

Profa. Dra. Raquel C. de Melo-Minardi
Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais



MÓDULO 2 – PROGRAMAÇÃO

Sequências – Tuplas

TIPOS DE VARIÁVEIS OU ESTRUTURAS DE DADOS EM PYTHON

- ▶ As **sequências** podem ser de dois tipos:
 - ▶ **sequências imutáveis**: são objetos ordenados e finitos
 - ▶ **strings**: cadeias de caracteres
 - ▶ **tuples**: dois ou mais elementos de qualquer tipo dentro de parênteses e separados por vírgula
 - ▶ **sequências mutáveis**
 - ▶ **lists**: conhecidas em outras linguagens como vetores ou arranjos
 - ▶ **sets**: coleções não ordenadas e não indexadas escritas entre chaves

TUPLAS

- ▶ **Tupla** é uma lista imutável `t = ('10', 'segunda-feira', 'Fevereiro', 2022)`
- ▶ O que diferencia a lista da tupla é que a primeira pode ter elementos adicionados a qualquer momento, enquanto que a segunda estrutura, após definida, **não permite a adição ou remoção de elementos**
- ▶ Uma tupla é definida da mesma forma que uma lista, o que as diferencia são os caracteres que as delimitam
- ▶ Listas tem seus elementos delimitados por colchetes, enquanto que a tuplas tem seus elementos delimitados por **parênteses**

TUPLAS

- ▶ A ordem dos elementos numa tupla será a ordem na qual estes foram definidos, ou seja, não é possível ordenar em tempo de execução os elementos
- ▶ O primeiro elemento de uma tupla também possui índice igual a 0 e o último índice igual $n-1$

TUPLAS

- ▶ Outras formas de se **declarar tuplas**:

```
#tupla declarada sem o uso de parenthesis
```

```
t1 = 1, 2, 3
```

```
#tupla declarada com o uso de parênteses
```

```
t2 = (1, 2, 3)
```

```
#tupla com um único elemento
```

```
t3 = 1,
```

```
#tupla vazia
```

```
t4 = ()
```

- ▶ É importante notar que não são os parênteses que definem uma tupla, mas as **vírgulas**

OPERADORES PARA MANIPULAÇÃO DE TUPLAS

- ▶ Há alguns operadores que podem ser usados para operar em objetos do tipo tupla

VERIFICAÇÃO DE PERTENCIMENTO EM TUPLAS

- ▶ O operador `in` pode ser usado para retornar se um determinado elemento existe (pertence a) em uma tupla:

```
5 in tupla # Retornara True ou False
```

SOMA E MULTIPLICAÇÃO DE TUPLAS

- ▶ É possível **somar** e “**multiplicar**” tuplas

```
t = ('a', 'b') + ('c',) # t = ('a', 'b', 'c')
```

```
t += (3,) # t = ('a', 'b', 'c', 3)
```

```
t = t * 3 # t = ('a', 'b', 'c', 'a', 'b', 'c', 'a', 'b', 'c')
```

- ▶ Note que essas operações se dão de forma diferente da convencional ou esperada
- ▶ Na realidade, elas não fazem operações aritméticas com as tuplas e sim concatenam a tupla

ENCONTRAR A POSIÇÃO DE UM ELEMENTO EM TUPLAS

- ▶ O método `index()` retorna a primeira ocorrência do elemento em uma tupla

```
tupla.index('A')
```

CONTAGEM DE OCORRÊNCIAS DE ELEMENTOS EM TUPLAS

- ▶ O método `count()` retorna a o **número de ocorrências** do elemento em uma tupla:

```
tupla.count('A')
```

OBTENDO PARTES DE TUPLAS

- ▶ Da mesma forma como ocorre com as *strings*, não se usa métodos para tal mas apenas indexação:

```
tupla[6:9]  
tupla[:9]  
tupla[6:]
```

- ▶ Pode-se utilizar índices negativos que começam a contar do final, ou seja, -2 equivale ao penúltimo item

```
tupla[-2:] # Retorna os elementos do penúltimo até o fim
```

TAMANHO DE UMA TUPLA

- ▶ O método `len()` nativo de Python retorna o tamanho de uma tupla:

```
len(tupla)
```