

Dans ce TP, nous allons découvrir la programmation à l'aide du langage Python.

On rappelle qu'un algorithme est une succession d'un nombre fini d'étapes réalisées dans un ordre précis et qui a pour but de résoudre un problème. Nous allons traduire certains algorithmes en langage Python et nous allons les exécuter à l'aide de l'ordinateur.

On peut faire beaucoup de choses avec Python : des graphiques, du traitement du son, traitement d'images, des jeux, des logiciels, des applications web, ...

Avant de commencer à programmer, une petite organisation est nécessaire :

## EXERCICE 0

- Lorsque votre session est ouverte, brancher votre clé usb.
- Ouvrir la clé dans l'explorateur windows (ou dans poste de travail) et créer un nouveau dossier nommé « SNT ».
- Dans ce nouveau dossier, créer un un nouveau dossier que l'on nommera « TP1 ».
- Dans ce dossier « TP1 », vous y mettrez tous les documents concernant la séance (feuille de TP, document réponse, programmes créés lors de la séance, ...)

On prendra l'habitude de créer, pour chaque TP, un nouveau dossier et dans lequel on mettra les documents correspondants.

.....

Pour programmer, nous allons utiliser un IDE (Integrated Development Environment). C'est un éditeur de texte un peu spécial qui va nous permettre d'écrire nos instructions en langage Python puis de les exécuter. Vous avez deux possibilités pour les TP : ISN-IDE (simple et efficace à utiliser) ou EduPython (un peu plus complet).

Lancer un de ces deux logiciels et vous êtes prêt à programmer!

.....

## EXERCICE 1 - Manipulation de la console

Nous allons dans cet exercice apprendre à utiliser la console.

- Si vous utilisez EduPython, la console est en bas de l'écran.
- Si vous utilisez ISN-IDE, la console n'apparaît pas immédiatement. Il faut l'ouvrir en faisant « Fichier » puis « Console Python ». Une nouvelle fenêtre apparaît.

La console nous permet d'entrer des instructions et de les exécuter en appuyant sur la touche « Entrée ».

1. On peut l'utiliser comme une calculatrice. Taper dans la console  $50+3$  et vérifier que l'on obtient :

```
>>>50+3
53
```

2. Taper les instructions suivantes dans la console et **noter le résultat obtenu** sur le document réponse :

50-3    50\*3    50/3    50//3    50 % 3    4.215 + 0.256    "Bonjour"    "Choco"+"lat"

3. A quoi correspondent les instructions 50//3 et 50 % 3?

4. En utilisant le tableau donné ci-dessous, taper les instructions dans la console, noter le résultat obtenu puis donner la signification en français.

type(5)    type(2,71)    type("Hola")    type(2==1)    type(4==4)

## EXERCICE 2 - Les variables

On souhaite parfois conserver en mémoire le résultat d'une expression arithmétique pour l'utiliser plus tard ou dans un autre calcul. On utilise alors une **variable**.

- Une variable peut-être vue, virtuellement, comme une boîte représentant un emplacement dans la mémoire de l'ordinateur, associé à un nom et permettant de stocker une valeur.
- **Affecter une valeur à une variable** consiste à stocker une certaine valeur dans une variable. En langage naturel, l'affectation se note avec une flèche.

Par exemple, si on veut stocker le nombre 4 dans la variable  $a$ , on écrira en langage naturel :  $a \leftarrow 4$ .

1. **Affecter une valeur à une variable** : Avec Python, on utilise le symbole  $=$  pour affecter une valeur à une variable.

Taper dans la console l'instruction  $a = 4$ .

*Remarque* : On constate que rien ne s'affiche. C'est normal, Python a fait ce qu'on lui a demandé, c'est à dire stocker le nombre 4 dans la variable  $a$ .

Si on veut afficher la valeur contenue dans la variable  $a$ , il suffit de taper  $a$  dans la console.

2. **Modifier une variable** : Si on veut modifier le contenu de la variable, c'est-à-dire, mettre une nouvelle valeur, il suffit de répéter l'instruction précédente.

Taper  $a = 10$  dans la console et afficher le contenu de la variable  $a$ .

3. **Utilisation des variables** : Il est possible d'utiliser le contenu d'une variable dans une autre variable.

Taper les instructions suivantes dans la console :

```
>>>a=6
>>>b=10
>>>c=a+b
```

Que contient la variable  $c$ ? L'afficher à l'aide de la console.

4. **Affectation de plusieurs variables en même temps** : on peut affecter plusieurs variables en même temps.

Taper  $a,b,c = 1,2,3$  dans la console.

*Remarque* : Cela signifie que dans la variable  $a$ , on stocke le nombre 1 ; dans la variable  $b$ , on stocke le nombre 2 ; dans la variable  $c$ , on stocke le nombre 3.

5. Taper les instructions suivantes dans la console Python :

```
>>>a=5
>>>b=7
>>>a=a+1
>>>b=b-2
```

Quelles sont les valeurs contenus dans les variables  $a$  et  $b$ ?

*Remarque* : Il ne faut pas confondre l'égalité mathématique et l'égalité servant d'affectation pour Python.

Pour comprendre l'instruction  $a = a + 1$ , il faut lire de la droite vers la gauche :

on regarde la valeur contenue dans  $a$ , on ajoute 1 à cette valeur et on stocke le tout dans  $a$ .

### EXERCICE 3 - Utilisation de l'éditeur

La console est utile lorsque l'on souhaite exécuter une ou deux instructions, ou pour vérifier le contenu d'une variable. Mais lorsqu'on a plusieurs instructions à exécuter, il est préférable de tout écrire puis de l'exécuter.

Pour cela, nous allons utiliser l'éditeur.

Tout d'abord, créer un nouveau fichier en faisant « Fichier » puis « Nouveau ». Et sauvegarder ce fichier dans votre dossier TP1 sur votre clé usb en le nommant Prog3TP1SNT.py

1. Écrire les instructions suivantes dans l'éditeur (attention à bien respecter les sauts de lignes) :

```
a = 2
b = 14
a = a + 1
b = b/2
c = a + b
c = c * b
```

Exécuter ensuite le programme :

- Si vous travaillez avec EduPython, cliquer sur la flèche verte en haut de la fenêtre.
- Si vous travaillez avec ISN-IDE, appuyer sur la touche F5 de votre clavier ou sélectionner « Exécuter » dans le menu « Action » .

*Remarque* : Rien ne s'affiche. Comme pour l'exercice précédent, c'est normal.

L'interpréteur a effectué ce qu'on lui a demandé, c'est à dire stocker des valeurs dans des variables.

Si on veut qu'il affiche les contenus des variables, il faut le lui demander. Pour cela, on utilise l'instruction « print » :

2. Rajouter à la fin du programme précédent, l'instruction « print(a) » et exécuter le programme.
3. Compléter le programme précédent pour qu'il affiche le contenu des variables *b* et *c* lorsque l'on exécute le programme.

*Remarque* : les contenus apparaissent mais on ne se rappelle pas quelle valeur se trouve dans quelle variable...

4. Rajouter l'instruction : print("a vaut ",a).

De cette façon, on se rappelle lors de l'affichage à quelle variable correspond la valeur que l'on obtient.

### EXERCICE 4 - Instruction input

Avant de commencer l'exercice, créer un nouveau fichier puis le sauvegarder dans le dossier TP1 en le nommant Prog4TP1SNT.py

1. Écrire le programme suivant dans l'éditeur.

```
a = "Bonjour "
b = input("Entrer votre nom : ")
c = a+b
print(c)
```

Exécuter ce programme.

2. À quoi sert l'instruction input("Entrer votre nom : ") ?

3. On remplace le programme précédent par le programme suivant :

```
a = input("Entrer un nombre")
b = 24
c = a+b
print(c)
```

Exécuter ce programme. Que se passe-t-il?

4. Modifier le programme précédent pour obtenir le bon résultat.

*Aide* : L'instruction `int()` permet de transformer une chaîne de caractère (un texte) en un nombre entier; l'instruction `float()` permet de transformer une chaîne de caractère (un texte) en un nombre à virgule.

### EXERCICE 5 - Mini-bilan

Compléter le tableau suivant, qui résume les instructions vues dans les exercices précédents :

Langage naturel	Langage Python	À quoi ça sert ?
$a \leftarrow 7$		
Afficher b		
	<code>print("b contient :",b)</code>	
Saisir une valeur		

### EXERCICE 6 - Programme de calcul

On considère le programme de calcul suivant :

```
Choisir un nombre
Ajouter 3
Multiplier par 4
Enlever 2
Afficher le résultat
```

1. Traduire le programme de calcul donné précédemment en langage naturel. *On complètera le document réponse.*
2. Traduire le programme donné langage naturel en un programme Python. *On complètera le document réponse.*
3. Programmer l'algorithme de la question 2. Pour cela, on créera un nouveau fichier `Prog6TP1SNT.py` et on sauvegardera le programme Python dans ce fichier.
4. Tester le programme précédent en choisissant les nombres 156 puis -975. Noter les résultats obtenus sur le document réponse.
5. Quel nombre doit-on choisir au départ pour obtenir 58?

**Pour les exercices suivants, vous rédigerez le programme obtenu au dos du document réponse.**

### **EXERCICE 7**

Écrire un programme Python (que l'on nommera Prog7TP1SNT.py) qui demande à l'utilisateur d'entrer successivement deux nombres, qui fait la somme et qui affiche « La somme des deux nombres vaut : » suivi du résultat.

### **EXERCICE 8**

Écrire un programme Python (que l'on nommera Prog8TP1SNT.py) qui demande à l'utilisateur d'entrer une longueur et une largeur, qui calcule l'aire du rectangle de longueur et de largeur données précédemment et qui affiche « Aire du rectangle : » suivi du résultat.

### **EXERCICE 9**

Écrire un programme Python (que l'on nommera Prog9TP1SNT.py) qui demande à l'utilisateur d'entrer un prénom et qui affiche « Bonjour et bienvenue : » suivi du prénom entré précédemment.

### **EXERCICE 10**

Écrire un programme Python (que l'on nommera Prog10TP1SNT.py) qui demande à l'utilisateur d'entrer un nombre et qui affiche « Le carré du nombre (ici on met le nombre entré) vaut (ici on met le résultat) : ».

Par exemple, si l'utilisateur entre le nombre 4, le programme doit afficher :

Le carré de 4 est 16.

### **EXERCICE 11**

Écrire un programme Python (que l'on nommera Prog11TP1SNT.py) qui demande à l'utilisateur d'entrer un nombre et qui affiche la table de multiplication de ce nombre (de 1 jusqu'à 10).

Par exemple, si l'utilisateur entre le nombre 4, le programme doit afficher :

$4 \times 1 = 4$   
 $4 \times 2 = 8$   
 $4 \times 3 = 12$   
...  
 $4 \times 10 = 40$