

Formation Python

Elie Dumas-Lefebvre

Institut des Sciences de la Mer
elie.dumas-lefebvre.1@ulaval.ca

12 juin 2017

1 Second Section

Histoire de Python



Pourquoi utiliser Python ?

- Language élégant
- Syntaxe rigoureuse (indentation, conventions, etc.)
- Logiciel gratuit et portable
- Plusieurs bibliothèques disponibles pour le traitement de données scientifiques (Numpy, Scipy, Matplotlib)

Déclaration de variables

```
# Voici un premier commentaire  
toto = 1 # En voila un deuxieme  
        # Et pourquoi pas un troisieme ?  
  
titi = 3.1415  
  
tata = '#Ceci n'est pas un commentaire'
```

Déclaration de variables

Bilan :

- toto est un entier
- titi est un réel
- tata est une chaîne de caractères

Nombres et opérations mathématiques de base

- interaction entre les réels et les entiers
- +, -, /, //, %, *, **

Exemples :

```
>>> 3 + 4
```

```
7
```

```
>>> 50 - 5*6
```

```
20
```

```
>>> (50 - 5.0*6) / 4.0
```

```
5.0
```

```
>>> 5 ** 2 # L'opérateur ** est l'exposant
```

```
25
```

```
>>> 8 / 5 # La division retourne toujours un réel
```

```
1.6
```

Exemples (suite) :

```
>>> 8 // 5 # La division entiere arrondi a  
1 # la baisse la division normale
```

```
>>> 8 // 5.0 # La division entiere effectuee  
1.0 # avec un reel retournera un reel
```

```
>>> 8 % 5 # L'operateur modulo donne le  
3 # reste de la division entiere
```


Exemples :

```
>>> 'ISMER'  
'ISMER'  
>>> 'L\'annexe A' == "L'annexe A"  
True  
>>> "'L\'annexe A", dit-il '  
'"L\'annexe A", dit-il '  
>>> print("'L\'annexe A", dit-il '  
'"L'annexe A", dit-il '
```

Caractères spéciaux

- `\n` : nouvelle ligne
- `\r` : retour de chariot
- `\t` : tabulation horizontale

Interprétation

```
>>> print('La premiere ligne.\nLa deuxieme.')  
La premiere ligne.  
La deuxieme.  
>>> print(r'La premiere ligne.\nLa deuxieme')  
'La premiere ligne.\nLa deuxieme'
```

Exemples :

```
>>> 3 * 'la' + 'si'  
'lalalasi'  
>>> 3 * 'la' 'si'  
'lasilasilasi'  
>>> 'la' 'si' 'do'  
'lasido'  
>>> note_1 = 'do'  
>>> note_1 + 're'  
'dore'
```

Matlab

- Séquences indexées de 1 à n
- Les tranches se font avec des " ()"

Python

- Séquences indexées de 0 à n-1
- Les tranches se font avec des " []"

Exemples :

```
>>> mot = 'bonjour '  
>>> mot[0]  
b  
>>> mot[6]  
r  
>>> mot[-1]  
r  
>>> mot[-7]  
b  
>>> mot[:3]  
bon  
>>> mot[1:4] # caracteres etant compris entre  
onj          # l'indice 1 (inclus) et  
              # l'indice 4 (exclus)  
>>> len(mot)  
7
```

Immuabilité des chaines de caractères :

```
>>> mot[:3] = 'ba'  
TypeError: 'str' object does not support item assignment  
>>> nouveau_mot = 'ba' + mot[3:]  
>>> print(nouveau_mot)  
'bajour'
```

Représentation d'une séquence Python :

+	+	+	+	+	+	+	+							
	b		o		n		j		o		u		r	
+	+	+	+	+	+	+	+							
0	1	2	3	4	5	6	7							
-7	-6	-5	-4	-3	-2	-1								

Listes

```
>>> ma_liste = [1, 2, 3, 4]
>>> print(ma_liste)
[1, 2, 3, 4]
```


Indexation et tranches

```
>>> ma_liste [0]
1
>>> ma_liste [1:3]
[2, 3, 4]
>>> ma_liste [-1]
4
>>> len ( ma_liste )
4
>>> min ( ma_liste )
1
>>> max ( ma_liste )
4
```

Copie d'une liste

```
# Creation d'une copie independante
>>> copie_de_ma_liste = ma_liste[:]
>>> copie_de_ma_liste
[1, 2, 3, 4]
>>> ma_liste[0] = 5
>>> ma_liste
[5, 2, 3, 4]
>>> copie_de_ma_liste
[1, 2, 3, 4]
```

```
#Creation d'une copie dependante
>>> copie_de_ma_liste = ma_liste
>>> copie_de_ma_liste
[1, 2, 3, 4]
>>> ma_liste[0] = 5
>>> ma_liste
```

Mutabilité de la liste

```
>>> fibonacci = [1, 1, 2, 3, 6, 8] # Il y a une erreur
>>> fibonacci[4] = 5 # On remplace la mauvaise valeur
>>> print(fibonacci)
[1, 1, 2, 3, 5, 8]
>>> liste_mixte = [1, 2, 3, 'a', 'b', 'c']
>>> liste_mixte[3:] = ['A', 'B', 'C'] # On substitue
>>> print(liste_mixte)
[1, 2, 3, 'A', 'B', 'C']
>>> liste_mixte[3:] = [] # On enleve les lettres minuscules
>>> liste_mixte
[1, 2, 3]
```

Concaténation

```
>>> ma_liste = [1, 2, 3, 4]
>>> DLC = [5, 6, 7, 8]
>>> ma_liste = ma_liste + DLC
>>> print(ma_liste)
[1, 2, 3, 4, 5, 6, 7, 8] # Concatenation directe
>>> ma_liste = [1, 2, 3, 4]
>>> ma_liste.append(DLC)
>>> print(ma_liste)
[1, 2, 3, 4, [5, 6, 7, 8]] # ajout de l'argument de
>>> ma_liste = [1, 2, 3, 4]
>>> ma_liste.extend(DLC)
>>> print(ma_liste)
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
# Methode alternative de concatenation avec la meth  
>>> ma_liste = ma_liste[:4]  
>>> for i in DLC:  
...     ma_liste.append(i)  
...  
>>> print(ma_liste)  
[1, 2, 3, 4, 5, 6, 7, 8]
```

TABLE – Comparaison entre les opérateurs logiques Python vs Matlab

Énoncé logique	Python	Matlab
plus petit que	<code>i</code>	<code>i</code>
plus petit ou égal	<code>i<=</code>	<code>i<=</code>
plus grand que	<code>i</code>	<code>i</code>
plus grand ou égal	<code>i<=</code>	<code>i<=</code>
égal à	<code>==</code>	<code>==</code>
n'égale pas	<code>!=</code>	<code>~=</code>
et	<code>and</code>	<code>&</code>
ou	<code>or</code>	<code>—</code>
non	<code>not</code>	<code>~</code>

En python il existe trois énoncés conditionnels : if, elif et else. Dans ce langage, l'indentation est très importante, voici donc quelques exemples.

```
>>> chiffre = int(input('Veuillez entrer un chiffre
... Veuillez entrer un chiffre: 42
>>> if chiffre < 42 :
...     print('Le chiffre doit etre plus grand')
... elif chiffre > 42 :
...     print('Le chiffre doit etre plus petit')
... else:
...     print('Voila la reponse a l'univers')
...
Voila la reponse a l'univers
```

Bref, les énoncés conditionnels s'expriment de façon générale comme

```
if condition1:  
    consequence  
elif condition2:  
    consequence  
else condition3:  
    consequence
```

De manière générale, la condition est un test logique ou une variable booléenne et la conséquence sera exécutée uniquement si la condition est satisfaite.

Note : Il est très important de finir chaque condition par un ' : '.

Blocks of Highlighted Text

Block 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Block 2

Pellentesque sed tellus purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum quis magna at risus dictum tempor eu vitae velit.

Block 3

Suspendisse tincidunt sagittis gravida. Curabitur condimentum, enim sed venenatis rutrum, ipsum neque consectetur orci, sed blandit justo nisi ac lacus.

Multiple Columns

Heading

- 1 Statement
- 2 Explanation
- 3 Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

TABLE – Table caption

Theorem

Theorem (Mass–energy equivalence)

$$E = mc^2$$

Verbatim

Example (Theorem Slide Code)

```
\begin{frame}  
\frametitle{Theorem}  
\begin{theorem}[Mass--energy equivalence]  
$E = mc^2$  
\end{theorem}  
\end{frame}
```

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

Citation

An example of the `\cite` command to cite within the presentation :

This statement requires citation [Smith, 2012].

References



John Smith (2012)

Title of the publication

Journal Name 12(3), 45 – 678.

The End