

Algorithmique et Programmation (TALA330A... L3 INALCO)

Crédits : M-Anne Moreaux - INALCO, et G. Galisot - Université de Tours

Caroline Parfait
caroline.parfait@sorbonne-universite.fr

2021-2022

Obtic, Sorbonne Center for Artificial Intelligence
STIH EA 4509, Sorbonne Université

Plan de la présentation

1. Programmer efficacement
2. Bonnes pratiques
3. Qu'est qu'une variable ?

Programmer efficaciment

Rappel : un algorithme = une recette

Déterminer :

- les données d'entrées
- le type de résultat attendu
- la procédure : actions élémentaires réalisées par la machine

Un algorithme est conçu :

- pour généraliser un calcul,
 - ** non pas procéder à un calcul particulier
- pour opérer sur une quantité de données de même type
 - ** et non une donnée particulière
- de manière différente selon le programmeur.
 - ** évaluer le travail de son collègue pour améliorer l'algorithme
- s'il est possible de trouver une solution algorithmique
 - ** évaluer théoriquement si le problème peut trouver une solution calculable

Construire un programme

- Définir un algorithme (peut être fait sur papier) :
 - spécifier les données et les résultats
 - définir la structure logique de la solution
 - ne pas anticiper les solutions de codage
- Encoder l'algorithme : Écrire le code source en langage de programmation
- Transformer ce programme source en programme exécutable
- Faire fonctionner effectivement son programme

Modifier, améliorer, compléter un programme

- Être capable de lire et de comprendre du code source
- Déterminer la structure logique de l'algorithme
- Écrire un texte dans le langage de programmation utilisé
- Faire fonctionner effectivement son programme

*Un programme est l'encodage d'un algorithme dans un langage
« compréhensible » par un système informatique.* M-Anne Moreaux

Faire fonctionner son programme

- connaître le langage de programmation utilisé
 - lexique, syntaxe, sémantique
- connaître son environnement de programmation (IDE)
- savoir utiliser son IDE dans le système d'exploitation utilisé
 - **Où et Comment sont rangés les fichiers** : code source, données, résultats
 - quelles sont **les commandes** à utiliser
 - pour vérifier l'organisation de ses fichiers
 - pour transformer le code source en code exécutable

Programmation Procédurale

Principe

- Le programme est conçu comme une **suite de procédures** à réaliser pour répondre au problème considéré
- Un programme principal fait appel à des fonctions qui peuvent être réutilisées plusieurs fois avec différentes variables

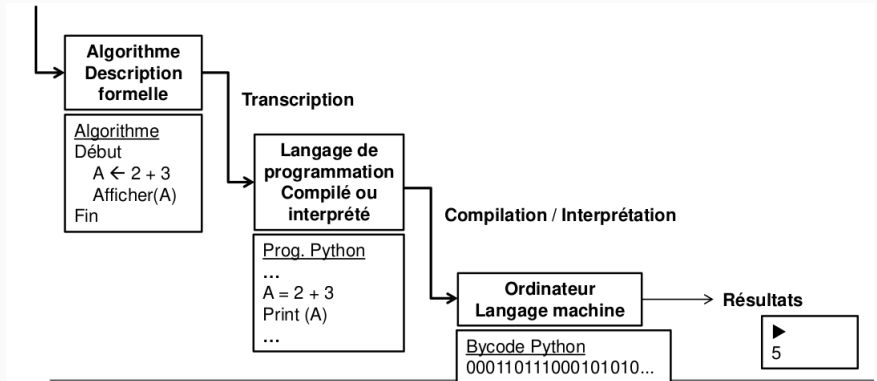
Exemple

- Je veux successivement lire deux nombres entrés au clavier
- Faire un calcul (calcul_R) à l'aide des deux nombres lus
- Écrire à l'écran le résultat1 du calcul

Questions : Quelles sont les briques à définir ? Comment les enchaîner ?

Programmation Procédurale

Problème : stocker le résultat de l'expression « $2 + 3$ » dans la variable **A puis l'afficher.**



Bonnes pratiques

Bonnes pratiques de programmation

- Réfléchir avant d'agir (programmation papier)
- Utiliser l'auto-complétion le copier/coller avec parcimonie :
 - se relire,
 - **faire relire par les pairs**
- Respecter le lexique, la syntaxe et la sémantique
- Définir les variables **explicitement** = NON AMBIGU
- Respecter le type de chaque variable
- Ne pas hésiter à écrire des commentaires
- ...

- Erreurs de syntaxe : Mauvaise utilisation du langage
- Erreurs de sémantique : Mauvaise conception de l'algorithme
- Erreur à l'exécution : circonstances particulières (installation défectueuse de l'IDE sur l'OS)

Déboguer avec les outils de votre IDE,

Faire des recherches sur le net

Qu'est qu'une variable ?

Définir une variable

Une variable est caractérisée par un nom

- Attention : les noms sont sans accent, sans espace
- Tient compte de la casse : Majuscule \neq minuscule
- Ne doit pas commencer par un chiffre

Définir une variable

Une variable renvoie :

- à une **position de la mémoire** dans l'ordinateur ("case")
- à un **type qui définit la taille** réservée en mémoire et les opérations possibles
- Peut prendre **successivement différentes valeurs** pendant l'exécution d'un programme

>>>> *age = 23*

>>>> *ville = "Tours"*

Adresse	Case Mémoire	Nom
...
10FFA3	23	age
10FFA4	"Tours"	ville
....		
10FFA9	

- La 1ere affectation s'appelle l'**initialisation**
- **Affectation** d'une valeur "=" : variable = valeur

Le contenu des variables (cases mémoires) évolue dans le temps (durant l'exécution du programme)

```
>>> a = 12345
```

```
>>> a = b = 123
```

```
>>> a, b = 3, 5
```

```
>>> a, b = b, a
```

Les Types

Nom type	Type	exemple
Nombre entier	Integer, int	<code>a = 1</code> (codé sur 32 ou 64 bits)
Nombre Réel (décimaux)	Float , float	<code>b = 3.14e-10</code> (codé 64 bits)
chaîne de caractères	String, str	<code>ch = "Bonjour"</code>
Liste	list	<code>[1,2,5,9,10,"Bonjour"]</code>
Booléens	bool	<code>a = True</code>

Il existe aussi des types construits, *Tuple* et les ensembles *Set*.
Nous les observerons plus tard.

Types numériques, Int, Float et opérateurs

Opérations	symbole	exemples
addition	+	$6+4 \rightarrow 10$
soustraction	-	$6-4 \rightarrow 2$
multiplication	*	$6*4 \rightarrow 24$ $1.2 * 1 \rightarrow 1.2$
exponentiation	**	$12**2 \rightarrow 144$
division	/	$6/4 \rightarrow 1$ (*) $6./4 \rightarrow 1.5$
quotient de division entière	//	$6//4 \rightarrow 1$
reste de division entière	%	$6\%4 \rightarrow 2$

Type String et opérateurs

- `ch = "Hello world"` # On peut utiliser `"` ou `'`
- longueur de chaîne : `>>> len(ch)`
- concaténation : `>>> ch + ch`
- accéder à un caractère de la chaîne : `>>> ch[0] = 'H'`
 - La numérotation commence toujours à 0
 - Les espaces comptent pour un caractère

>>> Salut = "Ceci est une chaîne plutôt longue contenant plusieurs lignes ... de texte (Ceci fonctionne de la même façon en C/C++. ... Notez que les blancs en début de ligne sont significatifs.

Types Liste et opérateurs

- Une liste est un ensemble fini d'objets de différents types

```
>>> lst = [1, 2, 3]
```

```
>>> jour =
```

```
['lundi', 'mardi', 'mercredi', 180, 20.3, 'jeudi', 'vendredi']
```

```
>>> lst = list(range(2, 10)) # donne [2, 3, 4, 5, 6, 7, 8, 9]
```

- Une liste peut être une liste de listes.

```
>>> liste4 = [1, 3, 5, 7, 9]
```

```
>>> liste5 = [0, 2, 4, 6, 8]
```

```
>>> liste6 = [liste4, liste5]
```

- **liste[n]** accède à l'élément n-1 d'une liste

*(numérotation à partir de 0)

- **liste[-1]** : désigne le dernier élément d'une liste non vide

*(liste[-2], l'avant dernier, etc)

Type Booléens et opérateurs

- valeurs possibles = **True** et **False**
- Les opérateurs sont : **and**, **or**, **not**

Les opérateurs de comparaison

<code>x == y</code>	<code># x est égal à y</code>
<code>x != y</code>	<code># x est différent de y</code>
<code>x > y</code>	<code># x est plus grand que y</code>
<code>x < y</code>	<code># x est plus petit que y</code>
<code>x >= y</code>	<code># x est plus grand que, ou égal à y</code>
<code>x <= y</code>	<code># x est plus petit que, ou égal à y</code>

Conversions de Types

** Les conversions de types sont possibles : `eval()`, `str()`, etc

*** Voir **automatique en Python = Danger !**

```
>>> ch = str(1323)
```

```
>>> f = float('13.23') + int('14')
```

```
>>> b = a + 12.3 # l'entier a est converti automatiquement en  
float pour effectuer le calcul
```


- **Mots-clés Python** (inutilisables pour nommer les variables)

Liste des mots-clefs en Python(V3)				
and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	-
class	exec	in	raise	-
continue	finally	is	return	-
def	for	lambda	try	-

Ce support de cours a été construit à partir des sources suivantes :

- Support de cours Licence 1 « Algorithmique et langages du Web » UE 102 - 2011-2012. Auteur : Thomas Devogèle
- Informatique et sciences du numérique. Spécialité ISN en Terminal S. Gilles Dowek. Eyrolles.
- Site OpenClassrooms.com. Tutoriel Python. Sept 2013
[http ://fr.openclassrooms.com/informatique/python/cours](http://fr.openclassrooms.com/informatique/python/cours)
- Site Débuter avec Python au lycée. Sept 2013.
[http ://python.lycee.free.fr/](http://python.lycee.free.fr/)
- Gérard Swinnen Apprendre Python 3 -
[http ://inforef.be/swi/python.htm](http://inforef.be/swi/python.htm)
- Cours sur Python 3 - Robert Cordeau
- M-Anne Moreaux, Licence LLCER, Parcours TAL – 2020/2021
- Wikipédia ...