# Tipos básicos

Programação I

2016.2017

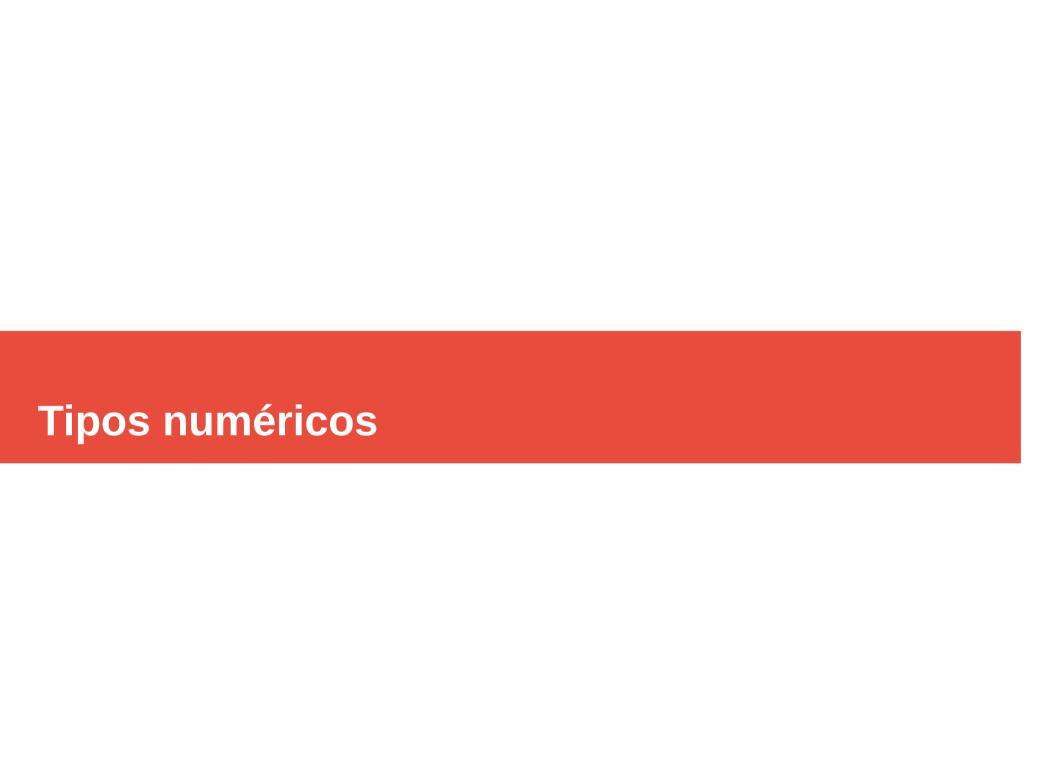
Teresa Gonçalves tcg@uevora.pt

Departamento de Informática, ECT-UÉ

# Sumário

**Tipos numéricos Tipo Booleano String** Conversão de tipos





# **Inteiros**

#### int

Inteiro que pode ser positivo ou negativo

Não tem limite máximo

Python 2: representado com 32 bits (ou 64 nas arquiteturas amd64)

Quantos valores se consegue representar com 32 bits?

Em inteiros com sinal, qual o maior e o menor números representáveis?



# Reais

### float

Valores numéricos com parte decimal

A melhor aproximação aos números reais

### **Sintaxe**

Simples: 0.000578

Forma científica:  $5.78e-4 \rightarrow 5.78 * 10^{-4}$ 

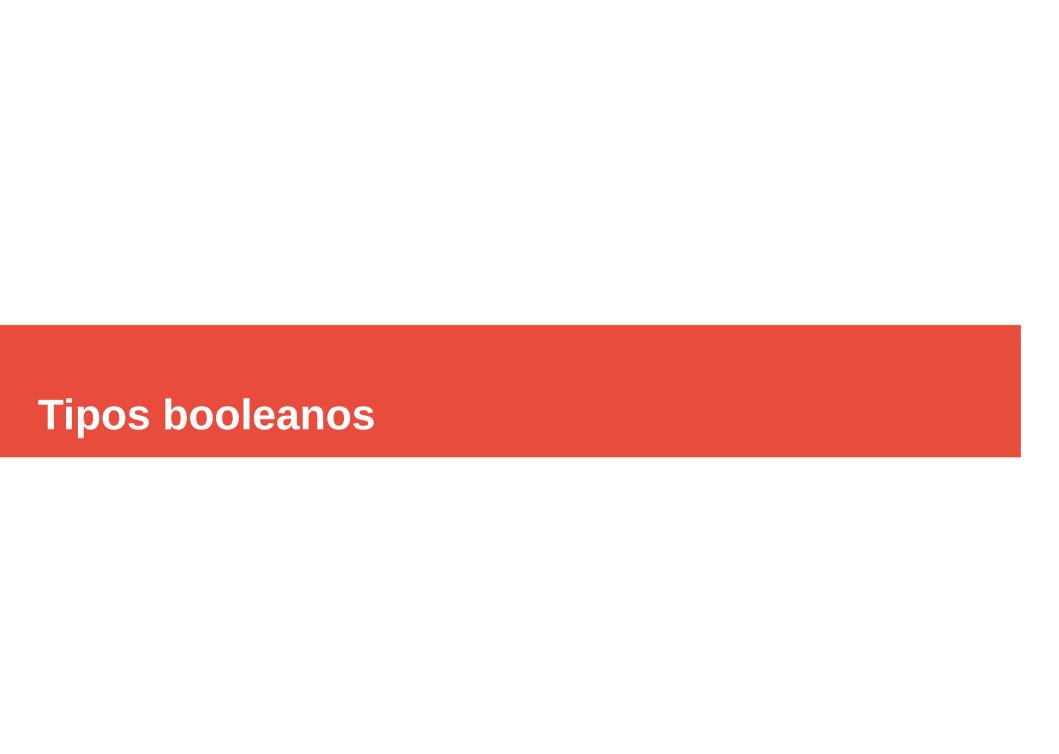
Outra:  $2^{**}-4 \rightarrow 2^{-4}$ 

# Complexos

### complex

Para representar números complexos (parte real e parte imaginária)

```
>>> z = complex(2,3)
>>> z
(2+3j)
>>> z.real
2.0
>>> z.imag
3.0
```



# **Booleano**

# Representam um valor verdade Constantes

True, False

### **Operadores lógicos (booleanos)**

```
X or y
Se x == False então y, senão x

X and y
Se x == False então x, senão y
not x
Se x == False então True, senão False
```



# **Operadores booleanos**

#### **Aceitam outros tipos**

#### **Expressões com valor False**

0 (em qualquer tipo numérico)

Sequência vazia: '', (), []

#### **Expressões com valor True**

Restantes

#### **Exemplos**



# **Operadores relacionais**

### Dão origem a um valor booleano

x == y → verdade se x é igual a y

Não confundir com a atribuição!

 $x != y \rightarrow verdade se x é diferente de y$ 

 $x < y \rightarrow verdade se x é menor que y$ 

 $x > y \rightarrow verdade se x é maior que y$ 

x <= y → verdade se x é menor ou igual a y

x >= y → verdade se x é maior ou igual a y

# Precedência

#### not > and > or

```
>>> False or True and False False
```

### Precedência menor que operadores não booleanos

not 0+1 equivalente a not (0+1)

### **Outros exemplos**

# Utilização

### Comparação ou teste

```
Igualdade
  idade == 18
Desigualdade
  altura > 1.80
```

### Expressar conjunto de características

Conjunções, disjunções e outras combinações

```
Inteiro par e menor que 10
type(x) == int and x<10 and x%2==0</pre>
```

# **String**

#### Cadeia de caracteres

### **Operações elementares**

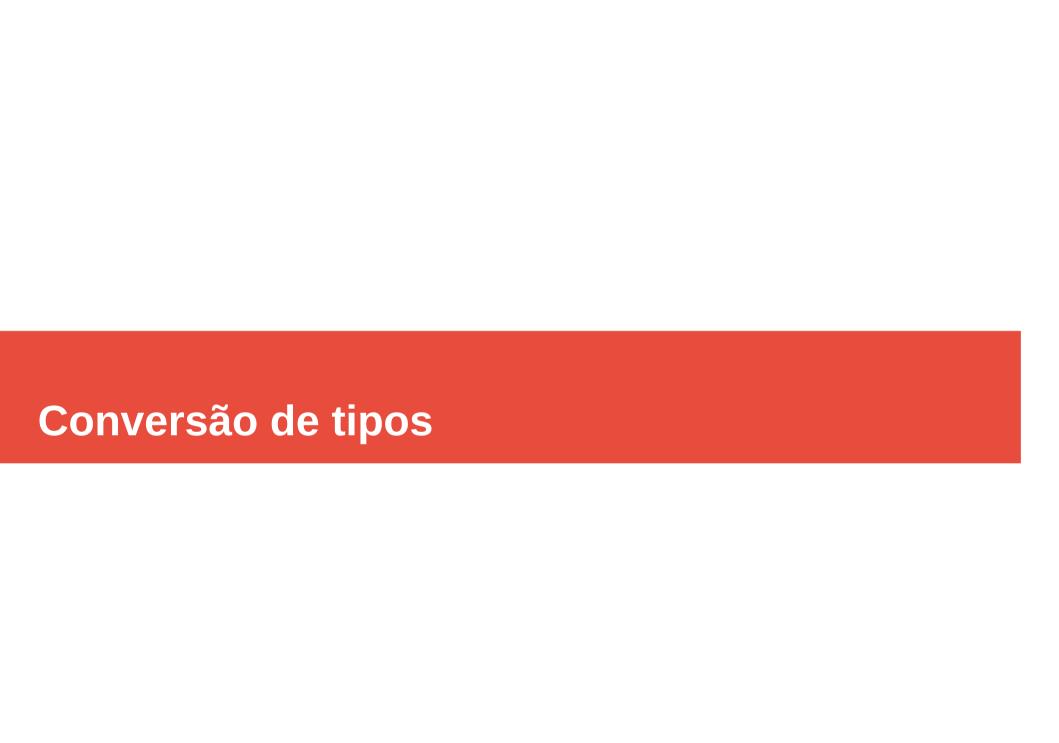
```
+ (concatenação)

'Hello'+'World'+'!' → 'Hello World!'

Que propriedade da soma de números não é válida na concatenação?
```

```
* (repetição)

'Spam' * 3
```



# Conversão explícita

### float(expr)

Converte expr para um valor em vírgula flutuante

### int (expr)

Converte expr para um valor inteiro

### str (expr)

Devolve uma cadeia de carateres, resultado da expressão expr

```
>>> str( 1+2/3 )
```

'1.66666666666665'

### eval (string)

Avalia string como expressão

```
>>> eval ('1+2/3')
```

1.666666666666665



# Conversão implícita

#### Conversão automática

Para permitir aritmética entre operandos de tipo diferente

### Conversão para o tipo "mais abrangente"

int → float

float → complex

# Outras funções built-in

### round(n [, ndig])

Arredonda para valor com ndig algarismos à direita da vírgula

# abs(n)

Valor absoluto de n

### max(), min()

Valor máximo e mínimo (respetivamente) de um conj. de valores

Permitem um nº variável de argumentos

# int(str [, base])

Converte a string str para um inteiro usando a base indicada; por omissão usa a base 10

