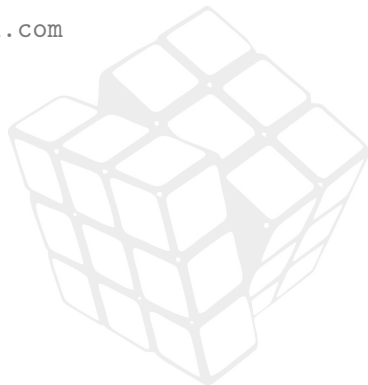


Algoritmos e Programação

Operadores de atribuição e Estruturas de Repetição

Gustavo Sávio
gsoprofessor@gmail.com

2017.2



Sabemos que para definirmos valores nas variáveis podemos utilizar o operador de atribuição

```
1 # coding: utf-8
2
3 texto = input('Digite qualquer texto')
```

+=	-=	=	%=
----	----	---	----

```
1  valor1 = 10
2  valor2 = 10
3
4  # Adição
5  valor1 = valor1 + valor2
6  valor1 += valor2
7
8
9  # Subtração
10 valor1 = valor1 - valor2
11 valor1 -= valor2
```

```
1  # Multiplicação
2  valor1 = valor1 * valor2
3  valor1 *= valor2
4
5  # Divisão
6  valor1 = valor1 / valor2
7  valor1 /= valor2
8
9  # Resto de Divisão - Módulo
10 valor1 = valor1 % valor2
11 valor1 %= valor2
```

A sintaxe da atribuição condicional é utilizada para expressões condicionais simples

```
1  # Como funciona?
2
3  <variável> = <valor1> if True else <valor2>
4
5  media = 7
6
7  situacao = 'Aprovado' if media >=7 else 'Reprovado'
```

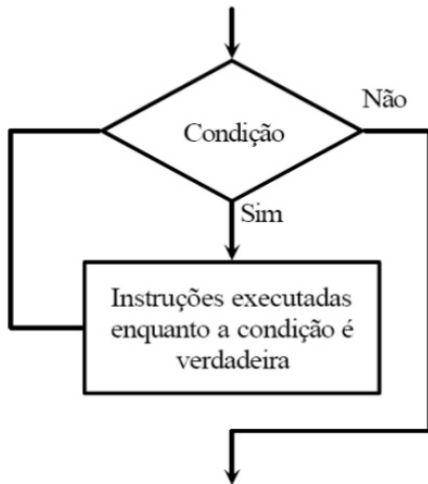
```
1 # atribuição condicional
2
3 valor_compra = 100
4 desconto = valor_compra - 20 if valor_compra > 100 else
   ↪ valor_compra - 10
5
6 print(desconto)
```

if vs Atribuição Condicional

```
1  # if
2
3  valor_compra = 100
4
5  if valor_compra > 100:
6      desconto = valor_compra - 20
7  else:
8      desconto = valor_compra - 10
9
10 print(desconto)
```

- ▶ Laço
- ▶ Loop
- ▶ Iteração
- ▶ Ciclo

É o processo de repetição de um bloco de código enquanto uma condição for verdadeira



Estruturas de Repetição - While

```
1  # Enquanto for verdadeiro repita
2
3  while <condicao>:
4      # Bloco de código
```

Escreva um algoritmo que calcule a média do usuário enquanto não for informado o valor -1.

Estruturas de Repetição - While

```
1 opcao = ''
2
3 while opcao != '-1':
4     nota1 = float(input('Informe a primeira nota: '))
5     nota2 = float(input('Informe a segunda nota: '))
6
7     print('A média do aluno é: ', (nota1 + nota2) / 2)
8
9     opcao = input('Digite qualquer tecla para continuar e -1 para
↵ finalizar...')
```

Estruturas de Repetição - While - else

```
1  # Python é tão lógico ...tsc
2  valor = 10
3
4  while valor <= 10:
5      print(valor)
6  else:
7      print('Finalizou')
8
9  # Eu preciso desse else?
10 while valor <= 10:
11     print(valor)
12
13 print('Finalizou')
```

Estruturas de Repetição - For in

```
1 for <valor> in <lista>:  
2     # bloco de código
```

Vocês lembram que uma **String** é uma **lista de caracteres**?

Estruturas de Repetição - For in

P	A	L	A	V	R	A
0	1	2	3	4	5	6

```
1 for p in "palavra":  
2     print(p)
```

- Quando devo usar o while ou o for?

```
1 i = 0
2 palavra = "palavra"
3
4 while i < len(palavra):
5     print(palavra[i])
6     i+=1
```

Estruturas de Repetição While - For in

- ▶ for ... in

Sempre que você "**souber**" quantas vezes o bloco de código for repetir

- ▶ while

Sempre que você **não** "**souber**" quantas vezes o bloco de código for repetir

```
1 # Gera uma lista em um intervalo de números definido
2 range([inicio], [fim-1], [incremento])
3
4 # Bastante utilizada com o laço de repetição for
5 for n in range(20):
6     print(n)
7
8 for n in range(10, 20):
9     print(n)
10
11 for n in range(0, 21, 2): # Tabuada de 2?
12     print(n)
```

- ▶ Utilizadas em um laço de repetição
 - ▶ **break** - finalizar todo o laço de repetição
 - ▶ **continue** - finalizar um loop / iteração / ciclo

```
1 # Uso do continue
2 for numero in range(0, 11):
3     if numero % 2 == 0:
4         continue
5
6     print(numero)
7
8 print('Fim do laço')
```

break e continue

```
1  # Uso do continue e break
2  for numero in range(0, 11):
3      if numero % 2 == 0:
4          continue
5      elif numero == 7:
6          break
7
8      print(numero)
9  print('Fim do laço')
```

```
1 import os
2
3 os.system('cls' if os.name=='nt' else 'clear')
```

-  Allen B. Downey; Think Python How to Think Like a Computer Scientist.