

> Ficha Prática Nº1 (Javascript – Introdução)

Notas:

- O editor selecionado para as aulas práticas é o *Visual Studio Code (VSCode)*, no entanto, os alunos podem optar por outro editor de preferência.
- O *download* da instalação do VSCode pode ser efetuado em <https://code.visualstudio.com/>. Caso não pretendam efetuar instalação, podem usar a versão online disponível em <https://vscode.dev/>
- As resoluções das fichas das aulas práticas **não são** para avaliação e, portanto, não é necessário efetuar qualquer entrega ao professor.

> Preparação do ambiente:

- Efetue o download e descompacte o ficheiro *ficha1.zip* disponível no *inforestudante*.
- Inicie o *Visual Studio Code*, abra a pasta no *workspace* e visualize a página *ficha1.html* no browser, no qual apenas deverá apresentar o texto “**Linguagens Script - Ficha Prática Nº1**”.
Nota: Caso não tenha disponível, pode instalar extensões como “Live Server” e/ou “Open in Browser”.
- Aceda às *Developer Tools* do browser (ex. clicando F12 Chrome).
- Selecione a tab “Console” onde poderá ver os erros de *Javascript* da página ou simplesmente utilizar essa área como um editor de *Javascript*. Neste momento, não existe qualquer erro assinalado na Figura 1. Na figura 2, como pode verificar, é identificado um erro existente na linha 2, do ficheiro *Ficha1.js*.

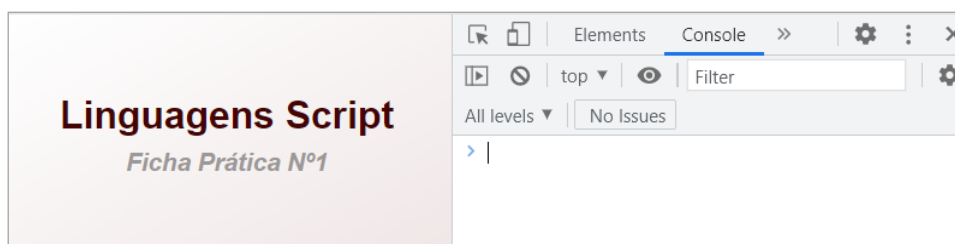


Figura 1 – Console sem Erros

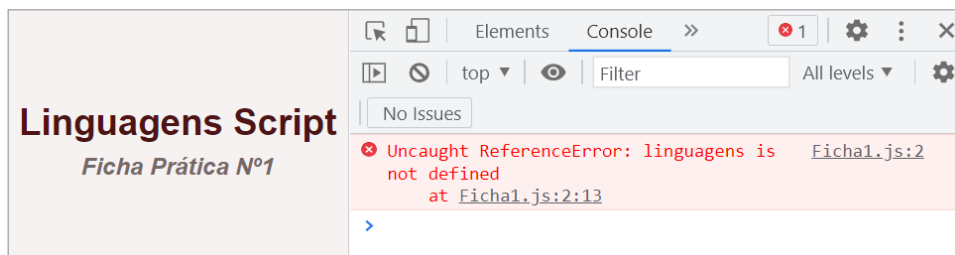


Figura 2 – Console com erros

Parte I – Introdução

- 1>** Faça uma análise dos trechos de código de cada uma das alíneas, antecipando qual será o resultado que será apresentado na consola do browser, devendo ter em conta que poderão existir erros. Por fim, copie o código para o ficheiro Ficha1.js (existente na pasta js), grave o documento, e confirme se o resultado apresentado na Consola do browser corresponde ao que previa. Não remova o código 'use strict' que já existe no ficheiro.

Retire as suas conclusões e esclareça as suas dúvidas.

- a.** Considere o seguinte código.

```
let n1=3;
let n2=6;
let n3;
console.log(n1+n2);           // 1)
console.log(n1+"n3");         // 2)
console.log(n1+"5");          // 3)
console.log(n1+"n2");         // 4)
console.log(n1+n3);           // 5)
console.log(n3);              // 6)
console.log(`Variavel n1*n2 = ${n1*n2} ( n1=${n1} e n2=${n2})`); //7)
n3=n1+n2;
console.log(n3);              //8)
n3=n1+"---"+n2;
console.log(n3);              //9)
n3="6";
console.log(n1+n3);           // 10
console.log(n2===n3);         // 11   True ou False ?
console.log(n2==n3);          // 12   True ou False ?
console.log(n2!==n3);         // 13   True ou False ?
console.log(n2!=n3);          // 14   True ou False ?
console.log(n1++);            // 15
console.log('n1='+n1);         // 16
n1=4;
console.log(++n1);            // 17
console.log('n1='+n1);         // 18
```

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

```
let variavel;
variavel = '3' + 2;
console.log(variavel)
variavel = '3' + true;
console.log(variavel);
variavel = '3' + undefined;
console.log(variavel);
variavel = '3' + null;
console.log(variavel);
```

- c. Existe algum erro no seguinte trecho de código?

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

- d. O que é apresentado na consola? Algum erro?

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

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

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

2> Resolva os seguintes exercícios.

- a. Implemente o código para calcular o maior de três números, devendo apresentar na consola a mensagem “O maior entre ??,?? e ?? é ??”. Declare as variáveis no início da seguinte forma:

```
const num1=5;
const num2=10;
const num3=7;
```

Teste o seu código agora valores obtidos pelo utilizador. Para isso declare as variáveis da seguinte forma:

```
const num1=prompt("Especifique o numero 1: ");
const num2=prompt("Especifique o numero 2: ");
const num3=prompt("Especifique o numero 3: ");
```

NOTA: `prompt(texto)` retorna uma *string* com os dados introduzidos pelo utilizador. Assim, caso pretendam obter o valor inteiro, devem usar o `parseInt(string)` para converter a string num inteiro.

- b. Implemente o código necessário para efetuar a soma de todos os números entre dois valores. Esses dois valores devem ser inicializados no início do programa. Por fim, o programa deverá apresentar a soma obtida.
- c. Implemente o código necessário para somar todos os números positivos obtidos do utilizador. Quando o valor introduzido for 0, o programa deverá terminar e apresentar uma mensagem com a **soma total**, e o número total de números (positivos e negativos) introduzidos.

Parte II – Global / Local / Block Scope e Hoisting

- 3>** Analise e diga qual 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 como funciona o Javascript.

- a.** Qual o output?

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

- b.** Qual o output?

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

- c.** Qual o output com ou sem o recurso à instrução 'use strict'; ?

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

- d.** Qual o output?

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

- e.** Qual o output?

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

f. Analise e verifique o que faz o 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();
```

g. 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);
```

h. Qual o resultado do seguinte trecho de código? E se mudar o var para let ? Alguma diferença?

```
console.log(str)
var str;
```

i. Qual o resultado do seguinte trecho de código? E se mudar o var para let ? Alguma diferença?

```
console.log(str)
var str='Linguagens Script';
```

j. Qual o resultado do seguinte trecho de código? E se mudar o var para let ? Alguma diferença?

```
str='Linguagens Script';
console.log(str)
var str;
```

k. Qual o resultado do seguinte trecho de código? E se mudar o var para let ? Alguma diferença?

```
function fazQualquerCoisa() {
    console.log(str);
}
fazQualquerCoisa();
var str = 'Linguagens';
```

l. Qual o resultado do seguinte trecho de código? Repare que neste caso, a variável str2 não se torna global, ainda que tenha sido declarada com o var.

```
var str="Linguagens"
function fazQualquerCoisa() {
    str2="Script"
    console.log(str2);
    var str2;
}
fazQualquerCoisa();
console.log(str2);
```

m. Qual o resultado do seguinte trecho de código?

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

n. Qual o resultado do seguinte trecho de código?

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

o. Qual o resultado do seguinte trecho de código?

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

p. Qual o resultado dos seguintes trechos de código?

- 1)

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

```
let n = 50
if (true) {
  console.log(n);
  n = 2
  console.log(n)
}
console.log(n);
```
- 3)

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

Parte III – Resolução de exercícios com funções

4> Resolva os seguintes exercícios, recorrendo a função, sempre que considerar o mais correcto.

- a.** Implemente a função `compara` que verifique se dois números são iguais. Deve retornar `true` se forem iguais e `false` caso contrário. Implemente a função com e sem recurso ao *operador ternário*.
- b.** Implemente a função `parOuImpar` que, recebendo como parâmetro um valor, escreva na consola se o número é par ou impar. Use o operador `%`

```
parOuImpar(5); //Deverá escrever 'O número é impar!'
parOuImpar(4); //Deverá escrever 'O número é par!'
```

- c.** Implemente uma função `maior` que devolva o quadrado de um valor.

```
console.log(obtemQuadrado(2)) //Será apresentado 4 na consola
console.log(obtemQuadrado(9)) //81
console.log(obtemQuadrado(10)) //100
```

- d. Implemente uma função maior que devolva a área e um retângulo. Se for passado apenas um valor, então, os dois valores devem ser considerados iguais. Resolva sem recorrer a qualquer if.

```
console.log(areaRetangulo(5,10)) // 50
console.log(areaRetangulo(10,20)) // 200
console.log(areaRetangulo(5)) // 25
```

- e. Implemente a função **contavogais** que receba por parâmetro uma *string* e devolva o número de vogais existentes nessa string. Pode recorrer aos métodos `charAt()`, `toLowerCase()`, métodos estes que permitem verificar o que se encontra numa determinada posição do carácter e converter tudo para minúsculas, respetivamente.

```
console.log(contaVogais("Ola")) //2
console.log(contaVogais("Linguagens Script")) //5
```

Parte IV– Arrays

5> Resolva os seguintes exercícios.

- a. O seguinte código apresenta a criação de um *array* (literal) de 6 elementos, no qual é apresentado na consola, o primeiro elemento do *array*, o texto angular. Implemente o código para que escreva todos os elementos do *array*, recorrendo a um ciclo for.

```
const palavras=['angular','bootstrap','javascript','vue','svelte','react'];
console.log(palavras[0]);
```

- b. Implemente a função **ImprimeArray** que recebe por parâmetro um *array* de *strings* e imprima os elementos desse *array* na consola (adaptar o código realizado na alínea anterior para uma função).
- c. Resolva o exercício anterior, recorrendo agora ao **for... of**. Sintaxe:

```
for (elemento of array ) {
    //...
}
```

- d. Resolva o exercício anterior, recorrendo agora ao *foreach*. Sintaxe:

```
palavras.forEach(imprimeArray);
function imprimeArray(el, index, arr) {
    //...
}
```

- e. Implemente código de forma a que adicione um novo elemento ao *array*, o elemento **'backbone'**.
- f. Existe um conjunto de métodos *built-in* que permitem a manipulação de elementos de um *array*, entre eles o **push()** e o **unshift()** que permitem adicionar de forma simples novos elementos no

array. Teste estes métodos, adicionando os elementos **ember** e **css** ao array. Por forma a verificar o que aconteceu, imprima o array na consola recorrendo à função usada nas alíneas anteriores. Dica: `nomearray.push('ember');`

- g.** Analise o comportamento dos seguintes métodos **pop()** e **shift()**, usando os *arrays* anteriormente definidos.
- h.** Ordene o array **palavras** por ordem alfabética, recorrendo ao método **sort()**.
- i.** Analise o seguinte trecho de código e antevêja o que aparece na consola do browser. Depois, copie o código e verifique o resultado.

```
const palavras1=['css','html'];
const palavras2=['angular','bootstrap','javascript'];
const palavras3=['vue','svelte','react'];
function fazQualquerCoisa(...arr) { // não é erro
    for (let i=0; i <arr.length;i++) {
        for (let j=0; j<arr[i].length;j++)
            console.log(arr[i][j]);
    }
    console.log('---FIM-----');
}
fazQualquerCoisa(palavras1);
fazQualquerCoisa(palavras1,palavras2);
fazQualquerCoisa(palavras1,palavras2, palavras3);
```

- j.** Implemente a função **somatorio** que calcule o somatório de um conjunto de números, passados por parâmetro. Note que podem ser passados 1, 2, ou n números.

```
console.log(somatorio(1,2,3,4,1));
console.log(somatorio(1,5,5));
console.log(somatorio(11));
```

- k.** Analise o seguinte código e antevêja o que aparece na consola do browser. Depois, copie o código e verifique o resultado. Esclareça as suas dúvidas.

```
let vogais = ['a','e','i','o','u'];
let letras = vogais;
letras.push('X');

console.log(vogais); // Escreve apenas vogais ou vogais + X ?
console.log(letras); // Escreve apenas vogais ou vogais + X ?

let letra = 'Z';
let letraZ = letra;
letra='Y';
console.log(letra); // Escreve Z ou Y?
console.log(letraZ); // Escreve Z ou Y?
```

1. O método **map** permite efetuar alguma operação em cada um dos elementos de um array e transformá-lo num novo array. Nesse sentido, implemente código que permita criar um novo array, onde além de apresentar o nome das palavras, também deve apresentar a sua posição no array como se apresenta no **arr_final**

```
original=['angular','bootstrap','javascript','vue','svelte','react'];  
arr_final=['[0]=angular','[1]=bootstrap','[2]=javascript','[3]=vue',  
          '[4]=svelte','[5]=react'];
```

Nota: Não se pretende que recorra a qualquer ciclo while, for, foreach, do.. while.

- m. Seguindo o raciocínio da alínea anterior, implemente o código necessário para que o array final apenas inclua palavras com menos que 7 caracteres, como apresentado de seguida. Tal como na alínea anterior, não se pretende que implemente com recurso a qualquer ciclo for, while, foreach...

```
arr_final=['[vue]','[svelte]','[react]'];
```

- 1) Implemente com recurso ao método **filter**
- 2) Implemente com recurso ao método **reduce**