

INSTITUTO FEDERAL

Catarinense

Campus Camboriú

AULA VIII - Condicional Composta e Escolha Caso

Professora: Lidiane Visintin

lidiane.visintin@ifc.edu.br

Professor: Rafael de Moura Speroni

rafael.speroni@ifc.edu.br

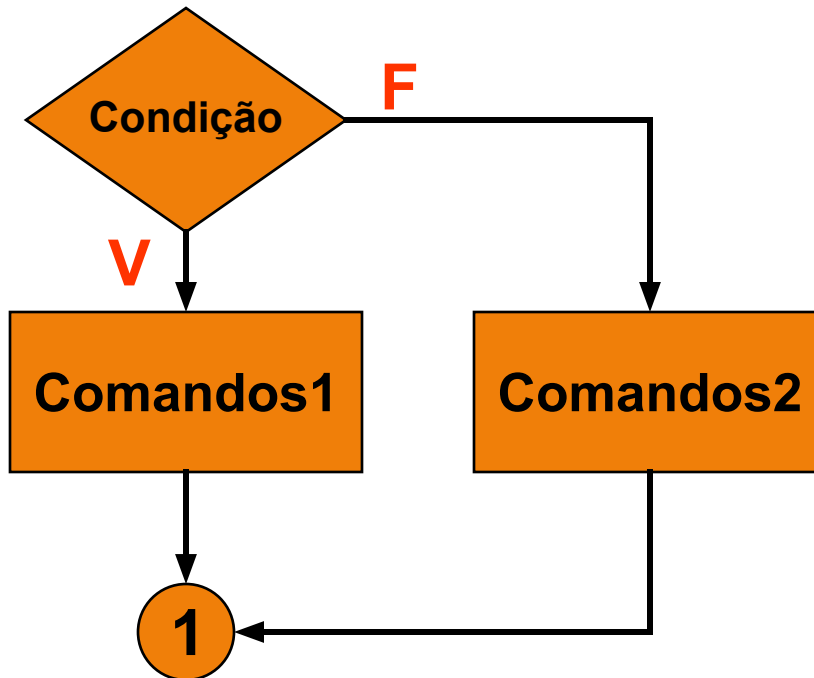
Objetivos da Aula:



- Compreender o uso de **ifs** encadeados;
- Compreender o uso do **elif** (estrutura escolha...caso).

Relembrando

- Representação da estrutura condicional composta em fluxograma



- “**Comandos1**” são executados se a condição for **VERDADEIRA**;
- “**Comandos2**” são executados quando a condição for **FALSA**.

Estrutura Condicional Encadeada

- Sabe-se que dentro de um comando de seleção (comando **if**), podemos colocar qualquer tipo de comando. Assim, dentro de um **if**, pode-se ter outros comandos **if**.
- Ao **utilizar uma condição dentro de outra condição**, estamos construindo uma estrutura condicional encadeada também chamada de **aninhamento** ou **encadeamentos de ses**.

Estrutura Condicional Encadeada

- Sintaxe da estrutura `if` encadeados:

```
if <condição1>:  
    Instrução1  
else:  
    if <condição2>:  
        Instrução2  
    else:  
        Instrução3
```

Observe que dentro do `if` existe uma outra estrutura: `if` que será executada **somente se** a segunda condição for **verdadeira**

Condicionais Encadeadas

Exemplo:

Exemplos > minutos.py > ...

```
1  #Programa conta de telefone com três faixas de preço
2  minutos = int(input("Quantos minutos você utilizou esse mês"))
3
4  √ if minutos < 200:
5      |   preco = 0.20
6  √ else:
7  √     |   if minutos < 400:
8          |       preco = 0.18
9  √     |   else:
10         |       preco = 0.15
11
12  print(f"Você vai pagar este mês: R${minutos * preco}")
```

Condicionais Encadeadas

Exemplo:

Exemplos > minutos.py > ...

```
1  #Programa conta de telefone com três faixas de preço
2  minutos = int(input("Quantos minutos você utilizou esse mês"))
3
4  ✓ if minutos < 200:
5      |     preco = 0.20
6  ✓ else:
7      |     if minutos < 400:
8          |         preco = 0.18
9      |     else:
10         |         preco = 0.15
11
12  print(f"Você vai pagar este mês: R${minutos * preco}")
```

Observe que o print está fora das estruturas aninhadas, portanto **sempre será exibida** esta mensagem

if - elif - else

- Adicionalmente, se existir mais de uma condição alternativa que precisa ser verificada, devemos utilizar o **elif** para avaliar as expressões intermediárias antes de usar o **else**, da seguinte forma:

```
if          <condição1>:  
    Instrução1  
  
elif <condição2>:  
    Instrução2  
  
elif <condição...n>:  
    Instrução...n  
  
else :  
    Instrução3
```

Observe que dentro do **if** existe uma outra estrutura: **elif** que será executada somente se a **condição2** for **verdadeira**. Caso essa condição ainda não seja atendida existem outras alternativas. Por fim, se nenhuma condição for atendida a instrução definida em **else** será executada.

if - elif - else

Exemplo:

Exemplos > categoria.py > ...

```
1  #Programa categoria x preço usando elif
2  categoria = int(input("Digite a categoria do produto"))
3
4  if categoria == 1:
5      preco = 10
6  elif categoria == 2:
7      preco = 18
8  elif categoria == 3:
9      preco = 23
10 elif categoria == 4:
11     preco = 26
12 elif categoria == 5:
13     preco = 31
14 else:
15     print("Categoria inválida, digite um valor entre 1 e 5!")
16     preco = 0
17 print(f"O preço do produto é: R${preco}")
```

if - elif - else

- Em outras linguagens de programação temos o comando **switch case(escolha caso)**, que apresenta uma estrutura muito similar ao **elif** do Python.
- Em português estruturado o comando escolha...caso equivale ao switch...case
- Veja a estrutura escolha ... caso:

```
Escolha (Variável)
Inicio
    Caso (Valor1):
        Instruções;
    Caso (Valor2):
        Instruções;
    Caso (ValorN):
        Instruções;
Fim;
```

Switch case(escolha caso) em Python

- Em Python na versão 3.10 há o **match... case**. Comando equivalente ao switch case de outras linguagens
- Veja a estrutura:

```
match (Variável):  
    Inicio  
        Caso Valor1:  
            Instruções;  
        Caso Valor2:  
            Instruções;  
        Caso _:  
            Instruções;  
    Fim;
```

Switch case(escolha caso) em Python

Exemplo:

```
Exemplo1.py > ...  
1  op = int(input("Informe um número de 1 a 5"))  
2  
3  match op:  
4      case 1:  
5          print("Opção 1")  
6      case 2:  
7          print("Opção 2")  
8      case 3:  
9          print("Opção 3")  
10     case 4:  
11         print("Opção 4")  
12     case 5:  
13         print("Opção 5")  
14     case _:  
15         print("Opção inválida")
```

Exercícios

1. Elabore um algoritmo que leia a idade e o sexo de uma pessoa. Mostre a mensagem conforme a tabela abaixo:

Sexo	Idade	Mensagem
F	0 - 12	Menina
	12 - 24	Moça
	Acima de 24	Mulher
M	0 - 12	Menino
	12 - 24	Rapaz
	Acima de 24	Homem

Exercícios

2. Faça um algoritmo que lê valores para 3 variáveis A, B e C e mostra as mesmas em ordem crescente.
3. Faça um algoritmo que lê valores para 5 variáveis A, B, C, D e E. Encontrar e mostrar o maior valor digitado.
4. Faça um algoritmo que lê valores para duas variáveis X e Y. Se o valor correspondente a 30% da soma de x por y for maior que 500 trocar os valores entre X e Y, senão mostrar o menor valor entre as duas variáveis.

Exercícios

5. Escreva um algoritmo que leia o número de litros vendidos, o tipo de combustível (codificado da seguinte forma: A-álcool, G-gasolina). Calcule e imprima o valor a ser pago pelo cliente sabendo-se que o preço do litro da gasolina é R\$ 7,20 o preço do litro do álcool é R\$ 6,50.

O posto está vendendo combustíveis com a seguinte tabela de descontos:

Álcool	até 20 litros, desconto de 3% por litro no valor
	acima de 20 litros, desconto de 5% por litro no valor
Gasolina	até 20 litros, desconto de 4% por litro no valor
	acima de 20 litros, desconto de 6% por litro no valor

Exercícios



6. Escreva um algoritmo que leia três valores que formam uma data: dia, mês e ano. O usuário deverá, então, informar o número de dias que deseja somar a esta data. Seu algoritmo deverá apresentar a data calculada.

Considere que:

- O usuário sempre informará uma data válida, não sendo necessário fazer este teste;
- Considere que os anos são todos não-bissextos

Referências



Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. Pearson Prentice Hall. 2005

FORBELLONE, André Luiz Villar; Eberspacher, Henri Frederico. Lógica de programação: A construção de algoritmos e estruturas de dados com aplicações em Python. Pearson_GrupoA 2022 331 p. ISBN 9788582605721.

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. Algoritmos: lógica para desenvolvimento de programação de computadores.. 27. ed.. Érica. 2014

Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>