

DICȚIONARE ȘI SETURI

BY SANJAY AND ARVIND SESHAN

OBIECTIVELE LECȚIEI

- Învățăm să cream dicționare și seturi

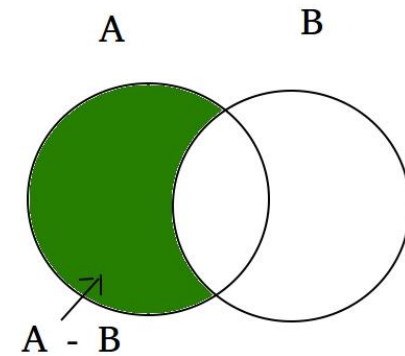
SETURILE

- Similarcu listele
- Stochează un set de elemente
- Toate elementele sunt unice și neordonate
 - Poți plasa un element în fiecare set
 - Nu este nicio ordine în set (chiar dacă introduci elementele într-o anumită ordine)
- Seterile sunt elemente în {a, b, ...} paranteze
- Poți adăuga unui set utilizând metoda add

```
>> s1 = set() # do not use {} to
initialize empty set
>> s1.add(5) # add to a set
>> s1.add(2)
>> print(s1)
{2, 5}
>> s2 = {1, 2, 4, 4, "hello"} #
define set
>> print(s2) # note only one 4 is
below
{1, 2, 4, 'hello'}
```

MAI MULTE DESPRE SETURI

- Poți găsi diferențele, intersecția, uniunea etc, dintre seturi
- Dacă încerci să adaugi o listă la un set , sau orice tîă mutabil , programul va bloca
- În general, este mult mai rapid să execuți lookups pe un set decât pe o listă datorită a ceva numit hashing



```
>> print(s2.difference(s1))
{'hello'}
>> print(s2.intersection(s1))
{1, 2, 4}
>> s2.add([4,1])
TypeError: unhashable type: 'list'
```

METODE SET

Method	Description
<u>add()</u>	Adăugă un element la un set
<u>clear()</u>	Scoate toate elementele din set
<u>copy()</u>	Returnează o copie a seturilor
<u>difference()</u>	Returnează un set conținând diferențele dintre 2 sau mai multe seturi
<u>difference_update()</u>	Scoate toate elementele din set care sunt incluse în alt set, set specific
<u>discard()</u>	Scoate elementele specificate
<u>intersection()</u>	Returnează un set, care este intersecția altor 2 seturi
<u>intersection_update()</u>	Scoate toate elementele din set care nu sunt prezente în altul, set specific

Cele evidențiate sunt cele mai importante

SET METHODS CONT.

<code>isdisjoint()</code>	Returnează dacă 2 seturi au o intersecție sau nu
<code>issubset()</code>	Returnează dacă alt set conține acest set sau nu
<code>issuperset()</code>	Returnează dacă setul conține alt set sau nu
<code>pop()</code>	Scoate un element din set
<code>remove()</code>	Scoate un element specific
<code>symmetric_difference()</code>	Returnează un set cu diferențe simetrice a două seturi
<code>symmetric_difference_update()</code>	Inserează diferențele simetrice din acest set și din altul
<code>union()</code>	Returnează un set care conține o uniune de seturi
<code>update()</code>	Updatează setul cu uniunea setului și a altora

COPIEREA SETURILOR

- La fel ca la listele `l`, utilizează metoda `copy`

```
s1 = {1, 2, 3}
```

```
s2 = s1.copy()
```

- Seturile sunt mutable, la fel ca listele, așa că e nevoie să fii atent când faci ceva de genul
`s1=s2`

DICTIONARELE

- Gandește-te la un dicționar de Engleză
 - Grupează elementele pe o definiție
- Definește utilizând {} linii și coloane
 - Forma pentru fiecare element este `item:definition` → numit tipic `key:value`
 - `d = {"hello":"a greeting", "red":"a color"}`
- Elementele cheie(e.g. "hello") trebuie să fie unică, dar multe chei pot avea aceeași definiție
- Cheile pot fi tipuri de date immutable (e.g. int, str)
- Valorile/definițiile pot fi orice (e.g. int, list, None)
- Utilizează `d2 = d.copy()` pentru a copia dicționarul (dicts are mutable)

METODELE DICȚIONAR

Cele evidențiate sunt cel mai importante

Method	Description
<code>clear()</code>	Elimină toate elementele din dicționar
<code>copy()</code>	Returnează o copie a unui dicționar
<code>fromkeys()</code>	Returnează un dicționar cu chei specifice și valori
<code>get()</code>	Returns the value of the specified key
<code>items()</code>	Returnează o listă ce conține tuple pentru fiecare pereche de valori cheie
<code>keys()</code>	Returnează o listă ce conține chei dicționar
<code>pop()</code>	Elimină elementele cu cheia specificată
<code>popitem()</code>	Elimină ultima pereche inserată de valori cheie
<code>setdefault()</code>	Returnează valorile cu cheia specificată. Dacă cheia nu există: inserează cheia, cu valoarea specifică
<code>update()</code>	Updatează dicționarul cu perechi de valori cheie
<code>values()</code>	Returnează o listă a tuturor valorilor în dicționar

OBȚINEREA UNEI VALORI

Sets:

```
s = {1, 2, 3}
3 in s # True
"hi" in s # False
```

Dicts:

```
d = {"a":1, "b":2}
d["a"] == 1 # True
d["b"] == 2 # True
```

- Se adresează unei liste → utilizează `[]` paranteze alături de variabilele ce conțin o cheie pentru a primi “value”

PROVOCAREA

- Tradu câteva cuvinte din Spaniolă în Engleză și printează rezultatele.
- Hola → hello
- Rojo → red
- Naranja → orange
- Verde → green

SOLUȚIA PROVOCĂRII

```
d = {"hola":"hello", "rojo":"red", "naranja":"orange", "verde":"green"}
```

```
data = "hola"
```

```
print(d[data]) # hello
```

```
data = "naranja"
```

```
print(d[data]) # orange
```

SĂ REVEDEM

- List → stochează valori [1, 2, 2, 3, "hello"] (mutable)
- Tuple → stochează valori (1, 2, 2, 3, "hello") (immutable)
- Set → stochează valori unice {1, 2, 3, "hello"} (mutable, but elements must be immutable)
- Dictionary → stochează valori care pot fi indexate cu o cheie {1:"a", 2:"b"} (mutable, but keys must be immutable)

CREDITS

- Această lecție de SPIKE Prime a fost realizată de Sanjay Seshan și Arvind Seshan.
- Mai multe lecții sunt disponibile pe www.primelessons.org
- Această lecție a fost tradusă în limba română de echipa de robotică FTC – ROSOPHIA #21455 RO20



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).