

Креирање сета

Сет је објекат који садржи елементе који се не могу понављати; елементи у сетовима су неуређеног редоследа и могу бити различитих типова података.

Да би се креирао један сет, користи се уграђена функција `set()`: `мој_set = set()`

После реализације овог кода, промењива `мој_set` указује на празан сет.

У загради као аргумент функције `set()` се може појавити један аргумент.

Тај аргумент мора бити објекат који садржи итерабилне елементе, попут листе, торке или стринга.

Појединачни елементи објекта који се придодаје као аргумент постају елементи сета.

Пример: `мој_set = set(['a', 'b', 'c'])`

Овако се придодаје листа у виду аргумента функције `set`.

После ове линије кода, промењива `мој_set` указује на сет који садржи елементе 'a', 'b' и 'c' .

Ако се придода стринг као аргумент функције `set`, сваки појединачни знак у стрингу постаје елемент сета:

`мој_set = set('abc')`

После реализације линије кода, промењива `мој_set` ће указати на сет који се састоји од три елемента: 'a', 'b' и 'c' .

Ако се покуша креирати сет са више истих елемената, само први од истих елемената ће постати елемент сета:

`мој_set = set('aaaabc')` #даје set који се састоји од 'a', 'b' и 'c'

Потребно је креирати сет који се састоји од елемената који су стрингови са једним или више знакова:

`мој_set = set(['jedan', 'dva', 'tri'])` #даје set који се састоји од елемената 'jedan', 'dva', 'tri'

Добијање броја елемената у сету

`мој_set = set([1, 2, 3, 4, 5])`

`print(len(мој_set))` #даје 5.

Додавање и одстрањивање елемената из сета

Сетови су промењиви објекти па је зато допуштено додавати и одстрањивати елементе из сетова.

За додавање се користи метода `add`.

Пример 01:

`мој_set = set()`

`мој_set.add(1)`

`мој_set.add(2)`

`мој_set.add(3)`

`print(мој_set)`

`мој_set.add(2)`

`print(мој_set)`

Даје:

`{1, 2, 3}`

`{1, 2, 3}`

Види се да после покушаја додавања истог елемента у сет (број 2) тај елемент није додат и није дошло до појаве грешке, већ се игнорише метода `add`.

Коришћењем методе `update`, може се истовремено додати група елемената у сет:

Пример 02:

`set1 = set([1, 2, 3])`

`set2 = set([8, 9, 10])`

`set1.update(set2)`

`print(set1)`

`print(set2)`

Даје:

`{1, 2, 3, 8, 9, 10}`

`{8, 9, 10}`

Пример 03:

`мој_set = set([1, 2, 3])`

`мој_set.update('abc')`

`print(мој_set)`

Даје:

`{1, 2, 3, 'a', 'b', 'c'}`

Да би се одстранио елемент из сета може се користити или метода `remove` или метода `discard`.

Разлика је у томе како се методе понашају ако се затражи одстрањивање елемента који не постоји у сету: метода `remove` подиже `KeyError` грешку док метода `discard` не подиже грешку.

Пример 04:

```
moj_set = set([1, 2, 3, 4, 5])
```

```
print(moj_set)
```

```
moj_set.remove(1)
```

```
print(moj_set)
```

```
moj_set.discard(5)
```

```
print(moj_set)
```

```
moj_set.discard(99)
```

```
moj_set.remove(99)
```

```
{1, 2, 3, 4, 5}
```

```
{2, 3, 4, 5}
```

```
{2, 3, 4}
```

```
Traceback (most recent call last):
```

```
File "C:\Users\nera\Desktop\proba\PythonApplication2\PythonApplication2\PythonApplication2.py", line 12, in <module>
```

```
    moj_set.remove(99)
```

```
KeyError: 99
```

Сви елементи у сету се могу одстранити позивањем методе `clear`.

Пример:

```
moj_set = set([1, 2, 3, 4, 5])
```

```
print(moj_set)
```

```
moj_set.clear()
```

```
print(moj_set)
```

Даје:

```
{1, 2, 3, 4, 5}
```

```
set() #prazan set
```