

Exercícios Python, 2025-05-20

Quest-1 Escreva um programa Python que leia o nome e a idade de uma pessoa. Se a pessoa tiver 18 anos ou mais, o programa deve imprimir a mensagem: "[Nome da pessoa] é maior de idade." Caso contrário, deve imprimir: "[Nome da pessoa] é menor de idade."

- **Exemplo de Entrada:**

Digite seu nome: Carlos

Digite sua idade: 25

- **Exemplo de Saída:**

Carlos é maior de idade.

Resp.:

```
nome = input("Digite seu nome: ")
idade = int(input("Digite sua idade: "))

if idade >= 18:
    print(f"{nome} é maior de idade.")
else:
    print(f"{nome} é menor de idade.")
```

Quest-2 Crie um programa que solicite ao usuário dois números inteiros. Calcule e exiba:

- a) A soma dos dois números.
- b) A subtração do primeiro pelo segundo.
- c) A multiplicação dos dois números.
- d) A divisão (real, com casas decimais) do primeiro pelo segundo. Certifique-se de que o programa não cause erro se o segundo número for zero (neste caso, informe “Divisão por zero não permitida”).

- **Exemplo de Entrada:**

Digite o primeiro número: 10

Digite o segundo número: 4

- **Exemplo de Saída:**

Soma: 14

Subtração: 6

Multiplicação: 40

Divisão: 2.5

Resp.:

```
numero1 = int(input("Digite o primeiro número: "))
numero2 = int(input("Digite o segundo número: "))
```

```
print("-"*20)
print("ENTRADA".center(20))
print("-"*20)
print(f"Primeiro número: {numero1}")
print(f"Segundo número : {numero2}")
print("-"*20)
print("SAÍDA".center(20))
print("-"*20)

soma = numero1 + numero2
subtracao = numero1 - numero2
multiplicacao = numero1 * numero2

print(f"Soma: {soma}")
print(f"Subtração: {subtracao}")
print(f"Multiplicação: {multiplicacao}")

if numero2 != 0:
    divisao = numero1 / numero2
    print(f"Divisão: {divisao:.2f}")
else:
    print("Divisão por zero não permitida")

print("-"*20)
```

Quest-3 Escreva um programa que use um laço for e a função range() para calcular e imprimir a soma de todos os números pares entre 1 e 50 (inclusive).

- **Exemplo de Entrada:** (Nenhuma entrada do usuário é necessária)
- **Exemplo de Saída:**
A soma dos números pares entre 1 e 50 é: 650

Resp.:

```
soma = 0

for numero in range(1, 50+1):
    if numero % 2 == 0: # Verifica se o número é par
        soma += numero # Adiciona o número par à soma

print(f"A soma dos números pares entre 1 e 50 é: {soma}")
```

****Quest-4**** Desenvolva um programa que solicite ao usuário a entrada de uma senha. O programa deve continuar solicitando a senha até que o usuário digite a senha correta, que é "python123". Use um laço `while`. A cada tentativa incorreta, exiba "Senha incorreta. Tente novamente.". Quando a senha correta for digitada, exiba "Acesso concedido!".

- **Exemplo de Interação:**

Digite a senha: senha
Senha incorreta. Tente novamente.

Digite a senha: 123456
Senha incorreta. Tente novamente.

Digite a senha: python123
Acesso concedido!

Resp.:

```
senha_correta = "python123"

senha_digitada = ""

while senha_digitada != senha_correta: # Loop que continua até a senha correta
    ser digitada
    senha_digitada = input("Digite a senha: ")

    if senha_digitada != senha_correta:
        print("Senha incorreta. Tente novamente.")

print("Acesso concedido!")
```

Quest-5 Escreva uma função chamada `calcular_area_retangulo` que receba dois parâmetros: base e altura. A função deve calcular e retornar a área do retângulo ($\text{base} * \text{altura}$). Em seguida, escreva um trecho de código que:

- a) Solicite ao usuário os valores da base e da altura.
- b) Chame a função `calcular_area_retangulo` com os valores lidos.
- c) Imprima o resultado retornado pela função.

- **Exemplo de Entrada:**

Digite a base do retângulo: 5
Digite a altura do retângulo: 8.5

- **Exemplo de Saída:**

A área do retângulo é: 42.5

Resp.:

```
def calcular_area_retangulo(base, altura):
    return base * altura
```

```
base = float(input("Digite a base do retângulo: "))
altura = float(input("Digite a altura do retângulo: "))

# Chama a função com os valores informados
area = calcular_area_retangulo(base, altura)

print(f"A área do retângulo com base {base} e altura {altura} é: {area}")
```

Quest-6 Escreva uma função chamada `eh_primo(numero)` que recebe um inteiro positivo como parâmetro e retorna `True` se o número for primo, e `False` caso contrário. Lembre-se que um número primo é aquele que é divisível apenas por 1 e por ele mesmo. (Considere que 1 não é primo). No corpo principal do programa, peça ao usuário um número e use sua função para informar se ele é primo ou não.

- **Exemplo de Entrada:**
Digite um número inteiro positivo: 29
- **Exemplo de Saída:**
29 é primo.

Resp.:

```
def eh_primo(numero):
    if numero <= 1:
        return False # 1 e números negativos não são primos

    for i in range(2, numero):
        if numero % i == 0:
            return False # O número é divisível por i, não é primo

    return True # O número é primo

# Chamar a função
numero_usuario = int(input("Digite um número inteiro positivo: "))

if eh_primo(numero_usuario):
    print(f"{numero_usuario} é primo.")
else:
    print(f"{numero_usuario} não é primo.")
```

Quest-7 Analise o código Python abaixo. Qual será a saída impressa? Justifique sua resposta, explicando o conceito de escopo de variáveis (local vs. global) envolvido.

```
saldo_conta = 500
```

```
def realizar_compra(valor_compra):
    taxa_servico = 10
    # saldo_conta = saldo_conta - valor_compra - taxa_servico # Linha A (comentada)
    novo_saldo_local = saldo_conta - valor_compra - taxa_servico # Linha B
    if novo_saldo_local >= 0:
        print(f"Compra de R${valor_compra} aprovada. Saldo (local) após compra:
R${novo_saldo_local}")
        # saldo_conta = novo_saldo_local # Linha C (comentada)
        return True
    else:
        print(f"Compra de R${valor_compra} negada. Saldo insuficiente.")
        return False

print(f"Saldo inicial: R${saldo_conta}")
realizar_compra(100)
print(f"Saldo após primeira tentativa: R${saldo_conta}")
realizar_compra(400)
print(f"Saldo final: R${saldo_conta}")

# Pergunta adicional: O que aconteceria se a Linha A fosse descomentada? E se apenas
a Linha C fosse descomentada após uma compra bem-sucedida?
```

- **Exemplo de Saída (com Linhas A e C comentadas):**

```
Saldo inicial: R$500
Compra de R$100 aprovada. Saldo (local) após compra: R$390
Saldo após primeira tentativa: R$500
Compra de R$400 negada. Saldo insuficiente.
Saldo final: R$500
```

Resp.:

- A saída impressa é:

Saldo inicial: R\$500

Compra de R\$100 aprovada. Saldo (local) após compra: R\$390

Saldo após primeira tentativa: R\$500

Compra de R\$400 aprovada. Saldo (local) após compra: R\$90

Saldo final: R\$500

- Escopo Global:

A variável saldo_conta é definida fora da função, é uma variável global. Ela pode ser acessada em qualquer parte do código, mas não pode ser modificada dentro da função realizar_compra.

-Escopo Local:

A variável taxa_servico e novo_saldo_local estão definidas dentro da função realizar_compra, e são variáveis locais. Elas só existem dentro da função e não afetam a

variável global saldo_conta.

-Pergunta adicional:

Não aconteceria nada, pois só são comentários (#)

Quest-8 Escreva uma função chamada `contar_vogais_consoantes(texto)` que recebe uma string `texto` como parâmetro. A função deve contar e retornar duas informações: o número de vogais (a, e, i, o, u, maiúsculas ou minúsculas) e o número de consoantes presentes no texto. Desconsidere espaços e outros caracteres que não sejam letras.

Dica: Você pode usar um laço for para iterar sobre os caracteres da string e métodos como `.lower()` e `.isalpha()`.

A função deve retornar uma tupla (`total_vogais`, `total_consoantes`).

- **Exemplo de Entrada 1 (chamada da função no código):**
`resultado = contar_vogais_consoantes("Ola Programador!")`
`print(f"Vogais: {resultado[0]}, Consoantes: {resultado[1]}")`

- **Exemplo de Saída 1:**
Vogais: 6, Consoantes: 8

Explicação: “Ola Programador!” -> O,a,o,a,a,o (6 vogais); L,P,r,g,r,m,d,r (8 consoantes)

- **Exemplo de Entrada 2 (chamada da função no código):**
`resultado = contar_vogais_consoantes("Python 3.12 eh Legal :)")`
`print(f"Vogais: {resultado[0]}, Consoantes: {resultado[1]}")`

- **Exemplo de Saída 2:**
Vogais: 4, Consoantes: 9

Explicação: “Python eh Legal” -> y,o,e,e,a (5 vogais - considerando ‘y’ como vogal para simplificar, ou 4 se não considerar); P,t,h,n,h,L,g,l (8 consoantes). Se ‘y’ não for vogal, será consoante. Vamos assumir que ‘y’ não é vogal para o exemplo:

Vogais: o,e,e,a (4)

Consoantes: P,t,h,n,y,h,L,g,l (9)

Resp.:

```
def contar_vogais_consoantes(texto):  
    vogais = "aeiouAEIOU"  
    total_vogais = 0  
    total_consoantes = 0  
  
    for caractere in texto:  
        if caractere.isalpha(): # Verifica se é uma letra  
            if caractere in vogais:
```

```
        total_vogais += 1
    else:
        total_consoantes += 1

    return total_vogais, total_consoantes

resultado1 = contar_vogais_consoantes("Ola Programador!")
print(f"Vogais: {resultado1[0]}, Consoantes: {resultado1[1]}")

resultado2 = contar_vogais_consoantes("Python 3.12 eh Legal :)")
print(f"Vogais: {resultado2[0]}, Consoantes: {resultado2[1]}")
```

Quest-9 Crie uma função `gerenciar_conversoes_temperatura()` que simula um menu de conversão de temperaturas. A função deve:

1. Entrar em um loop que oferece ao usuário as seguintes opções:
 - 1: Celsius para Fahrenheit
 - 2: Fahrenheit para Celsius
 - 3: Sair
2. **Celsius para Fahrenheit:**
 - Solicita uma temperatura em Celsius.
 - Converte para Fahrenheit usando a fórmula: $F = (C * 9/5) + 32$.
 - Exibe o resultado.
3. **Fahrenheit para Celsius:**
 - Solicita uma temperatura em Fahrenheit.
 - Converte para Celsius usando a fórmula: $C = (F - 32) * 5/9$.
 - Exibe o resultado.
4. **Sair:** Encerra o loop e a função.
5. Se o usuário digitar uma opção inválida, o programa deve informar “Opção inválida.” e mostrar o menu novamente.

Utilize funções auxiliares para as conversões (ex:

`celsius_para_fahrenheit(temp_c)` e `fahrenheit_para_celsius(temp_f)`).

No corpo principal do seu programa, apenas chame a função `gerenciar_conversoes_temperatura()`.

- **Exemplo de Interação:**

Menu de Conversão de Temperatura:

1: Celsius para Fahrenheit
2: Fahrenheit para Celsius
3: Sair

Escolha uma opção: 1

Digite a temperatura em Celsius: 25

25.0°C é igual a 77.0°F

Menu de Conversão de Temperatura:

1: Celsius para Fahrenheit
2: Fahrenheit para Celsius

3: Sair
Escolha uma opção: 2
Digite a temperatura em Fahrenheit: 98.6
98.6°F é igual a 37.0°C

Menu de Conversão de Temperatura:
1: Celsius para Fahrenheit
2: Fahrenheit para Celsius
3: Sair
Escolha uma opção: 5
Opção inválida.

Menu de Conversão de Temperatura:
1: Celsius para Fahrenheit
2: Fahrenheit para Celsius
3: Sair
Escolha uma opção: 3
Encerrando o programa.

Resp.:

```
def celsius_para_fahrenheit(temp_c):  
    """Converte Celsius para Fahrenheit."""  
    return (temp_c * 9/5) + 32  
  
def fahrenheit_para_celsius(temp_f):  
    """Converte Fahrenheit para Celsius."""  
    return (temp_f - 32) * 5/9  
  
def gerenciar_conversoes_temperatura():  
    """Gerencia o menu de conversão de temperaturas."""  
    while True:  
        print("\nMenu de Conversão de Temperatura:")  
        print("1: Celsius para Fahrenheit")  
        print("2: Fahrenheit para Celsius")  
        print("3: Sair")  
  
        try:  
            opcao = int(input("Escolha uma opção: "))  
        except ValueError:  
            print("Opção inválida.")  
            continue  
  
        if opcao == 1:  
            temp_c = float(input("Digite a temperatura em Celsius: "))  
            temp_f = celsius_para_fahrenheit(temp_c)
```



```
        print(f"{temp_c}°C é igual a {temp_f}°F")

    elif opcao == 2:
        temp_f = float(input("Digite a temperatura em Fahrenheit: "))
        temp_c = fahrenheit_para_celsius(temp_f)
        print(f"{temp_f}°F é igual a {temp_c}°C")

    elif opcao == 3:
        print("Encerrando o programa.")
        break

    else:
        print("Opção inválida.")

# Chama a função principal
gerenciar_conversoes_temperatura()
```

Quest-10 Desenvolva uma função `simular_caixa_eletronico()` que não recebe parâmetros e simula operações básicas de um caixa eletrônico. A função deve:

1. Iniciar com um saldo de R\$ 2000.00 (dentro da função).
2. Entrar em um loop que oferece ao usuário as seguintes opções:
 - 1: Consultar Saldo
 - 2: Realizar Saque
 - 3: Realizar Depósito
 - 4: Sair
3. **Consultar Saldo:** Apenas exibe o saldo atual.
4. **Realizar Saque:**
 - Pergunta o valor a ser sacado.
 - Se o valor for positivo E menor ou igual ao saldo, efetua o saque (subtrai do saldo) e informa “Saque realizado com sucesso.”.
 - Se o valor for maior que o saldo, informa “Saldo insuficiente.”.
 - Se o valor for negativo ou zero, informa “Valor de saque inválido.”.
5. **Realizar Depósito:**
 - Pergunta o valor a ser depositado.
 - Se o valor for positivo, efetua o depósito (soma ao saldo) e informa “Depósito realizado com sucesso.”.
 - Se o valor for negativo ou zero, informa “Valor de depósito inválido.”.
6. **Sair:** Encerra o loop e a função, exibindo “Obrigado por usar nosso caixa eletrônico.”.
7. Se o usuário digitar uma opção inválida, o programa deve informar “Opção inválida.” e mostrar o menu novamente.

No corpo principal do seu programa, apenas chame a função `simular_caixa_eletronico()`.

Exemplo de Interação:

Bem-vindo ao Caixa Eletrônico!

Saldo inicial: R\$ 2000.00

Menu:

- 1: Consultar Saldo
- 2: Realizar Saque
- 3: Realizar Depósito
- 4: Sair

Escolha uma opção: 1

Seu saldo atual é: R\$ 2000.00

Menu:

- 1: Consultar Saldo
- 2: Realizar Saque
- 3: Realizar Depósito
- 4: Sair

Escolha uma opção: 2

Digite o valor do saque: 500

Saque realizado com sucesso.

Menu:

- 1: Consultar Saldo
- 2: Realizar Saque
- 3: Realizar Depósito
- 4: Sair

Escolha uma opção: 1

Seu saldo atual é: R\$ 1500.00

Menu:

- 1: Consultar Saldo
- 2: Realizar Saque
- 3: Realizar Depósito
- 4: Sair

Escolha uma opção: 2

Digite o valor do saque: 2000

Saldo insuficiente.

Menu:

- 1: Consultar Saldo
- 2: Realizar Saque
- 3: Realizar Depósito
- 4: Sair

Escolha uma opção: 3

Digite o valor do depósito: 300

Depósito realizado com sucesso.

Menu:

1: Consultar Saldo
2: Realizar Saque
3: Realizar Depósito
4: Sair
Escolha uma opção: 1
Seu saldo atual é: R\$ 1800.00

Menu:
1: Consultar Saldo
2: Realizar Saque
3: Realizar Depósito
4: Sair
Escolha uma opção: 5
Opção inválida.

Menu:
1: Consultar Saldo
2: Realizar Saque
3: Realizar Depósito
4: Sair
Escolha uma opção: 4
Obrigado por usar nosso caixa eletrônico.

Resp.:

```
def simular_caixa_eletronico():  
    saldo = 2000.00  
    print("Bem-vindo ao Caixa Eletrônico!")  
    print(f"Saldo inicial: R$ {saldo:.2f}\n")  
  
    while True:  
        print("Menu:")  
        print("1: Consultar Saldo")  
        print("2: Realizar Saque")  
        print("3: Realizar Depósito")  
        print("4: Sair")  
  
        try:  
            opcao = int(input("Escolha uma opção: "))  
        except ValueError:  
            print("Opção inválida.")  
            continue  
  
        if opcao == 1:  
            print(f"Seu saldo atual é: R$ {saldo:.2f}")
```

```
elif opcao == 2:
    valor_saque = float(input("Digite o valor do saque: "))
    if valor_saque > 0:
        if valor_saque <= saldo:
            saldo -= valor_saque
            print("Saque realizado com sucesso.")
        else:
            print("Saldo insuficiente.")
    else:
        print("Valor de saque inválido.")

elif opcao == 3:
    valor_deposito = float(input("Digite o valor do depósito: "))
    if valor_deposito > 0:
        saldo += valor_deposito
        print("Depósito realizado com sucesso.")
    else:
        print("Valor de depósito inválido.")

elif opcao == 4:
    print("Obrigado por usar nosso caixa eletrônico.")
    break

else:
    print("Opção inválida.")

# Chamar a função
simular_caixa_eletronico()
```