Lista de Exercícios – Criação de Funções em Python (Com Explicações)

1. Crie uma função chamada saudacao() que exiba 'Bem-vindo(a)!'.
Explicação: Use a palavra-chave def para definir a função e dentro dela use print para exibir a mensagem.

2. Crie uma função chamada ola(nome) que receba um nome e exiba 'Olá, nome!'.
Explicação: A função precisa de um parâmetro 'nome' e deve usar f-string para formatar a mensagem.

3. Crie uma função chamada dobro(n) que receba um número e retorne o dobro dele.
Explicação: A função deve receber um valor, multiplicar por 2 e retornar o resultado com return.

4. Crie uma função soma(a, b) que some dois números e retorne o resultado.
Explicação: Defina dois parâmetros e use return para devolver a soma dos dois valores.

5. Crie uma função eh_par(n) que verifique se o número é par.
Explicação: Use o operador % para verificar se n % 2 == 0 e retorne True ou False.

6. Crie uma função que retorne o comprimento de uma string.
Explicação: Receba a string como parâmetro e use a função len para retornar seu tamanho.

7. Crie uma função maior_valor(lista) que retorne o maior número de uma lista.
Explicação: Use a função max() sobre a lista recebida como parâmetro.

8. Crie uma função media(lista) que retorne a média dos elementos.
Explicação: Some os elementos com sum() e divida por len(lista).

9. Crie uma função contar_letra(texto, letra) que conte quantas vezes uma letra aparece em um texto.
Explicação: Use o método count() da string.

10. Crie uma função inverte(texto) que retorne o texto ao contrário.
Explicação: Use fatiamento com texto[::-1] para inverter.

11. Crie uma função multiplicar_lista(lista, n) que multiplica todos os elementos por n.
Explicação: Use um laço for para multiplicar cada item da lista pelo valor n e retorne uma nova lista.

12. Crie uma função tabuada(n) que imprima a tabuada do número de 1 a 10.
Explicação: Use um laço for para multiplicar o número de 1 a 10 e exibir os resultados formatados.

13. Crie uma função soma_pares(lista) que retorne a soma dos números pares.
Explicação: Use um laço para verificar quais números são pares e acumular a soma em uma variável.

14. Crie uma função fatorial(n) que retorne o fatorial de n.
Explicação: Inicialize uma variável como 1 e multiplique ela por cada número de 1 até n usando um laço.

15. Crie uma função eh_primo(n) que verifica se n é primo.
Explicação: Use um laço de 2 até n-1 para verificar se n é divisível por algum número além de 1 e ele mesmo.

16. Crie uma função chamada saudacao_personalizada(nome, mensagem='Olá') que exiba a saudação com nome.
Explicação: Use parâmetro com valor padrão para mensagem e concatene com o nome usando f-string.

17. Crie uma função positivos(lista) que retorne somente os valores positivos da lista.
Explicação: Use um laço for e condicional para verificar se os valores são maiores que 0.

18. Crie uma função quadrados(lista) que retorne uma lista com os quadrados dos elementos.
Explicação: Use list comprehension ou um laço tradicional com item**2.

19. Crie uma função para converter uma string em maiúscula.
Explicação: Use o método upper() sobre a string recebida como parâmetro.

20. Crie uma função calcular_area_retangulo(base, altura).
Explicação: Retorne base * altura diretamente na função.

21. Crie uma função que receba uma lista de nomes e retorne o nome com mais letras.
Explicação: Use a função max() com key=len para obter o maior nome.

22. Crie uma função que receba uma frase e conte quantas palavras ela possui.
Explicação: Use split() para separar as palavras e len() para contar.

23. Crie uma função que receba um número e retorne se é positivo, negativo ou zero.
Explicação: Use if, elif e else para retornar a string correspondente.

24. Crie uma função que receba uma lista de números e retorne a quantidade de números ímpares.
Explicação: Use um contador e verifique com % 2 != 0.

25. Crie uma função que verifique se um número é múltiplo de outro.
Explicação: Receba dois números e use if n1 % n2 == 0 para retornar True.

26. Crie uma função que receba dois nomes e retorne o que tiver mais letras.
Explicação: Compare com len() e retorne o nome mais longo.

27. Crie uma função que receba um texto e retorne o número de vogais.
Explicação: Use um laço para contar as vogais usando in 'aeiouAEIOU'.

28. Crie uma função que receba uma lista de números e retorne uma nova lista com os números ao quadrado.
Explicação: Retorne uma lista onde cada elemento é elevado ao quadrado com list comprehension.

29. Crie uma função que receba uma lista de palavras e retorne uma nova lista com as palavras em maiúsculo.
Explicação: Use um laço ou list comprehension com upper().

30. Crie uma função que calcule o perímetro de um retângulo dado a base e altura.
Explicação: Use a fórmula 2 * (base + altura) e retorne o valor.