

# Computação I - Python

## Aula 4 - Variáveis e Atribuição, Strings e Tuplas

### Tuplas: mais exemplos

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



# Tuplas

- Mais exemplos de uso de Tuplas.

# Tuplas

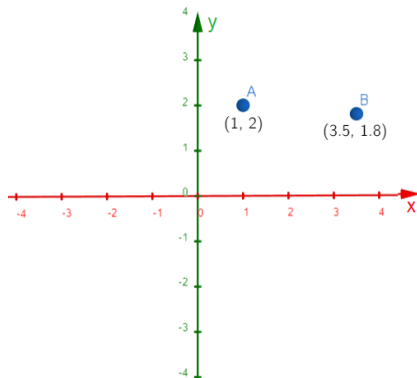
- Vamos analisar um exemplo um pouco mais elaborado!



- Distância entre dois prédios conhecidos da cidade do Rio de Janeiro  
Igreja da Candelária < -- > CCBB

# Tuplas

- Transportando o cenário do problema de localização para o plano cartesiano



- Pontos dispostos no espaço bi-dimensional

# Tuplas

- Distância Euclidiana

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Uma função para calcular a distância euclidiana entre os pontos A e B

```
1 import math
2 def distancia(A,B):
3     ''' Funcao que calcula distancia
4     euclidiana dados dois pontos
5     Parametros de entrada: tipo tupla(float , float)
6     Retorno: tipo float '''
7     dist = math.sqrt(pow(A[0]-B[0],2)+
8                     pow(A[1]-B[1],2))
9     return round(dist,2)
10
11 >>> distancia((1.0,2.0),(3.5,1.8))
12 2.51
```

**Teste: distancia((1.0,2.0),(3.5,1.8))**  
**Valor de retorno: 2.51**

# Tuplas

- Uma outra forma de definir uma função para calcular a distância euclidiana entre os pontos A e B

```
1 from math import *
2 def distanciaPontos(A,B):
3     ''' Funcao que calcula distancia
4     euclidiana dados dois pontos
5     Parametros de entrada: tipo tupla(float , float)
6     Retorno: tipo float '''
7     return round(hypot(B[0]-A[0],B[1]-A[1]),2)
8
9 >>> distanciaPontos((1.0,2.0),(3.5,1.8))
10 2.51
```

**Teste: distanciaPontos((1.0,2.0),(3.5,1.8))**  
**Valor de retorno: 2.51**

# Tuplas

- a. Escreva uma função que recebe uma tupla e retorna **True** se o primeiro elemento for igual ao último elemento da tupla.

# Tuplas

- a. Escreva uma função que recebe uma tupla e retorna **True** se o primeiro elemento for igual ao último elemento da tupla.

```
1 def igual_if(tup):
2
3     """Funcao que retorna True se o inicio de uma tupla e igual
4     ao seu final.
5     O parametros de entrada e do tipo tupla.
6     O valor de retorno e do tipo booleano."""
7
8     return tup[0] == tup[-1]
9
10 >>> igual_if((1,2,1))
11 True
```



# Tuplas

- a. Escreva uma função que recebe uma tupla e retorna **True** se o primeiro elemento for igual ao último elemento da tupla.

```
1 def igual_if(tup):
2
3     """Funcao que retorna True se o inicio de uma tupla e igual
4     ao seu final.
5     O parametros de entrada e do tipo tupla.
6     O valor de retorno e do tipo booleano."""
7
8     return tup[0] == tup[-1]
9
10 >>> igual_if((1,2,1))
11 True
```

- b. Escreva uma função *inverte* que recebe uma tupla de três elementos e retorna uma nova tupla com os elementos na ordem reversa.

# Tuplas

- b. Escreva uma função *inverte* que recebe uma tupla de três elementos e retorna uma nova tupla com os elementos na ordem reversa.

```
1 def inverte(tup):
2     """Funcao que inverte elementos de uma tupla de tamanho 3.
3     O parametros de entrada e uma tupla de tamanho 3.
4     O valor de retorno e uma tupla de tamanho 3."""
5     return tup[2], tup[1], tup[0]
6
7 >>> inverte((1,2,3))
8 (3,2,1)
```

```
1 def inverte(tup):
2     """Funcao que inverte elementos de uma tupla de tamanho 3.
3     O parametros de entrada e uma tupla de tamanho 3.
4     O valor de retorno e uma tupla de tamanho 3."""
5     return tup[::-1]
6
7 >>> inverte((1,2,3))
8 (3,2,1)
```

- Resumo: neste vídeos vimos mais alguns exemplos de uso de Tuplas.

## Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

## Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

# Computação I - Python

## Aula 4 - Variáveis e Atribuição, Strings e Tuplas

### Tuplas: mais exemplos

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>