

1 – Introduction à Python

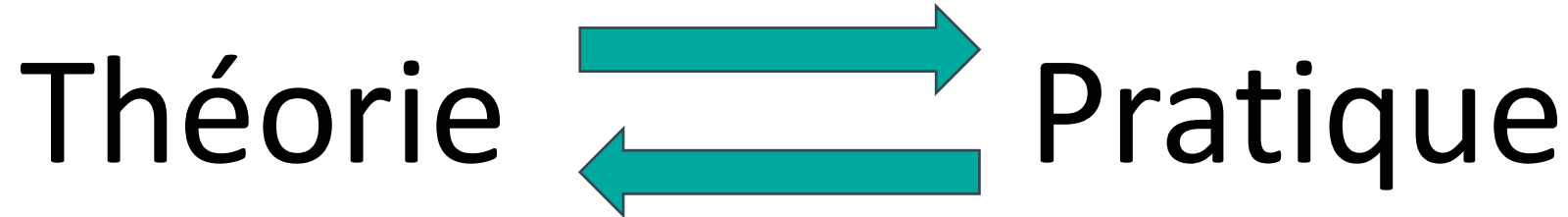


Quelques mots avant de commencer... (1/2)

- Ce langage de programmation a été créé par Guido van Rossum et ce langage porte ce nom en hommage de la troupe des « Monty Python ».
- Python est un langage **interprété** et jouit donc d'une grande portabilité.
- Deux versions majeures de Python existent : la 2 et la 3. Un programme étant codé en version 2 **peut ne pas marcher en 3.**

Quelques mots avant de commencer... (2/2)

Un cycle entre théorie et pratique : « *c'est en forgeant que l'on devient forgeront !* »



Ce cours part du principe que
vous ayez déjà des
connaissances en Java !

Téléchargeons Python

Se rendre ici : <https://www.python.org/downloads/>

En téléchargeant la version 3.7.X et en veillant à bien sélectionner votre système d'exploitation !

Comment savoir si cela a bien fonctionné ?

- Si vous êtes sous Windows, dirigez-vous vers votre invite de commande et tapez 'python'
- Dans le cas où vous êtes sous Mac ou Linux, lancez votre terminal et tapez également 'python'
- Quand cela fonctionne, vous devriez obtenir quelque chose de similaire à ceci

```
Python 3.6.5 |Anaconda, Inc.| (default, Apr 26 2018, 08:42:37)  
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

L'indétrônable 'Hello World!'

```
print( 'Hello World! ' )
```

Cela affichera sur votre écran la phrase 'Hello World!'.

1 – Les variables

Amusons-nous avec des variables

```
age = 24  
name = 'Grimm'
```

Nous créons 2 variables SANS spécifier leur type. Python le fait à notre place.

```
print('My name is', name, 'and I am', age, 'years old')
```

On affiche le tout.

Besoin d'un scanner pour lire le clavier ?

```
age = int(input('How old are you ? '))
```

On demande à l'utilisateur son âge et on **cast** directement sa réponse en **int**.

```
print('You are {} years old'.format(age))
```

Nouvelle manière d'afficher et d'inclure une variable.

Code complet

```
age = 24
name = 'Grimm'

print(age)
print(name)

print('My name is', name, 'and I am', age, 'years old')

age = int(input('How old are you ? '))

print('You are {} years old'.format(age))
```

Développons : un logiciel bien curieux

Créez un logiciel qui :

- Demande le nom de l'utilisateur
- Et l'affiche à l'écran

2 – Les alternatives

Avoir un commentaire sur ses points ?

```
points = int(input('Type your points '))

if points == 20:
    print('Perfect!')
elif points >= 10:
    print('You have to study more!')
elif points < 10:
    print('Sorry... This is a failure!')
else:
    print('Your points are so weird!')
```

Développons : le jeu du chiffre mystère

Écrire un programme qui :

- Engendre un chiffre aléatoire entre 1 et 10 (voir sur google) ;
- Demande un chiffre à l'utilisateur ;
- Compare le chiffre mystère et le chiffre de l'utilisateur ;
 - S'il le devine, afficher des félicitations ;
 - Sinon, lui dire qu'il s'est trompé.

3 – Les itératives

La boucle « for »

```
number = 10
```

```
for i in range(number):  
    print(i)
```

```
# range can help you to loop through a number  
# it will print from 0 to 9
```

La boucle « while »

```
i = 0
while i < 10:
    print(i)
    i += 1
```

Développons : épelons un prénom

Écrire un programme qui permet d'épeler un prénom.

Vous demanderez donc à l'utilisateur d'encoder préalablement son prénom.

4 – Les tableaux

Comment en créer ?

```
array = [5, 1, 4, 6, 4]
```

Comment accéder aux éléments ?

```
# access to the third element  
print(array[2])
```

```
# display all  
for num in array:  
    print(num)
```

Comment ajouter un élément ?

```
array.append(7)
```

Comment supprimer un élément ?

Via son index

```
del array[2]
```

```
array.pop(1)
```

Via son contenu

```
array.remove(4)
```


Développons : créer et filtrer un tableau

Initialiser un tableau de 10 éléments. Celui-ci contiendra, au départ, que des nombres aléatoires compris entre 1 et 10.

Ensuite, vous devez développer un algorithme qui garde uniquement les nombres pairs.

Enfin, incrémentez de 1 toutes les valeurs restantes.

Le slicing (1/3)

| | | | | | | | | | | | | |
|----------|----------|----------|-----|----------|----------|----------|----------|----------|----|----------|----------|----------|
| H | i | ! | | S | I | i | c | e | | m | e | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

`tableau[index_depart : index_fin]`

Comment récupérer « me » ?

`tableau[10:12]`

Le slicing (2/3)

| | | | | | | | | | | | | |
|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| H | i | ! | | S | l | i | c | e | | m | e | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

Comment récupérer « Hi » ?

tableau[0:2]

Comment récupérer « Slice » ?

tableau[4:9]

Comment récupérer **facilement** « ! » ?

tableau[-1]

Comment récupérer « me » via la numérotation inversée ?

tableau[-3:-1]

Le slicing (3/3)

| H | i | ! | | S | I | i | c | e | | m | e | ! |
|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

Comment récupérer « Slice me! » ?

tableau[4:]

Comment récupérer « me! » ?

tableau[10:]

5 – Les fonctions

Comment en créer une ? (1/2)

Supposons que l'on souhaite créer une fonction qui affiche tous les éléments d'un tableau :

```
def print_array(array):
```

Commencer par
def

Nom de la
fonction

Les arguments

On termine par
« : »

Comment en créer une ? (1/2)

Supposons que l'on souhaite créer une fonction qui affiche tous les éléments d'un tableau :

```
def print_array(array):  
    for num in array:  
        print(num)
```

Comment les utiliser ?

En reprenant ce que l'on a fait :

```
print_array(array)
```


6 – Les classes

Comment créer une classe ?

Nous allons créer une classe « Person ». Grâce à elle, nous pourrions connaître le nom d'une personne, le nombre de personnes qui ont été créées et les afficher.

```
class Person:  
    pass
```

Les attributs et le constructeur

```
class Person:  
    count = 0  
  
    def __init__(self, name):  
        self._name = name  
        Person.count += 1
```

Getter et setter

```
@property
def name(self):
    return self._name

@name.setter
def name(self, value):
    self._name = value
```

Comment afficher la représentation d'un objet ?

```
def __str__(self):  
    return 'Hello ! I am {}'.format(self.name)
```

Comment savoir si un objet est égal à un autre ?

```
def __eq__(self, obj):  
    return self.name == obj.name
```

Création d'une nouvelle instance

```
p1 = Person('Cyrus')
p2 = Person('Primrose')

# is p1 is ... p1 ?
print(p1 == p1)
# is p2 is equals to p2 ?
print(p1 == p2)
# we print p1
print(p1)
```

Comment faire de l'héritage ?

```
class Warrior(Person):  
    def __init__(self, name, has_a_sword):  
        Person.__init__(self, name)  
        self._has_a_sword = has_a_sword  
  
    @property  
    def has_a_sword(self):  
        return self._has_a_sword  
  
    @has_a_sword.setter  
    def has_a_sword(self, value):  
        self._has_a_sword = value
```


Comment appeler une méthode du parent ?

```
def __str__(self):  
    return super().__str__() + ' {}'.format(self.has_a_sword)
```

Iconographie

- Python - Image du langage Python lors de l'introduction. Téléchargée sur <https://www.python.org/>
- Chanut is Industries - Image signifiant le moment de théorie dans le cours. Téléchargée sur https://www.iconfinder.com/icons/3533404/development_education_improvement_knowledge_learning_potential_self_icon
- First Styles - Image signifiant le moment de pratique dans le cours. Téléchargée sur [https://www.iconfinder.com/icons/2135927/monitor_television_tv_work_i
con](https://www.iconfinder.com/icons/2135927/monitor_television_tv_work_icon)