

Python - Adicionar itens de conjunto

Mesmo que um conjunto mantenha juntos apenas objetos imutáveis, o próprio conjunto é mutável. Podemos adicionar novos itens nele de qualquer uma das seguintes maneiras -

Método add()

O método add() na classe set adiciona um novo elemento. Se o elemento já estiver presente no conjunto, não haverá alteração no conjunto.

Sintaxe

```
set.add(obj)
```

Parâmetros

- **obj** - um objeto de qualquer tipo imutável.

Exemplo

Dê uma olhada no exemplo a seguir -

```
lang1 = {"C", "C++", "Java", "Python"}
lang1.add("Golang")
print (lang1)
```

Ele produzirá a seguinte **saída** -

```
{'Python', 'C', 'Golang', 'C++', 'Java'}
```

Método update()

O método update() da classe set inclui os itens do conjunto dados como argumento. Se os elementos do outro conjunto possuírem um ou mais itens já existentes, eles não serão incluídos.

Sintaxe

```
set.update(obj)
```

Parâmetros

- **obj** - um conjunto ou objeto de sequência (lista, tupla, string)

Exemplo

O exemplo a seguir mostra como funciona o método `update()` -

```
lang1 = {"C", "C++", "Java", "Python"}  
lang2 = {"PHP", "C#", "Perl"}  
lang1.update(lang2)  
print (lang1)
```

Ele produzirá a seguinte **saída** -

```
{'Python', 'Java', 'C', 'C#', 'PHP', 'Perl', 'C++'}
```

Exemplo

O método `update()` também aceita qualquer objeto de sequência como argumento. Aqui, uma tupla é o argumento para o método `update()`.

```
lang1 = {"C", "C++", "Java", "Python"}  
lang2 = ("PHP", "C#", "Perl")  
lang1.update(lang2)  
print (lang1)
```

Ele produzirá a seguinte **saída** -

```
{'Java', 'Perl', 'Python', 'C++', 'C#', 'C', 'PHP'}
```

Exemplo

Neste exemplo, um conjunto é construído a partir de uma string e outra string é usada como argumento para o método `update()`.

```
set1 = set("Hello")  
set1.update("World")
```

```
print (set1)
```

Ele produzirá a seguinte **saída** -

```
{'H', 'r', 'o', 'd', 'W', 'l', 'e'}
```

Método união()

O método union() da classe set também combina os itens exclusivos de dois conjuntos, mas retorna um novo objeto de conjunto.

Sintaxe

```
set.union(obj)
```

Parâmetros

- **obj** - um conjunto ou objeto de sequência (lista, tupla, string)

Valor de retorno

O método union() retorna um objeto definido

Exemplo

O exemplo a seguir mostra como funciona o método union() -

```
lang1 = {"C", "C++", "Java", "Python"}  
lang2 = {"PHP", "C#", "Perl"}  
lang3 = lang1.union(lang2)  
print (lang3)
```

Ele produzirá a seguinte **saída** -

```
{'C#', 'Java', 'Perl', 'C++', 'PHP', 'Python', 'C'}
```

Exemplo

Se um objeto de sequência for fornecido como argumento para o método union(), o Python o converte automaticamente em um conjunto primeiro e depois executa a união.

```
lang1 = {"C", "C++", "Java", "Python"}  
lang2 = ["PHP", "C#", "Perl"]  
lang3 = lang1.union(lang2)  
print (lang3)
```

Ele produzirá a seguinte **saída** -

```
{'PHP', 'C#', 'Python', 'C', 'Java', 'C++', 'Perl'}
```

Exemplo

Neste exemplo, um conjunto é construído a partir de uma string e outra string é usada como argumento para o método union().

```
set1 = set("Hello")  
set2 = set1.union("World")  
print (set2)
```

Ele produzirá a seguinte **saída** -

```
{'e', 'H', 'r', 'd', 'W', 'o', 'l'}
```