

Aula 3

Professores:

Dante Corbucci Filho
Leandro A. F. Fernandes

Conteúdo:

- Estruturas de Controle
 - Sequência
 - Estruturas de Seleção
 - Estruturas de Repetição

Sequência de Comandos

Em Python, para construirmos uma sequência de comandos, devemos colocá-los um por linha, um abaixo do outro, mantendo-os na mesma tabulação (na mesma indentação).

```
x = int(input("Digite o primeiro valor: "))
y = int(input("Digite o segundo valor: "))
soma = x + y
prod = x * y
print("A Soma =", soma, "e o Produto =", prod)
```

Bloco de Comandos (*Suite*)

Em Python, um bloco de comandos contendo um ou mais comandos é chamado de **suite**. No exemplo abaixo, temos uma suite com cinco comandos em sequência.

```
x = int(input("Digite o primeiro valor: "))
y = int(input("Digite o segundo valor: "))
soma = x + y
prod = x * y
print("A Soma =", soma, "e o Produto =", prod)
```

Comentários

- Comentários simples são iniciados com o caractere #
- Comentários com múltiplas linhas podem ser delimitados em seu início e término por:
 - Três aspas simples, isto é: ““ e ””
 - Três aspas duplas, isto é: “”” e ””” (os docStrings)

““

Programa que lê do teclado um par de números inteiros e escreve na tela a soma e o produto dos números lidos.

””

```
x = int(input("Digite o primeiro valor: "))
y = int(input("Digite o segundo valor: "))
soma = x + y
prod = x * y
print("A Soma =", soma, "e o Produto =", prod)
```

Comentários

- Comentários simples são iniciados com o caractere #
- Comentários com múltiplas linhas podem ser delimitados em seu início e término por:
 - Três aspas simples, isto é: “” e ””
 - Três aspas duplas, isto é: “””” e ”””” (os docStrings)

““

Programa que lê do teclado um par de números inteiros e escreve na tela a soma e o produto dos números lidos.

””

```
x = int(input("Digite o primeiro valor: "))      # Lê o primeiro número
y = int(input("Digite o segundo valor: "))      # Lê o segundo número
soma = x + y                                     # Calcula a soma
prod = x * y                                    # Calcula o produto
print("A Soma =", soma, "e o Produto =", prod)  # Escreve resultados
```

Execução do Programa

Exemplo:

```
x = int(input("Digite o primeiro valor: "))  
y = int(input("Digite o segundo valor: "))  
soma = x + y  
prod = x * y  
print("A Soma =", soma, "e o Produto =", prod)
```

Execução do Programa

Exemplo:

```
x = int(input("Digite o primeiro valor: "))  
y = int(input("Digite o segundo valor: "))  
soma = x + y  
prod = x * y  
print("A Soma =", soma, "e o Produto =", prod)
```

Execução:

```
Digite o primeiro valor: 2 <enter>
```

Execução do Programa

Exemplo:

```
x = int(input("Digite o primeiro valor: "))  
y = int(input("Digite o segundo valor: "))  
soma = x + y  
prod = x * y  
print("A Soma =", soma, "e o Produto =", prod)
```

Execução:

```
Digite o primeiro valor: 2 <enter>  
Digite o segundo valor: 13 <enter>
```


Execução do Programa

Exemplo:

```
x = int(input("Digite o primeiro valor: "))  
y = int(input("Digite o segundo valor: "))  
soma = x + y  
prod = x * y  
print("A Soma =", soma, "e o Produto =", prod)
```

Execução:

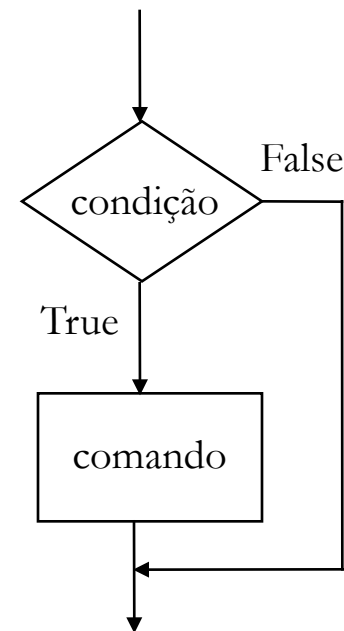
```
Digite o primeiro valor: 2 <enter>  
Digite o segundo valor: 13 <enter>  
A Soma = 15 e o Produto = 26 <enter>
```

Estrutura de Seleção com Um Ramo (if)

A estrutura de seleção com um ramo **if** é utilizada quando se deseja executar um comando (ou uma suite) apenas no caso de uma condição ser satisfeita.

```
if <condição>:  
    <comando>
```

<condição> representa uma expressão booleana, ou seja, expressão cujo resultado é verdadeiro (True) ou falso (False).



Estrutura de Seleção com Um Ramo (*if*)

Exemplo:

```
valor = float(input("Entre com um valor: "))  
if valor>0:  
    print(valor, "é maior do que zero.")
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplo:

```
valor = float(input("Entre com um valor: "))  
if valor>0:  
    print(valor, "é maior do que zero.")
```

Execução:

```
Entre com um valor:  4.7 <enter>  
  
4.7 é maior do que zero. <enter>
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplo:

```
salario = float(input("Diga seu salário atual: "))  
if salario > 10000:  
    inps = salario * 0.10  
    impostoRenda = salario * 0.15  
    print("Valor do INPS:", inps, "e do Imposto de Renda:", impostoRenda)
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplo:

```
salario = float(input("Diga seu salário atual: "))  
if salario > 10000:  
    inps = salario * 0.10  
    impostoRenda = salario * 0.15  
    print("Valor do INPS:", inps, "e do Imposto de Renda:", impostoRenda)
```

Execução:

```
Diga seu salário atual: 11000 <enter>  
  
Valor do INPS: 1100.0 e do Imposto de Renda: 1650.0 <enter>
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplos de condições compostas:

```
if (valor >= 10) and (valor <= 20):  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplos de condições compostas:

```
if (valor >= 10) and (valor <= 20):  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

```
if 10<=valor<= 20:  
    print(valor, "ocorre dentro do intervalo [10,20]")
```


Estrutura de Seleção com Um Ramo (*if*)

Exemplos de condições compostas:

```
if (valor >= 10) and (valor <= 20):  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

```
if 10<=valor<= 20:  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

```
if (valor < 0) or (valor > 30):  
    print(valor, "ocorre fora do intervalo [0,30]")
```

Estrutura de Seleção com Um Ramo (*if*)

Exemplos de condições compostas:

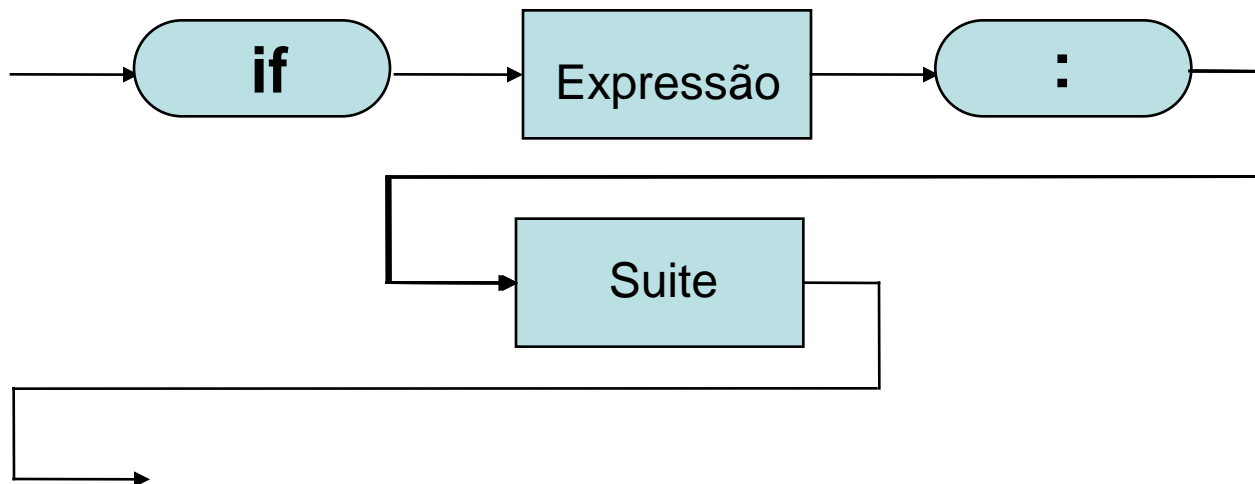
```
if (valor >= 10) and (valor <= 20):  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

```
if 10<=valor<= 20:  
    print(valor, "ocorre dentro do intervalo [10,20]")
```

```
if (valor < 0) or (valor > 30):  
    print(valor, "ocorre fora do intervalo [0,30]")
```

```
if ((valor >= 0) and (valor < 10)) or ((valor > 20) and (valor <= 30)):  
    print(valor, "ocorre dentro do intervalo [0,10) ou em (20,30]")
```

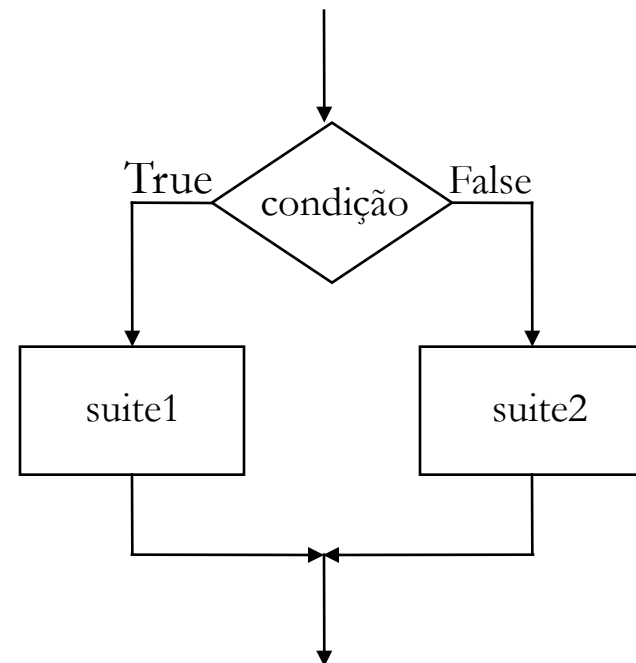
Diagrama Sintático da Seleção *if*



Estrutura de Seleção com Dois Ramos (*if-else*)

A estrutura de seleção **if-else** é utilizada quando se deseja executar uma entre duas suites, dependendo do resultado da avaliação de uma condição.

```
if <condição>:  
    <suite1>  
else:  
    <suite2>
```



Estrutura de Seleção com Dois Ramos (*if-else*)

Exemplo:

```
valor = float(input("Entre com um valor:"))  
if valor>0:  
    print(valor, "é maior do que zero.")  
else:  
    print(valor, "é menor ou igual a zero.")
```

Estrutura de Seleção com Dois Ramos (*if-else*)

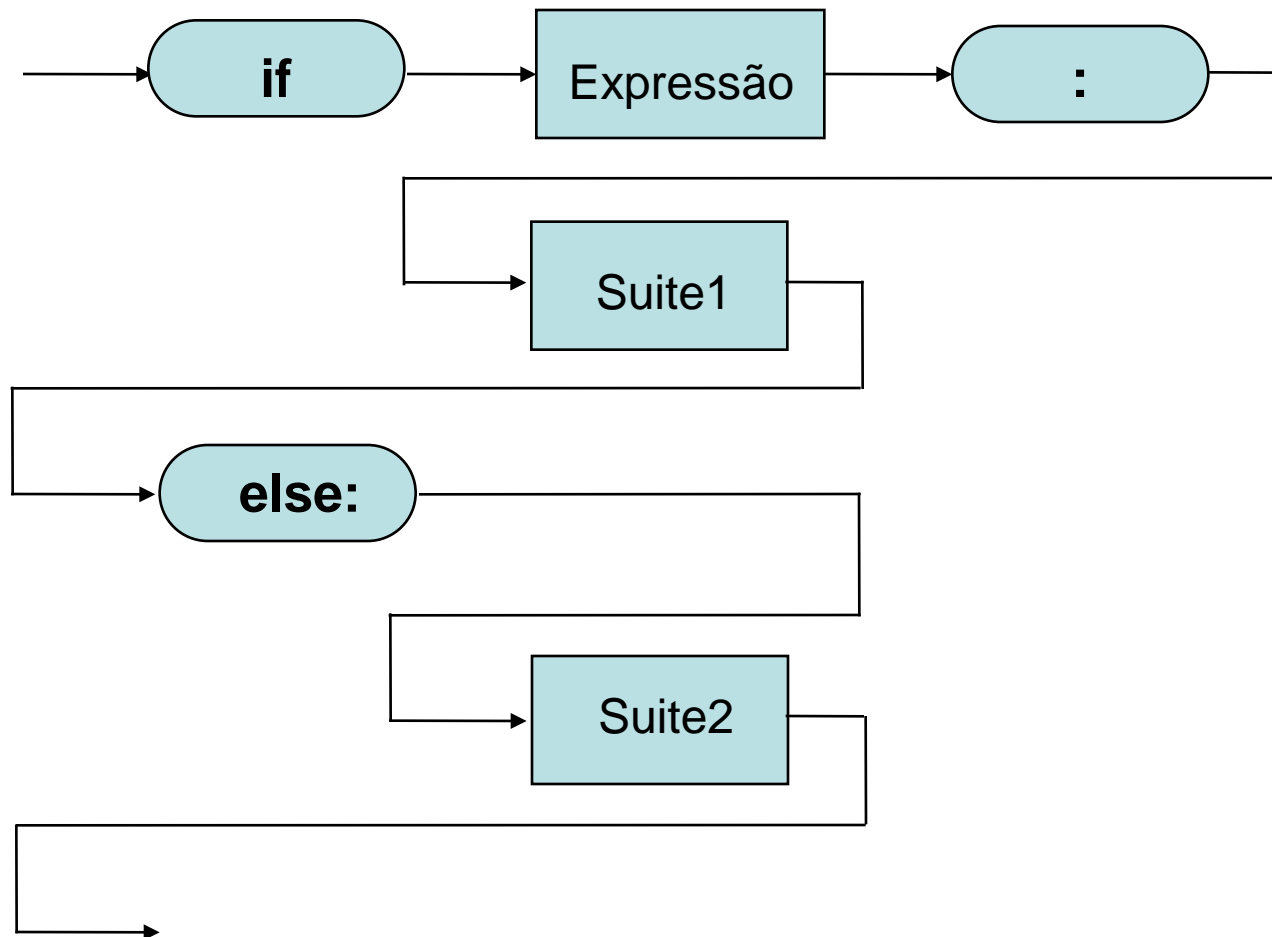
Exemplo:

```
valor = float(input("Entre com um valor:"))  
if valor>0:  
    print(valor, "é maior do que zero.")  
else:  
    print(valor, "é menor ou igual a zero.")
```

Execução:

```
Entre com um valor: -13.22 <enter>  
  
-13.22 é menor ou igual a zero.<enter>
```

Diagrama Sintático de Seleção *if-else*



Estrutura de Seleção Aninhadas

O programa a seguir calcula o abono salarial que uma empresa concederá aos seus funcionários com mais de um ano de tempo de serviço. Os que têm menos de dez anos ganharão abono de 10%. Os demais ganharão de 25%.

“

Abono Salarial

”

```
salario = float(input("Diga seu salário atual: "))
tempo = int(input("Diga quantos anos completos tem de serviço: "))
if tempo < 1:
    print("Seu salário se mantém o mesmo:", salario)
else:
    if tempo < 10:
        salario = salario * 1.10
    else:
        salario = salario * 1.25
    print("Seu novo salário, com abono:", salario)
```


Estrutura de Seleção Aninhadas

O programa a seguir calcula o abono salarial que uma empresa concederá aos seus funcionários com mais de um ano de tempo de serviço. Os que têm menos de dez anos ganharão abono de 10%. Os demais ganharão de 25%.

““

Abono Salarial

””

```
salario = float(input("Diga seu salário atual: "))
tempo = int(input("Diga quantos anos completos tem de serviço: "))
if tempo < 1:
    print("Seu salário se mantém o mesmo:", salario)
else:
    if tempo < 10:
        salario = salario * 1.10
    else:
        salario = salario * 1.25
    print("Seu novo salário, com abono:", salario)
```

Estrutura de Seleção Aninhadas

O programa a seguir calcula o abono salarial que uma empresa concederá aos seus funcionários com mais de um ano de tempo de serviço. Os que têm menos de dez anos ganharão abono de 10%. Os demais ganharão de 25%.

```
““
```

```
Abono Salarial
```

```
””
```

```
salario = float(input("Diga seu salário atual: "))
```

```
tempo = int(input("Diga quantos anos completos tem de serviço: "))
```

```
if tempo<1:
```

```
    print("Seu salário se mantém o mesmo:", salario)
```

```
else:
```

```
    if tempo<10:
```

```
        salario = salario * 1.10
```

```
    else:
```

```
        salario = salario * 1.25
```

```
    print("Seu novo salário, com abono:", salario)
```

Estrutura de Seleção com Múltiplos Ramos (if-elif ou if-elif-else)

As estruturas de seleção **if-elif** e **if-elif-else** são utilizadas quando se deseja executar uma entre várias suites, dependendo dos valores das expressões.

```
if <expressao1>:  
    <suite1>  
elif <expressao2>:  
    <suite2>  
...  
elif <expressaoN>:  
    <suiteN>
```

Estrutura de Seleção com Múltiplos Ramos (if-elif ou if-elif-else)

As estruturas de seleção **if-elif** e **if-elif-else** são utilizadas quando se deseja executar uma entre várias suites, dependendo dos valores das expressões.

```
if <expressao1>:  
    <suite1>  
elif <expressao2>:  
    <suite2>  
  
...  
elif <expressaoN>:  
    <suiteN>
```

```
if <expressao1>:  
    <suite1>  
elif <expressao2>:  
    <suite2>  
  
...  
elif <expressaoN>:  
    <suiteN>  
else:  
    <suiteCasoContrário>
```

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
x = int(valores[0])      # primeira substring da lista
y = int(valores[1])      # segunda substring da lista
op = input("Informe o operador (+, -, *, / ou **): ")
if op=="+":
    resultado = x + y
elif op=="-":
    resultado = x - y
elif op=="*":
    resultado = x * y
elif op=="/":
    resultado = x / y
elif op=="**":
    resultado = x ** y
print(x, op, y, "=", resultado)
```

Operação a ser vista que divide a string, considerando o espaço em branco como separador, em uma lista de substrings.

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
x = int(valores[0])      # primeira substring da lista
y = int(valores[1])      # segunda substring da lista
op = input("Informe o operador (+, -, *, / ou **): ")
```

```
if op=="+":
    resultado = x + y
elif op=="-":
    resultado = x - y
elif op=="*":
    resultado = x * y
elif op=="/":
    resultado = x / y
elif op=="**":
    resultado = x ** y
print(x, op, y, "=", resultado)
```

Operação a ser vista que divide a string, considerando o espaço em branco como separador, em uma lista de substrings.

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
```

```
x = int(valores[0])      # primeira substring da lista
```

```
y = int(valores[1])      # segunda substring da lista
```

```
op = input("Informe o operador (+, -, *, / ou **): ")
```

```
if op=="+":
```

```
    resultado = x + y
```

```
elif op=="-":
```

```
    resultado = x - y
```

```
elif op=="*":
```

```
    resultado = x * y
```

```
elif op=="/":
```

```
    resultado = x / y
```

```
elif op=="**":
```

```
    resultado = x ** y
```

```
print(x, op, y, "=", resultado)
```

Operação a ser vista que divide a string, considerando o espaço em branco como separador, em uma lista de substrings.

Execução:

```
Entre com dois inteiros positivos: 2 9 <enter>
```

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
```

```
x = int(valores[0])      # primeira substring da lista
```

```
y = int(valores[1])      # segunda substring da lista
```

```
op = input("Informe o operador (+, -, *, / ou **): ")
```

```
if op=="+":
```

```
    resultado = x + y
```

```
elif op=="-":
```

```
    resultado = x - y
```

```
elif op=="*":
```

```
    resultado = x * y
```

```
elif op=="/":
```

```
    resultado = x / y
```

```
elif op=="**":
```

```
    resultado = x ** y
```

```
print(x, op, y, "=", resultado)
```

Operação a ser vista que divide a string, considerando o espaço em branco como separador, em uma lista de substrings.

Execução:

```
Entre com dois inteiros positivos: 2 9 <enter>
```

```
Informe o operador (+, -, *, / ou **): ** <enter>
```


Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
x = int(valores[0])      # primeira substring da lista
y = int(valores[1])      # segunda substring da lista
op = input("Informe o operador (+, -, *, / ou **): ")
```

```
if op=="+":
    resultado = x + y
elif op=="-":
    resultado = x - y
elif op=="*":
    resultado = x * y
elif op=="/":
    resultado = x / y
elif op=="**":
    resultado = x ** y
print(x, op, y, "=", resultado)
```

Operação a ser vista que divide a string, considerando o espaço em branco como separador, em uma lista de substrings.

Execução:

```
Entre com dois inteiros positivos: 2 9 <enter>
Informe o operador (+, -, *, / ou **): ** <enter>
2 ** 9 = 512 <enter>
```

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
x = int(valores[0])
y = int(valores[1])
op = input("Informe o operador (+, -, *, / ou **): ")
if op=="+":
    resultado = x + y
elif op=="-":
    resultado = x - y
elif op=="*":
    resultado = x * y
elif op=="/":
    resultado = x / y
elif op=="**":
    resultado = x ** y
else:
    resultado = None
if resultado == None:
    print(op, ": Operador inexistente!!")
else:
    print(x, op, y, "=", resultado)
```

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
x = int(valores[0])
y = int(valores[1])
op = input("Informe o operador (+, -, *, / ou **): ")

if op=="+":
    resultado = x + y
elif op=="-":
    resultado = x - y
elif op=="*":
    resultado = x * y
elif op=="/":
    resultado = x / y
elif op=="**":
    resultado = x ** y
else:
    resultado = None

if resultado == None:
    print(op, ": Operador inexistente!!")
else:
    print(x, op, y, "=", resultado)
```

Estrutura de Seleção com Múltiplos Ramos

```
valores = input("Entre com dois inteiros positivos: ").split()
```

```
x = int(valores[0])
```

```
y = int(valores[1])
```

```
op = input("Informe o operador (+, -, *, / ou **): ")
```

```
if op=="+":
```

```
    resultado = x + y
```

```
elif op=="-":
```

```
    resultado = x - y
```

```
elif op=="*":
```

```
    resultado = x * y
```

```
elif op=="/":
```

```
    resultado = x / y
```

```
elif op=="**":
```

```
    resultado = x ** y
```

```
else:
```

```
    resultado = None
```

```
if resultado == None:
```

```
    print(op, ": Operador inexistente!!")
```

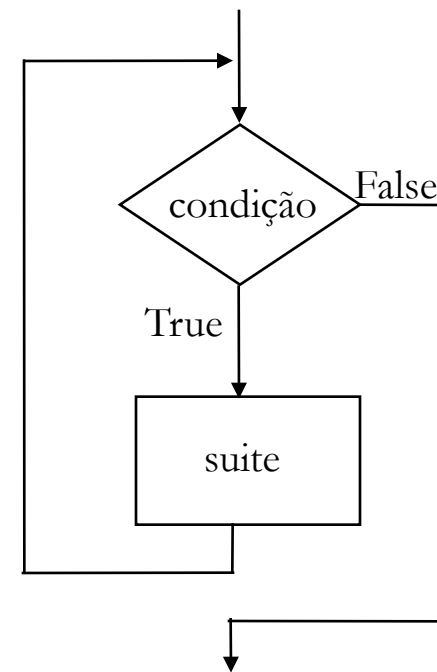
```
else:
```

```
    print(x, op, y, "=", resultado)
```

Estrutura de Repetição Indefinida (*while*)

A estrutura de repetição **while** é utilizada quando se deseja executar uma suite enquanto uma condição for verdadeira (zero ou mais vezes).

```
while <condição>:  
    <suite>
```



Estrutura de Repetição Indefinida (*while*)

Exemplo:

```
indice = 1                                # inicializa variável
while indice <= 10:                       # testa condição
    print(indice, end=" ")                # escreve o valor da variável seguido de espaço " "
    indice = indice + 1                   # incrementa a variável que controla a repetição
print()                                   # pula linha no vídeo
```

Estrutura de Repetição Indefinida (*while*)

Exemplo:

```
indice = 1                # inicializa variável
while indice <= 10:        # testa condição
    print(indice, end=" ") # escreve o valor da variável seguido de espaço " "
    indice = indice + 1    # incrementa a variável que controla a repetição
print()                   # pula linha no vídeo
```

Execução:

```
1 2 3 4 5 6 7 8 9 10 <enter>
```

Outro Exemplo: Cálculo do Fatorial

$$\begin{aligned} N! &= N * (N-1) * (N-2) * \dots * 2 * 1, & \text{para } N > 0. \\ N! &= 1, & \text{para } N = 0. \end{aligned}$$

```
num = int(input("Digite um valor inteiro e positivo: "))
i = 1
fat = 1
while i <= num:
    fat = fat * i
    i = i + 1
print("O fatorial de", num, "=", fat)
```


Outro Exemplo: Cálculo do Fatorial

$$N! = N * (N-1) * (N-2) * \dots * 2 * 1, \quad \text{para } N > 0.$$
$$N! = 1, \quad \text{para } N = 0.$$

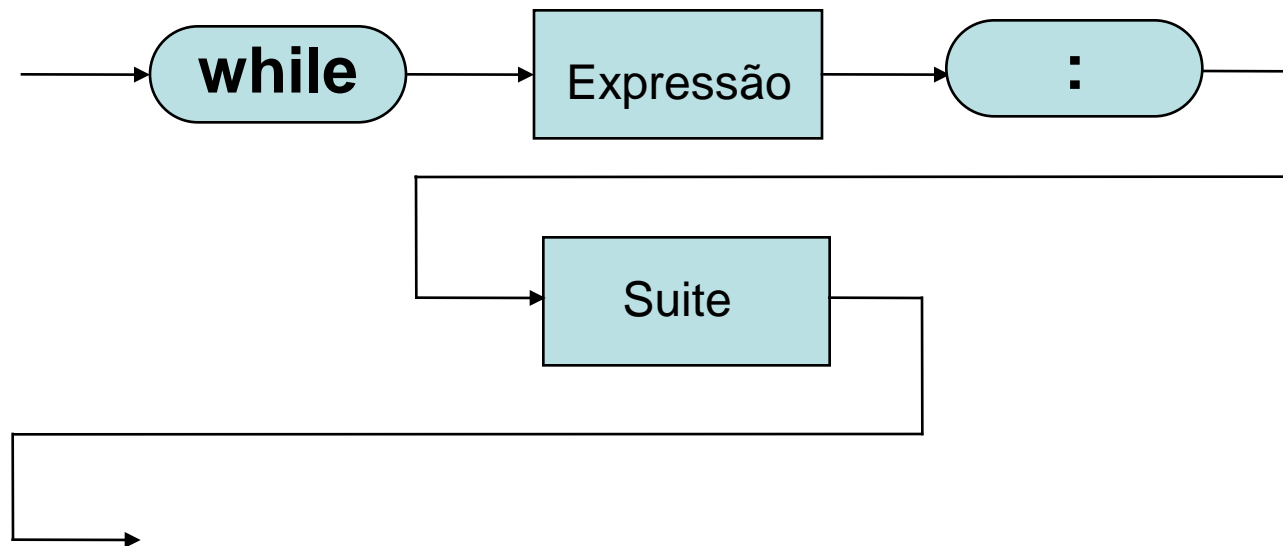
```
num = int(input("Digite um valor inteiro e positivo: "))
i = 1
fat = 1
while i <= num:
    fat = fat * i
    i = i + 1
print("O fatorial de", num, "=", fat)
```

Execução:

```
Digite valor inteiro e positivo: 5 <enter>

O fatorial de 5 = 120 <enter>
```

Diagrama Sintático de Repetição *while*



Estrutura de Repetição Definida (*for*)

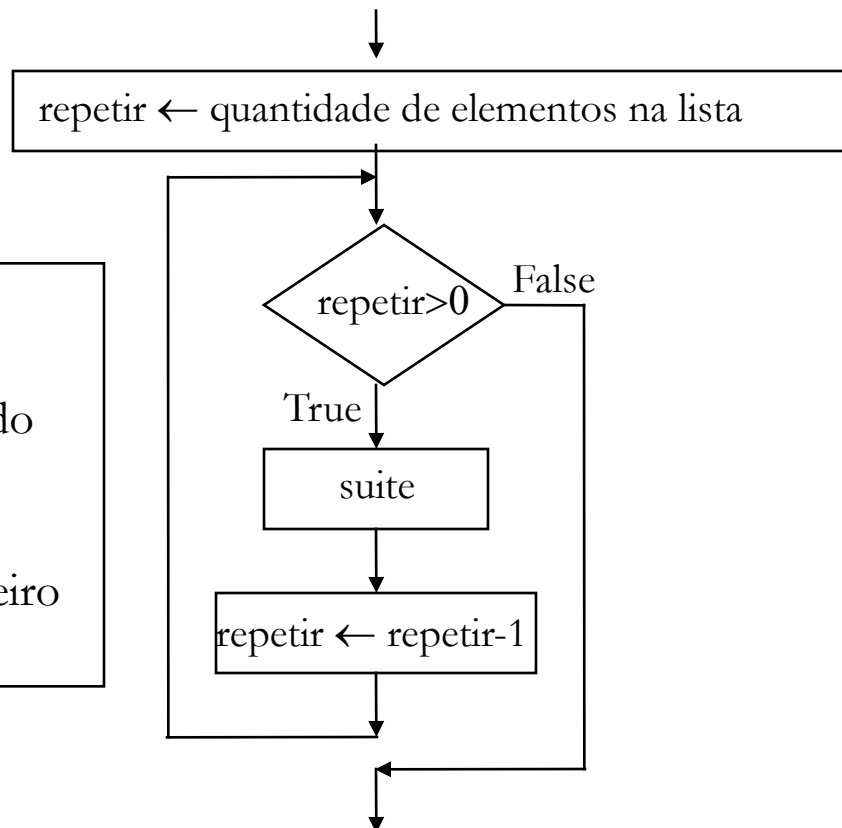
A estrutura de repetição **for** é utilizada quando se deseja executar uma mesma suite um número específico de vezes, enumerado em uma lista de valores.

```
for <var> in <lista de valores>:  
    <suite>
```

Estrutura de Repetição Definida (for)

```
for <var> in <lista>:  
    <suite>
```

A cada ciclo da repetição, <var> assume o valor referenciado pela posição do elemento na lista. Isto é, o valor referenciado progressivamente do primeiro até o último elemento.



Estrutura de Repetição Definida (*for*)

Exemplo:

```
for item in [3,4,5,6,7,8,9]: # item assume progressivamente os valores na lista
    print(item, end=" ")      # cada valor de item é escrito na tela, separado por um espaço
print()                     # pula linha (imprime "\n")
```

Estrutura de Repetição Definida (*for*)

Exemplo:

```
for item in [3,4,5,6,7,8,9]: # item assume progressivamente os valores na lista
    print(item, end=" ")      # cada valor de item é escrito na tela, separado por um espaço
print()                     # pula linha (imprime "\n")
```

Execução:

```
3 4 5 6 7 8 9 <enter>
```

Listas de Valores para uma Repetição **for**

- Para criar uma lista com uma progressão aritmética de elementos, bastante comum em repetições **for**, podemos utilizar a função `range`

Listas de Valores para uma Repetição **for**

- Para criar uma lista com uma progressão aritmética de elementos, bastante comum em repetições **for**, podemos utilizar a função `range`
- **`range(valor limite)`**
Cria uma lista com progressão aritmética de itens de razão 1, iniciada pelo valor 0 e terminada no valor que antecede o *limite*
 - Exemplo: **`range(5)`** cria a lista [0, 1, 2, 3, 4]

Listas de Valores para uma Repetição **for**

- Para criar uma lista com uma progressão aritmética de elementos, bastante comum em repetições **for**, podemos utilizar a função `range`
- **`range(valor limite)`**
Cria uma lista com progressão aritmética de itens de razão 1, iniciada pelo valor 0 e terminada no valor que antecede o *limite*
 - Exemplo: **`range(5)`** cria a lista [0, 1, 2, 3, 4]
- **`range(valor inicial, valor limite)`**
Cria uma lista com progressão aritmética de itens de razão 1, iniciada pelo *valor inicial* e terminada no valor que antecede o *limite*
 - Exemplo: **`range(8, 13)`** cria a lista [8, 9, 10, 11, 12]

Listas de Valores para uma Repetição **for**

- Para criar uma lista com uma progressão aritmética de elementos, bastante comum em repetições **for**, podemos utilizar a função `range`
- **`range(valor limite)`**
Cria uma lista com progressão aritmética de itens de razão 1, iniciada pelo valor 0 e terminada no valor que antecede o *limite*
 - Exemplo: **`range(5)`** cria a lista [0, 1, 2, 3, 4]
- **`range(valor inicial, valor limite)`**
Cria uma lista com progressão aritmética de itens de razão 1, iniciada pelo *valor inicial* e terminada no valor que antecede o *limite*
 - Exemplo: **`range(8, 13)`** cria a lista [8, 9, 10, 11, 12]
- **`range(valor inicial, valor limite, avanço)`**
Cria uma lista com progressão aritmética de itens de razão *avanço*, iniciada pelo *valor inicial* e terminada no valor que antecede o *limite*
 - Exemplo: **`range(1, 30, 5)`** cria a lista [1, 6, 11, 16, 21, 26]
 - Exemplo: **`range(5, -14, -3)`** cria a lista [5, 2, -1, -4, -7, -10, -13]

Outro Exemplo: Cálculo do Fatorial

$$\begin{aligned} N! &= N * (N-1) * (N-2) * \dots * 2 * 1, & \text{para } N > 0. \\ N! &= 1, & \text{para } N = 0. \end{aligned}$$

```
num = int(input("Digite valor inteiro e positivo: "))
fat = 1
for i in range(1,num+1):
    fat = fat * i
print("O fatorial de", num, "=", fat)
```

Outro Exemplo: Cálculo do Fatorial

$$\begin{aligned} N! &= N * (N-1) * (N-2) * \dots * 2 * 1, & \text{para } N > 0. \\ N! &= 1, & \text{para } N = 0. \end{aligned}$$

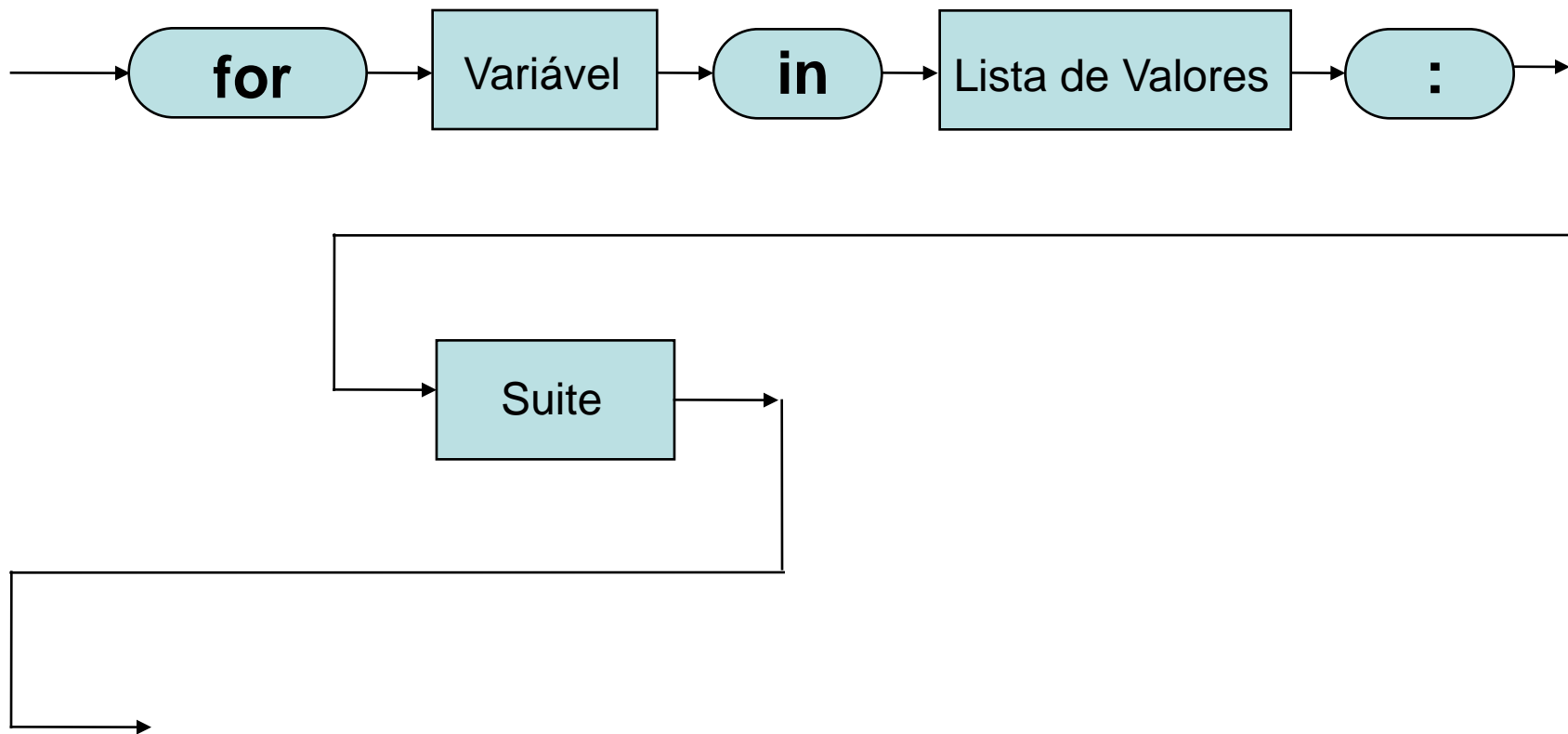
```
num = int(input("Digite valor inteiro e positivo: "))
fat = 1
for i in range(1,num+1):
    fat = fat * i
print("O fatorial de", num, "=", fat)
```

Execução:

```
Digite valor inteiro e positivo: 5 <enter>

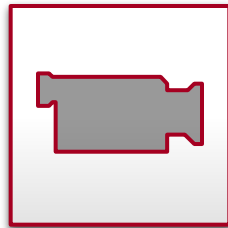
O fatorial de 5 = 120 <enter>
```

Diagrama Sintático de Repetição *for*



Exemplos de Aplicação dos Conteúdos Vistos

Clique no botão para assistir ao tutorial:



Faça os Exercícios Relacionados a essa Aula

Clique no botão para visualizar os enunciados:



Aula 3

Professores:

Dante Corbucci Filho
Leandro A. F. Fernandes

Conteúdo apresentado:

- Estruturas de Controle
 - Estrutura de Sequência
 - **suite**
 - Estruturas de Seleção
 - **if, if-else, if-elif, if-elif-else**
 - Estruturas de Repetição
 - **while e for**