

Introdução ao Python

Estrutura de dados - Conjuntos (Set)

Henrique Y. Shishido

Departamento de Computação
Universidade Tecnológica Federal do Paraná

Introdução

- Um Set é usado para armazenar múltiplos itens em uma única variável
- É uma coleção não ordenada, itens existentes são imutáveis, não são indexáveis e não permitem valores repetidos
 - Não-ordenado: os itens não possuem uma ordem definida;
 - Imutável: os itens já adicionados são imutáveis, contudo é possível adicionar ou remover novos elementos no conjunto
 - Valores não-duplicados: não são permitidos dois itens com o mesmo valor

Inicialização de uma Set

- Um Set é inicializado envolvendo o conjunto de valores dentro de chaves { e }

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pastor Alemão", "Pequinês"}  
# ou  
>>> racasCachorro = set(("Bulldog", "Pincher", "Pastor Alemão", "Pequinês"))  
  
>>> print(racasCachorro)  
{"Bulldog", "Pincher", "Pastor Alemão", "Pequinês"}
```

- Valores duplicados são desconsiderados:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Bulldog"}  
  
>>> print(racasCachorro)  
{"Bulldog", "Pincher"}
```

Tamanho de um Set

- O número de elementos de um Set pode ser obtido pela função `len()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}  
  
>>> print(len(racasCachorro))  
3
```

Itens de um Set

- Semelhante às listas e tuplas, os elementos que compõem um Set podem ser de diferentes tipos de dados
- Um Set também pode ser composta por valores de diferentes tipos de dados

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}  
>>> numeros = {1, 5, 15, 30, 80}  
>>> meuSet = {1, "Bola", True}
```

Acesso a itens de um Set

- Não é possível acessar itens de um set por meio de índice ou chave
- A única forma de acessar seus elementos é por meio do comando for

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> for raca in racasCachorro:
>>>     print(raca)
"Bulldog"
"Pincher"
"Pequinês"
```

Adicionando itens no Set

- A adição de novos itens em um Set pode ser realizado por meio da função `add()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> racasCachorro.add("Shitzu")
>>> print(racasCachorro)
{"Bulldog", "Pincher", "Pequinês", "Shitzu"}
```

- Adicionar elementos de um Set para outro é possível por meio da função `update()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> novasRacas = {"Shitzu", "Labrador"}
>>> racasCachorro.update(novasRacas)
>>> print(racasCachorro)
{"Bulldog", "Pincher", "Pequinês", "Shitzu", "Labrador"}
```

Removendo itens no Set

- Para remover um item de um Set, pode-se usar o método `remove()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> racasCachorro.remove("Bulldog")
>>> print(racasCachorro)
{"Pincher", "Pequinês"}

>>> racasCachorro.remove("Doberman")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: "Doberman"
```

- Para evitar o erro na remoção de um item inexistente em um Set, pode utilizar o método `discard()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> racasCachorro.discard("Doberman")
>>> print(racasCachorro)
{"Bulldog", "Pincher", "Pequinês"}
```


Limpando todos os itens de um Set

- Para remover todos os itens de um Set, pode-se utilizar o método `clear()`:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> racasCachorro.clear()
>>> print(racasCachorro)
set()
```

Iterando nos itens de um Set

- Para percorrer os itens de um set, pode-se usar o comando for:

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> for raca in racasCachorro:
>>>     print(raca)
"Bulldog"
"Pincher"
"Pequinês"
```

Unindo dois sets

- Há diferentes formas de unir dois ou mais sets:
 - `union()`: retorna um novo conjunto contendo todos os itens dos dois sets.
 - `update()`: insere todos os itens de um set em outro.

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> novasRacas = {"Doberman", "Golden"}
>>> todasRacas = racasCachorro.union(novasRacas)
>>> print(todasRacas)
{"Bulldog", "Pincher", "Pequinês", "Doberman", "Golden"}
```

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}
>>> novasRacas = {"Doberman", "Golden"}
>>> racasCachorro.update(novasRacas)
>>> print(racasCachorro)
{"Bulldog", "Pincher", "Pequinês", "Doberman", "Golden"}
```

Unindo dois sets - Parte 2

```
#Matendo somente os duplicados
```

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}  
>>> novasRacas = {"Bulldog", "Golden", "Pincher"}  
>>> racasCachorro.intersection_update(novasRacas)  
>>> print(racasCachorro)  
{"Bulldog", "Pincher"}
```

```
#Mantendo somente os não duplicados
```

```
>>> racasCachorro = {"Bulldog", "Pincher", "Pequinês"}  
>>> novasRacas = {"Bulldog", "Golden", "Pincher"}  
>>> racasCachorro.symmetric_difference_update(novasRacas)  
>>> print(racasCachorro)  
{"Pequinês", "Golden"}
```