Programmation orientée objet avec Python

Mars 2021

académie Versailles
GRETA
DES HAUTS-DE-SEINE
FORMEZ-VOUS À DEMAIN
RÉGION ACADÉMIQUE ÎLE-DE-FRANCE
LUDIKSCIENCES

Tuple

En français, n-uplet, généralisation de couple, triplet, quadruplet – par définition, un élement d'un produit cartésien – par nature, donc : ordonné. Un couple est formellement différent d'une paire (ensemble à 2 éléments, non ordonné).

En Python, non modifiable

Noté entre ( )

Virgule , séparatrice

Hétérogène

LUDIKSCIENCES

Opérateurs sur les séquences

appartenance	in
concaténation	+
répétition	*
slicing	s[start: stop:step]
longueur	len(s)
Plus petit	min(s)
Plus grand	max(s)
Recherche	s.index(x)
Compte	s.count(x)

LUDIKSCIENCES

Slicing

S'applique à toutes les séquences

Slice : Tranche

Exemples de tranches d'une séquence s

- s[0] : le 1er élément, index positif, numérotation à partir du début, au numéro 0
- s[-1] : le dernier, index négatif, numérotation à partir de la fin, au numéro len(s) -1
- s[2:4] : la sous chaîne du 2ème élément inclus au 4ème élément exclus
- s[:4] : sous-chaîne du début au 4ème élément exclus
- s[2:] : sous chaîne allant du 2ème élément compris jusqu'à la fin

Forme générale s [start:stop:step]

Tranches particulières

- s[::-1] retourne une séquence
- s[::3] prend un élément sur 3

LUDIKSCIENCES

LUDIKSCIENCES

Exercice Slicing

A partir des deux objets suivants :

texte = 'anticonstitutionnellement'

chiffre = list(range(10)) # Q1 : A quoi correspond chiffre ? Quel est son type ?

Q2 : Trouver les slices qui permettent d'obtenir les résultats suivants :

- [0,1,2,3,4,5]
- 'anti'
- 'constitution'
- [0,2,4,6,8]
- 'tnemellennoitutitsnocitna'
- [3,7]
- [9, 7, 5, 3, 1]

LUDIKSCIENCES

Range

Crée une liste de nombres entiers en progression arithmétique

range(stop)

- les nombres de 0 à stop-1

range(start, stop [,step])

- les nombres de start à stop-1 par pas de step

Python3 optimise

- une liste n'est vraiment créée que si on a besoin de tous les éléments à la fois.
- sinon, ils sont créés un à un, au fil et à mesure des besoins
- itérateurs et générateurs

```
range(20)
```

```
list(range(20))
```

LUDIKSCIENCES

Références

Les variables qui référencent les mêmes objets modifiables subissent les mêmes modifications.

Affectation : référence au même objet

Copie : référence à une copie

```
import copy
y = copy.copy(x)
z = copy.deepcopy(x)
```

LUDIKSCIENCES

Set

Ensemble

non redondant, pas de doublons

non ordonné

itérable

modifiable

- mais frozenset est non modifiable

Littéraux

- { élément, … }
- set( )

Opérateur

- appartenance in
- union |
- intersection &
- différence –
- différence symétrique ^

Méthodes

- inclusion .issubset(), issuperset()

```
ens_vide = set()

couleurs = {'Rouge','Vert','Bleu'}
```

LUDIKSCIENCES

Dict

Dictionnaire

Ensemble plutôt que séquence car il n'est pas ordonné. Mais il est modifiable

Noté {clé : valeur, … }

- La clé doit être hachable
- La valeur peut être quelconque

```
dict_vide = {}

notes = {Pierre':18,'Paul':12,'Marie':20}
```

Opérateur

- appartenance in
- del

Méthodes

- .keys() regarder quel est le type renvoyé (set ou list)
- .values()
- .items() liste de couples clé-valeur

sorted

Classeur Jupyter Dictionnaire.ipynb

LUDIKSCIENCES

Del

Effacer un élément de liste

```
del l[3]
```

Effacer un élément de dictionnaire

```
del note['Pierre']
```

Effacer une variable locale

```
del x
```

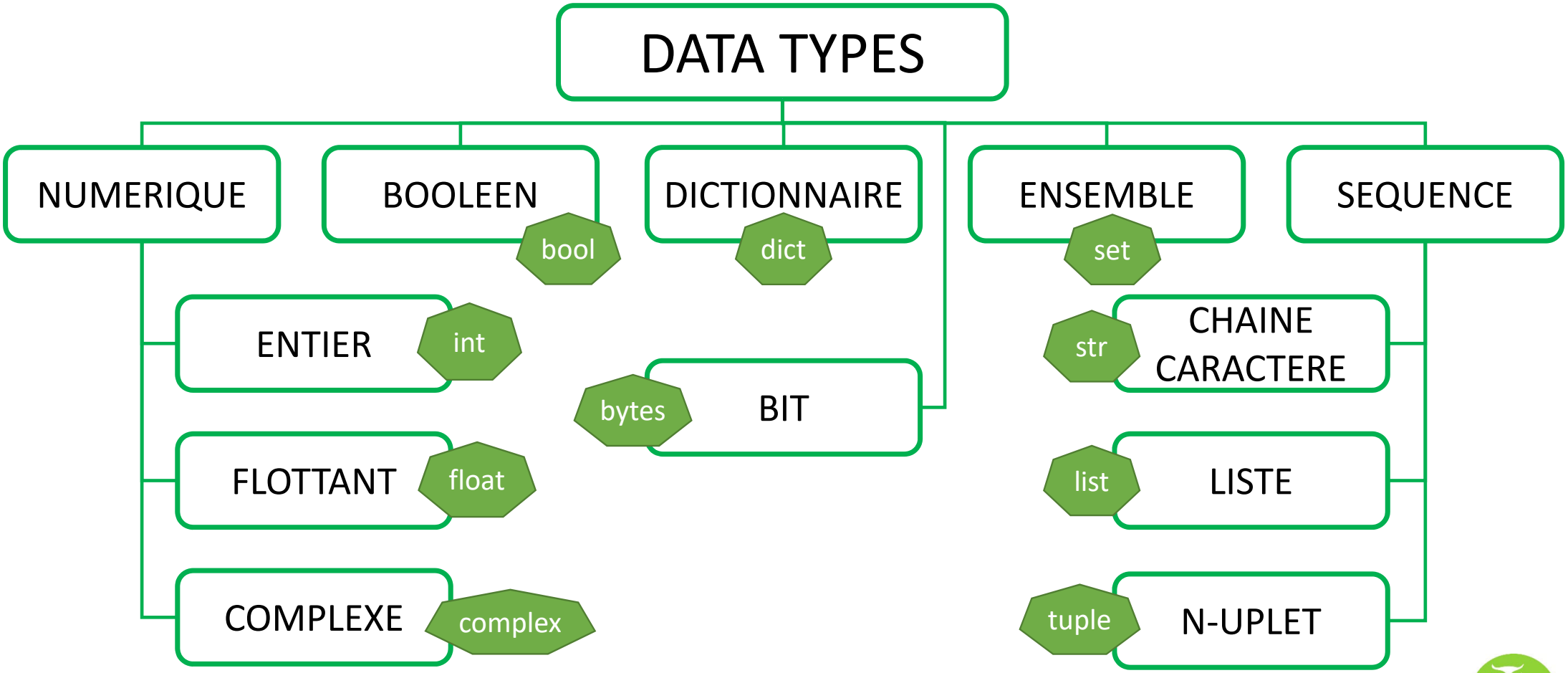
Après un del, la variable devient non définie

del est différent de

```
x = None
```

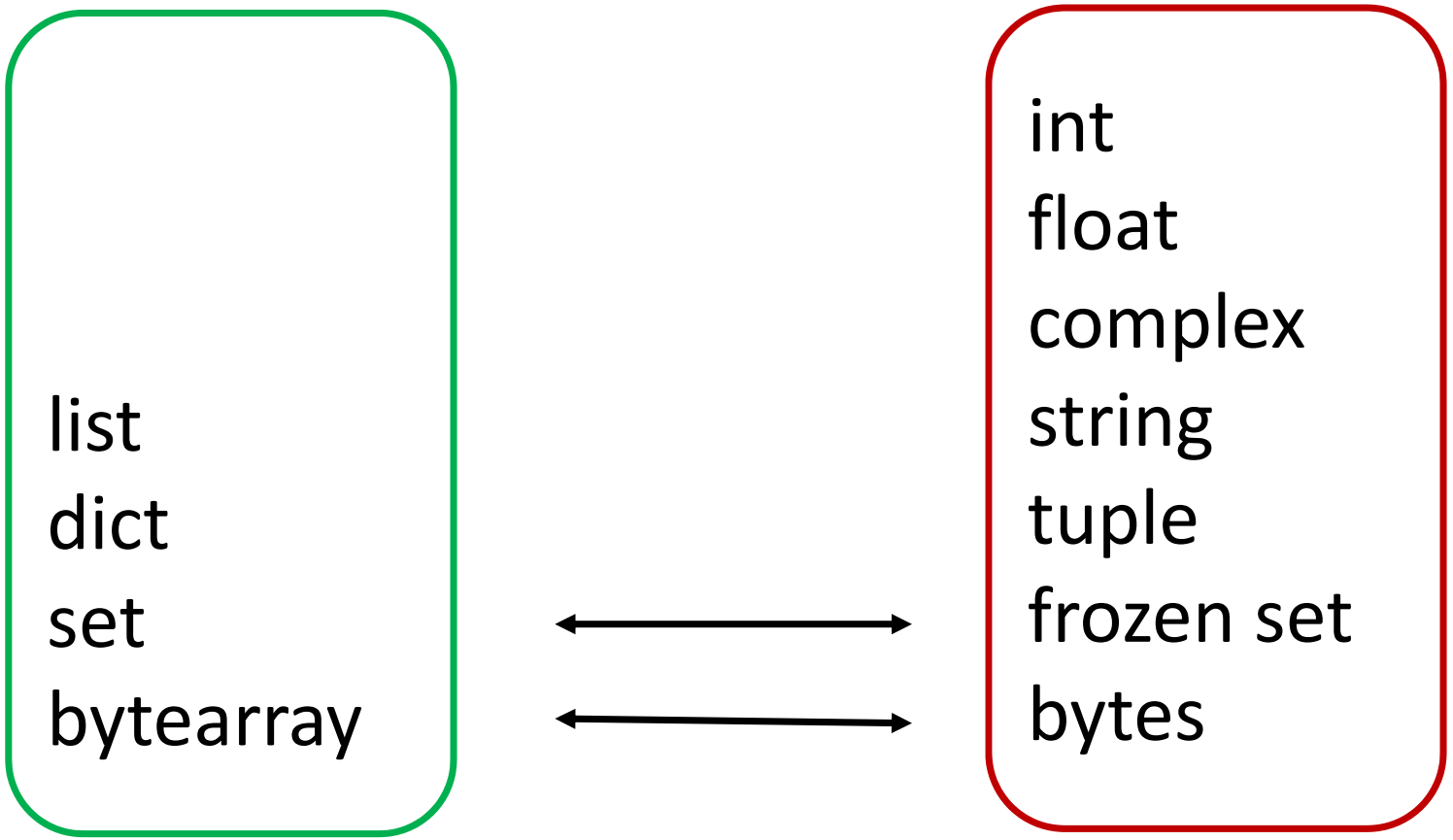
LUDIKSCIENCES

Type de données



LUDIKSCIENCES

Synthèse Classes Mutables



LUDIKSCIENCES

None

Une constante prédéfinie qui signifie *l'absence de valeur* pour une variable qui est définie

La variable pointe alors vers un objet None du type NoneType (et c'est le seul de ce type)

Lorsqu'une fonction ne renvoie pas de résultat, elle renvoie None

Utile pour les arguments optionnels à des fonctions

- On compare l'argument qui manque par is

```
if arg is None :
```

LUDIKSCIENCES

While

Boucle très générale, proche de ce qui se fait dans d'autres langages de programmation

```
while condition:
    action
```

"Tant que"

La boucle "Répéter jusqu'à" n'existe pas en Python (do… while)

Une clause else ajoute une action qui s'exécute quand on sort de la boucle.

```
while condition:
    action
else:
    action_finale
```

- savoir que cela existe
- l'action finale est toujours exécutée
- sauf en cas de "break" voir plus loin

LUDIKSCIENCES

For

Parcourir un intervalle de nombres entiers

```
for i in range (a, b, step):
    action
```

mais l'on peut parcourir d'autres itérables, par ex: une liste

```
for x in ['Riri', 2, 3.14]:
    action
```

ou une chaîne

```
for c in "Nabuchodonosor":
```

ou un ensemble

```
for c in { 'R', 'G', 'B', 'Y', M', 'C' }:
```

Pour parcourir un dictionnaire, on boucle sur ses éléments constitutifs

```
for x in dict.keys():

for x in dict.values():

for k,v in dict.items():
```

Peut prendre une clause else, exécutée en fin de boucle

LUDIKSCIENCES

Compréhension

Maths

Définir un ensemble *en extension* : citer tous ses éléments

Définir un ensemble *en compréhension* :

à l'aide d'autres ensembles,

par ex "sous ensemble de"

en caractérisant les éléments par une propriété.

```
l = [ n**2 for n in primes] # liste
```

```
noms_propres = {mot for mot in dico \
if mot[0].isupper()} # ensemble
```

Python:

à l'aide de for et if

S'applique aux listes, ensembles, dictionnaires – tout objet itérable

```
from random import randint
N1=1000
N2=9999
pseudo = {nom:randint(N1,N2) for \
nom in noms_propres} # dictionnaire
```

LUDIKSCIENCES

Ruptures

Sortir de la boucle et passer à la suite avec break

En cas de break, l'action finale introduite par else: n'est pas exécutée

```
for i in range (a, b, step):
    action
    if condition:
        break
    autre action
suite
```

Rester dans la boucle mais passer à l'itération suivante avec continue

```
while condition1:
    action
    if condition2:
        continue
    autre action
suite
```

LUDIKSCIENCES

Sort & Sorted

Si l est une liste

l.sort() réarrange le contenu de l sur place

sorted(l) crée une nouvelle liste triée

LUDIKSCIENCES

Enumerate

S'applique à tout objet itérable

affecte un numéro à chaque élément parcouru

- donner le numéro de départ

renvoie un tuple

- numéro
- élément

Classeur Jupyter Enumerate.ipynb

LUDIKSCIENCES

Exercice Recherche dans une chaîne

Rechercher si la sous-chaîne "pattern" apparaît dans au moins l'un des mots de la liste "words"

```
def match(words, pattern):
        pass
```

LUDIKSCIENCES

Exercice Recherche dans une chaîne

Correction :

```
def match(words, pattern) :
    for word in words:
        if pattern in word :
            return True
    return False

uneliste = ['Riri', 'Fifi', 'Loulou']
unpattern = 'fi'

print(match(uneliste,unpattern))

unpattern = ‘Zou'
print(match(uneliste,unpattern))
```

LUDIKSCIENCES