Tuplas

Alessandro Botelho Bovo

Tuplas

- Estruturas **imutáveis** (não podem ser alteradas após criadas).
- Armazenam múltiplos elementos em uma única variável.
- Sintaxe com parênteses `()` (parênteses opcionais).

```
coordenada = (10, 20)
print(coordenada)
```

```
(10, 20)
```

Cuidado ao criar tuplas!

- Cuidado 1: Parênteses não são obrigatórios

```
tupla1 = (1, 2, 3)
tupla2 = 1, 2, 3

print(tupla1)          # (1, 2, 3)
print(type(tupla1))    # <class 'tuple'>

print(tupla2)          # (1, 2, 3)
print(type(tupla2))    # <class 'tuple'>
```

Cuidado ao criar tuplas!

- Cuidado 2: Tupla com 1 elemento

```
tupla1 = (4)            # Não é tupla
print(tupla1)           # 4
print(type(tupla1))     # <class 'int'>

tupla2 = (4,)           # Isso é tupla
print(tupla2)           # (4,)
print(type(tupla2))     # <class 'tuple'>
```

Importante lembrar

- Tuplas são definidas **pela vírgula**, e não pelos parênteses.
- Parênteses são usados apenas para **organizar** ou melhorar a **legibilidade**.

```
tupla = 4,              # Isso é tupla
print(tupla)            # (4,)
print(type(tupla))      # <class 'tuple'>
```

Diferença em relação às listas

- Listas usam [] e são **mutáveis**.
- Tuplas usam () e são **imutáveis**.

```
lista = [1, 2, 3]
tupla = (1, 2, 3)

lista[0] = 10    # funciona
# tupla[0] = 10  # ERRO! Não é permitido
```

Vantagens das tuplas

- Maior **segurança** dos dados (não podem ser alterados por engano).
- **Mais rápidas** para acessar elementos.
- Úteis como chaves de dicionários.
- Representam dados que não devem mudar, ex.: coordenadas, datas.

Acesso em tuplas

- Por índice

```
ponto = (10, 20, 30)
print(ponto[0])
print(ponto[-1])
```

```
10
30
```

Acesso em tuplas

- Fatiamento (*slicing*)

```
ponto = (10, 20, 30)
print(ponto[1:3])
print(ponto[::2])
```

```
(20, 30)
(10, 30)
```

Acesso em tuplas

- Desempacotamento

```
coordenada = (3, 4)
x, y = coordenada
print(x, y)
```

```
3 4
```

Iteração

- Com índice

```
ponto = (10, 20, 30)
for i in range(len(ponto)):
    print(f"Posição {i} -> {ponto[i]}")
```

```
Posição 0 -> 10
Posição 1 -> 20
Posição 2 -> 30
```

Iteração

- Direto nos valores

```
ponto = (10, 20, 30)
for valor in ponto:
    print(valor)
```

```
10
20
30
```

Iteração

- Com *enumerate*

```
ponto = (10, 20, 30)
for i, valor in enumerate(ponto):
    print(f"Posição {i} -> {ponto[i]}")
```

```
Posição 0 -> 10
Posição 1 -> 20
Posição 2 -> 30
```

Exemplos

- Coordenadas: `(3, 4)`
- Datas: `(15, 8, 2000)`
- Registros fixos: `("Ana", "ana@email.com", "(43) 99999-9999")`
- Cores: `(255, 100, 0)`
- Meses: `('Janeiro', 'Fevereiro', 'Março', 'Abril', ...)`

Contato

Alessandro Botelho Bovo

alessandrobovo@utfpr.edu.br