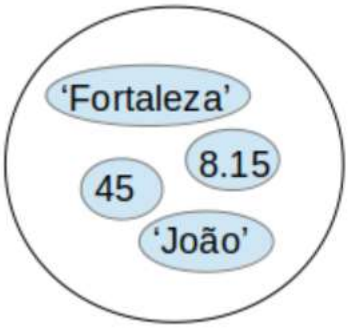
Conjuntos em Python

Listas são coleções ordenadas (a ordem dos elementos na lista é significativa: o 1º elemento, o 2º elemento, etc), indexadas (cada elemento da lista é identificável por um índice) e que permitem a duplicação de elementos, a exemplo de:

'João'	45	8.15	'Fortaleza'	45
0	1	2	3	4

Conjuntos (sets), por outro lado, são coleções não ordenadas, não indexadas e sem duplicação de elementos, pode ser representada por algo como:



Sets (ou, como iremos chamar daqui para a frente, conjuntos) são estruturas disponíveis como *builtins* do Python (identificadores embutidos do Python), utilizadas para representar **coleções desordenadas de elementos únicos**.

> Leitura adicional
> Funções embutidas
> https://docs.python.org/pt-br/3/library/functions.html#built-in-funcs

É importante sempre lembrar dos conjuntos por suas duas principais características:

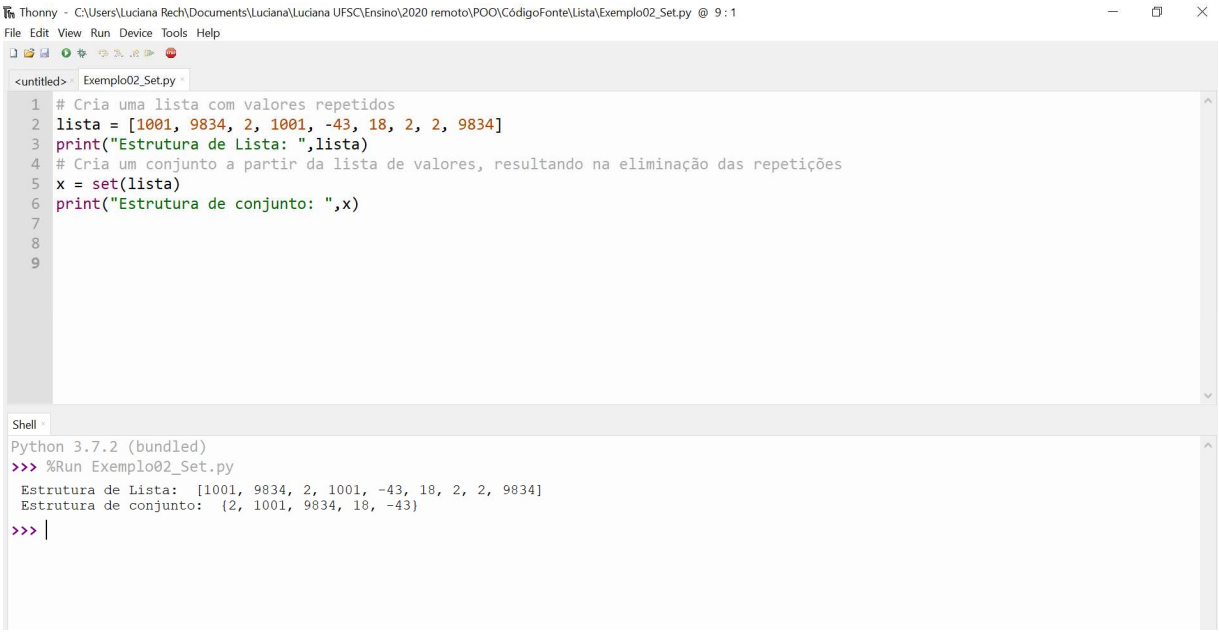
1. Os elementos não são armazenados em uma ordem específica e confiável;
2. Conjuntos não contém elementos repetidos.

A característica número 1 é importante, porque o desenvolvedor **jamais deve confiar na ordenação de um conjunto**, visto que a ordem em que os elementos são mantidos nos

conjuntos varia de implementação para implementação do interpretador Python. Conjuntos não suportam indexação nem fatiamento (*slicing*).

Criando um conjunto em Python:

Também é possível criar um conjunto a partir de uma lista já existente.



Algumas das funções e ações sobre conjuntos que são similares às vistas sobre listas, tuplas e strings:

- len(c) → número de elementos do conjunto (cardinalidade)
- x in c → verificar se um elemento **x** pertence ao conjunto **c**
- x not in c → verificar se um elemento **x** NÃO pertence ao conjunto **c**
- for x in c: → percorrer os elementos do conjunto **c**

Pertinência

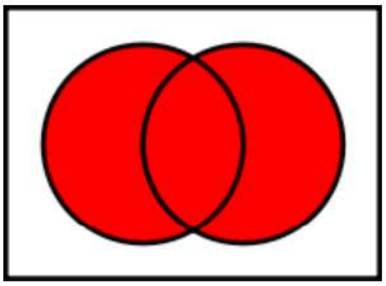
Além das operações tradicionais de união, interseção e diferença, também temos operações de verificação de pertinência. A seguir veremos algumas.

Para verificar se um determinado elemento pertence a um conjunto, podemos usar o já conhecido operador de pertinência `in`:

```
>>> a = {1, 2, 3, 4}
>>> b = {3, 4, 5, 6}
>>> 1 in a
True
>>> 5 in a
False
```

Operações me conjuntos:

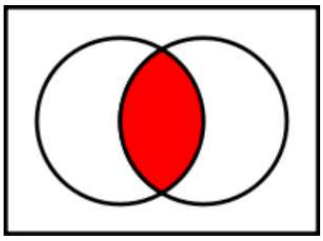
União



A U B (Crédito da imagem: Wikipedia)

```
>>> a = {1, 2, 3, 4}
>>> b = {3, 4, 5, 6}
>>> print a.union(b)
set([1, 2, 3, 4, 5, 6])
```

Interseção



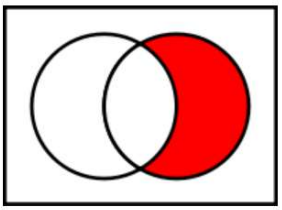
A ∩ B (Crédito da imagem: Wikipedia)

```
>>> print a.intersection(b)
set([3, 4])
```

Essa operação é muito útil quando precisamos descobrir elementos que duas listas possuem em comum:

```
>>> l1 = [1, 2, 3]
>>> l2 = [2, 4, 3]
>>> print set(l1).intersection(l2)
set([2, 3])
```

Diferença



B \ A (Crédito da imagem: Wikipedia)

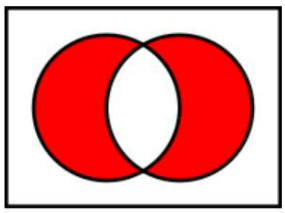
A diferença entre dois conjuntos A e B retorna somente os elementos de A que não estão em B, ou seja, retira de A todos os elementos comuns a ambos os conjuntos:

```
>>> a = {1, 2, 3, 4}
>>> b = {3, 4, 5, 6}
>>> print a.difference(b)
set([1, 2])
>>> print b.difference(a)
set([5, 6])
```

Diferença simétrica

Diferença simétrica é uma operação sobre os dois conjuntos, que retorna todos os elementos (de ambos os conjuntos `a` e `b`) que pertencem a somente um dos conjuntos.

```
>>> a = {1, 2, 3, 4}
>>> b = {3, 4, 5, 6}
>>> print a.symmetric_difference(b)
set([1, 2, 5, 6])
```



A △ B (Crédito da imagem: Wikipedia)

Sintaxe	Descrição
s.difference_update(t)	Remove os itens que estão no conjunto t do conjunto s.
s.intersection_update(t)	Faz com que o conjunto s contenha a interseção dele com t.
s.symetric_difference_update(t)	Faz com que o conjunto s contenha a diferença simétrica dele com t.
s.isdisjoin(t)	Retorna True se s e t não tem nenhum item em comum.
s.issubset(t) (s <= t)	Retorna True se s é igual a ou um subconjunto de t.
s.issuperset(t) (s >= t)	Retorna True se s é igual a ou t é um subconjunto de s.
s.discard(x)	Remove o item x do conjunto s.
s.remove(x)	Remove o item x se ele estiver em s se não retorna um KeyError.
s.update(t)	Adiciona cada item do conjunto s que não está no conjunto t para o conjunto t.

Exercício :

1) Considere um Centro de Treinamento Esportivo que oferece cursos de Futebol, Natação, Vôlei e Basquete. Em função do grande número de desistência de seus alunos, a administração resolveu oferecer um desconto de 50% na segunda modalidade.

   Para isso eles precisarão verificar os alunos em comum (matriculados em mais de um esporte).

   Você foi contratado para desenvolver um sistema que faça essa verificação e retorne as seguintes informações:

   1- Mostre (imprimir na tela) a relação de alunos matriculados por modalidade: Futebol, Basquete, Natação e Vôlei.
   2- Permita que novos alunos possam ser matriculados em qualquer uma das modalidades.
   3- Faça a verificação se existem alunos matriculados em pelo menos 2 modalidades. E indique quais os alunos que possuem direito ao desconto.
   4- Indique o número total de alunos por modalidade, e também o número total de alunos do Centro de Treinamento.

2) Considere que você foi contratado para desenvolver um sistema de integração entre 2 instituições financeiras (devido a uma fusão entre estas instituições). Em função do grande número de clientes, a administração resolveu oferecer algumas vantagens para clientes em comum (um desconto de 70% na tarifa de manutenção de contas correntes por exemplo), na expectativa que novos clientes sejam atraídos com a novidade.

   Para isso eles precisarão verificar quais os clientes em comum (cadastrados tanto na empresa 1 como na empresa2).

   Você foi contratado para desenvolver um sistema que faça essa verificação e retorne as seguintes informações:

   1) Mostre (imprimir na tela) a relação de clientes cadastrados por empresa.
   2) Permita que novos clientes possam ser cadastrados em qualquer uma das modalidades.
   3) Faça a verificação se existem clientes já cadastrados nas 2 empresas. E indique quais são estes clientes.
   4) Indique o número de clientes:

      Da empresa 1
      Da empresa 2
      Clientes já pertencentes a ambas as empresas
      Clientes pertencentes a apenas 1 das empresas
      Número total de clientes