

Bem-vindo ao roteiro de Introdução ao Python - Parte 2

*Confeccionado por Marcílio F. Oliveira Neto (marciliofoneto@gmail.com)

Github: github.com/marciliooliveira

Neste roteiro você irá aprender sobre a sintaxe do Python. Este roteiro tem o intuito de seguir o conceito *do-it-yourself*.

Neste roteiro você encontrará os seguintes tópicos sobre Python:

- Operadores relacionais
- if/else
- if/elif
- Operadores Lógicos

Instruções para utilizar este roteiro:

- Importar o arquivo deste roteiro no Jupyter;
- Clicar no menu superior **Cell**;
- Clicar na opção **Run All**;
- Para executar célula individualmente utilizar:
 - Selecionar a célula
 - Clicar no botão **Run** - menu superior - ou;
 - Clicar na opção **Cell** e **Run Cells** - ou;
 - Utilizar o atalho **Ctrl + Enter**.

Este roteiro foi inspirado em metodologias ativas de ensino.

Este conteúdo foi confeccionado pelo professor: Marcílio Francisco de Oliveira Neto (marciliofoneto@gmail.com)

Este roteiro é aberto para todos, só não esqueça de citar a fonte!

In [1]:

```
1 == 1 #Testa se 1 é igual a 1
```

Out[1]:

True

In [2]:

```
3 > 5 #Testa se 3 é maior que 5
```

Out[2]:

False

In [3]:

```
numero = 4 #Atribui o valor 4 para a variável numero  
numero >= 4 #Testa se o valor da variável numero é maior igual a 4
```

Out[3]:

True

In [4]:

```
fruta = "maça" #Atribui o texto maçã a variável fruta  
fruta != "maça" #Testa se o valor da variável fruta é diferente ao texto maçã
```

Out[4]:

False

In [5]:

```
palavra_um = "araras" #Atribui o texto araras a variável palavra_um  
palavra_dois = "sarara" #Atribui o texto sarara a variável palavra_dois  
palavra_um == palavra_dois #Realiza a comparação (igualdade) entre as variáveis
```

Out[5]:

False

In [6]:

```
numero_teste = 5  
numero_teste > 10+10
```

Out[6]:

False

Do-it-yourself

Tente executar suas comparações nos quadros abaixo. Vou sugerir algumas para você! :D

Lembre-se, para executar as células utilize o botão **Run** no menu ou o atalho **Ctrl + Enter**.

Verifique se sua idade é maior que 18:

Atribua o valor 600 a uma variável qualquer (pode dar o nome que quiser). Faça um teste se essa variável é maior que a diferença 49209 - 48630:

Crie uma variável chamada *palavra*, atribua o valor "arara" à ela. Faça um teste para checar se essa variável é um palíndromo, ou seja, teste se "arara" escrito de trás para frente tem a mesma grafia:

O IF

Você já conheceu os *babysteps* do Python, mas programar é muito além de simples textos na tela e entrada de dados. Programar é garantir ao computador a realização de tarefas da melhor forma, podendo escolher qual a melhor solução, o que deve ser feito...O massa disso tudo é que **você** define o que deve ser feito, como é feito, qual a melhor escolha! Você deve saber a lógica por baixo do capô!

Assim, a fim de dar mais liberdade ao programa e o que ele deve fazer, temos os comandos que alteram o fluxo de execução de um programa.

Um desses comandos é o **if**. Esse comando está acostumado a aparecer junto a vários outros, então, atente-se que, daqui em diante, seus *scripts* Python ficarão maiores.

Vale lembrar que utilizaremos os conceitos abordados na parte 1 desses roteiros.

Sintaxe do IF

Veja no quadro abaixo como é simples escrever um **if**:

In [7]:

```
if 5 > 3:
    #do something
    pass
```

Neste momento você pode desconsiderar o comando **pass** acima. Foque apenas na linha do comando **if**.

O comentário *do something* sugere que você poderá utilizar os espaços abaixo da linha do comando **if** para desenvolver scripts necessários, **se e somente se, a condição 5 > 3 for VERDADEIRA**. Entenda abaixo! ;)

In [8]:

```
if 5 > 3:
    print("O número 5 é maior que o 3!")
```

O número 5 é maior que o 3!

O comando **print** só foi executado porque a condição **5 > 3** é verdadeira, caso contrário, nada seria exibido.

Você deve ter notado que a linha do comando **print** está mais recuada se comparada ao **if**, certo? Se não, essa é a hora!

Indentação

Isso significa que o comando **print** está na hierarquia do comando **if**, ou seja, é algo a ser executado apenas se, e somente se, a condição do comando **if** for satisfeita. Normalmente, indentação é feita pelo uso da tecla **TAB** do seu teclado.

Indentação é OBRIGATÓRIO em Python.

Do-it-yourself

Teste você mesmo o caso contrário, ou seja, modifique os comandos mostrados acima e teste se 5 é menor igual a 3. Execute este teste, note que não haverá saída de texto.

Voltando ao exemplo do palíndromo, veja um caso interessante para testar essa se palavras escritas de forma inversa são iguais às grafias normais.

In [9]:

```
original = "arara"
invertida = "arara"
if original == invertida: #Testa SE as variáveis são idênticas
    print("A palavra arara é um palíndromo!") #Em caso afirmativo, este comando
    print é executado
```

A palavra arara é um palíndromo!

Do-it-yourself

Modifique o exemplo do quadro acima - palíndromo - para que seja possível o usuário digitar duas palavras. Faça o teste e, caso as grafias forem iguais - escritas original e na ordem inversa - exiba que as palavras são palíndromo. Atenção no uso de **indentação**!

IF/ELSE

Para toda condição, há a possibilidade desta ser **VERDADEIRA** ou **FALSA**, lembra?! Pois bem, tratar uma condição verdade você já sabe - utilizando o comando **if**, mas e se o teste da condição resultar em falso? Eis que surge o comando **else** (senão)!

Para todo comando **ELSE** deve existir um comando **IF**. Lembre-se, o else é o caso contrário do que foi testado no if, portanto, **não há condições a serem testadas no else**!

In [10]:

```
original = "arara"
invertida = "araras"
if original == invertida: #Testa SE as variáveis são idênticas
    print("É palíndromo!") #Se VERDADEIRO, este comando print é executado
else:
    print("NÃO é palíndromo!") #Se o teste do IF resultar em FALSO, este comando print é executado
```

NÃO é palíndromo!

In [11]:

```
if 5 < 3:
    print("O número 5 é maior que 3!")
else: #Se o teste do IF resultar em FALSO
    print("O número 5 NÃO É MENOR que 3!") #este comando print é executado
```

O número 5 NÃO É MENOR que 3!

Do-it-yourself

Faça um script que leia uma idade do usuário e verifique se ela é maior de idade. Se sim, mostre uma mensagem de sua criatividade, caso contrário, mostre uma mensagem de "erro". Atenção no uso de **indentação**!

Faça um script que contenha uma variável senha - com valor já atribuído de sua vontade (será sua senha). Após isso, crie um simulador de login. Peça ao usuário para digitar a senha, se a senha digitada corresponder com a salva no sistema, exiba uma mensagem amigável. Caso contrário, exiba uma mensagem de senha inválida.

IF/ELIF

Este jogo de comandos - **IF/ELIF** - é uma variação dos conhecidos **IF/ELSE**. Comumente, em aplicações complexas ou extensas, precisamos de várias condições a serem testadas, o que gera aninhamentos de *if/else*. E como Python veio para tornar a sua vida produtiva, criou-se o **ELIF**, que nada mais é do que a junção de *else* + *if*.

Uma nota importante é que o **ELIF** só pode existir se houver um **IF** ou um **ELIF** precedendo-os.

A ideia base é a seguinte: caso condição(ões) não seja(m) satisfeita(s) para um dado *if*, esta(s) possa(m) ser testada(s) - novamente - por **elif** subsequentes.

Veja como é fácil a sintaxe:

In [2]:

```
if 5 > 10: #Testa se 5 é maior que 10 - o que é falso!
    print("Frase 1")
elif 5 == 4: #Testa se 5 é igual a quatro - o que é falso!
    print("Frase 2")
elif 5 <= 5: #Testa se 5 é igual ou menor que 5 - o que é VERDADEIRO!
    print("Frase 3")
else: #Caso todas as condições anteriores falhassem
    print("Frase 4")
```

Frase 3

Do-it-yourself

Faça um script que leia uma idade do usuário e verifique qual categoria ele pertence, seguindo a tabela abaixo.

| Mirim | Infantil | Juvenil | Adulto | Idoso |
|------------|-------------|--------------|--------------|------------------|
| até 7 anos | 8 a 13 anos | 14 a 17 anos | 18 a 60 anos | acima de 60 anos |

Operadores lógicos

Os operadores lógicos têm o intuito de relacionar condições de modo que tais relações resultam em UM de dois estados possíveis: **VERDADEIRO** ou **FALSO**.

Em Python, os operadores lógicos são palavras reservadas em inglês:

- Operador E - **and**
- Operador OU - **or**
- Operador NÃO - **not**

A tabela abaixo mostra a relação das condições e os resultados de saída para os operadores:

| Condição A | Condição B | A and B | A or B | not A |
|------------|------------|------------|------------|------------|
| Verdadeiro | Verdadeiro | Verdadeiro | Verdadeiro | Falso |
| Verdadeiro | Falso | Falso | Verdadeiro | Falso |
| Falso | Verdadeiro | Falso | Verdadeiro | Verdadeiro |
| Falso | Falso | Falso | Falso | Verdadeiro |

Veja como fica a sintaxe dos operadores lógicos, que são comumente utilizados com os comandos condicionais:

In [3]:

```
5 == 5 and 10 > 9
```

Out[3]:

True

In [4]:

```
5 == 5 and 10 > 15
```

Out[4]:

False

In [5]:

```
5 == 5 or 10 > 9
```

Out[5]:

True

In [6]:

```
5 != 5 or 10 < 15
```

Out[6]:

True

Você poderá utilizar os operadores lógicos com os comandos **IF/ELSE** e **IF/ELIF**. Veja um exemplo abaixo antes de você mesmo programar.

In [9]:

```
#Programa do Tempo
condicao_tempo = input("Digite se o tempo está chuvoso ou ensolarado: ")
horario_tempo = int(input("Digite a hora atual - sem os minutos: "))
if horario_tempo > 5 and horario_tempo < 12 and condicao_tempo == "chuvoso":
    print("É manhã e está chovendo")
elif horario_tempo > 5 and horario_tempo < 12 and condicao_tempo == "ensolarado":
    print("É manhã e está ensolarado")
elif horario_tempo > 12 and horario_tempo < 18 and condicao_tempo == "chuvoso":
    print("É tarde e está chuvoso")
elif horario_tempo > 12 and horario_tempo < 18 and condicao_tempo == "ensolarado":
    print("É tarde e está ensolarado")
```

Digite se o tempo está chuvoso ou ensolarado: chuvoso

Digite a hora atual - sem os minutos: 11

É manhã e está chovendo

Do-it-yourself

Faça um script que leia uma idade do usuário e verifique qual categoria ele pertence, seguindo a tabela abaixo. Desta vez, utilize operadores lógicos para verificar os intervalos.

| Mirim | Infantil | Juvenil | Adulto | Idoso |
|------------|-------------|--------------|--------------|------------------|
| até 7 anos | 8 a 13 anos | 14 a 17 anos | 18 a 60 anos | acima de 60 anos |

Parabéns! Você finalizou o roteiro 2!

Agora é hora de você praticar com exercícios.

Este roteiro foi inspirado em metodologias ativas de ensino.

Este conteúdo foi confeccionado pelo professor:

Marcílio Francisco de Oliveira Neto (marciliofoneto@gmail.com)

