Fundamentos da Progamação

Aula 5

Elementos básicos de programação

Predicados e condições. Comunicação com o exterior.

Alberto Abad, IST, 2024-25

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 `True` se e só se os valores das expressões e_1 e e_2 são iguais.
e_1 != e_2	Números	Tem o valor `True` se e só se os valores das expressões e_1 e e_2 são diferentes.
e_1 > e_2	Números	Tem o valor `True` 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 `True` 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 `True` 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 `True` se e só se o valor da expressão e_1 é menor ou igual ao valor da expressão e_2.

In [ ]:
```
not ""
```

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 [ ]:
```

Elementos básicos de programação - Leitura e escrita

Leitura de dados (do teclado)

BNF

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

- 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 [ ]:
```
input('Escreva a\n\tsua idade:')
```

Elementos básicos de programação - Leitura e escrita

Função de avaliação de *strings*

BNF

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

Examples

```
eval('200 + 2')   # avalia a expressão e retorna inteiro
type(eval('200 + 2'))
eval('2 > 1')     # avalia a expressão e retorna lógico
x = eval(input("Introduza uma expressão:\n->\t"))
```

In [ ]:
```
eval('2 > 1')
```

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 [ ]:
```
a = 2
b = 10
print("a = ", a, "b = ", b)
print("a = ", a, "\n\tb = ", b)
```

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)
```

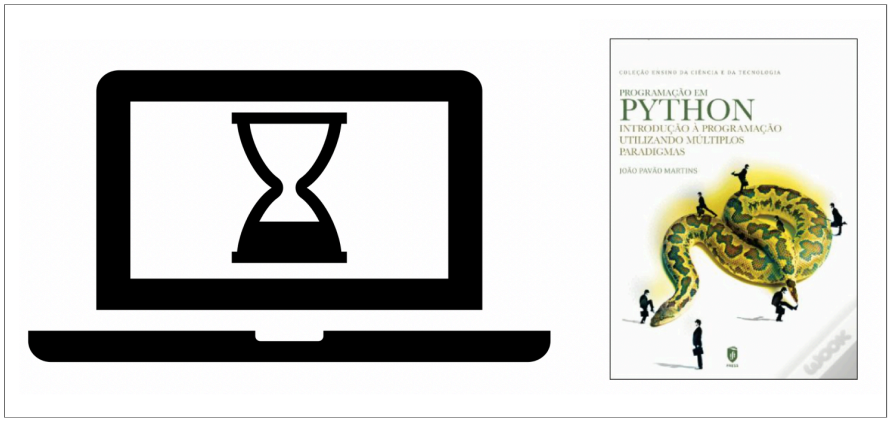
- Qual é o valor resultante de avaliar a funçaõ `print()` ?

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

```
In [ ]: print(x)
```

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 programanção I
    - L04: Elementos básicos de programanção II




```
In [ ]:
```