

Коришћење петље за итерацију преко сета

Пример:	Даје:
<code>moj_set = set(['a', 'b', 'c'])</code>	a
<code>for x in moj_set:</code>	b
<code>print(x)</code>	c

Петља итерира једном за сваки елемент из сета.

Коришћење оператора in и not in за тестирање вредности у сету

```
moj_set = set([1, 2, 3])
if 1 in moj_set:
    print("Element 1 se nalazi u setu.")
if 99 not in moj_set:
    print("Element 99 se ne nalazi u setu.")
```

Добијање уније сетова

Унија два сета је нов сет који садржи све елементе оба сета.

У Пајтону се позива метода union или се користи оператор | за добијање уније два сета.

```
set1.union(set2)      set1 | set2
```

Пример:	Даје:
<code>set1 = set([1, 2, 3])</code>	{1, 2, 3, 4, 5, 6}
<code>set2 = set([10, 20, 30])</code>	{1, 2, 3, 10, 20, 30}
<code>set3 = set([4, 5, 6])</code>	
<code>set4 = set1.union(set3)</code>	
<code>set5 = set2 set1</code>	
<code>print(set4)</code>	
<code>print(set5)</code>	

У првом делу примера позива се метод union објекта set1 који има за аргумент објекат set3.

У другом делу примера користи се бинарни оператор | који над два објекта set2 и set1 реализује операцију уније сетова.

Добијање пресека сетова

Пресек (intersection) два сета је сет који садржи само оне елементе који се налазе у оба сета.

У Пајтону се за добијање пресека може користити метод intersection или оператор &:

```
set1.intersection(set2)      set1 & set2
```

Пример:	Даје:
<code>set1 = set([1, 2, 10])</code>	{10}
<code>set2 = set([10, 20, 5])</code>	set()
<code>set3 = set([4, 5, 6])</code>	
<code>set4 = set1.intersection(set2)</code>	
<code>set5 = set1 & set3</code>	
<code>print(set4)</code>	
<code>print(set5)</code>	

Добијање разлике сетова

Разлика (difference) између објеката set1 и set2 су елементи који се појављују у set1 али се не појављују у set2 објекту.

У Пајтону се за добијање разлике може користити метод difference или оператор -:

```
set1.difference(set2)      set1 - set2
```

Пример:	Даје:
<code>set1 = set([1, 2, 3, 4, 5])</code>	{1, 3, 5}
<code>set2 = set([2, 4, 6, 8])</code>	{2, 4}
<code>set3 = set([6, 7, 8, 9])</code>	
<code>set4 = set1.difference(set2)</code>	
<code>set5 = set2 - set3</code>	
<code>print(set4)</code>	
<code>print(set5)</code>	

Добијање симетричне разлике сетова

Симетрична разлика (symmetric difference) између два сета јесте сет који садржи елементе који нису заједнички за та два сета.

У Пајтону се за добијање симетричне разлике може користити метод `symmetric_difference` или оператор `^`:

```
Пример:
set1 = set([1, 2, 3, 4, 5])
set2 = set([2, 4, 6, 8])
set3 = set([6, 7, 8, 9])
set4 = set1.symmetric_difference(set2)
set5 = set2 ^ set3
print(set4)
print(set5)
```

Даје:

```
{1, 3, 5, 6, 8}
{2, 4, 7, 9}
```

Добијање субсета и суперсета

Ако постоје `set1` и `set2` а `set1` садржи све елементе који припадају `set2`, онда се каже да је `set2` субсет (подсет) од `set1`. Такође, може се рећи да је у том случају `set1` суперсет (надсет) од `set2`.

```
Пример:
set1 = set([1, 2, 3, 4, 5])
set2 = set([2, 4])
set3 = set([6, 7, 8, 9])
print("set1 =", set1)
print("set2 =", set2)
print("set3 =", set3)
print("Da li je set2 podset od set1?", set2.issubset(set1))
print("Da li je set1 superset od set2?", set1.issuperset(set2))
print("Da li je set2 podset od set3?", set2.issubset(set3))
print("Da li je set3 superset od set2?", set3.issuperset(set2))
```

Даје:

```
set1 = {1, 2, 3, 4, 5}
set2 = {2, 4}
set3 = {8, 9, 6, 7}
Da li je set2 podset od set1? True
Da li je set1 superset od set2? True
Da li je set2 podset od set3? False
Da li je set3 superset od set2? False
```