



Modularização

EXERCÍCIOS: Modularização Python (comando DEF)

(OBRIGATÓRIO: NÃO USAR: PRINT ou INPUT DENTRO DAS FUNÇÕES, OU SEJA, EXIBIR E LER OS DADOS PARA O USUÁRIO SEMPRE NA CÉLULA DO PROGRAMA PRINCIPAL)

- 1) Escrever uma **função em Python** para cada equação de $f(n) \in R$. Em seguida, crie um programa que **exiba os termos**, a **soma dos termos** e a **média dos termos** das equações de $f(n) \in R$.

Testar os valores inválidos da variável n lida do usuário..

k e p são constantes a sua escolha, por exemplo: $k = \text{math.pi}$ e $p = \text{math.e}$.

a) $f(n) = \frac{k}{n!+k}$, para os n primeiros termos.

b) $f(n) = \frac{k^n - k}{p * n}$ para os n primeiros termos.

- 2) Criar uma função e protótipo que receba como argumento um *Número* inteiro (*Número* > 0) e retorne a multiplicação dos Pares ou dos Ímpares entre $[1, \text{Número}]$ de acordo com a Opção do usuário. Na *célula do programa principal*, use a função 100 vezes para valores de *Número* e Opção escolhidos pelo usuário.
- 3) Criar duas funções e seus protótipos que calculam a **Combinação** e o **Arranjo** de n elementos combinados p a p . Após isso, na *célula do programa principal*, use as funções criadas várias vezes e exiba o resultado da Combinação e do Arranjo enquanto os valores lidos do usuário: n e p **estiverem corretos**. Sabe-se que:

$$\text{Arranjo}(n, p) = \frac{n!}{(n-p)!} \quad \text{Combinação}(n, p) = \frac{\text{Arranjo}(n, p)}{p!} \quad ; \text{ sendo que: } n \text{ e } p \geq 0 \text{ e } n \geq p$$

- 4) Criar uma função e seu protótipo que receba o valor de um ano e retorne da função: Zero (0): Ano NÃO é Bissexto ou Um (1): Ano é Bissexto. Regras do ano Bissexto (**Ver exercício 11**):

Na célula do programa principal, use a função 100 vezes para valores de anos lidos do usuário (um de cada vez).

- 5) Criar uma função que receba um número inteiro qualquer positivo e retorne: *True* , se N for PRIMO ou retorne: *False* , se N for NÃO for PRIMO. Na célula do programa principal, use a função 500 vezes para valores de N lidos do usuário (um de cada vez).

Regras do Número Primo, considere sendo:

→ 1, 2 e 3 Números Primos e

→ Os números que somente são múltiplos de 1 e dele mesmo.



- 6) Crie uma função que calcule o IMC (Índice de Massa Corpórea) de uma pessoa com base na sua idade, massa e sexo conforme tabela. Na *célula do programa principal*, use a função e exiba o IMC de 100 entrevistados (um por um).

Sexo	Fórmula
1 : Feminino	$IMC = \left(\frac{0.95 * Massa}{Altura^2} \right)$
2: Masculino	$IMC = \left(\frac{1.05 * Massa}{Altura^2} \right)$

- 7) Crie uma função para uma "mini" calculadora (somente de inteiros), ou seja, passe como argumento:

- ☐ Dois (2) números inteiros: **Número1** e **Número2** e
- ☐ Um (1) **Operador**: 1: Soma (+) ou 2: Subtração (-) ou 3: Multiplicação (*) ou 4: Divisão (/) ou 5: MOD (%) ou 6: DIV (//)

Retorne desta função o resultado da operação matemática solicitada pelo usuário. Na *célula do programa principal*, use a função em 100 operações matemáticas com valores de **Número1**, **Número2** e **Operador** lidos do usuário.

- 8) Faça um programa Python modularizado que calcule o somatório dos n primeiros termos da série de Fibonacci (0, 1, 1, 2, 3, 5, 8, ...). A função deve receber o valor de n e retornar o valor do somatório para ser impresso na célula do programa principal.
- 9) Crie uma função e seu protótipo que receba como argumento o preço de uma mercadoria e seu reajuste em porcentagem, retorne o preço reajustado e o reajuste calculado em reais. Após isso, use a função no programa principal e exiba o preço reajustado e o reajuste de 100 mercadorias informadas pelo usuário.
- 10) Crie somente uma (1) função e seu protótipo, conforme para a descrição dos exercícios: 3 acima.
- 11) Crie uma função e seu protótipo que receba como argumento um **ano qualquer** e um dos três números citados: 4 ou 100 ou 400 e retorne **zero ou um** (se o ano for ou não bissexto), **quociente** e **resto da divisão do ano pelo número escolhido**, conforme as regras de teste do ano bissexto. Após isso, use a função no programa principal e exiba as informações sobre o ano conforme os dados ditos pelo usuário.

Regras de teste do ano bissexto:

- ✓ São bissextos todos os anos múltiplos de **400**. Exemplo: 1600, 2000, 2400, 2800...
- ✓ São bissextos todos os múltiplos de **4** e não múltiplos de **100**. Exemplo: 1996, 2004, 2008, ...
- ✓ Não são bissextos todos os demais anos.

- 12) Crie uma função e seu protótipo que receba como argumento um número inteiro qualquer representando os segundos de realização de um teste de laboratório e retorne as horas, minutos e segundos convertidos deste número. Use a função (100 vezes) no programa principal e exiba a informação, conforme o exemplo a seguir:

Exemplo: Tempo: **10000** Segundos = **2** Horas(s) + **46** Minuto(s) + **40** Segundos(s).



- 13) Crie uma função e protótipo que receba como argumento um valor real do **Tempo T** (Horas) e converta em: **Horas + Minutos + Segundos**. Na *célula do programa principal*, use a função 50 vezes para valores de **T** lidos do usuário.

Exemplo: T (12.47 Horas) = 12 Horas(s) + 28 Minuto(s) + 12 Segundo(s).

- 14) Crie uma função e seu protótipo que receba a hora de entrada e saída de um estacionamento e retorne o **tempo de permanência e total a pagar**. Na célula do programa principal o usuário informa:

Hora de entrada (HH:MM): Exemplo: 13:37

Hora de saída (HH:MM): Exemplo: 18:41 (Menus)

Diferença: (HH:MM): Exemplo: 04:64 → 5 Horas de **permanência** no estacionamento.

Preço: R\$ 7.00 por hora.

- 15) Crie uma função e seu protótipo que determine An e Sn de uma Progressão Geométrica (P.G). Após isso, use a função 50 vezes para calcular os dados de 50 P.G's com valores de a_1 , q e n lidos do usuário.

$$\text{Enésimo termo : } An = a_1 * q^{(n-1)}$$

$$\text{Soma dos termos: } Sn = \frac{a_1 * (q^n - 1)}{q - 1}$$

Sendo que:

$$a_1 = \text{primeiro termo da P.G}$$

$$q = \text{Razão da P.G}$$

$$n = \text{Quantidade de termos da P.G}$$

- 16) Faça um programa Python modularizado que:

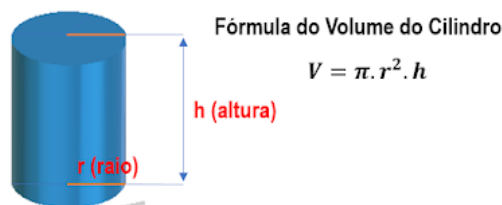
- (i) Tenha uma função que calcule e retorne a área de um círculo de raio R;
- (ii) Tenha outra função que calcule e retorne o volume do cilindro reto.

A função de (ii) deve usar a função de (i).

O raio e a altura de um cilindro devem ser lidos do teclado na célula do programa principal.

Obs.: Área do círculo: $A = \pi \cdot r^2$

Volume do Cilindro: *Área do Círculo da Base vezes a Altura.*





17) Faça um programa Python modularizado que exiba o público total de um jogo de futebol e forneça a arrecadação do jogo, sabendo que:

- ✓ Crianças abaixo de 10 anos não pagam;
- ✓ Jovens de 11 a 17 pagam $\frac{1}{2}$ entrada;
- ✓ Acima dos 18 anos paga $\frac{1}{2}$ entrada se doarem um quilo de alimento não perecível.
- ✓ O valor inteiro do ingresso é lido do usuário em reais (R\$).

Na *célula do programa principal*, use a função várias vezes criando um Menu para o usuário.

18) Escrever um algoritmo modularizado em Python que leia o preço de uma mercadoria e exiba o preço na tela o preço reajustado de X%. O usuário escolherá para reajuste, (1): Acréscimo ou (2): Desconto para o reajuste de X %.

Na *célula do programa principal*, use a função várias vezes, criando um Menu para o usuário.

19) Faça uma função em Python que receba três valores A, B, C e verifique se os mesmos podem formar um triângulo ou não. Caso possam, dizer que tipo: Escaleno, Isóscele ou Equilátero (0, 2 ou 3 lados iguais, respectivamente).

Condição obrigatória: É triângulo se, somente se $(A < B + C)$ e $(B < A + C)$ e $(C < A + B)$.

Na *célula do programa principal*, use a função várias vezes, criando um Menu para o usuário.

20) Faça uma função em Python que calcule o contracheque de um funcionário. Essa função receberá como parâmetros:

- Salário Bruto (R\$)
- Número de Dependentes: Esposa + Filhos;

E, retornará da função:

- Salário Líquido (R\$)
- Total de Descontos (R\$)

Conforme as regras a seguir:

DESCONTOS:

- INSS: 11% do Salário Bruto;
- Plano de Saúde: 2% do Salário Bruto;

BENEFÍCIOS:

- Vale Alimentação: R: Salário Mínimo + 5% do Salário Bruto por dependente.

Na *célula do programa principal*, use a função várias vezes, criando um Menu para o usuário.