

Funções

1. Faça uma função que recebe um número e imprime seu dobro.
2. Faça uma função que recebe o valor do raio de um círculo e retorna o valor do comprimento de sua circunferência: $C = 2 \cdot \pi \cdot r$.
3. Faça uma função para cada operação matemática básica (soma, subtração, multiplicação e divisão). As funções devem receber dois números e retornar o resultado da operação.
4. Faça uma função que recebe um nome e imprime "olá, [nome]".
5. Faça uma função que recebe um nome e um horário e imprime "Bom dia, [nome]", caso seja antes de 12h, "Boa Tarde, [nome]", caso seja entre 12h e 18h e "Boa noite, [nome]" se for após às 18h.
6. Faça uma função que recebe um número e retorna *True* se ele é par ou *False*, se ele é ímpar.
7. Faça uma função que sorteia 10 números aleatórios entre 0 e 100 e retorna o maior entre eles.
8. Faça uma função que recebe um número n de entrada, sorteia n números aleatórios entre 0 e 100 e retorna a média deles.
9. Faça uma função que recebe uma lista de palavras e retorna uma lista contendo as mesmas palavras da lista anterior, porém escritas em caixa alta.
10. Faça uma função que recebe duas listas e retorna a soma item a item dessas listas.

Exemplo: Se a função receber as listas [1,4,3] e [3,5,1], então a função deve retornar [1+3, 4+5, 3+1] = [4, 9, 4].
11. Faça uma função que receba duas listas e retorne o produto item a item dessas listas.

Exemplo: Se a função receber as listas [1,4,3] e [3,5,1], então a função deve retornar [1x3, 4x5, 3x1] = [3, 20, 3].

12. Faça uma função que recebe um número x e uma lista numérica e retorna uma lista cujos elementos são os itens da lista de entrada multiplicado por x .

Exemplo: se a função receber o número 5 e a lista [3,5,1], então a função deve retornar [5x3, 5x5, 5x1] = [15, 25, 5].

13. Faça uma função que recebe uma lista de números e retorna a soma dos elementos dessa lista.
14. Faça uma função que recebe uma lista de números e retorna a média aritmética dos elementos dessa lista.

Desafio

1. Faça uma função que receba um número e calcule seu fatorial.
2. Super Desafio! - Repita o exercício anterior usando recursão, ou seja, uma função que chame ela mesma, lembrando que $3! = 3 \cdot 2!$, que $2! = 2 \cdot 1!$, que $1! = 1 \cdot 0!$ e que $0! = 1$.
3. Faça uma função que recebe duas entradas: um input dado pelo usuário e um string que informa o tipo de dado ("idade", "salário" ou "sexo"), e verifica se os dados digitados foram válidos, usando os seguintes critérios:
 - a. Idade: entre 0 e 150;
 - b. Salário: maior que 0;
 - c. Sexo: M, F ou Outro.

4. A sequência Fibonacci é a sequência cujos dois primeiros termos são 1 e os demais são obtidos através da soma de seus dois antecessores, isso é:
- a. $\text{Fibonacci}(1) = 1$ e $\text{Fibonacci}(2) = 2$;
 - b. dado qualquer número $n \geq 3$, $\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$

Assim, os 10 primeiros termos da sequência Fibonacci são:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55...

Faça uma função que receba um número n e calcule o termo de número n da sequência Fibonacci.

5. Super Desafio! Refaça o desafio 4 usando recursão.
6. Super Desafio! - Faça um jogo de *BlackJack* usando funções: o *BlackJack*, ou *Vinte e Um*, é um jogo em que os jogadores podem comprar cartas livremente, enquanto tiverem menos de 21 pontos. No nosso jogo, o Ás vale um ponto; as cartas de 2 a 10 valem o número de pontos que elas representam; e Valete, Dama e Rei valem 10 pontos cada. Ganha o jogador que tiver o maior número de pontos, desde que este seja menor ou igual a 21. Nosso jogo deve ter as seguintes funções:
- a. Função principal: a função que vamos chamar para iniciar o jogo. Essa função não irá receber nem retornar nenhuma variável. Ela deve perguntar o número de jogadores participantes e o nome de cada um. Em seguida ela chama as outras funções do jogo.
 - b. Função para criar o baralho: essa função deve criar um baralho (uma lista) com as cartas do baralho.

- c. Função para a jogada: essa função deve receber o nome do jogador que irá realizar a jogada e, caso ele ainda esteja ativo (tenha menos de 21 pontos e ainda não tenha desistido de comprar cartas) deve perguntar se ele quer comprar uma carta. Se ele responder que sim, a função deve chamar a próxima função para sortear uma carta e somar o valor retornado na pontuação do jogador; se ele responder que não, a função deve desativar o jogador para que ele não possa mais comprar cartas; Essa função só deve ser chamada enquanto houver jogadores ativos.
 - d. Função para o sorteio: essa função retira uma carta aleatória do baralho e retorna o número de pontos que essa carta vale.
 - e. Função verificação: verifica e indica qual/ quais jogador/jogadores tem o maior número de pontos, que seja menor ou igual a 21.
7. Super desafio! Faça um sistema de cadastro de clientes. Modele cada cliente como uma lista de três elementos: nome, CPF e e-mail.
- a. Faça uma função que peça o nome, o CPF e o e-mail da pessoa e retorne uma lista contendo esses elementos nessa ordem.
 - b. Os clientes cadastrados devem ser armazenados em uma lista (uma lista de listas, já que cada cliente será uma lista tal como produzido no item a).
 - c. Faça uma função que recebe a lista do item b) e um CPF e, se esse cliente estiver na lista de cadastro, sua função deve devolver a lista de dados desse cliente; caso contrário, sua função deve imprimir “não encontrado”.

- d. Enquanto não for digitado 0, o seu programa deve continuar rodando. Se digitado 0, seu programa deve cadastrar um novo cliente; se digitado 1, seu programa deve pedir um CPF e procurá-lo na lista de clientes (item c); se digitado 3, seu programa deve imprimir todos os clientes cadastrados.