

CPBx 2016-2017

BIOINFORMATICS & PYTHON PROGRAMMING

VARIABLES & TYPES

JEAN-CHRISTOPHE TAVEAU & PATRICIA THÉBAULT

UN LANGAGE DE PROGRAMMATION: LES COMMENTAIRES

Pour documenter le programme, on peut mettre des commentaires qui ne seront pas exécutés par le “moteur” du langage.

→ Pour définir des commentaires, il faut placer un symbole en début de commentaire (Python : un dièse #). Le reste de la ligne est alors commenté.

```
# Ceci est un commentaire  
print('Ceci est une instruction') # Un autre commentaire
```



```
# Ceci décrit mon programme  
# Le(s) nom(s) du(es) auteur(s) à l'origine du script  
# Lien(s) web  
# Date de dernière modification
```

LANGAGE DE PROGRAMMATION: LES BASES

- Les variables
- Les structures conditionnelles
- Les boucles



LES BASES: VARIABLES

Une variable est une “boîte” permettant de stocker une valeur.

Cette valeur est:

- un nombre (*Number*)
- un booléen (*True/False*)
- une chaîne de caractères (*String*)
- un tableau (*List* en Python)
- un dictionnaire (*Dictionary*)
- un objet



myVar

(1) Déclaration

Variable *undefined*

3



myVar

(2) Définition

signe d'affectation =

LES BASES: VARIABLES



Le nom des variables ont certaines contraintes parmi lesquelles:

- seuls les caractères de l'alphabet américain (ascii) sont autorisés.
- majuscules et minuscules sont autorisés.
- ne peut pas commencer par un chiffre.
- ne doit pas contenir des symboles comme `-+/*@`
- le symbole souligné `'_'` est autorisé.
- ne doit pas correspondre à un mot clé réservé au langage Python.

LES BASES: VARIABLES & NOMBRES

La variable peut contenir un nombre (entier ou réel):

```
myVar = 3
```

```
i = 3.145
```

```
j = 1e+16
```

... ou le résultat d'un calcul

```
k = 3 + 2
```

```
m = 3 / 4
```

```
p = 4 + 2 * 3
```

Les opérateurs arithmétiques sont:

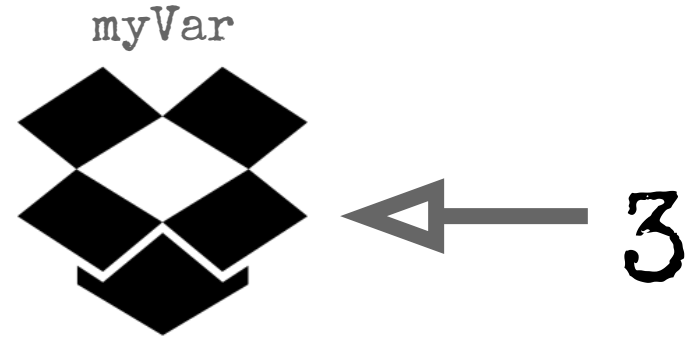
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo (reste de la division entière)

Autres opérateurs

**	Puissance
//	Division entière

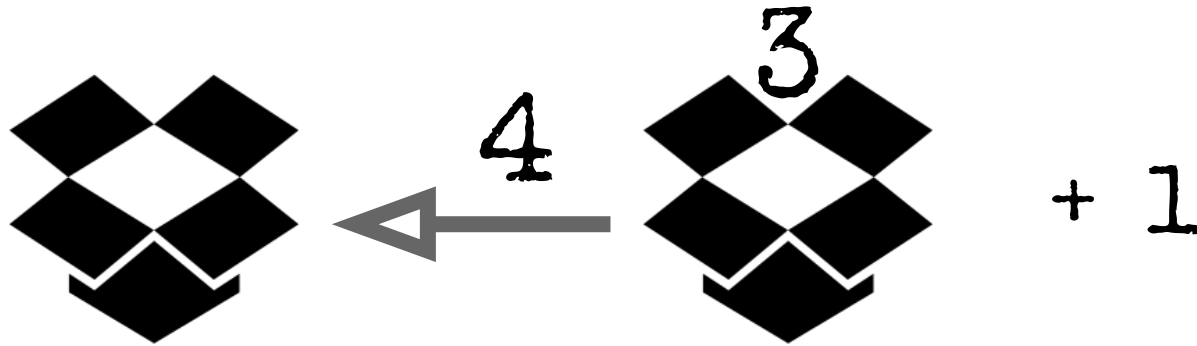


LES BASES: VARIABLES



myVar = 3

myVar = myVar + 1



LES BASES: BOOLÉENS (TRUE / FALSE) & COMPARAISON

$x > y$ est vrai quand x est strictement supérieur à y ,

$x < y$ est vrai quand x est strictement inférieur à y ,

$x \geq y$ est vrai quand x est supérieur ou égal à y , et

$x \leq y$ est vrai quand x est inférieur ou égal à y .

$x == y$ est vrai quand x est égal à y ,

$x != y$ est vrai quand x est différent de y ,

Opérateurs logiques

→ X **or** Y : OU logique.

→ X **and** Y : ET logique.

→ **not** X : valeur booléenne opposée de X

```
value = 5
```

```
#Tests
```

```
test1 = (value == 5)
```

```
# test1 ← True
```

```
test2 = (value >= 3)
```

```
# test2 ← True
```

```
test3 = (value < 6)
```

```
# test3 ← True
```

```
test4 = (value != 5)
```

```
# test4 ← False
```

```
test5 = (value > 3 and value <= 5)
```

```
# test5 ← True
```



LES BASES: VARIABLES & TEXTE (STRING)

La variable peut contenir du texte:

```
myVar = 'Ceci est du texte'
```

```
i = "Ceci est aussi du texte"
```

```
j = """Ceci  
est un texte  
sur plusieurs lignes"""
```

Concaténation et Répétition

```
m = 'py' + 'thon' # m ← 'python'
```

```
k = 2 * 'cou' # k ← 'coucou'
```

Les opérateurs de String:

+ Concaténation

Autres opérateurs

* Répétition



```
k = 'cou' * 2 # ERROR
```

LES BASES: VARIABLES & TABLEAUX (LIST EN PYTHON)

```
myVar = [1, 2, 3, 4]
```

```
lang = ['python', 'java', 'C++', 'JS']
```

```
i = [1, 5.43, 'coucou', [0,1,2]]
```

```
j = myVar + [5,6,7] # j ← [1, 2, 3, 4, 5, 6, 7]
```

Lire un élément du tableau:

```
k = lang[2] # k ← 'C++'
```

```
m = lang[-3] # m ← 'java'
```

Modifier un élément du tableau:

```
lang[2] = 'PHP'
```

Ajouter un élément du tableau:

```
lang[4] = 'ruby' #ERROR
```



Le tableau ou liste est:

→ délimité par des crochets

→ chaque item/élément est séparé par des virgules

Lecture d'un item

→ list[index] index positif ou négatif

index	0	1	2	3
lang	'python'	'java'	'C++'	'JS'
	-4	-3	-2	-1

Intervalle

→ list[index_start:index_end_excluded]

LES BASES: STRING EST AUSSI UN TABLEAU (ARRAY OU LIST)

une *String* est un tableau:

```
myVar = 'python'
```

```
first_c = myVar[0] # first_c ← 'p'
```

```
last_c = myVar[-1] # last_c ← 'n'
```

```
pair = myVar[2:4] # pair ← 'th'
```

Modification d'une lettre:

```
myVar[0]='J' #ERROR
```



index	0	1	2	3	4	5
myVar	'p'	'y'	't'	'h'	'o'	'n'
	-6	-5	-4	-3	-2	-1

LES BASES: DICTIONNAIRES

un *Dictionnaire* est un tableau contenant des paires (clé / valeur). Il est défini avec des **accolades** :

```
myVar = { 'Glucose': 'C6H12O6', 'Lactose': 'C12H22O11' }
```

```
compound = myVar['Glucose'] # compound ← 'C6H12O6'
```

```
isHere = 'Lactose' in myVar # isHere ← True
```

```
noThere = 'Lactose' not in myVar # noThere ← False
```

Ajout/Suppression d'une paire clé/valeur:

```
myVar['Pyruvate']='C3H4O3'
```

```
del myVar['Lactose']
```

LES BASES: OBJET

L'objet est une structure de données particulière contenant:

→ des attributs (ou propriétés)

→ des fonctions (appelées méthodes)

En Python,

les *String*, *List* et *Dictionary* sont aussi des objets