

Institut Préparatoire aux Études d'Ingénieurs de Tunis



Leila Ben Othman

Première année de classes préparatoires
MP-PC

Année universitaire: 2016 - 2017

Les dictionnaires et les ensembles en PYTHON

Plan

① Les ensembles

② Les dictionnaires: tableau associatif

Les ensembles

- Un ensemble est une collection d'objets **distincts** et de types **hétérogènes**.
- Un ensemble est une structure **non ordonnée**: il n'est pas possible d'utiliser un indice (rang) ni de sélectionner un sous-ensemble.
- Un ensemble en PYTHON est un objet **mutable**.

Comment initialiser/créer un ensemble?

- Initialisation: Il n'existe pas de syntaxe particulière pour les ensembles comme il en existe par exemple pour les listes et les tuples (crochets vides ou parenthèses vides). Les accolades vides sont relatives aux dictionnaires.
- La seule méthode possible consiste à utiliser la méthode `set()` pour l'initialisation.
- La méthode `set(iterable)` permet de créer un ensemble avec comme éléments, les éléments de l'itérable.

Comment initialiser/créer un ensemble?

```
In [1]: E=set(); E# ensemble vide  
Out [1]: set()
```

```
In [2]: type(E);  
Out [2]: <class 'set'>
```

```
In [3]: E={}; type(E)# Attention, ceci n'est pas un ensemble vide  
Out [3]: <class 'dict'>
```

```
In [4]: E={4,7,8};E  
Out [4]: {4,7,8}
```

```
In [5]: E={'Bonjour', 56, 56};E # éléments distincts  
Out [5]: {'Bonjour', 56}
```

Comment initialiser/créer un ensemble?

```
In [6]: E=set('Bonjour'); E # L'itérable est une chaîne
Out [6]: {'B','j','o','n','r','u'}
```

```
In [7]: E=set([1,4,8,8]);E # L'itérable est une liste
Out [7]: {1,4,8}
```

```
In [8]: E=set((4,7,8,8));E # L'itérable est un tuple
Out [8]: {4,7,8}
```

```
In [9]: E=set(range(11));E # L'itérable est un range
Out [9]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In [10]: E=set((i,i) for i in range(11) if i%2==0);E # par compréhension
Out [10]: {(0,0), (6,6), (4,4), (10,10), (8,8), (2,2)}
```

Opérations sur les ensembles

```
In [1]: E=set(range(11))
```

```
In [2]: len(E);
```

```
Out [2]: 11
```

```
In [3]: 5 in E
```

```
In [3]: True
```

```
In [4]: 6 not in E
```

```
Out [4]: False
```

```
In [5]: for i in E:
```

```
    print(i)
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

Opérations sur les ensembles

Intersection - Union

In [1]: $\{1,2,3\} \mid \{4,5,6\}$ # le symbole $\mid$ pour l'union

Out[1]: $\{1,2,3,4,5,6\}$

In [2]: $\{1,2,3\}.union(\{4,5,6\})$ # la méthode union

Out [2]: $\{1,2,3,4,5,6\}$

In [3]: $\{4,5\} \& \{4,5,6\}$ # le symbole $\&$ pour l'intersection

Out[3]: $\{4,5\}$

In [4]: $\{4,5\}.intersection(\{4,5,6\})$ # la méthode intersection

Out [4]: $\{4,5\}$

Opérations sur les ensembles

Différence - Différence symétrique

In [1]: $\{4, 5, 6, 7, 8\} - \{4, 5, 6, 10\}$ # le symbole - pour la différence
 Out[1]: $\{8, 7\}$

In [2]: $\{4,5,6,7,8\}.difference(\{4, 5, 6, 10\})$ # la méthode difference
 Out [2]: $\{8, 7\}$

In [3]: $\{4,5,6,7,8\} \wedge \{4, 5, 6, 10\}$ # le symbole $\wedge$ pour pour la difference symétrique
 Out [3]: $\{7,8,10\}$

In [4]: $\{4,5,6,7,8\}.symmetric_difference(\{4, 5, 6, 10\})$ # la méthode symmetric_difference
 Out [4]: $\{7,8,10\}$

Opérations sur les ensembles

```
In [1]: {4, 5}.issubset({4, 5, 6, 10}) # teste si un ensemble est un  
sous-ensemble d'un autre ensemble
```

```
Out[1]: True
```

```
In [2]: {4,5,6,10}.issuperset({4, 5}) # teste si un ensemble est un  
super-ensemble d'un autre ensemble
```

```
Out [2]: True
```

```
In [3]: {4,5,6}.isdisjoint({4, 5}) # teste si deux ensembles sont disjoints
```

```
Out [3]: False
```

Opérations sur les ensembles

Ajout - suppression

```
In [1]: E={1, 2, 3}
```

```
In [2]: E.add(4);E
```

```
Out [2]: {1,2,3,4}
```

```
In [3]: E.remove(3);E
```

```
Out [3]: {1,2,4}
```

Les dictionnaires

- Les dictionnaires sont le seul type associatif en PYTHON. Ces objets associatifs permettent de réaliser une correspondance entre valeurs de hashage (clés) et objets associés (valeurs). On parle de couples <clef, valeur>
- Le dictionnaire n'est pas une structure ordonnée $\Rightarrow$ Pas de position (rang) d'une valeur dans le dictionnaire. Par contre, l'accès à une valeur se fait à travers la clef.
- Un dictionnaire est un objet **mutable**

Comment initialiser/créer un dictionnaire

```
In [1]: D=dict(); D # dictionnaire vide  
In [1]: {}
```

```
In [2]: type(D)  
Out [2]: <class 'dict'>
```

```
In [3]: D={}; type(D) # dictionnaire vide  
In [3]: <class 'dict'>
```

```
In [4]: D={1:'A',2:'B',3:'C',4:'D',5:'E'}; D # ensemble de clef:valeur  
Out [4]: {1:'A',2:'B',3:'C',4:'D',5:'E'}
```

```
In [5]: D=dict((i,i) for i in range(1,6)) # par compréhension  
Out [5]: {1:1,2:2,2:2,4:4,5:5}
```

Opérations sur les dictionnaires

```
In [1]: D={1:'A',2:'B',3:'C',4:'D',5:'E'};D
```

```
Out [1]: {1:'A',2:'B',3:'C',4:'D',5:'E'}
```

```
In [2]: len(D)
```

```
In [2]: 5
```

```
In [3]: 5 in D #Attention la recherche se fait par clef
```

```
Out [3]: True
```

```
In [4]: 'A' in D
```

```
Out [4]: False
```

```
In [5]: D.get(1) # ou bien D[1], attention 1 est la clef et non pas la  
position
```

```
Out [5]: 'A'
```

Les méthodes `keys()`, `values()` et `items()`

```
In [1]: D.keys()
```

```
Out [1]: dict_keys([1,2,3,4,5])
```

```
In [2]: D.values()
```

```
In [2]: dict_values(['A','B','C','D','E'])
```

```
In [3]: D.items()
```

```
Out [3]: dict_items([(1,'A'),(2,'B'),(3,'C'),(4,'D'),(5,'E')])
```

Les méthodes keys(), values() et items()

```
for i in D.keys():  
    print(i)
```

1
2
3
4
5

```
for i in D.values():  
    print(i)
```

A
B
C
D
E

```
for i in D.items():  
    print(i)
```

(1, 'A')
(2, 'B')
(3, 'C')
(4, 'D')
(5, 'E')

```
for i,j in D.items():  
    print(i,j)
```

1 A
2 B
3 C
4 D
5 E

Les dictionnaires

Ajout/Modification/suppression

In [1]:

```
D={'MP6-1':'Ala','MP6-2':'Azhar','MP6-3':'Youssef','MP6-4':'Aymen',  
'MP6-5':'Iheb','MP6-6':'Khaoula','MP6-7':'Nesrine','MP6-8':'Ali',  
'MP6-9':'Haykel','MP6-10':'Fatma'}
```

In [2]: D['MP6-11']='Leila'; D # si la clef n'existe pas, la paire va être ajoutée

```
Out [2]:{'MP6-1':'Ala','MP6-2':'Azhar','MP6-3':'Youssef',  
'MP6-4':'Aymen','MP6-5':'Iheb','MP6-6':'Khaoula','MP6-  
7':'Nesrine','MP6-8':'Ali',  
'MP6-9':'Haykel','MP6-10':'Fatma','MP6-11':'Leila'}
```

Ajout/Modification/suppression

In [3]: D['MP6-10']='Fattouma'; D # si la clef existe, la valeur va être modifiée

Out [3]:{'MP6-1':'Ala','MP6-2':'Azhar','MP6-3':'Youssef',
'MP6-4':'Aymen','MP6-5':'Iheb','MP6-6':'Khaoula','MP6-7':'Nesrine',
'MP6-8':'Ali',
'MP6-9':'Haykel','MP6-10':'Fattouma','MP6-11':'Leila'}

In [4]: D.pop('MP6-11');D # suppression de la valeur dont la clef est 'MP6-11'

Out [4]:{'MP6-1':'Ala','MP6-2':'Azhar','MP6-3':'Youssef',
'MP6-4':'Aymen','MP6-5':'Iheb','MP6-6':'Khaoula','MP6-7':'Nesrine',
'MP6-8':'Ali',
'MP6-9':'Haykel','MP6-10':'Fattouma'}