



Fundamentos de Programação

Estruturas CONDICIONAIS

Variações do *If*

Aula 10

Post It da Aula Passada

Tomada de Decisão

- Processo cotidiano
- Decisões simples e complexas
- Cadeias de decisão

O Comando If

- Ponto de decisão em um programa
- Fluxo de execução
- Teste condicional
 - Expressões
- Hierarquia de código
 - Indentação

Até Agora...

Até agora, vimos que o comando *if* executa um conjunto de instruções **caso o seu teste condicional seja verdadeiro**

Assim, nesse momento, **caso o teste condicional seja falso**, o bloco de instruções relacionadas ao *if* é simplesmente ignorado

Porém, e se eu quiser prever um conjunto de instruções para serem executadas apenas quando o teste condicional não é satisfeito?

O Comando *Else*

Para fazer isso, podemos utilizar o comando ***else***

if teste:

Instruções executadas caso o teste seja verdadeiro (***True***)

else:

Instruções executadas caso o teste seja falso (***False***)

Instruções executadas independentemente do condicional

Todo o *else* é vinculado a um *if* definido imediatamente antes. A presença de um *else* indica que sempre algum dos blocos de instruções será executado (ou do *if*, ou o do *else*)

O Comando *Else*

Resumindo, blocos condicionais do tipo *if-else*:

- Permitem a criação de blocos de código distintos para quando os testes condicionais resultam em valores de verdadeiro (*True*) ou falso (*False*)
- Sempre desviam o fluxo de execução para um, e apenas um, dos blocos definidos na estrutura condicional
- Se limitam a avaliar um teste condicional, ignorando qualquer outra possível situação específica

O Comando *Else*

Analise o trecho de código abaixo:

```
if (x == 10):  
    print("Aqui temos o número 10")  
else:  
    print("Isso definitivamente não é o número 10")  
if (x == "batata"):  
    print("Que tal fritar elas?")  
print("E encerramos por aqui!")
```

Execute para (i) x=10, (ii) x=25, (iii) x="batata", quais são as saídas?

O Comando *Elif*

Ainda, existem cenários em que apenas um teste condicional não é suficiente para tratar todas as possíveis situações em um bloco condicional

Para tratar mais de uma situação específica, podemos utilizar o comando *elif* junto ao *if* ou ao *if-else*

O *elif*, diferente do *else*, permite a definição de um **novo teste condicional que será realizado apenas se os anteriores falharam**

O Comando *Elif*

if teste #1:

Instruções executadas caso o teste #1 seja verdadeiro (**True**)

elif teste #2:

Instruções executadas caso o teste #1 seja falso (**False**) e o teste #2 seja verdadeiro (**True**)

else:

Instruções executadas caso o teste #1 e #2 sejam falsos (**False**)
Instruções executadas independentemente do condicional

Podemos ter mais de um *elif* no bloco condicional, sendo que apenas o primeiro com teste condicional verdadeiro terá suas instruções executadas

O Comando *Elif*

Um exemplo comum do uso de *elif* é verificar o maior entre dois números:

```
a = float(input("Digite um número: "))  
b = float(input("Digite outro número: "))  
if a > b:  
    print("O maior número é {}".format(a))  
elif b > a:  
    print("O maior número é {}".format(b))  
else:  
    print("Os números são iguais!")
```

O Comando *Elif*

Resumindo, blocos condicionais do tipo *if-elif-else*:

- Permitem a criação de fluxos de execução distintos para quando existem múltiplos testes condicionais e também para quando nenhum deles é considerado verdadeiro
- O *else* pode ser removido, sendo possível criar blocos *if-elif*
- Apenas o conjunto de instruções do primeiro teste condicional dado como verdadeiro será executado
- Podem existir múltiplos *elif*, porém sempre existe um único *if* e apenas pode existir um único *else* em um bloco condicional

Condicionais Aninhados

Podemos utilizar blocos condicionais (sejam eles *if*, *if-else*, *if-elif* ou *if-elif-else*) dentro de outros blocos condicionais

A existência de condicionais hierarquizados damos o nome de **condicionais aninhados**

Todas as regras e particularidades dos blocos condicionais, valem quando o mesmo é definido de forma aninhada

Podemos pensar em condicionais aninhados como uma sequência de perguntas

Condicionais Aninhados

```
idade = int(input("Digite a sua idade: "))
```

```
if idade > 15:
```

```
    print("Você já pode votar no Brasil!")
```

```
    if idade > 17:
```

```
        print("Você já é maior de idade no Brasil!")
```

```
        if idade >= 35:
```

```
            print("Você pode se candidatar à presidência do  
Brasil!")
```

```
    else:
```

```
        print("Mas você NÃO é maior de idade no Brasil!")
```

Condicionais Aninhados

Considere uma loja fazendo a seguinte promoção:

- Os descontos são progressivos segundo o número de itens comprados, ao máximo de três itens
- Se o pagamento for à vista via cartão de crédito, cada peça acumula 3% de desconto, via débito, 5%, e se via PIX ou dinheiro, 7%
- Se o pagamento for a prazo, cada peça acumula 2% de desconto

Faça um programa que leia a quantidade de peças adquiridas por um cliente e a forma de pagamento das mesmas e apresente como resultado a **% de desconto que será aplicado ao valor total** da compra

Exercício #10

Receba a idade do cliente pela entrada padrão, defina o dia do mês em uma variável e faça um programa que calcule o **valor do ingresso** de cinema, considere as seguintes condições:

- O valor padrão do ingresso é R\$ 20,00
- Menores de 5 anos não pagam
- Crianças entre 5 e 12 anos pagam meia entrada
- Adolescentes entre 13 e 17 anos têm 20% de desconto
- Maiores de 70 anos não pagam
- Todo o dia 10 todos pagam meia entrada, exceto menores de 5 anos e maiores de 70 (que não pagam)



Fundamentos de Programação

Aula 10

Obrigado e
ATÉ A
PRÓXIMA!