

Fundamentos da Programação

@ LEIC/LEGM

Aula 5

Elementos básicos de programação

Predicados e condições. Comunicação com o exterior.

ALBERTO ABAD, IST, 2022-23

Elementos básicos de programação - Predicados e condições

- Um *predicado* é uma operação cujo valor é lógico: *True* or *False*
- Uma *condição* é uma expressão cujo valor é lógico
- As condições podem ser combinadas com os operadores lógicos, ex:
and, or
- Operadores relacionais em Python:

Operação	Tipo dos argumentos	Valor
$e_1 == e_2$	Números	Tem o valor <i>True</i> se e só se os valores das expressões e_1 e e_2 são iguais.
$e_1 != e_2$	Números	Tem o valor <i>True</i> se e só se os valores das expressões e_1 e e_2 são diferentes.
$e_1 > e_2$	Números	Tem o valor <i>True</i> se e só se o valor da expressão e_1 é maior do que o valor da expressão e_2 .
$e_1 >= e_2$	Números	Tem o valor <i>True</i> se e só se o valor da expressão e_1 é maior ou igual ao valor da expressão e_2 .
$e_1 < e_2$	Números	Tem o valor <i>True</i> se e só se o valor da expressão e_1 é menor do que o valor da expressão e_2 .
$e_1 <= e_2$	Números	Tem o valor <i>True</i> se e só se o valor da expressão e_1 é menor ou igual ao valor da expressão e_2 .

```
In [135]: x == 4
```

Out[135]:

True

Elementos básicos de programação - Predicados e condições

Exemplos

- `nota = 17` (é isto um predicado?)
- `nota > 10`
- `3 < nota % 2`
- `3 < nota // 2`
- `nota < 9*2 and nota > 10`
- `nota < 9*2 < 25` (*syntactic sugar*)
- `not 10` (qq expressão em Python pode ser tomado por condição)

```
In [137]: a += 1
```

Elementos básicos de programação - Leitura e escrita

Leitura de dados (do teclado)

BNF

<leitura de dados> ::= input() | input(<informação>)
<informação> ::= <cadeia de caracteres>

- A função `input` retorna a *string* introduzida
 - A *string* pode conter caracteres de escape, e.g., `\n`, `\r`, `\t`, `\v`, etc.
- Exemplos:

```
input()
input('Escreva alguma coisa\n\t ->')
```

```
In [143]: n = input('Escreva alguma coisa\n\t ->')
          type(int(n))
```

```
Escreva alguma coisa
->34.5
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-143-7d268b462116> in <module>
      1 n = input('Escreva alguma coisa\n\t ->')
----> 2 type(int(n))

ValueError: invalid literal for int() with base 10: '34.5'
```

Elementos básicos de programação - Leitura e escrita

Função de avaliação de *strings*

BNF

<função de avaliação> ::= eval(<cadeia de caracteres>)

EXAMPLES

```
eval('200 + 2')
type(eval('200 + 2'))
x = eval(input("Introduza uma expressão:\n->\t"))
```

```
In [151]: eval('x + 1')
```

Out[151]:

5

Elementos básicos de programação - Leitura e escrita

Função de escrita (no ecrã)

BNF

<saída> ::= print() | print(<expressões>)

<expressões> ::= <expressão> | <expressão>, <expressões>

EXEMPLOS

```
a = 2
b = 10
print("a = ", a, "b = ", b)
print("a = ", a, "\nb = ", b)
```

```
In [153]: print(24, True, 45.6, "FP")
```

24 True 45.6 FP

Elementos básicos de programação - Leitura e escrita

Outro exemplo:

```
x = eval(input("Introduza uma expressão:\n\t"))
y = input("Introduza uma string:\n\t")
print(x, "e", y)
```

- Qual é o valor resultante de avaliar a função `print()` ?

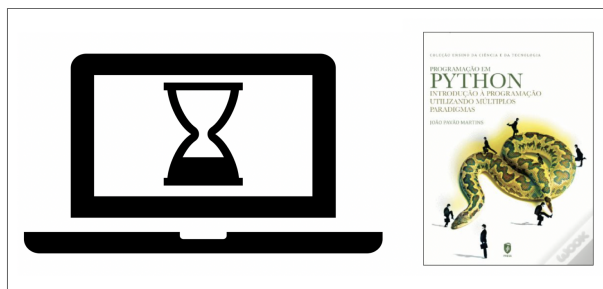
```
val = print (x, "e", y)
print(val)
```

```
In [360]: val = print (x, "e", y)
          print(val)
```

154 e boa tarde
None

Elementos básicos de programação - Tarefas próxima semana

- Trabalhar matéria apresentada esta semana
- Ler seções 2.6-2.9 do livro da UC
- Nas aulas de problemas da próxima semana:
 - Mini-teste BNF no início da primeira aula (L03)
 - L03: Elementos básicos de programação I
 - L04: Elementos básicos de programação II



In []: