

Instrução condicional

NUMWORKS

Dentro de uma função, pode ser interessante executar instruções sob certas condições.

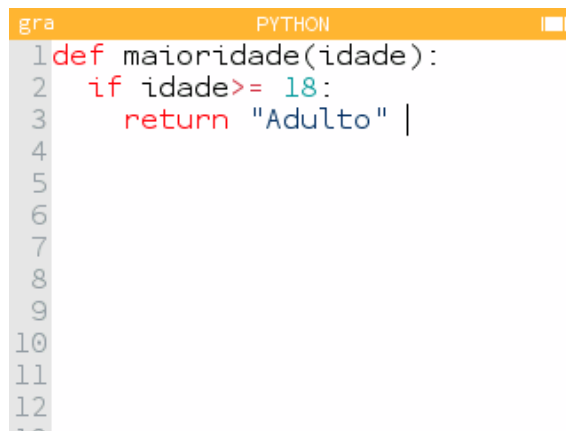
1 If ... elif ... else

Por exemplo, se quiser escrever uma função `maioridade(idade)` que devolve "Adulto"/"Menor" de acordo com a idade introduzida, é necessário separar os casos em que a idade é superior a 18 dos casos em que a idade é inferior.

Em Python, um bloco com uma instrução condicional é escrito da seguinte forma:

```
if condition:
    instruction
```

A instrução só é executada se a condição `if` se verificar. Os dois pontos são utilizados para se indicar a Python que está a iniciar um bloco de instruções.

A screenshot of a Python code editor window titled 'PYTHON'. The code defines a function 'maioridade(idade)'. The function has a docstring 'gra' on the first line. The function body starts with an 'if' statement: 'if idade >= 18:'. Inside this 'if' block, there is a 'return' statement: 'return "Adulto" |'. The code is displayed with line numbers from 1 to 12 on the left side of the editor.

```
1 def maioridade(idade):
2     """gra"""
3     if idade >= 18:
4         return "Adulto" |
5
6
7
8
9
10
11
12
```

Por exemplo, podemos escrever:

```
gra PYTHON
>>> from exemplo import *
>>> maioridade(25)
'Adulto'
>>> |
```

A função em cima devolve então o texto “Adulto” se a idade indicada entre parênteses for igual ou superior a 18. Vejamos ver a aplicação ao lado:

```
gra PYTHON
1 def maioridade(idade):
2     if idade >= 18:
3         return "Adulto"
4     else:
5         return "Menor"
6
7
8
9
10
11
12
13
```

Se quisermos adicionar a possibilidade de responder “Menor”, é necessário adicionar um novo caso. Desejamos então que a função responda “Adulto” se a idade for igual ou superior a 18 e “Menor” caso contrário. Para darmos esta instrução CASO CONTRÁRIO a Python, utilizamos **else** seguido de dois pontos com a mesma indentação (avanço do texto) que a primeira instrução **if**.

Em suma:

- Se houver apenas um caso a distinguir, usamos:

```
if condition:
    instruction
```

- Se houver dois casos a distinguir, usamos:

```
if condition:
    instruction_1
else:
    instruction_2
```

- Se houver mais do que dois casos a distinguir, usamos **elif** para adicionar condições:

```

if condition_1:
    instruction_1
elif condition_2:
    instruction_2
elif condition_3:
    instruction_3
...

```

2 As condições

Aqui está a sintaxe das condições em Python:

Condição	Sintaxe Python
Se x é igual a y	<code>if x == y:</code>
Se x é diferente de y	<code>if x != y:</code>
Se x é estritamente superior a y	<code>if x > y:</code>
Se x é estritamente inferior a y	<code>if x < y:</code>
Se x é superior ou igual a y	<code>if x >= y:</code>
Se x é inferior ou igual a y	<code>if x <= y:</code>

Se quisermos verificar duas condições, usamos **and** entre as duas condições. Por exemplo:

```

if x == 1 and y > 0:

```

Se quisermos verificar uma de duas condições, utilizamos **or** entre as duas condições. Por exemplo:

```

if x >= 1 or x == 0:

```

Selecione a tecla da Caixa de Ferramentas da sua calculadora para fazer surgir um menu com atalhos. Em **Laços e testes** encontra blocos de instruções pré-preenchidos para não ter de introduzir todo o texto no teclado.

3 Exercício

Escreva uma função **vabsoluto(x)** que tome um argumento real e devolva o seu valor absoluto.

4 Outro exercício

Escreva uma função **mediana** que tome uma série de números de qualquer tamanho como argumento e devolve a mediana da série.