**Exercício 1**

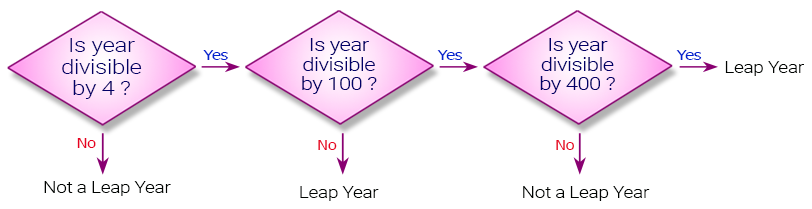
Definir uma variável com um número e calcular o factorial desse número, isto é, a multiplicação de todos os números inteiros entre 1 e o número.  
Por exemplo o factorial de 5 é o resultado da operação 1x2x3x4x5=120  
Não existe factorial de números negativos e 0! = 1.

Testar estas hipóteses.

**Exercício 2**

Dada uma variável com um ano,let ano=2004;determinar se o ano é bissexto.

Ver imagem anexa.NÃO PROCURAR A SOLUÇÃO NA INTERNET!!!!!!



**Exercício 3**

Dada uma variável do tipo string, validar se obedece às seguintes condições, apresentando uma mensagem de erro:

* não pode ser vazia
* tem de ter, no mínimo, 3 caracteres
* o primeiro caracter tem de ser A ou B

Utilizar os seguintes valores para testar:

let valor1=""; // false

let valor2="Primavera"; // false

let valor3="Be"; // false;

let valor4="Amesterdão"; // true

let valor5="Benin"; // true

**Exercício 4**

Simular o método repeat, isto é, dadas duas variáveis,

let repeticoes=3;

let texto="Primavera";

Criar uma variável (e mostrar o resultado no ecrã) que contenha o valor da variável "texto" repetido 3 vezes (é o valor da variável "repeticoes"):

PrimaveraPrimaveraPrimavera

Testar o algoritmo para valores diferentes das variáveis "repetições" e "texto".

**Exercício 5**

Simular o método trim() que remove os espaços que ocorrem no início e no fim de uma string.Exemplo:let frase=" Frase com espaços no início e no fim ";Deveremos ter como resultado "Frase com espaços no início e no fim".

**Exercício 6**

Validar se um email fornecido pelo utilizador tem o formato correcto, apresentando uma mensagem de erro, caso não seja válido.NÃO UTILIZAR EXPRESSÕES REGULARES!Condições:

* Tem de possuir uma @
* Antes da @ tem de existir 1 caracter
* Depois da @ tem de existir um caracter
* Depois da @, mas espaçado, pelo menos 1 caracter, tem de existir 1 ponto
* Depois do ponto, têm de existir, pelo menos 2 caracteres

Caso não seja fornecido nenhum valor deve apresentar a mensagem "É necessário fornecer um email".Utilizar os seguintes valores para testar:

* let email1=""; // "É necessário fornecer um email"
* let email2="primavera"; // false
* let email3="a@primavera"; // false
* let email4="premoaldo@.primavera."; // false
* let email5="premoaldo@primavera"; // false
* let email6="premoaldo@primavera."; // false
* let email7="premoaldo@primavera.b"; // false
* let email8="[premoaldo@primavera.pt](mailto:premoaldo@primavera.pt)"; // true

**Exercício 7**

Dada uma variável com o nome de um ficheiro,

let ficheiro="teste.txt";

extrair a extensão do ficheiro.O resultado deve ser ".txt" ou "txt". No caso de ser um ficheiro Word seria "doc" ou ".doc".Convém testar todas as hipóteses para a variável "ficheiro". Por exemplo, se não existe extensão, se a variável está vazia, se não contém uma string, etc.

**Exercício 8**

Sem recorrer ao método toUpperCase() converta uma string para maiúsculas.Exemplo, fornecendo a seguinte variável,

let nome="Pedro Remoaldo";

o resultado deve ser:

PEDRO REMOALDO

Ideia: cada caractere tem um código ASCII/Unicode específico - por exemplo, a letra "a" tem o código 97 e a letra "A" tem o código 65. Existe uma função em JavaScript que permite saber o código ASCII de um caracter.

**Exercício 9**

Identificar a maior palavra existente numa frase e o respectivo número de caracteres.Exemplo:

let frase="Exemplo de uma frase com palavras";

Deve apresentar como resultado

"palavras" - 8 letras

Podem utilizar métodos de strings.

**Exercício 10**

Gerar uma chave aleatória do euromilhões (5 números + 2 estrelas) - não podem existir números repetidos!

Os números podem variar entre 1 e 50, e as estrelas entre 1 e 12.

Utilizar o método Math.random().

A chave deve ser apresentada com os números/estrelas ordenados.

**Exercício 11**

Dado um array de temperaturas,

let temperaturas=[ 12, 34, 45, 3, 8, 23, 56 ];

calcular a temperatura mínima, máxima e a média das temperaturas.

**Exercício 12**

Remova os números repetidos de um array.

Por exemplo, partindo do array,

[12,6,3,12,8,6,5]

o resultado deverá ser:

[12,6,3,8,5]

**Exercício 13**

Criar uma função que receba como parâmetro o nome de um ficheiro, e que determine se a extensão é válida para o fim em causa - deverá devolver true ou false.

function validaExtensao(nomeFicheiro) {

let extensoesValidas=...

...

}

let ficheiro="dados.xls";

console.log(validaExtensao(ficheiro));

Deverá existir uma lista com as extensões válidas.

Exemplo de lista de extensões válidas: xls, doc, bmp, jpg  
Caso não seja passado o nome do ficheiro, deve ser devolvido false.

**Exercício 14**

Criar uma função, validaNome(), que simule a validação dos nomes de variáveis em JavaScript.

Isto é, a função recebe um parâmetro "nomeVariavel" e tem de determinar se o nome da variável é autorizado.Regras:

* só pode conter letras (maiúsculas e minúsculas, mas não pode conter os caracteres acentuados e o ç), algarismos, e os caracteres $ e \_
* não pode começar por um algarismo
* não pode ser uma palavra reservada de JavaScript - break, case, catch, continue, debugger, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with, class, const, enum, export, extends, import, super, implements, interface, let, package, private, protected, public, static, yield, true, false, null, NaN, Infinity e undefined

**Exercício 15**

Criar uma função, remove\_elementos(), que localiza e remove elementos de um array.  
Exemplo:

let nomes=["Pedro", "Mário", "João", "Filipe", "Fernando"];

remove\_elementos("Mário");

O array "nomes" tem agora os seguintes elementos:

["Pedro", "João", "Filipe", "Fernando"]

**Exercício 16**

Sem utilizar nenhum dos métodos dos arrays para encontrar elementos (como find, findIndex, includes e indexOf), criar uma função, chamada existe(), que verifica se um determinado elemento existe num array. Esta função recebe dois parâmetros - o array e o elemento a encontrar - e deve devolver true ou false.  
É necessário validar se a função foi invocada com dois parâmetros e com o tipo de dados correcto (array, string).Exemplo de utilização da função:

let formandos = ["Edgar","Frederico","Anísio","Djony","Pedro"];

if (existe(formandos,"Luís")) {

console.log("Existe");

} else {

console.log("Não existe");

}

Criar outra versão da função em que existem as seguintes alterações:

* case insensitive - não distinguir maiúsculas das minúsculas
* mostrar a posição em que o nome ocorre no array

**Exercício 17**

Criar uma função palavraAEncontrar, que devolve o número de ocorrências de uma letra (passada como parâmetro) numa frase.  
Por exemplo:

function palavraAEncontrar() {

...

}

let frase="O filho foi visitar o pai.";

palavraAEncontrar("o");

O resultado será 4, que é o número de vezes que a letra "o" ocorre na frase (em maiúsculas ou minúsculas).  
Podem utilizar métodos de strings.

**Exercício 18**

Criar uma função que receba como parâmetro uma string e devolva o número de palavras existentes na string.  
Como variante deste exercício, devolver um array com as palavras existentes na string.

**Exercício 19**

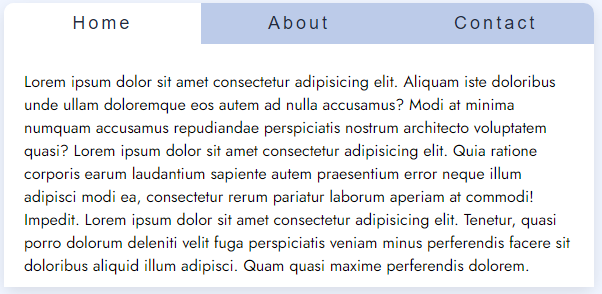
Criar uma função que recebe como parâmetro um número, por ex: 1586, e que devolve a sua representação por extenso ("mil quinhentos e oitenta e seis").  
O número máximo que a função pode receber é 9999.

**Exercício 20**

Tendo como base o exercício de gestão de tarefas realizado na sessão de formação, criar uma gestão de utilizadores, em que cada utilizador tem a seguinte informação associada: nome, email e telefone.  
A informação dos utilizadores deve ser apresentada numa tabela HTML (<table>) e também deve ser guardada num array.  
Implementar a pesquisa de utilizadores por nome.

**Exercício 21**

Implementar um componente tabs (ver imagem em anexo).  
Quando se clica num dos tabs é mostrado o respectivo conteúdo, e o tab fica "realçado" para indicar que é o tab activo. Os conteúdos dos outros tabs não seleccionados devem ser escondidos e os tabs devem aparecer com as cores "normais".



**Exercício 22**

Implementar um componente accordion - na prática, é um conjunto de collapsibles.  
Ver exemplo em:<https://getbootstrap.com/docs/5.1/components/accordion/#example>

Apenas um dos conteúdos deve estar visível. Isto é, quando se clica num dos títulos para abrir um conteúdo, todos os outros conteúdos devem ser fechados.  
O título clicado deve ficar com uma cor de fundo e cor de texto diferentes.

**Exercício 23**

Dado um array com números,

let dados = [90,190,20,240,340,220,470,60,430,460,400];

ordená-los por ordem crescente sem recorrer ao método sort().Podem fazer a ordenação no próprio array ou utilizar um array auxiliar.Investigar o método bubble sort - ver imagem em anexo.

bubble-sort.png

