

Estruturas de Controle - Exercícios Práticos

Exercício 01: uso de `if`

Você está ajudando a gerenciar a entrada de pessoas em um bar noturno. O programa deve perguntar ao usuário sua idade. Se a idade for igual ou maior que 18 anos, o programa deve imprimir a mensagem **"Entrada liberada"**. Caso contrário, deve imprimir **"Entrada negada"**.

Exemplo de saída:

```
Digite sua idade: 18
```

```
Entrada liberada
```

Exercício 02: uso do `if` aninhado

Você trabalha em um site de vendas e precisa calcular o desconto de um produto com base no valor da compra. O programa deve funcionar assim:

- Se o valor da compra for **maior que R\$500**, o desconto será de **10%**.
- Se o valor for **menor ou igual a R\$500**, mas **maior que R\$100**, o desconto será de **5%**.
- Se o valor for **menor ou igual a R\$100**, não haverá desconto.

Implemente um código que faça esse cálculo e exiba o valor do desconto.

Exemplo de saída 01:

```
Digite o valor da Compra (R$): 80
```

```
O valor do desconto é R$ 0,00
```

Exemplo de saída 02:

```
Digite o valor da Compra (R$): 500
```

```
O valor do desconto é R$ 25,00
```

Exemplo de saída 03:

```
Digite o valor da Compra (R$): 600
```

```
O valor do desconto é R$ 60,00
```

Exercício 03: Diferença entre `if` Conjuntos e `elif`

Você está criando um sistema de notas para uma escola. O programa deve avaliar a nota de um aluno e imprimir a seguinte mensagem:

1. Se a nota for **acima de 9**, o aluno receberá **"Aprovado com Mérito"**.
2. Se a nota for **maior ou igual a 7**, o aluno será **"Aprovado"**.
3. Se a nota for **menor que 7**, o aluno será **"Reprovado"**.

Sua tarefa é implementar dois programas para realizar essa avaliação de formas diferentes:

1. No arquivo **q3a.py**, crie o programa utilizando **apenas `if s conjuntos`** (sem usar `elif`). Ou seja, use várias verificações `if` para cada condição, sem interromper as verificações com `elif`.
2. No arquivo **q3b.py**, implemente o programa utilizando a estrutura **`if e elif`**. Isso significa que você deve usar `elif` para encadear as verificações de forma que, uma vez que uma condição seja satisfeita, as demais não sejam avaliadas.

Objetivo: Compare os dois códigos e observe se há diferenças no comportamento. Caso haja, descreva qual é a diferença e explique o motivo.

Dica: Lembre-se de que, no código com `if` conjuntos, todas as condições serão verificadas, enquanto no código com `elif`, a verificação de condições para após a primeira condição verdadeira ser encontrada.

Exemplo de saída **q3a.py**:

```
Digite a nota do aluno: 10
```

```
Aprovado com Mérito
```

```
Aprovado
```

Exemplo de saída **q3b.py**:

```
Digite a nota do aluno: 10
```

```
Aprovado com Mérito
```

Exercício 04: Vários `elif`

Você está desenvolvendo uma aplicação que informa o estado do tempo com base na temperatura em graus Celsius. O programa deve mostrar uma mensagem dependendo da temperatura:

1. Se a temperatura for **abaixo de 0°C**, a mensagem deve ser **"Frio extremo"**.
2. Se a temperatura estiver **entre 0°C e 10°C**, a mensagem será **"Frio"**.
3. Se estiver **entre 10°C e 20°C**, a mensagem será **"Ameno"**.
4. Se a temperatura estiver **entre 20°C e 30°C**, a mensagem será **"Quente"**.
5. Se a temperatura for **acima de 30°C**, a mensagem será **"Muito quente"**.

Implemente um código que verifique as condições e imprima a mensagem correspondente

Exemplo de saída:

```
Digite a temperatura do dia (em °C): 30
```

```
Muito quente
```

Exercício 05: condições com operadores

Você está criando um sistema de descontos para clientes. O programa deve calcular o desconto baseado no valor da compra e no status do cliente:

1. Se o cliente for **VIP** e a compra for **igual ou acima de R\$1000**, ele receberá **15% de desconto**.
2. Se o cliente não for VIP, mas a compra for **igual ou acima de R\$1000**, o desconto será **10%**.
3. Se a compra for **abaixo de R\$1000**, o desconto será **5%** para todos os clientes, independentemente de serem VIP ou não.

Implemente esse código que calcule o valor do desconto.

Exemplo de saída 01:

```
Informe o valor da compra (em R$): 1000.00
```

```
Informe se o cliente é VIP: não
```

```
O valor do desconto é R$ 100.00
```

Exemplo de saída 02:

```
Informe o valor da compra (em R$): 1000.00
```

```
Informe se o cliente é VIP: sim
```

```
O valor do desconto é R$ 150.00
```

Exemplo de saída 03:

```
Informe o valor da compra (em R$): 900.00
```

```
Informe se o cliente é VIP: não
```

```
O valor do desconto é R$ 50.00
```

Exercício 06: Verificando Múltiplas Condições (Uso de `and` / `or`)

Você está criando um programa que verifica se uma pessoa pode ser aprovada para um empréstimo bancário. Para que o empréstimo seja aprovado, a pessoa deve atender a uma das seguintes condições:

1. Ter uma **renda mínima de R\$3000** e um **histórico de crédito "bom"**.
2. Caso tenha uma **renda inferior a R\$3000**, ela deve ter um **histórico de crédito "excelente"**.

Implemente um programa que verifique essas condições e imprima **"Empréstimo aprovado"** ou **"Empréstimo negado"**.

Exemplo de saída 01:

```
Informe sua renda mínima: 1000.00
Informe seu histórico de crédito:
1 - Excelente
2 - Bom
3 - Médio
4 - Ruim
1

Empréstimo aprovado
```

Exemplo de saída 02:

```
Informe sua renda mínima: 1000.00
Informe seu histórico de crédito:
1 - Excelente
2 - Bom
3 - Médio
4 - Ruim
2

Empréstimo negado
```

Exercício 07: Sequência de Gastos Diários

Você está criando um programa para ajudar uma pessoa a gerenciar seus gastos diários. O programa deve funcionar da seguinte maneira:

1. O programa deve pedir ao usuário para informar a quantidade total de gastos que ele teve no dia.
2. Em seguida, o programa deve solicitar o valor de cada um desses gastos, um por vez.
3. Ao final, o programa deve calcular e exibir o **valor total de todos os gastos** daquele dia.

O programa deve ser capaz de somar todos os valores inseridos e mostrar o total ao usuário.

Dica: Para pedir ao usuário o valor de cada gasto repetidamente, utilize um laço `for` para percorrer a quantidade de gastos informada pelo usuário.

Exemplo de saída:

```
Quantos gastos você teve hoje? 3
Informe o valor do gasto 1: 15.50
Informe o valor do gasto 2: 30.00
Informe o valor do gasto 3: 45.75

O valor total dos gastos de hoje é: R$ 91.25
```

Exercício 08: Caixa de supermercado

Você está criando um sistema para o caixa de um supermercado. O programa deve simular o processo de escaneamento dos itens de um cliente. O funcionamento deve ser o seguinte:

1. O programa deve pedir ao usuário que digite o **número de um item** sempre que ele escanear um produto. Esse número pode ser qualquer valor, representando o código do produto.
2. O programa continuará pedindo números até que o usuário digite a palavra **"fim"**. Quando o usuário digitar "fim", o programa deve parar de pedir mais números.
3. Quando o cliente terminar de registrar todos os itens e digitar "fim", o programa deve **contar quantos itens foram escaneados** e exibir essa quantidade.

O programa deve contar o número de itens escaneados e mostrar o total de produtos registrados.

Dica: Use um laço `while` para continuar pedindo números até que o usuário digite "fim".

Exemplo de saída:

```
Digite o número do item: 12345
Digite o número do item: 67890
Digite o número do item: 11223
Digite o número do item: fim

Total de itens escaneados: 3
```

Exercício 09: Contagem de Vogais em uma Frase

Crie um programa que leia uma frase e conte quantas vogais (a, e, i, o, u) existem nela. O programa deve percorrer cada caractere da frase e, se o caractere for uma

vogal, ele deve incrementar um contador.

Dica: Use um laço `for` para percorrer os caracteres da frase e contar as vogais.

Exemplo de saída:

```
Digite uma frase: Olá, como vai você?  
Número de vogais: 8
```

Exercício 10: Tabuada

Crie um programa que mostre a tabuada de um número fornecido pelo usuário. O programa deve exibir o número multiplicado de 1 a 10.

Dica: Use um laço `for` para gerar a tabuada de um número.

Exemplo de saída:

```
Digite um número para ver a tabuada: 3  
3 x 1 = 3  
3 x 2 = 6  
3 x 3 = 9  
3 x 4 = 12  
3 x 5 = 15  
3 x 6 = 18  
3 x 7 = 21  
3 x 8 = 24  
3 x 9 = 27  
3 x 10 = 30
```