

Programação II

Roteiro de Atividades- 01

Prof. Dr. Adilson Ferreira da Silva

Linguagem de programação Python

1. Operadores

Operadores de uma linguagem de programação correspondem aos símbolos que devem ser utilizados pelo programador para atribuir valor a uma variável, elaborar sentenças condicionais simples, compostas ou efetuar cálculos através de expressões aritméticas.

O operador de atribuição = é utilizado para armazenar um valor em uma variável, e caso não exista, O interpretador Python vai cria-la, de forma a poder armazenar o tipo de informação que o programador deseja.

Exemplo:

```
fase = "Teste"          # Tipo string
horas = 15               # Tipo inteiro
salario = 2100.50        # Tipo real
total = horas * 30       # Tipo inteiro (multiplicação entre valores inteiros)
desconto = salario - 10  # Tipo real
```

A tabela 1 apresenta os operadores aritméticos disponibilizados pela linguagem de programação Python.

Tabela 1 - Operadores Aritméticos

Operador	Descrição	Exemplo
+	Soma	$5 + 5 = 10$
-	Subtração	$7 - 2 = 5$
*	Multiplicação	$2 * 2 = 4$
/	Divisão	$5 / 2 = 2.5$
%	Resto da divisão	$10 \% 3 = 1$
**	Exponenciação	$4 ** 2 = 16$
//	Parte inteira do quociente	$5 ** 2 = 2$

A tabela 2 contém os operadores relacionais, que podem ser utilizados em comandos da linguagem Python para efetuar comparações.

Tabela 2 - Operadores relacionais (Comparações)

Operador	Descrição	Exemplo
<	Menor que	a < 10
<=	Menor ou igual	b <= 5
>	Maior que	c > 2
>=	Maior ou igual	d >= 8
==	Igual	e == 5
!=	Diferente	f != 12

Os operadores lógicos, apresentados na tabela 3, são utilizados para a construção de sentenças condicionais compostas. Exemplo:

Se **fizer Sol no Domingo** e **receber pagamento no Sábado** então **vou ao Parque**
(condição 1) (condição 2)

Tabela 3 - Operadores Lógicos

Operador	Descrição	Exemplo
and	E	(a <=10) and (c = 5)
or	OU	(a <=10) or (c = 5)

1.1. Exercícios para fixação

1. São operadores aritméticos da linguagem Python:

- > < >= <= == !=
- =
- + - * / % // ****
- and or
- Todas as anteriores

2. São operadores relacionais:

- > < >= <= == !=**
- =
- + - * / %
- and or
- Todas as anteriores

3. São operadores lógicos:

- > < >= <= == !=
- =
- + - * / %
- and or**
- Todas as anteriores

4. Os operadores relacionais servem para:
- Efetuar comparações somente entre constantes
 - Efetuar comparações somente entre variáveis
 - Efetuar comparações somente entre variáveis e constantes
 - Efetuar comparações entre variáveis, constantes ou expressões aritméticas**
 - Todas as anteriores
5. O operador lógico **and** é utilizado para:
- Verificar se duas ou mais condições são todas verdadeiras**
 - Verificar se duas ou mais condições são todas falsas
 - Verificar se uma das condições é verdadeira ou não.
 - Não é um operador relacional
 - Todas as anteriores
6. O operador lógico **or** é utilizado para:
- Verificar se duas ou mais condições são todas verdadeiras
 - Verificar se duas ou mais condições são todas falsas
 - Verificar se entre duas ou mais condições há uma verdadeira**
 - Todas as anteriores
7. Complete as seguintes tabelas verdade indicando verdadeiro ou falso:

Condição 1	Condição 2	Condição 1 and Condição 2
Verdadeiro	Verdadeiro	verdadeiro
Verdadeiro	Falso	falso
Falso	Verdadeiro	falso
Falso	Falso	falso

Condição 1	Condição 2	Condição 1 or Condição 2
Verdadeiro	Verdadeiro	verdadeiro
Verdadeiro	Falso	verdadeiro
Falso	Verdadeiro	verdadeiro
Falso	Falso	falso

8. Sabendo que A=8, B=12 e C=4, informe se as expressões abaixo são verdadeiras (V) ou falsas (F)
- a) $(A+C) > B$ (F)
 $(8 + 4) > 12$
 $12 > 12$

b) $B \geq (A + 2)$ (V)

$12 \geq (8 + 2)$
 $12 \geq 10$

c) $C == (B - A)$ (V)

$4 == (12 - 8)$
 $4 == 4$

d) $(B + A) \leq C$ (F)

$(12 + 8) \leq 4$

$20 \leq 4$

9. Sabendo que $A=5$, $B=4$ e $C=3$ e $D=6$, informe se as expressões abaixo são verdadeiras (V) ou falsas (F).

a) $(A > C) \text{ and } (C \leq D)$ (V)

$(5 > 3) \text{ and } (3 \leq 6)$
 $V \text{ and } V$

b) $(A+B) > 10 \text{ and } (A+B) == (C+D)$ (F)

$(5 + 4) > 10 \text{ and } (5 + 4) == (3 + 6)$
 $9 > 10 \text{ and } 9 == 9$
 $F \text{ and } V$

c) $(A > C) \text{ or } (C \leq D)$ (V)

d) $(A+B) > 10 \text{ or } (A+B) == (C+D)$ (V)

2. Comunicação com o usuário

Há situações em que o usuário interage diretamente com o programa de computador, fornecendo dados para serem processados e recebendo os resultados. Neste sentido, uma linguagem de programação deve proporcionar comandos que permitam ao programador tratar desta questão. Na linguagem Python temos os comandos **print** e **input**.

2.1. Comando print

O comando print serve para exibir informações no monitor. Segue exemplo de uso:

```
ano = 2023
disciplina = 'Programação'
print ('Estudo na Fatec São Caetano do Sul')
print ('Em', ano, 'Estou tendo aula de ', disciplina)
```

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: C:/@Fatec/PII20231S/aa.py =====
Estudo na Fatec São Caetano do Sul
Em 2023 Estou tendo aula de Programação
```

2.2. Comando input

O comando input serve para entrada de dados via teclado. Tomemos como exemplo, uma situação em que o usuário deve ser instruído pelo programa a digitar seu nome completo para que possa ser armazenado na memória. Segue exemplo 1:

```
nome = ''
print('Informe seu nome: ')
nome = input()
```

Exemplo 2:

```
nome = input('Informe seu nome: ')
print(nome)
```

2.3. Exercícios para fixação

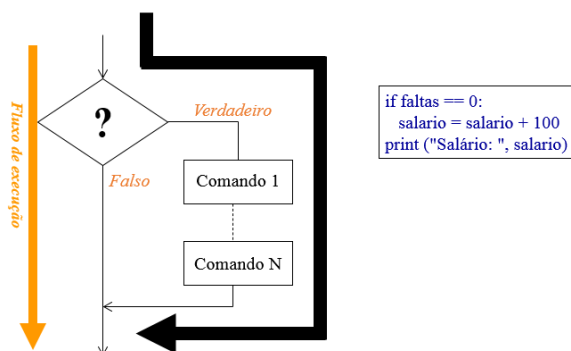
10. Faça um programa para ler uma medida em metros e mostrar o equivalente em centímetros.
11. Faça um programa que solicite ao usuário os dados necessários para calcular a área de um quadrado. Mostrar em seguida valor correspondente ao dobro da área calculada.
12. Em um jogo de Fazenda existe a opção de compras de frutas para o jogador manter em sua despensa. No empório virtual as frutas são vendidas pelos seguintes preços: Maçã: R\$ 2,30 por Kg, Melancia: R\$ 4,00 a unidade e Laranja R\$ 0,30 a unidade. Escreva um programa para ler a quantidade de maçãs, laranjas e melancias compradas, calcular e mostrar na tela o valor a pagar.

3. Estruturas de seleção

Uma estrutura de seleção corresponde a um comando que faz comparações entre valores para decidir sobre a execução ou não de um conjunto de comandos. As linguagens de programação fornecem estes comandos, pois sem eles não será possível

resolver problemas que envolvem tomada de decisão. A figura 1 ilustra a estrutura de seleção por um caminho alternativo. (IF ... THEN)

Figura 1 – Estrutura de seleção por um caminho alternativo



Codificação em Python

Sintaxe: **if condição:**

Comando 1

Comando N

Exemplo 01:

```
if num>10:
    print ("O número é maior que 10")
```

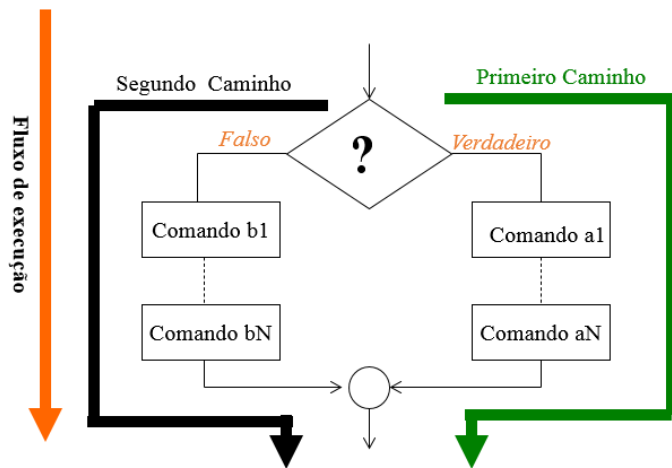
Exemplo 02:

```
if num==10:
    print("O número digitado é Dez!")
    print ("O programa será encerrado!")
```

Importante: A linguagem Python exige que os comandos a serem executados sejam iniciados na mesma coluna, o que é chamado de indentação.

A estrutura de seleção por dois caminhos alternativos (IF .. THEN .. ELSE ..) executa um conjunto de comandos se o resultado da condição for verdadeiro ou outro conjunto de comandos se o resultado for falso. A figura 2 ilustra a estrutura de seleção por dois caminhos alternativos.

Figura 2 – Estrutura de seleção por dois caminhos alternativos



Codificação em Python

Sintaxe: **if condição:**
 comando
 else:
 comando

Exemplo:

```

if n>10:
    print("Maior que dez")
else:
    print("Menor ou igual a dez");
  
```

3.1.Exercícios para fixação

13. Escreva um programa para ler as notas N1 e N2 obtidas por um aluno que cursou a disciplina Programação II, e em seguida calcular e mostrar média obtida. Caso a nota da N1 tenha sido maior ou igual a 8.0, mostrar também a mensagem “Foi bem na N1”.
14. Faça um programa para ler uma medida em metros. Mostrar o equivalente em centímetros somente caso a medida seja menor que 100.
15. Faça um programa que solicite ao usuário os dados necessários para calcular a área de um quadrado. Mostrar em seguida o valor correspondente ao dobro da área calculada caso todos os dados fornecidos sejam maiores que zero.
16. Ler as notas de avaliação de um jogador sobre a 1ª e 2ª. fases de um Game e em seguida escrever uma mensagem informando a média aritmética destas notas e uma mensagem informando “Experiência boa”, caso a média seja maior ou a 8.0 ou a mensagem “Experiência ruim” caso contrário.

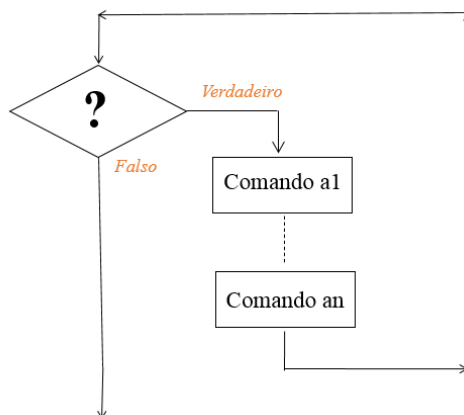
4. Estruturas de repetição

Há situações em que a solução de um problema requer que um conjunto de comandos seja executado mais de uma vez em seguida, o que obriga o programador a fazer uso de um laço de repetição, que permite executar um conjunto de comandos, sendo que o encerramento do processo de repetição é determinado pelo resultado de uma condição. Veremos a seguir comandos para repetição da linguagem Python.

4.1. Enquanto ... Faça .. → while

A estrutura de repetição **while** possibilita ao programador repetir a execução de um conjunto de comandos enquanto uma condição de controle retornar verdadeiro. A figura 1 ilustra esta estrutura.

Figura 1 – Estrutura de repetição while



A linguagem de programação Python oferece aos programadores o comando para repetição de mesmo nome: **while**

Exemplo:

```
i = 1
while i < 6:
    print(i)
    i = i + 1
```

Resultado:

1
2

3
4
5

O comando **break** interrompe o processo de repetição. Segue exemplo:

```
i = 1
while i < 6:
    print(i)
    i = i + 1
    if i > 3:
        break
```

Resultado:

1
2
3

Exemplo: Login e Senha (1)

```
login = input("Login: ")
senha = ""
while senha != "abc123":
    senha = input("Senha:")
```

Observações sobre o exemplo Login e Senha (1):

- A senha é fixa pois trata-se de um exemplo
- Não há limite de tentativas para acertar a senha

4.2. For – definição de parâmetros para repetição

Sintaxe:

```
for <variável> in range(<valor inicial>,<Valor limite>,<incremento>):
    <bloco de comandos>..
```

Exemplo:

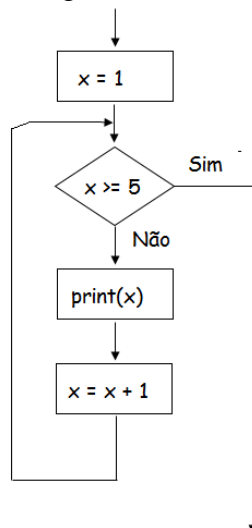
```
for x in range(1,5,1):
    print(x)
```

Resultado:

```
>>>
=====
1
2
3
4
>>>
```

X → Variável de controle. Armazena o número que será gerado em cada iteração
range(1, 5, 1) → função para controlar as repetições. O primeiro parâmetro informado (**1**) corresponde ao valor que será armazenado na variável de controle **X** antes de iniciar o processo de repetição. O segundo parâmetro (**5**), corresponde ao valor limite para encerramento da repetição, caso a variável de controle **X** o ultrapasse. O terceiro parâmetro corresponde ao incremento a ser aplicado na variável de controle **X** após o término de uma repetição.

Fluxograma



4.3. For – acesso a elementos de uma coleção de dados

for <variável> in <Coleção>:
 <bloco de comandos>

Exemplo:

```
pomar = ("Laranja", "Banana", "Uva")
for fruta in pomar:
    print(fruta)
```

Resultado:

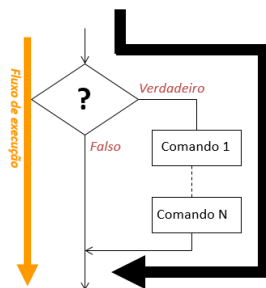
Laranja

Banana

Uva

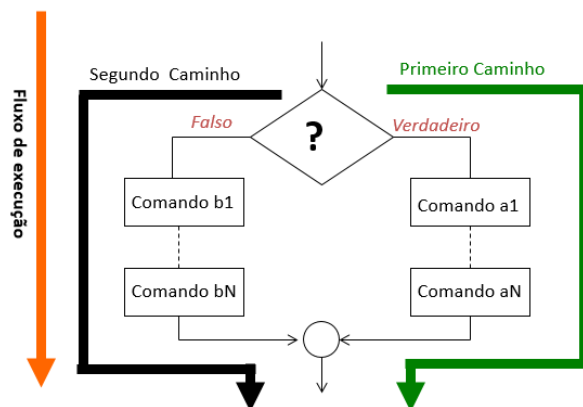
4.4. Exercícios para fixação

17. Dada a figura:



- Qual é o nome desta estrutura?
- Explique como é o funcionamento desta estrutura.

18. Dada a figura:



- Qual é o nome desta estrutura?
- Explique como é o funcionamento desta estrutura.