

Langages de Script - Memento Python

C. Tasson

I) Introduction à PYTHON

a) Installer PYTHON et se documenter

Nous utiliserons PYTHON version 3.0. Dans les systèmes Linux, PYTHON est généralement déjà installé par défaut. Pour installer PYTHON sous Windows ou Mac, il suffit de télécharger la version 3.0 disponible à l'adresse : <http://www.Python.org/>.

Parmi les nombreuses ressources disponibles sur Internet, on peut citer :

1. La traduction française du livre *A bite of PYTHON* : http://www.swaroopch.com/notes/Python_fr:Table_des_Matières.
2. l'ouvrage *Apprendre à programmer avec PYTHON* de Gérard Swinnen : <http://inforef.be/swi/python.htm>.
3. le tutoriel <http://docs.python.org/py3k/tutorial/index.html>.

b) Écrire des programmes

Pour écrire un programme PYTHON , on ouvre un fichier texte dans un éditeur de texte à côté du terminal. On écrit les instructions, puis on sauvegarde le fichier en choisissant un nom de la forme `nom_du_fichier.py`. Remarquez que l'extension est indispensable. Enfin, on lance l'interprétation dans le terminal via la ligne de commande `python3.0 nom_du_fichier.py`.

L'encodage des caractères par un ordinateur n'est pas encore unifiée. La version PYTHON 3.0 est prévue pour fonctionner avec l'UTF-8. Pour éviter tout problème, on veillera à configurer son éditeur de texte dans cet encodage.

c) Règles de bonne programmation

Un programme doit pouvoir être lu et relu, pour cela :

- * Utiliser des noms de variables et de fonctions *explicites*.
- * *Commenter* les différentes étapes du programme avec le caractère `#`.
- * *Documenter* les fonctions.

Enfin, quelques règles de présentation doivent être respectées :

- * Placer un espace après une virgule, un point-virgule ou deux-points ;
- * Ne pas placer d'espace avant une virgule, un point-virgule ou deux-points ;
- * Placer un espace de chaque côté d'un opérateur ;
- * Ne pas placer d'espace entre le nom d'une fonction et sa liste d'arguments ;
- * Indenter (python est un langage indenté, ce qui signifie que l'indentation est obligatoire, elle est normalement de 4 espaces à paramétrer dans son éditeur de texte).

II) Calcul numérique

Opérations			
Addition	+	Soustraction	-
Multiplication	*	Division	/
Puissance	**		
Division Entière			
Partie entière	//	Reste	%
Conversions			
En entier	int()	En flottant	float()

III) Variables et Fonctions

Affectation	
x	= 2
Retourner la valeur d'une variable	
return	x
Définition d'une fonction	
<pre>def <nom_fonction>(<argument_1>,<argument_2>,...,<argument_n>): """Description sur plusieurs lignes de la fonction""" <instructions></pre>	
Documentation sur une fonction	
help	(nom_fonction)

IV) Calcul propositionnel et conditionnelles

Constantes booléennes			
True		False	
Comparaisons			
Égalité	==	Différence	!=
Strictement supérieur	>	Strictement inférieur	<
Supérieur ou égal	>=	Inférieur ou égal	<=
Opérateurs booléens			
Non	not		
Ou	or	Et	and

Instruction conditionnelle

```

if <condition_1>:
    <instructions_1>
elif <condition_2>:
    <instructions_2>
elif <condition_3>:
    <instructions_3>
...
else:
    <instructions_n>

```

V) Types énumératifs et algorithmes itératifs**Types list et string**

Liste vide	<code>[]</code>	Chaîne vide	<code>''</code>
Longueur	<code>len(T)</code>	Concaténation	<code>+</code>
Extraction	<code>T[i]</code>	Extraction	<code>T[deb:fin:pas]</code>

Spécifique au type list

Ajout d'un élément	<code>L.append(a)</code>	Suppression	<code>del(L[i])</code>
Copie sans alias	<code>L1 = L2[:]</code>		

Type range

```
range(deb:fin+1:pas)
```

Boucle de répétition

```

for <element> in <list_string_range>:
    <instructions>

```

Boucle *Tant que*

```

while <condition>:
    <instructions>

```