Algoritmos e Estruturas de Dados I
Aula15

Tuplas e Dicionários

Prof. MSc. Adalto Selau Sparremberger


assparremberger@senacrs.com.br


@adaltoss


/assparremberger

Fecomércio RS | Senac

Estruturas de Dados

- Listas -> [ ] | x = [1 , 2]
- Tuplas -> ( ) | y = (1 , 2)
- Dicionários -> { } | z = { 1 , 2 }

Tuplas

- Uma tupla é uma coleção que é ordenada e **imutável** .
- Tuplas em Python são escritas com parênteses.
- A partir do Python 3, não precisa de parênteses.

```
carros = ("Uno", "Doblo", "Toro")
print( carros )
```

```
('Uno', 'Doblo', 'Toro')
```

Tuplas

```
carros = ("Uno", "Doblo", "Toro", "Hilux")
print( carros )


print( carros[1] )


print( carros[1:3] )


for car in carros:
    print( car )
```

```
('Uno', 'Doblo', 'Toro', 'Hilux')

---------

Doblo

---------

('Doblo', 'Toro')

---------

Uno
Doblo
Toro
Hilux
```

Tuplas

```
carros = ("Uno", "Doblo", "Toro", "Hilux")

for cont in range( len(carros) ):
    print( "Posição ", cont , " - Carro: ", carros[cont] )
```

```
Posição  0  - Carro:  Uno
Posição  1  - Carro:  Doblo
Posição  2  - Carro:  Toro
Posição  3  - Carro:  Hilux
```

```
for index, car in enumerate(carros):
    print( "Posição ", index , " - Carro: ", car )
```

```
Posição  0  - Carro:  Uno
Posição  1  - Carro:  Doblo
Posição  2  - Carro:  Toro
Posição  3  - Carro:  Hilux
```

Tuplas

```
carros = ("Uno", "Doblo", "Toro", "Hilux")

print( sorted( carros ) )


print( carros )
```

```
['Doblo', 'Hilux', 'Toro', 'Uno']

--------------------------------

('Uno', 'Doblo', 'Toro', 'Hilux')
```

Dicionários

- Um dicionário é uma coleção que é desordenada, mutável e indexada.
- Em Python, dicionários são escritos com chaves, e sua estrutura é composta por:

{ “key” : “value” }

Dicionário

```
pessoa = {
   "nome" : "Maria" ,
   "idade" : 25 ,
   "altura" : 1.68  ,
   "temFilhos" : True
}

print( pessoa )


print( pessoa["nome"] )
```

```
{'nome': 'Maria', 'idade': 25, 'altura': 1.68, 'temFilhos': True}

----------

Maria
```

Dicionário
Editando e Adicionando elemento

```
pessoa = {
   "nome" : "Maria" ,
   "idade" : 25
}
print( pessoa )
```

```
{'nome': 'Maria', 'idade': 25}
```

```
print( pessoa["nome"] )
```

```
------------
Maria
```

```
pessoa["nome"] = "Júlia"
print( pessoa )
```

```
------------
{'nome': 'Júlia', 'idade': 25}
```

```
pessoa["email"] = "j@a.com"
print( pessoa )
```

```
------------
{'nome': 'Júlia', 'idade': 25, 'email': 'j@a.com'}
```

Dicionário – Excluindo elemento

```
pessoa = {
   "nome" : "Maria" ,
   "idade" : 25 ,
   "email" : "m@mail.com"
}
print( pessoa )



del pessoa["email"]
print( pessoa )
```

```
{'nome': 'Maria', 'idade': 25, 'email': 'm@mail.com'}

------------

{'nome': 'Maria', 'idade': 25}
```

Map

- A função **map()** executa uma função especificada para cada item em um iterável (array, list, ...).
- O item é enviado para a função como um parâmetro.

Map

```
def getSizeArray(a):
  return len(a)

x = map( getSizeArray, ('maçã', 'banana', 'laranja') )

print(x)
print("----------------------------")
#convertendo o map para uma lista, para facilitar a leitura:
print(list(x))
```

```
<map object at 0x14aafb66c130>

----------------------------

[4, 6, 7]
```

Exercícios

- Construa um algoritmo que possua uma tupla com os números escritos por extenso de “zero” a “nove”. Peça ao usuário para digitar um número de 0 a 9 e retorne a ele o número por extenso, sem usar estruturas condicionais (if e switch).

- Construa um algoritmo que peça ao usuário para informar o nome, a nota01 e a nota02 de um aluno. Guarde estas informações em um dicionário. Após, calcule a nota final deste aluno [(nota01 + nota02) /2] e adicione ao dicionário. Ao final, imprima todos os dados do dicionário.