros de código. Por fim, copie o código para o ficheiro `exercicios.js` (recorrendo ao VSCode), execute-o, e confirme com o resultado apresentado no *output do Visual Studio Code*.

Nota: Use o código `'use strict'` no início do código durante a resolução de exercícios.

- a.** Analise o seguinte código e confirme o resultado na consola do VSCode.

```
'use strict';
let a=3;
let b=6;
let c;
console.log("1 =",a+b);
console.log("2 =",a+"c");
console.log("3 =",a+"5");
console.log("4 =",a+"b");
console.log("5 =",a+c);
console.log("6 =",c);
console.log(`Variavel a*b = ${a*b} ( a=${a} e b=${b})`);
```

- b.** Faça o mesmo procedimento para o seguinte bloco de código.

```
'use strict';
let a=3;
let b=6;
let c;
c=a+b;
console.log("1 =",c);
c=a+"---"+b;
console.log("2 =",c);
c="6";
console.log("3 =",a+c);
console.log("4 =",b===c);
console.log("5 =",b==c);
console.log("6 =",b!=c);
console.log("7 =",b!=c);
console.log("8 =",a++);
console.log('9 = ',a);
a=4;
console.log('10 = ',++a);
console.log('11 = ',+a);
```

c. Existe algum erro no seguinte trecho de código? Se sim, qual?

```
const disciplina;
disciplina="Linguagens Script";
console.log(disciplina);
```

d. O que é apresentado na consola? Identifica algum erro?

```
'use strict';
let uc='Linguagens';
uc+='Script';
console.log('Disciplina:'+uc +' - 2 semestre');
```

e. Faça o mesmo procedimento para o seguinte bloco de código.

```
'use strict';
const nome1='Nuno'
const nome2='Ricardo'
const resultado = `Os nomes são ${nome1} e ${nome2}`
console.log(resultado);
console.log(resultado+' e Filipe');
```

**2>** Analise e antecipe o resultado de cada um dos seguintes trechos de código. **Nota:** poderão existir erros que impeçam a execução completa do código. Confirme se o resultado obtido na consola é o esperado. Esclareça as suas dúvidas.

**Tome especial cuidado ao uso/declaração das variáveis e à sua localização, tendo em mente o funcionamento e características da linguagem JavaScript.**

a. Qual o resultado do seguinte trecho de código?

```
'use strict';
let n = 50
if (true) {
  let n = 2
  console.log(n)
}
console.log(n);
```

b. Qual o resultado do seguinte trecho de código?

```
'use strict';
let n = 50
if (true) {
  console.log(n);
  n = 2
  console.log(n)
}
console.log(n);
```

c. Qual o resultado do seguinte trecho de código?

```
'use strict';
let n = 50
if (true) {
  console.log(n);
  let n = 2
  console.log(n)
}
console.log(n);
```

d. Qual o output?

```
'use strict';
let str = 'Linguagens Script';
function fazQualquerCoisa() {
  console.log(str);
}
fazQualquerCoisa();
```

e. Qual o output?

```
'use strict';
let str = 'Linguagens';
function fazQualquerCoisa() {
  str = 'Script';
}
console.log(str);
fazQualquerCoisa();
console.log(str);
```

f. Veja a diferença com ou sem o recurso à instrução 'use strict';

```
'use strict';
function fazQualquerCoisa() {
  str = 'Script';
}
fazQualquerCoisa();
console.log(str);
```

g. Qual o output?

```
'use strict';
function fazQualquerCoisa() {
  let str = 'Script';
}
fazQualquerCoisa();
console.log(str);
```

**h.** Qual o output?

```
'use strict';
let str = 'Linguagens';
function fazQualquerCoisa() {
    let str2 = ' Script';
    console.log(str+str2);
}
fazQualquerCoisa();
console.log(str+str2);
```

**i.** Qual o output na consola?

```
var str = 'Linguagens';
function fazQualquerCoisa() {
    var str2 = ' Script';
    if (str==='Linguagens') {
        var dim='ok';
        console.log("->" +dim);
    }
    console.log(str+str2+"- "+dim);
}
fazQualquerCoisa();
console.log(str+str2);
```

**j.** Analise e verifique qual o objetivo do seguinte bloco de código.

```
'use strict';
let str = 'Linguagens';
function fazQualquerCoisa() {
    let str2 = ' Script';

    if (str.length > str2.length) {
        let dim="Primeira é maior!"
        console.log(dim);
    }
    else if (str.length === str2.length) {
        let dim="São iguais!"
        console.log(dim);
    }
    else {
        let dim="Segunda é maior!"
        console.log(dim);
    }

    console.log(str+str2+"-"+dim);
}
fazQualquerCoisa();
```

k. Qual o resultado do seguinte trecho de código?

```
function mensagem() {
  let nome='José';
  console.log(`Olá ${nome}`);
}
mensagem();
mensagem('Maria');
mensagem('Maria','Jose','Vieira');
```

l. Qual o resultado do seguinte trecho de código?

```
mensagem();
function mensagem() {
  let nome='José';
  console.log(`Olá ${nome}`);
}
```

m. Qual o resultado do seguinte trecho de código?

```
function mensagem(nome='') {
  console.log(`Olá ${nome}`);
}
mensagem();
mensagem('Maria');
mensagem('Jose');
mensagem('Cristiana','Areias');
```

## Jogo do Galo

3> Implemente as seguintes alíneas em JavaScript de forma a simular o tradicional **jogo do galo**, sem recurso a qualquer interface gráfica para o mesmo.

Para isso, considere as seguintes variáveis:

```
let jogadorAtual = "0";

let tabuleiro = [
  [" ", " ", " ", " ", " "],
  [" ", " ", " ", " ", " "],
  [" ", " ", " ", " ", " "],
];
```

- a.** Implemente a função **imprimeTabuleiro** cujo objectivo é imprimir o tabuleiro na consola. Deverá, como exemplo, apresentar o seguinte aspeto(neste caso sem qualquer jogada):

```

| | |
| | |
| | |

```

- b.** Implemente a função **jogada** que recebe por parâmetro o valor da **linha** e **coluna**, e altere o tabuleiro, com o valor do jogador atual, nessa coordenada.

- > A jogada só poderá ser aceite se, na posição em questão, ainda não existirem jogadas efetuadas.
- > Quando uma jogada é aceite, será necessário alterar o jogadorAtual.
- > Deverão ser apresentadas as mensagens “Posição ocupada. Jogue novamente!” e “Jogador Atual=??” no momento correcto.

- c.** Implemente a função **verificaFimJogo**.

- > A função deverá devolver 1 caso seja fim de jogo, caso contrário, deverá devolver 0.
- > Altere a função **jogada** de forma a verificar o fim de jogo sempre que uma jogada é efetuada e apresente a devida mensagem.

- d.** Implemente a função **verificaVencedor**.

- > A função deverá devolver o código do vencedor ou 1 se houver empate.
- > Altere a função **jogada** de forma a verificar o vencedor e apresente a devida mensagem.

- e.** Teste o seu código, invocando as seguintes funções, o qual deverá terminar o jogo e apresentar o vencedor.

```
jogada(1, 1);
```

```

| | |
| X |
| | |

```

```
jogada(0, 0);
```

Proximo Jogador -> O

```

O | | |
| X |
| | |

```

```
jogada(0, 0);
```

Proximo Jogador -> X  
Posição ocupada. Jogue novamente!

```

O | X |
| X |
| | |

```

Proximo Jogador -> O

jogada(0, 1);

|   |   |  |
|---|---|--|
| O | X |  |
|   | X |  |
| O |   |  |

Proximo Jogador -> X

jogada(2, 0);

|   |   |   |
|---|---|---|
| O | X | X |
|   | X |   |
| O |   |   |

Proximo Jogador -> O

Posição ocupada. Jogue novamente!

|   |   |   |
|---|---|---|
| O | X | X |
|   | X | O |
| O |   |   |

Proximo Jogador -> X

jogada(2, 1);

|   |   |   |
|---|---|---|
| O | X | X |
| X | X | O |
| O |   |   |

Proximo Jogador -> O

|   |   |   |
|---|---|---|
| O | X | X |
| X | X | O |
| O | O |   |

Proximo Jogador -> X

|   |   |   |
|---|---|---|
| O | X | X |
| X | X | O |
| O | O | X |

Fim de Jogo! Jogo empatado!