# Revisões

Programação I

2016.2017

Teresa Gonçalves tcg@uevora.pt

Departamento de Informática, ECT-UÉ

### Como programar?

#### Processo de desenvolvimento

- 1.Compreender o problema
- 2.Conceber o algoritmo
- 3.Implementar o algoritmo
- 4.Testar

### **Como aprender?**

Estudar, estudar, ...

Praticar, praticar, ...

Cometer erros, cometer erros, ...

Aprender com os erros, ...





### Sumário

#### Revisões

Conceitos básicos

**Tipos** 

Operadores aritméticos

Operadores lógicos e relacionais

Funções

Instruções

### Atribuição



# Revisões

### **Conceitos básicos**

**Valor** 

Variável

**Operador** 

**Expressão** 

Instrução

Comentário



# Tipos

### int

inteiro

#### float

real

### complex

complexo

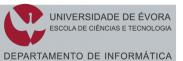
#### bool

Booleano

True, False

#### str

cadeia de caracteres



# **Operadores aritméticos**

```
Soma, concatenação
Subtração
Multiplicação, repetição
Potenciação
```

```
Divisão
Divisão inteira
Divisão inteira
modulus (resto da divisão inteira)
```

# Operadores lógicos e relacionais

### **Operadores lógicos**

and

Conjunção, e lógico

or

Disjunção, ou lógico

not

Negação

### **Constantes lógicas**

True

False

### **Operadores relacionais**

==

Igual

!=

**Diferente** 

< , <=

Menor, menor ou igual

>,>=

Maior, maior ou igual



# Precedência de operadores

### Não lógicos > lógicos

#### **Numéricos**

potenciação

multiplicação, divisão

soma, subtração

### Lógicos

*Major* not

and

*Menor* or



# **Funções – conceitos importantes**

Definição vs. invocação

Cabeçalho vs. corpo

Parâmetros vs. argumentos

Resultado

Visibilidade

Fluxo de execução

Módulo



# Funções – definição

```
def nome_funcao(par1, par2, ...):
    instrucao1
    instrucao2
    ...
    [return valor]
```

# Funções built-in

#### Entrada/saída

input([prompt])

print(obj1 [, obj2, ...])

### Conversão de tipos

float(expr)

int(expr)

int(str, base)

str(expr)

eval(str)

#### **Matemáticas**

round(expr)

abs(expr)

max()

min()

#### **Outras**

type(nome)

help(nome)



### Instruções

### Atribuição

### Retorno de valor

return

#### **Condicional**

```
if cond1:
    instrucoes1
[elif cond2:
    instrucoes2]
...
[else:
    instrucoesn]
```

### **Exercícios**

### 1. Implemente a função menos\_significativo

Argumentos: int

Resultado: algarismo menos significativo do inteiro dado como argumento

### 2. Implemente a função ou\_exclusivo

Argumentos: bool, bool

Resultado: valor verdade resultado da aplicação da operação ou\_exclusivo aos dois booleanos

### 3. Implemente a função porta\_logica

Argumentos: string, bool, bool

Resultado: valor verdade resultado da aplicação da porta lógica (AND ou OR) indicada pela string aos dois booleanos





# Re-atribuição

### Uma nova atribuição faz com que uma variavel existente passe a referenciar um novo valor

### **Exemplo**

$$>>> x=7$$

7

$$>> x=5$$

5

# Atribuição não é teste de igualdade!

$$= vs ==$$



# Atualização de variáveis

### Numa atualização, o novo valor depende do anterior

```
>>> x = x+1
>>> y=y+1
Traceback (most recent call last):
  File "<pyshell#59>", line 1, in <module>
    y=y+1
NameError: name 'y' is not defined
```

### É necessário inicializar antes de atualizar!

```
>>> y=0
>>> y=y+1
```



### Incrementar e decrementar

#### Incrementar

Atualizar uma variavel adicionando 1

$$>>> x = x+1$$

#### **Decrementar**

Atualizar uma variavel **subtraindo** 1

$$>>> x = x-1$$