



# Programação I

Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# TUPLAS

- ▶ Vimos até agora **tipos primitivos** da linguagem e seus **valores elementares**:
  - ▶ Inteiro: 1 -6 389
  - ▶ Real: 0.9 -2.3 345.789
  - ▶ Caracter: 'r' '#' '5'
- ▶ Percebemos porém que, em alguns casos, precisávamos tratar ao mesmo tempo de mais de um valor elementar:
  - ▶ Raízes da equação de 2º grau
  - ▶ Pontos no plano cartesiano

# TUPLAS

- ▶ Uma tupla é um **tipo composto** que nos permite representar, **valores estruturados**, ou seja, uma combinação de mais de um valor elementar.
- ▶ A representação de uma tupla é feita com a seguinte sintaxe:
  - ▶  $(t_1, t_2, t_3, \dots, t_n)$
- ▶ Exemplos:
  - ▶  $(27, 04, 2011)$
  - ▶  $(\text{'A'}, \text{'espada'})$
  - ▶  $(x, y, x > y)$

# TUPLAS

- ▶ Os valores de uma tupla são imutáveis !!
- ▶ Tupla vazia
- ▶ `>>> tupla = ()`
- ▶ Tupla com um único elemento (note a necessidade da vírgula, mesmo sendo um único elemento)
- ▶ `>>> tupla = (1,)`

# TUPLAS

```
>>> def maiorQue(x, y): return (x, y, x > y)
>>> maiorQue(1,3)
(1, 3, False)
```

```
>>> def divisaolnt(x, y): return (x//y, x%y)
>>> divisaolnt(10,3)
(3, 1)
```



# TUPLAS

- Podemos ainda usar tuplas como parâmetros

```
>>> def distancia(p1, p2):  
    def prim((x,y)): return x  
    def seg((x,y)): return y  
    def dx(): return prim(p1) - prim(p2)  
    def dy(): return seg(p1) - seg(p2)  
    return sqrt(dx()2 + dy()2)
```

```
>>> distancia((0,0),(3,4))
```

```
5.0
```



# TUPLAS

► Ou ainda ....

```
>>> def distancia((x1,y1), (x2, y2)):
def dx(): return x1 - x2
def dy(): return y1 - y2
return sqrt(dx()**2 + dy()**2)
>>> distancia((0,0),(3,4))
5.0
```

# Operadores Básicos sobre Tuplas

Expressão	Resultado	Descrição
<code>len((1,2,3))</code>	3	Número de elementos que a tupla contém
<code>(1, 2, 3) + (4, 5, 6)</code>	<code>(1, 2, 3, 4, 5, 6)</code>	Concatenação
<code>(1,) * 4</code>	<code>(1,1,1,1)</code>	Repetição
<code>3 in (1, 2, 3)</code>	True	Pertencimento



# Exercícios

- ▶ Escreva uma função que recebe uma tupla e retorna True se o primeiro elemento for igual ao último elemento da tupla
- ▶ Defina uma função para determinar as raízes de uma equação de 2º grau
- ▶ Escreva a função **opera** que recebe uma tupla com uma string e dois números; se a string for 'SOMA', retorna a soma dos dois números, se for 'MULT', retorna a multiplicação, se for 'DIV', retorna a divisão, se for 'SUB', retorna a subtração, se não for nenhuma das anteriores retorna None.

That's all Folks!



nemo