

ESMAD | TSIW | POO
Ficha de Exercícios nº2 – Funções

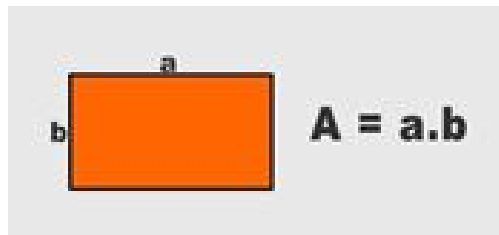
Crie uma função para resolver cada uma destas alíneas. Para a codificação é aconselhado que use um playground Web:

1. Declaração de função

- Crie uma função que apresente numa caixa de alerta a frase “OLÁ MUNDO!”. Dê um bom nome à função. Execute a função 3 vezes

2. Parâmetros

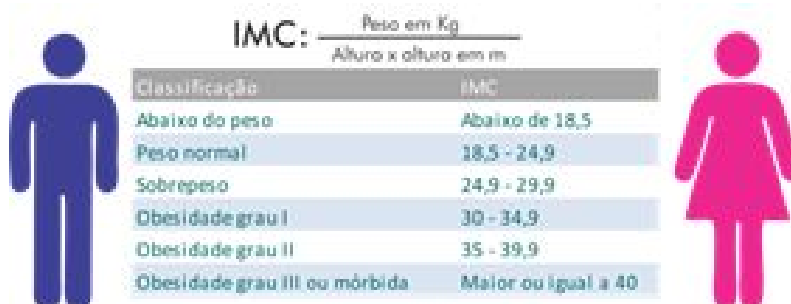
- Crie uma função que imprima “OLÁ [name]!”, onde **name** é uma variável passada como um parâmetro. A variável **name** deve ser inicializada “hard-coded”
- Crie uma função que imprima o resultado do cálculo da área de um retângulo passados por parâmetro o comprimento dos dois lados



- Crie uma função que apresente numa caixa de alerta o resultado de operações aritméticas. A função deve receber dois valores inteiros e um operador (+, -, *, /). Todos os valores devem ser previamente obtidos através de funções **prompt**. Salvaguarde a divisão por 0 apresentando ao utilizador uma mensagem a informar que não é possível efetuar a divisão



- d. Crie uma função que apresente numa caixa de alerta a classificação do IMC (Índice de Massa Corporal) de uma pessoa recebendo como parâmetros o peso e a altura respetivos. Ambos os parâmetros devem ser recolhidos através de funções **prompt**



- e. Crie uma função que simule um eco. Dada uma string **s** e um número **n** deve imprimir **s** escrito **n** vezes
- f. Crie uma função que comece por ler do utilizador dois valores inteiros **a** e **b**, e que escreva todos os valores inteiros pertencentes ao intervalo **[a,b]**
- g. Crie uma função que calcule a soma dos múltiplos de 3 existentes num intervalo **[a,b]**, em que **a** e **b** são passados como parâmetros
- h. Crie uma função que devolva a tabuada de um determinado número passado como parâmetro. Se nenhum número for passado a função deve imprimir a tabuada do 1

1 X 1 = 1	2 X 1 = 2	3 X 1 = 3	4 X 1 = 4	5 X 1 = 5
1 X 2 = 2	2 X 2 = 4	3 X 2 = 6	4 X 2 = 8	5 X 2 = 10
1 X 3 = 3	2 X 3 = 6	3 X 3 = 9	4 X 3 = 12	5 X 3 = 15
1 X 4 = 4	2 X 4 = 8	3 X 4 = 12	4 X 4 = 16	5 X 4 = 20
1 X 5 = 5	2 X 5 = 10	3 X 5 = 15	4 X 5 = 20	5 X 5 = 25
1 X 6 = 6	2 X 6 = 12	3 X 6 = 18	4 X 6 = 24	5 X 6 = 30
1 X 7 = 7	2 X 7 = 14	3 X 7 = 21	4 X 7 = 28	5 X 7 = 35
1 X 8 = 8	2 X 8 = 16	3 X 8 = 24	4 X 8 = 32	5 X 8 = 40
1 X 9 = 9	2 X 9 = 18	3 X 9 = 27	4 X 9 = 36	5 X 9 = 45
1 X 10 = 10	2 X 10 = 20	3 X 10 = 30	4 X 10 = 40	5 X 10 = 50
6 X 1 = 6	7 X 1 = 7	8 X 1 = 8	9 X 1 = 9	10 X 1 = 10
6 X 2 = 12	7 X 2 = 14	8 X 2 = 16	9 X 2 = 18	10 X 2 = 20
6 X 3 = 18	7 X 3 = 21	8 X 3 = 24	9 X 3 = 27	10 X 3 = 30
6 X 4 = 24	7 X 4 = 28	8 X 4 = 32	9 X 4 = 36	10 X 4 = 40
6 X 5 = 30	7 X 5 = 35	8 X 5 = 40	9 X 5 = 45	10 X 5 = 50
6 X 6 = 36	7 X 6 = 42	8 X 6 = 48	9 X 6 = 54	10 X 6 = 60
6 X 7 = 42	7 X 7 = 49	8 X 7 = 56	9 X 7 = 63	10 X 7 = 70
6 X 8 = 48	7 X 8 = 56	8 X 8 = 64	9 X 8 = 72	10 X 8 = 80
6 X 9 = 54	7 X 9 = 63	8 X 9 = 72	9 X 9 = 81	10 X 9 = 90
6 X 10 = 60	7 X 10 = 70	8 X 10 = 80	9 X 10 = 90	10 X 10 = 100

- i. Crie uma função que soma **N** números passado por parâmetros, Use o objeto **arguments** para resolver este problema
- j. Crie uma função que receba o primeiro e último nome de uma criança e um conjunto de strings que representam o nome de cada uma das suas amigas. A função deve apresentar a seguinte frase: "A **[primeiroNome]** **[ultimoNome]** tem **[nAmigas]** amigas!". Use o parâmetro **Rest** para resolver esta questão

3. Retorno de valor

- a. Crie uma função **min(a, b)** que retorne o mínimo de dois números **a** e **b**.
Casos de teste:

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

- b. Crie uma função **pow(x, n)** que retorne **x** na potência **n**. Ou, em outras palavras, multiplica **x** por si **n** vezes e retorna o resultado.
Casos de teste:

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ... * 1 = 1
```

Antes de chamar a função, deve solicitar **x** e **n** ao utilizador. A função deve suportar apenas valores naturais de **n**: inteiros acima de 1

- c. Crie uma função que verifique se um número passado como parâmetro é ou não primo. Relembre-se que um número primo **N** é um número natural maior do que 1 que não possui divisores além de 1 e de si mesmo. Retorne **true** caso seja primo e **false**, caso contrário. A função deve ser invocada dentro de uma estrutura **IF** e caso seja verdadeiro deve ser apresentado "O número **[N]** é primo". Caso contrário, "O número **[N]** não é primo"
- d. Crie uma função que retorne o fatorial de um valor inteiro positivo passado como parâmetro. Exemplos: $0!=1$; $1!=1$; $5!=5 \times 4 \times 3 \times 2 \times 1 = 120$; $6!=6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$;

4. Expressão de função

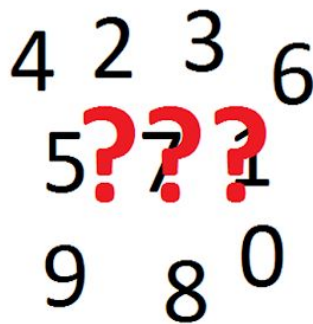
- a. Crie uma expressão de função chamada **isLeapYear** que deve retornar **true** se um ano passado pelo utilizador é bissexto e **false** caso contrário.



- b. Crie uma expressão função chamada **isPerfect** que verifique se um número é perfeito. Um número perfeito é um número natural para o qual a soma de todos os seus divisores naturais próprios (excluindo ele mesmo) é igual ao próprio número.

Por exemplo, o número 28 é, pois: $28=1+2+4+7+14$. Se for perfeito deve retornar **true**. Caso contrário, deve retornar **false**. Use a função num ciclo que deve continuar a pedir um número ao utilizador até que o número digitado seja perfeito.

- c. Crie uma expressão de função abreviada (função arrow) que dado um número de 100 a 999 verifique se o mesmo é um número palíndromo (capicua). Caso seja, deve devolver **true**. Caso contrário, deve devolver **false**. Escolha um bom nome para a função.
5. Faça o jogo da adivinha. Comece por gerar um número aleatório entre 1 e 100. Depois vá perguntando ao utilizador para adivinhar o número. Caso o utilizador digite um número superior deve exibir o seguinte texto **"PARA BAIXO"**. Caso contrário, deve indicar: **"PARA CIMA"**. Caso acerte, deve exibir a mensagem **"ADIVINHO, PARABÉNS!"**.



Neste jogo deve ter uma função que recebe dois parâmetros: o número a adivinhar e a tentativa do utilizador. A função deve retornar:

- 1 se a tentativa for INFERIOR ao número gerado inicialmente
- 1 se a tentativa for SUPERIOR ao número gerado inicialmente
- 0 se a tentativa for IGUAL ao número gerado inicialmente

Torne o jogo mais interessante, e forneça apenas 5 tentativas ao jogador. Caso atinja o limite, deve indicar: **"PACIÊNCIA, JOGUE OUTRA VEZ!"**.