

# Programação I

Licenciatura em Engenharia Informática

2015–2016

O operador “modulus”

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função `range()`

Vitor Beires Nogueira

Escola de Ciências e Tecnologia  
Universidade de Évora

### Definição

O operador “modulus” ou resto da divisão como o nome indica calcula o resto da divisão entre dois inteiros. Em Python é denotado por %.

### Exemplo

```
>>> quociente = 7 / 3
>>> print(quociente)
2.3333333333333335
>>> resto = 7 % 3
>>> print(resto)
1
```

Como é que conseguimos

- obter o algarismo menos significativo de um número inteiro?  
E os dois menos significativos?
- verificar se um número é divisível por outro?

O operador “modulus”

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função range()

## Definição

Uma expressão booleana é uma expressão que ou é verdadeira (True) ou é falsa (False)

## Exemplo

```
>>> 1 == 1
True
>>> 1 == 2
False
>>> type(True)
<class 'bool'>
>>> type(1 == 2)
<class 'bool'>
```

- $x == y$  é verdade se  $x$  é igual a  $y$ . Não confundir com atribuição ( $x = y$ )!!!
- $x != y$  é verdade se  $x$  é diferente de  $y$
- $x < y$  é verdade se  $x$  é menor que  $y$
- $x > y$  é verdade se  $x$  é maior que  $y$
- $x <= y$  é verdade se  $x$  é menor ou igual a  $y$
- $x >= y$  é verdade se  $x$  é maior ou igual a  $y$

## Definição

Existem três operadores lógicos: `and` (e), `or` (ou) e `not` (negação). A semântica destes operadores é semelhante ao seu significado em Inglês (Português).

## Exemplo

```
>>> not(False)
True
>>> n = 123
>>> n % 2 == 0 or n % 3 == 0
True
>>> n % 2 == 0 and n % 3 == 0
False
```

- Como poderíamos obter o comportamento do “ou exclusivo”?
- Tendo os operadores lógicos qual será o conjunto mínimo de operadores relacionais?

O operador “modulus”

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função `range()`

## Definição

As instruções de execução condicional permitem-nos verificar determinadas condições e alterar o comportamento do programa.

## Exemplo (Instrução if)

```
if x > 0:  
    print('x é positivo')
```

O número de instruções no “corpo” do if tem de ser maior que 1.

## Exemplo (Instrução pass)

```
if x < 0:  
    pass
```

A instrução pass até pode ser útil ...

O operador “modulus”

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função range()

## if-else

Uma outra forma para o if é aquela em que se utiliza o else para especificar o conjunto de instruções a executar caso a condição do if seja falsa.

## Estrutura do if-else

```
if <condição>:  
    <instruções para o caso da condição ser verdadeira>  
else:  
    <instruções para o caso da condição ser falsa>
```

Escreva uma função que recebe um número inteiro e escreve se é esse número é par ou ímpar.

## Exemplo

```
def par_impar(n):  
    if n % 2 == 0:  
        print(n, 'é par')  
    else:  
        print(n, 'é ímpar')
```



### Exemplo

```
def comparacao(x, y):  
    if x < y:  
        print( x, ' é menor que ', y)  
    elif x > y:  
        print( x, ' é maior que ', y)  
    else:  
        print(x, ' é igual a ', y)
```

- Apenas um dos ramos é que é executado.
- Podemos ter inúmeros elif (abreviatura de else if)
- Mesmo que existam diversas condições verdadeiras, somente o “ramo” referente à 1ª condição verdadeira é que é executada.
- Podemos não ter nenhum else

O operador “modulus”

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função range()

### Exemplo

```
def comparacao(x, y):  
    if x < y:  
        print(x, ' é menor que ', y)  
    else:  
        if x > y:  
            print(x, ' é maior que ', y)  
        else:  
            print(x, ' é igual a ', y)
```

- Apesar da indentação facilita a compreensão, os condicionais encaixados podem tornar-se muito facilmente difíceis de ler.
- Por vezes são utilizados indevidamente ...

### Exemplo

```
if 0 < x:  
    if x < 10:  
        print(x, 'é um inteiro positivo com um algarismo')
```

Qual é a forma “correcta”?

A função range() permite-nos iterar sobre uma sequência de números. Vejamos alguns exemplos:

### Exemplo

```
>>> list(range(10))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

### Exemplo

```
>>> list(range(5, 10))  
[5, 6, 7, 8, 9]  
>>> list(range(0, 10, 3))  
[0, 3, 6, 9]  
>>> list(range(-10, -100, -30))  
[-10, -40, -70]
```

O operador "modulus"

Expressões Booleanas

Operadores Lógicos

Execução Condicional

Execução alternativa

Condicionais  
encadeados

Condicionais  
encaixados

A função range()