

Unidade 3

Estruturas de Seleção Python

Prof. Me. Renata Cristina Laranja Leite
Disciplina Laboratório de Programação I
Unidade de Computação e Sistemas



Conteúdo da Unidade 3

3.1 Desvio Condicional Simples

3.2 Desvio Condicional Composto

3.3 Desvio Condicional Encadeado

3.4 Desvio Condicional Seletivo

Desvio Condicional Simples

O comando **if** é sempre utilizado quando você precisa escolher entre executar blocos diferentes de códigos dependendo do resultado de uma condição definida (condição lógica).

Seleção Simples: testa uma condição, se esta for verdadeira executa uma ação ou um conjunto de ações.

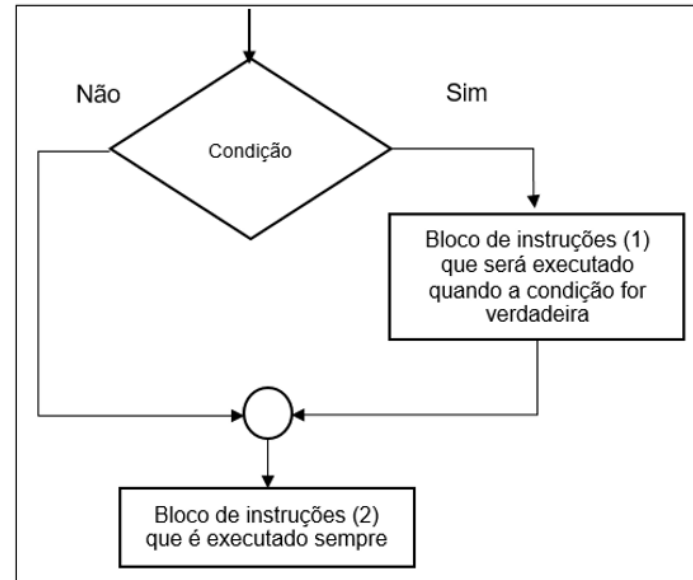
Sintaxe:

```
if <condição>:  
    instrução1
```

Importante:

- A linha do if deve ser encerrada com os dois pontos(:);
- A(s) linha(s) a ser executada(s), caso a condição seja verdadeira, deverá(ão) estar com um recuo da margem esquerda. Este recuo deve ser realizado através da tecla <tab>. Tudo que estiver recuado, abaixo do if, será executando somente se a condição for verdadeira.

Exemplo Desvio Condicional Simples



```
nome=input("Digite o nome: ")
idade=int(input("Digite a idade: "))
prioridade="NÃO"
if idade>=65:
    prioridade="SIM"
print("O paciente " + nome + " possui atendimento prioritário? " + prioridade)
```

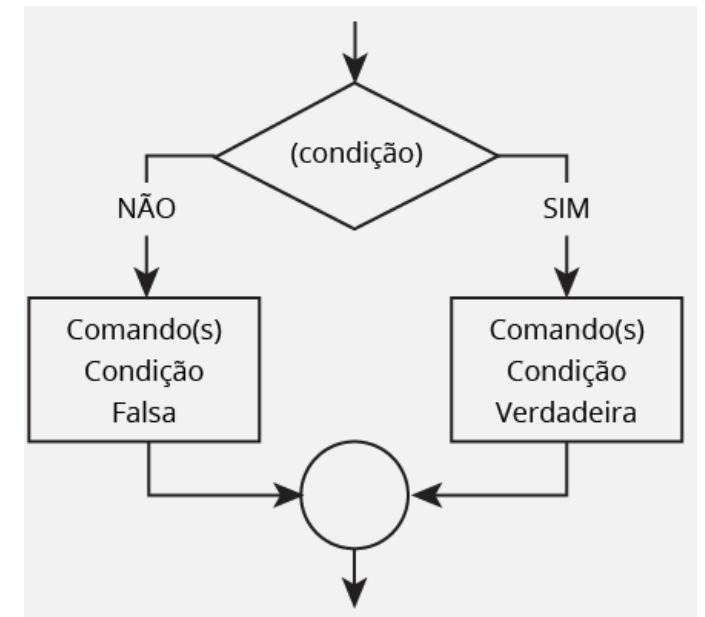
Desvio Condicional Composto

Seleção Composta: testa uma condição, se esta for verdadeira executa uma ação ou um conjunto de ações. Caso contrário, executa outro conjunto de ações. Sintaxe:

```
if <condição>:  
    instrução1  
else:  
    instrução2
```

Se a condição testada for verdadeira, a instrução1 será executada.

A instrução2 só será executada se a condição for falsa.



Exemplo Desvio Condicional Composto

```
nome=input("Digite o nome: ")
idade=int(input("Digite a idade: "))
if idade>=65:
    print("O paciente " + nome + " POSSUI atendimento prioritário!")
else:
    print("O paciente " + nome + " NÃO possui atendimento prioritário!")
```

Desvio Condicional Composto

Seleção Encadeada ou Aninhada: agrupa várias alternativas a fim de inspecionar mais de duas condições.

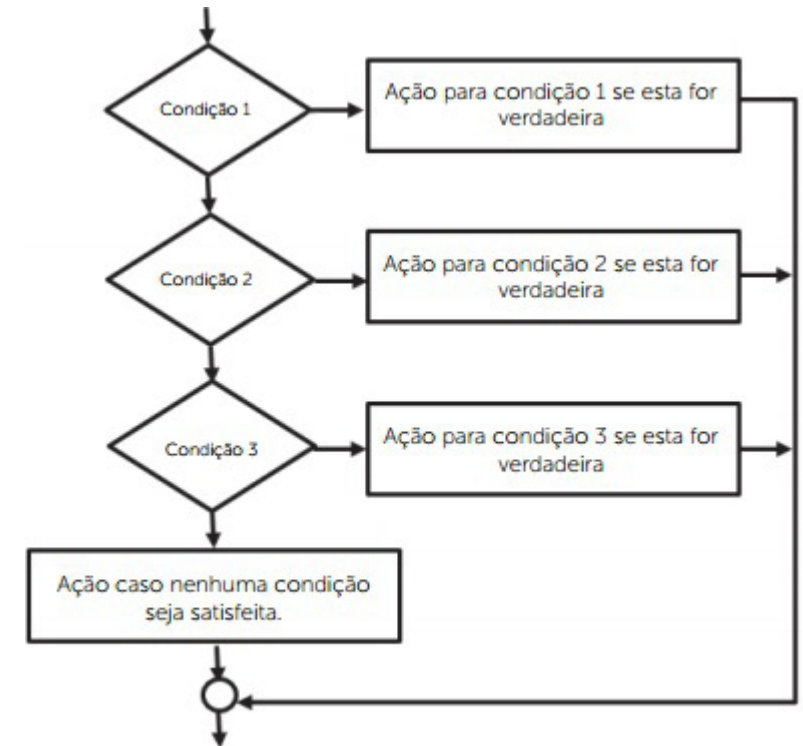
Sintaxe:

```
if <condição>:  
    instrução1  
elif <condição2>:  
    instrução2  
elif <condição3>:  
    instrução3  
else:  
    instrução4
```

Nesta composição do comando if, temos várias condições para serem testadas.

A condição que primeiro for verdadeira, executará(ão) a(s) instrução(ões) que estiver(em) dentro do bloco.

As demais condições não serão testadas e nem executarão instruções.



Exemplo 1: Desvio Condicional Composto e aninhado

Transforma todos os caracteres em maiúsculo.

```
nome=input("Digite o nome: ")
idade=int(input("Digite a idade: "))
doenca_infectocontagiosa=input("Suspeita de doença infecto-contagiosa?").upper()
if idade>=65:
    print("O paciente " + nome + " POSSUI atendimento prioritário!")
elif doenca_infectocontagiosa=="SIM":
    print("O paciente " + nome + " deve ser direcionado para sala de espera reservada.")
else:
    print("O paciente " + nome + " NÃO possui atendimento prioritário e pode aguardar na
```


Exemplo 2: Desvio Condicional Composto e aninhado

Bloco de instruções

Bloco de instruções

```
nome=input("Digite o nome: ")
idade=int(input("Digite a idade: "))
doenca_infectocontagiosa=input("Suspeita de doença infectocontagiosa?").upper()
if idade >= 65:
    print("Paciente COM prioridade")
    if doenca_infectocontagiosa=="SIM":
        print("Encaminhe o paciente para sala AMARELA")
    elif doenca_infectocontagiosa=="NAO":
        print("Encaminhe o paciente para sala BRANCA")
    else:
        print("Responda a suspeita de doença infectocontagiosa com SIM ou NAO")
else:
    print("Paciente SEM prioridade")
    if doenca_infectocontagiosa=="SIM":
        print("Encaminhe o paciente para sala AMARELA")
    elif doenca_infectocontagiosa=="NAO":
        print("Encaminhe o paciente para sala BRANCA")
    else:
        print("Responda a suspeita de doença infectocontagiosa com SIM ou NAO")
```

Exemplo 3: Desvio Condicional Composto e Aninhado

Pode colocar
quantos **elif** forem
necessários no
comando if.

```
nome=input("Digite o nome: ")
idade=int(input("Digite a idade: "))
doenca_infectocontagiosa=input("Suspeita de doença infectocontagiosa? sim-nao-talvez ").upper()
if idade >= 65:
    print("Paciente COM prioridade")
    if doenca_infectocontagiosa=="SIM":
        print("Encaminhe o paciente para sala AMARELA")
    elif doenca_infectocontagiosa=="NAO":
        print("Encaminhe o paciente para sala BRANCA")
    elif doenca_infectocontagiosa=="TALVEZ":
        print("Encaminhe o paciente para sala de TRIAGEM")
    else:
        print("Responda a suspeita de doença infectocontagiosa com SIM, NAO ou TALVEZ")
else:
    print("Paciente SEM prioridade")
    if doenca_infectocontagiosa=="SIM":
        print("Encaminhe o paciente para sala AMARELA")
    elif doenca_infectocontagiosa=="NAO":
        print("Encaminhe o paciente para sala BRANCA")
    elif doenca_infectocontagiosa=="TALVEZ":
        print("Encaminhe o paciente para sala de TRIAGEM")
    else:
        print("Responda a suspeita de doença infectocontagiosa com SIM, NAO ou TALVEZ")
```

Estruturas de Condição com Operadores Lógicos

Podemos avaliar duas ou mais expressões por meio de operadores lógicos com o `and`, `or` e o `not`. Para que a condição do `if` seja verdadeira e execute a instrução, é preciso que satisfaça as condições de cada operador lógico.

Sintaxe:

```
if <condição1> and <condição2>:  
    instrução1
```

```
if <condição1> or <condição3>:  
    instrução2
```

```
if not <condição1>:  
    instrução3
```

Exemplo de Estruturas de Condição com operadores lógicos

AND

```
a = 6
if (a <= 10) and ((a % 2) == 0):
    print("O valor", a, "é menor ou igual a 10 e é um número par.")
else:
    print("O valor", a, "não é menor que 10 ou não é par")
```

OR

```
a = 12
if (a <= 10) or ((a % 2) == 0):
    print("O valor", a, "é menor ou igual a 10 ou é um número par.")
else:
    print("O valor", a, "não é menor que 10 ou não é par")
```

NOT

```
a = 12
if not (a > 10):
    print("O valor", a, "é menor ou igual a 10")
else:
    print("O valor", a, "não é menor que 10")
```

Comando MATCH-CASE no Python

Na versão 3.10 do Python podemos encontrar o comando MATCH-CASE. Este comando equivalente ao SWITCH-CASE, comando encontrado em várias linguagens como C, Java, Javascript, C++, C#, entre outras.

Acesse o link para mais informações deste comando:

<https://www.hashtagtreinamentos.com/match-case-no-python>

```
dia = int(input("Digite o número do dia da semana:"))
match dia:
    case 1:
        print("Domingo")
    case 2:
        print("Segunda-Feira")
    case 3:
        print("Terça-Feira")
    case 4:
        print("Quarta-Feira")
    case 5:
        print("Quinta-Feira")
    case 6:
        print("Sexta-Feira")
    case 7:
        print("Sábado")
    case _:
        print("Valor",dia,"inválido")
```

Simulando o SWITCH-CASE com o MATCH-CASE do Python usando if-elif

```
dia = int(input("Digite o número do dia da semana:"))
if dia == 1:
    print("Domingo")
elif dia == 2:
    print("Segunda-Feira")
elif dia == 3:
    print("Terça-Feira")
elif dia == 4:
    print("Quarta-Feira")
elif dia == 5:
    print("Quinta-Feira")
elif dia == 6:
    print("Sexta-Feira")
elif dia == 7:
    print("Sábado")
else:
    print("Valor", dia, "inválido")
```

Hora de praticar
com a lista de
exercícios sobre
este assunto.





- Apostila Programação Python de Prof. Rogério Coelho
- <https://blog.betrybe.com/python/if-else/#:~:text=If%20em%20Python,se%20a%20condi%C3%A7%C3%A3o%20for%20verdadeira.>
- <https://pt.stackoverflow.com/questions/94540/problema-ao-utilizar-switch-case-no-python-3>
- <https://www.programandocomcarlos.com.br/2020/01/duas-formas-de-criar-um-switch-case-em.html>

Referências