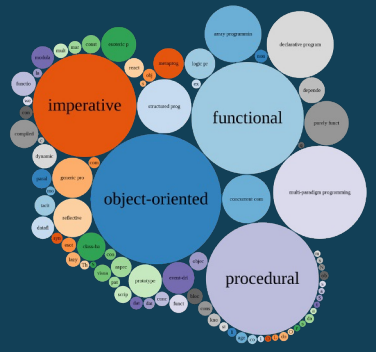
Paradigmas de Linguagem de Programação em Python



Estruturas Condicionais

condicionais.ipynb

Hoje
Execução de código!

A **ordem de execução** de um programa é denominado **fluxo de controle**;

Exceto quando especificado de outra forma, a ordem de execução é linear → as **instruções** são **executadas uma após a outra**, em **sequência**, de **cima** para **baixo**;

Hoje
Tomada de Decisão

Nesta aula, vamos aprender a controlar o fluxo de controle do programa.

Alguns comandos em programação nos permitem decidir se a execução de uma instrução deve ou não ser feita;

Útil para que certas instruções só sejam executadas sob determinadas condições;

Esta tomada de decisão é baseada em expressões booleanas

Estruturas Condicionais
if ... else

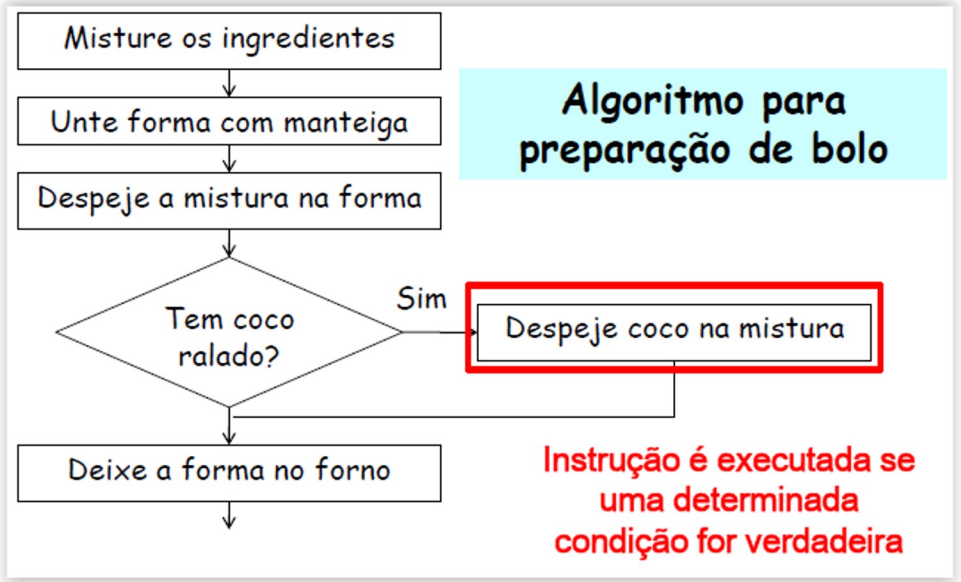
Tomada de decisão é baseada em expressões booleanas:

- Se determinada expressão for verdadeira/falsa, serão executados alguns comandos.
- Se outras expressões forem verdadeiras/falsas, serão executados outros comandos.

Como?

- Estruturas condicionais em Python, capitaneadas por blocos de if e else.

Estruturas Condicionais
if ... else



Estruturas Condicionais
if

O if nada mais é que o nosso "se"

Podemos entendê-lo da seguinte forma:

Se a condição for verdadeira, faça alguma coisa

Estruturas Condicionais
if

```
a = 33
b = 200
if b > a:
    print("b é maior que a")
```

Estruturas Condicionais
if

```
a = int(input("Primeiro valor: "))
b = int(input("Segundo valor: "))

if a > b:
    print("Primeiro eh maior." )

if b > a:
    print("Segundo eh maior." )
```

Estruturas Condicionais
if

Note que o **bloco** if é iniciado após os dois pontos (:).

O primeiro comando do bloco está **indentado**, i.e., avançado alguns espaços.

Esses espaços podem ser um simples *tab*.

O bloco continua enquanto a linha não retorna à indentação anterior.

Estruturas Condicionais
if

```
a = int(input("Primeiro valor: "))
b = int(input("Segundo valor: "))

if a > b:
     print("Primeiro eh maior." )
     print("a = %d, b = %d" % (a, b))

if b > a:
          print("Segundo eh maior." )
     print("b = %d, a = %d" % (b, a))

print("C'est fini.")
```

Estruturas Condicionais
if

E o que acontece, caso `a == b` ?

Estruturas Condicionais
if

E o que acontece, caso **`a == b`** ?

No exemplo anterior, só o ultimo **`print()`** seria executado

Estruturas Condicionais
else

Quando temos um cenário onde uma condição é simplesmente a negação da primeira, podemos usar uma forma de para simplificar os programas

Essa forma é a comando else, que significa "caso contrário"

```
pc = "linux"
if pc == "linux":
    print("Eh Linux!" )
else:
    print("Nao eh Linux ...")
```

Estruturas Condicionais
Estruturas Aninhadas

E se tivermos múltiplas condições (if's)?

```
categoria = int(input("Digite a categoria do produto"))
if categoria == 1:
      preco = 10
else:
      if categoria == 2:
                  preco = 20
      else:
                  if categoria == 3:
                              preco = 30
                   else:
                              if categoria == 4:
                                          preco = 40
print("Preco = %d" % preco)
```

Estruturas Condicionais

No exemplo anterior, temos o seguinte:

- Alinhamento (**indentação**) **se tornou** um problema
- Legibilidade prejudicada

Estruturas Condicionais
elif

Python apresenta uma solução para o uso de múltiplos if's

Utilizar outro if dentro um else, através do comando elif

O elif substitui um par else if, mas sem criar outro nível de estrutura, evitando indentações desnecessárias

Estruturas Condicionais
elif

```
categoria = int(input("Digite a categoria do produto"))
if categoria == 1:
     preco = 10
elif categoria == 2:
     preco = 20
elif categoria == 3:
          preco = 30
elif categoria == 4:
          preco = 40
else:
          preco = 50

print("Preco = %d" % preco)
```

Estruturas Condicionais
Complexidade das Expressões

A condição avaliada pelo if pode conter expressões lógicas e/ou relacionais de quaisquer complexidades.

Basta que seja uma expressão que retorne **`True`** ou **`False`**.

```
calor = True
pc = "linux"

if (pc == "linux") and calor:
    print("To no Linux e ta um calor danado")
elif calor:
    print("Nao to no linux e o clima ta quente!")
```

Estruturas Condicionais
Complexidade das Expressões

```
calor = True
pc = "linux"
experiencia = 3

if (pc == "linux") and experiencia < 5 and calor:
    print("Não da pra usar o linux no calor e com menos de 5 anos de experiencia")
elif (pc == "linux") and experiencia >= 5 and calor:
    print("Ta calor, mas eu entendo de linux!")
```

Estruturas Condicionais
Estruturas Aninhadas

As vezes precisamos aninhar comandos por necessidade!

```
p_max = 20
p1, p2, p3 = 30, 10, 40

if p1 < p_max:
      if p1 > 10:
                  print("Aljava comprada por mais de 10 moedas!")
      else:
                  print("Aljava comprada por um valor menor ou igual a 10 moedas!")
elif p2 < p_max:
      print("Aljava comprada por %f moedas!" % p2)
elif p3 < p_max:
      print("Aljava comprada por %f moedas!" % p3)
else:
      print("Aljavas muito caras!")
```

Estruturas Condicionais
if curto (short hand)

Se você tiver **apenas 1 comando** para o if, é possível colocá-lo na mesma linha.

```
if a > b: print("a eh maior que b")
```

Estruturas Condicionais
If ... else curto (short hand)

Se você tiver apenas 1 comando tanto para o if, quanto para o else...

```
a = 2
b = 330
print("A") if a > b else print("B")
```

Essa técnica também é conhecida como operador ternário

Estruturas Condicionais
If ... else curto (short hand)

```
a, b = 2, 330

if a > b:
        print("A")
else:
        print("B")
```

```
a, b = 2, 330
print("A")if a > b else print("B")
```

Obrigado!

Alguma dúvida?