SCC 124 - Introdução à Programação para Engenharias

Tuplas

python

Professor: André C. P. L. F. de Carvalho, ICMC-USP
Pos-doutorando: Isvani Frias-Blanco
Monitor: Henrique Bonini de Britto Menezes

Aula de hoje

- Introdução
- Tuplas
- Enpacotamento e desempacotamento
- Operações para sequências
- Conversão de lista para tupla e de tupla para lista

Tuplas

- Outro tipo de sequência
  - Também é uma coleção de elementos
- Sintaxe
  - Separados por ", " (vírgula)
  - Entre parênteses
    - Quando atribuídos, podem vir sem parênteses
- Possíveis usos:
  - Coordenadas de um ponto, registros em um banco de dados, etc.

Exemplo

```
>>> t = 12345, 54321, 'alo!' # pode terminar com ,
>>> t[0]

>>> t

>>> # Podemos ter tuplas de tuplas
>>> u = t, (1, 2, 3, 4, 5)
>>> u

>>> z = t, 1, 2, 3, 4, 5
>>> z
```


Exemplo

```
>>> t = 12345, 54321, 'alo!' # pode terminar com ,
>>> t[0]
12345
>>> t
(12345, 54321, 'alo!')
>>> # Podemos ter tuplas de tuplas
>>> u = t, (1, 2, 3, 4, 5)
>>> u
((12345, 54321, 'alo!'), (1, 2, 3, 4, 5))
>>> z = t, 1, 2, 3, 4, 5
>>> z
((12345, 54321, 'alo!'), 1, 2, 3, 4, 5)
```


Tuplas

- Tipo imutável
  - Como em *strings*, não é possível atribuir valores a elementos de uma tupla
    - Pode simular com fatia e concatenação
- Podem ser formadas por elementos mutáveis
  - Ex. Tupla de listas
- Observações:
  - Tupla vazia: ()
  - Tupla com apenas um elemento: (x,)

Exemplo 1

```
>>> vazio = ()
>>> tup = 'olá',
>>> len(vazio)

>>> len(tup)

>>> tup

>>> a = [1, 2], [3, 4]
>>> a
```

Para indicar que é uma tupla

Exemplo 1

```
>>> vazio = ()
>>> tup = 'olá',
>>> len(vazio)
0
>>> len(tup)
1
>>> tup
('ola',)
>>> a = [1, 2], [3, 4]
>>> a
([1, 2],[3,4])
```

Para indicar que é uma tupla

Exemplo 2

```
>>> a = [1, 2], [3, 4]
>>> a[0]

>>> a[0] = 1

>>> a[0] = [18, 9]

>>> a[0][1] = 3
>>> a
```


Exemplo 2

```
>>> a = [1, 2], [3, 4]
>>> a[0]
[1,2]
>>> a[0] = 1
Erro
>>> a[0] = [18, 9]
Erro
>>> a[0][1] = 3
>>> a
([1, 3][3,4])
```


Tuplas

- Empacotamento de tupla
  - Empacota elementos em uma tupla
  - Sempre cria tupla
- Desempacotamento de sequência
  - Recupera elementos de uma tupla
  - Número de variáveis no lado esquerdo deve ser igual ao número de elementos da tupla
- Operações válidas para qualquer tipo sequência

Exemplo

```
>>> t = 12345, 54321, 'alo!' # Empacotamento de tupla
>>> x, y, z = t              # Desempacotamento de tupla
>>> x

>>> t
```

Exemplo

```
>>> t = 12345, 54321, 'alo!' # Empacotamento de tupla
>>> x, y, z = t              # Desempacotamento de tupla
>>> x
12345
>>> t
(12345, 54321, 'alo‘)
```


Operadores para sequências

- Comparação se sequências
  - Utiliza operadores convencionais de comparação
    - ==, !=, <, >, <=, >=,
- *seq1 + seq2*
  - *Constrói uma nova tupla unindo os operandos*

Exemplo

```
>>> a = (1, 2, 3)
>>> b = ( 5, 6, 7)
>>> a < a

>>> a < b

>>> a != b

>>> a + b
```


Exemplo

```
>>> a = (1, 2, 3)
>>> b = ( 5, 6, 7)
>>> a < a
False
>>> a < b
True
>>> a != b
True
>>> a + b
(1, 2, 3, 5, 6, 7)
```


Comparação de sequências

- Segue ordem lexicográfica
  - Comparação começa com os dois primeiros itens das sequências
  - Se forem diferentes, as sequencias são diferentes
  - Senão, passa para os itens seguintes das sequências

Operadores para sequências

- *max(seq)*
  - Retorna o maior elemento de *seq*
- *min(seq)*
  - Retorna o menor elemento de *seq*
- *Seq*5*
  - Gera lista com 5 repetições de *seq*
- *x in seq*
  - Verifica se o valor x pertence a *seq*
    - Retorna *True* se x pertence a *seq* e *False* caso contrário

Exemplo 1

```
>>> a = (1, 2, 5)
>>> min (a)

>>> max (a)

>>> a*4

>>> 3 in a
```


Exemplo 1

```
>>> a = (1, 2, 5)
>>> min (a)
1
>>> max (a)
5
>>> a*4
(1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 2, 5)
>>> 3 in a
False
```


Exemplo 2

```
a = (0, 1, 2, 6, 8, 9)
for x in a:
    y = x + 1
    y += 1
    if y in a:
        print ("x = %d y = %d" % (x, y))
b = a*3
print (len (b))
```


Exemplo 2

```
a = (0, 1, 2, 6, 8, 9)
for x in a:
    y = x + 1
    y += 1
    if y in a:
        print ("x = %d y = %d" % (x, y))
b = a*3
print (len (b))
```

Saída:
x = 0 y = 2
x = 6 y = 8
18

Conversão de sequências

- list(tup)
  - Converte tupla *tup* em uma lista
  - Também pode converter string em lista
- tuple(lta)
  - Converte lista *lta* em uma tupla
  - Também pode converter string em tupla

Exemplo

```
>>> a = (1, 2, 5)
>>> b = list (a)
>>> b

>>> c = tuple (b)
>>> c
```

Exemplo

```
>>> a = (1, 2, 5)
>>> b = list (a)
>>> b
[1, 2, 5]
>>> c = tuple (b)
>>> c
(1, 2, 5)
```


Conclusão

- Introdução
- Tuplas
- Enpacotamento e desempacotamento
- Operações para sequências
- Conversão de lista para tupla e de tupla para lista

Perguntas