

DEV1 – DEV1L – Laboratoires Python**TD 01 – Introduction à Python**

Lors de cette première séance d'exercices, vous ferez vos premiers pas en langage Python. Vous apprendrez à lancer Python en mode interactif de façon à exécuter quelques commandes simples et commencerez à vous familiariser avec les types de données.

Table des matières

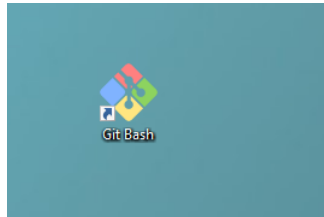
1	Lancer l'interpréteur Python	2
2	Manipuler des nombres	4
3	Manipuler du texte	5
4	Manipuler des listes de données	5
5	En résumé : les types de données en Python	6
6	Obtenir de l'aide sur Python	7
7	En résumé ...	8

1 Lancer l'interpréteur Python

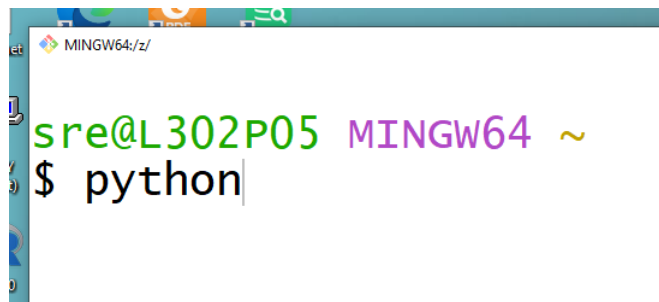
La première chose à faire pour pouvoir travailler avec Python est de lancer l'interpréteur qui se chargera d'exécuter vos commandes :

Tutoriel 1 Lancer et utiliser l'interpréteur Python

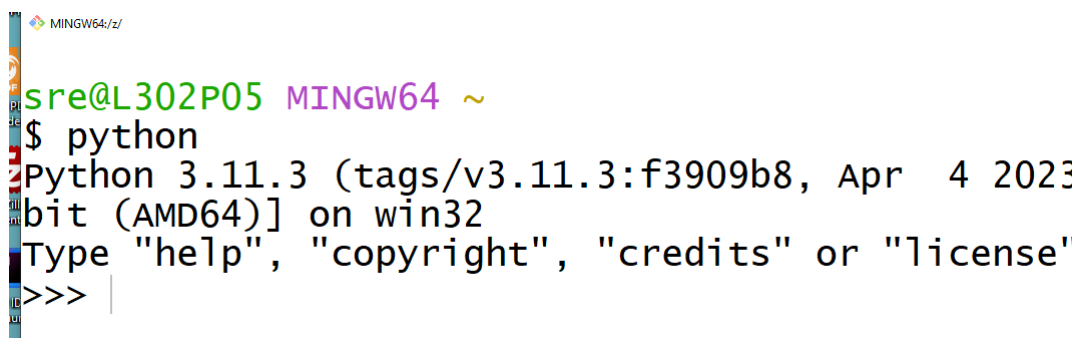
✍ Cliquez sur l'icône 'Git Bash'¹, se trouvant sur votre bureau :



✍ Dans la fenêtre qui s'affiche, tapez simplement 'python' :



✍ L'interpréteur se lance et attend que nous entrions une commande :



✍ Demandons lui par exemple d'effectuer une addition :

1. Nous avons ici besoin d'une interface en ligne de commande pour lancer Python, et utilisons Git Bash par facilité. D'autres interfaces (exemple : PowerShell) font aussi l'affaire

```
sre@L302P05 MINGW64 ~
$ python
Python 3.11.3 (tags/v3.11.3:f3c
bit (AMD64)] on win32
Type "help", "copyright", "crec
>>> 1234+56779
58013
>>>
```

Le résultat de l'addition s'affiche sur la ligne du dessous.

✎ Entrons à présent une commande comportant une erreur de syntaxe :

```
>>> 1234+56779
58013
>>> 1234+
      File "<stdin>", line 1
        1234+
            ^
SyntaxError: invalid syntax
>>>
```

Un message d'erreur est renvoyé et affiché sous la commande.

✎ Pour quitter l'interpréteur, tapez simplement `exit()`.

```
>>> 1234+
      File "<stdin>", line 1
        1234+
            ^
SyntaxError: invalid syntax
>>> exit()

sre@L302P05 MINGW64 ~
$
```

L'interpréteur Python

Nous venons de voir comment lancer Python en *mode interactif* : l'interpréteur lit et exécute les commandes à partir du terminal. Il est également possible d'utiliser Python en *mode script* : les commandes à exécuter se trouvent dans un fichier qui est passé à l'interpréteur. Nous nous contenterons du mode interactif lors de cette première séance.

Exercice 1 Utiliser Python en mode interactif

Exécutez les commandes Python suivantes en mode interactif. Certaines d'entre-elles sont erronées et renvoient un message d'erreur. Complétez alors le tableau ci-dessous avec la cause, selon vous, de l'erreur (demandez à votre enseignant si la raison de l'erreur ne semble pas évidente).

Comande	Erreur (oui/non) ?	Cause de l'erreur (le cas échéant)
456 - 723		
4/3		
4//3		
4/0		
len('Bonjour')		
'Bonjour'*5		
Bonjour*5		
10 == 5		
523 < 500		
523 <		

2 Manipuler des nombres

Les principaux opérateurs permettant de manipuler des nombres sont repris dans le tableau ci-dessous :

Opérateur	Signification
+	addition
-	soustraction
*	multiplication
**	exponentiation
/	division
//	division entière
%	reste de la division

Exercice 2 Opérateurs arithmétiques

Exécutez les commandes Python suivantes et notez le résultat obtenu dans le tableau. Dans la dernière colonne, donnez brièvement (avec vos mots) une description du calcul effectué par Python.

Comande	Résultat	Description du calcul
5*6		
2**4		
10/3		
10//3		
23%5		

Exercice 3 Opérateurs arithmétiques (bis)

Utilisez une commande Python pour calculer :

- $(10.54)^5 - 234$
- $\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$

Remarquez que les nombres à virgule se notent avec un point (10.54, 1.5, 0.6, etc.).

Exercice 4 L'aire du cercle

Utilisez une commande Python pour calculer l'aire d'un cercle de 6.5 centimètre de rayon².

2. L'aire A d'un cercle de rayon r est donné par la formule $A = \pi r^2$, vous pouvez ici utiliser l'approximation $\pi \approx 3.14$.

L'aire demandée vaut : cm²

Exercice 5 Secondes en minutes

Combien de minutes sont contenues dans 567 734 secondes ? Affichez la réponse à l'écran au moyen d'une commande Python.

Le nombre de secondes vaut :

Exercice 6 Les diviseurs

Utilisez Python pour déterminer les nombres entiers compris entre 2 et 9 qui divisent³ le nombre entier 952 426 626.

Les nombres recherchés sont :

3 Manipuler du texte

Les chaînes de caractères en Python sont toujours délimitées par des simples guillemets (exemple : 'salut !') ou des doubles guillemets (exemple : "Bonjour !"), au choix⁴.

Exercice 7 Opérations sur les chaînes de caractères

Exécutez les commandes suivantes et complétez le tableau avec une petite description de l'intérêt de chaque commande :

Comande	A quoi sert la commande ?
<code>len('Bonjour !')</code>	
<code>'EN MAJUSCULES'.lower()</code>	
<code>'en minuscule'.upper()</code>	
<code>'Bonjour ' + 'tout le monde !'</code>	
<code>'Haha'*10 + '!'</code>	
<code>"Papa, maman".count('a')</code>	

Les opérateurs + et *, en plus d'être des opérateurs mathématiques, permettent donc de *concaténer* des chaînes de caractères (les joindre, bout à bout).

Exercice 8 Concaténer des chaînes de caractère

Utiliser une commande Python pour afficher à l'écran votre prénom 50 fois de suite, suivi d'un espace, suivi du prénom de votre voisin 50 fois de suite.

4 Manipuler des listes de données

Pour définir une liste en Python, il suffit de préciser les différents éléments qui la composent, séparés par des virgules et entourés de crochets. On peut également lui donner un nom :

```
ma_liste = [5,1,2,8,1,4,6]
```

Voici quelques exemples de commandes qu'il est alors possible d'exécuter sur la liste ainsi créée :

3. Rappel : un nombre n divise un nombre m si le reste de la division de n par m vaut zéro. Par exemple, 2 divise 10 mais pas 3.

4. ... mais il convient d'être cohérent : utilisez soit systématiquement des simples guillemets, soit systématiquement des doubles.

Commandes	Signification
<code>ma_liste.append(10)</code>	ajoute 10 à la fin de la liste
<code>ma_liste.extend([18,19,20])</code>	ajoute la liste [18,19,20] à la fin la liste
<code>sorted(ma_liste)</code>	renvoie une version triée (par ordre croissant) de la liste
<code>len(ma_liste)</code>	renvoie le nombre d'éléments de la liste
<code>max(ma_liste)</code>	renvoie le plus grand élément de la liste

Exercice 9 Définir et utiliser des listes

Fournir et exécuter une commande Python permettant d'effectuer les actions suivantes (complétez le tableau avec la commande appropriée pour chaque action) :

Action	Commande Python
Créer une liste contenant les chiffres de votre matricule,	
Ajoutez à cette liste les chiffres du matricule de votre voisin	
Afficher la liste à l'écran	
Afficher la liste triée à l'écran	
Afficher le plus grand élément de la liste à l'écran	

5 En résumé : les types de données en Python

Les valeurs avec lesquelles il est possible de travailler en Python possèdent toutes un *type*, nous avons pour le moment abordé :

1. Les nombres entiers,
2. Les nombres à virgules,
3. Les chaînes de caractère,
4. Les listes.

Mais il en existe beaucoup d'autres. Il est possible d'obtenir le type de donnée correspondant à une valeur au moyen de la commande :

`type(valeur)`

Exercice 10 Afficher les types de données

Exécutez les commandes suivantes et inscrivez le résultat dans la colonne de droite :

Commande	Résultat
<code>type(1.586)</code>	
<code>type('salut')</code>	
<code>type('23')</code>	
<code>type(23)</code>	
<code>type([1,2,3])</code>	

Chaîne de caractère ou nombre entier ?

Attention : Comme illustré dans l'exercice ci-dessus, Python fait une différence entre la donnée '23' (une chaîne de caractère contenant les caractères 2 et 3) et 23 (le nombre entier 23).

Ces données sont gérées différemment en mémoire. En particulier, il est exclu d'appliquer une opération n'ayant de sens que pour les nombres entiers (comme l'exponentiation) sur une chaîne de caractère.

6 Obtenir de l'aide sur Python

Comme vous avez pu le constater lors des sections précédente, le langage Python est très vaste et comprend de nombreuses commandes et fonctionnalités. Il est à peu près impossible de tout connaître, et il est donc nécessaire de se renseigner en allant lire la documentation appropriée lorsqu'on cherche des informations sur une partie du langage. Un moyen de faire est de consulter la documentation officielle en ligne :

<https://www.python.org/>

Une autre façon de faire est d'utiliser la fonction `help()` de Python qui vous permet d'obtenir de l'aide sur une partie du langage. La commande :

`help(len)`

permet par exemple d'obtenir de l'aide sur la façon d'utiliser la fonction `len`.

Exercice 11 Utiliser la fonction d'aide de Python

Utilisez la fonction `help()` de Python pour afficher l'aide sur les fonctions `bin` et `abs` de Python. Parvenez-vous à comprendre l'utilité de ces deux fonctions ?

Fonction	Utilité ?
<code>bin</code>	
<code>abs</code>	

Taper simplement `help()` permet d'afficher le système d'aide de Python. Vous pouvez alors taper n'importe quel mot clé pour obtenir de l'aide sur celui-ci. Tapez `q` pour quitter le système d'aide.

Exercice 12 Utiliser la fonction d'aide de Python (bis)

Lancer le système d'aide de Python pour obtenir de l'aide sur les sujets suivants.

1. Les nombres entiers (tapez `INTEGER`). Quelle est la taille maximum d'un nombre entier sous Python ? Votre réponse :
2. Les listes (tapez `LISTS`). Comment faire pour supprimer tous les éléments d'une liste donnée ? Votre réponse :

Lire la documentation

Lire la documentation propre à un langage informatique ou un logiciel donné fait partie des pratiques courantes d'un informaticien professionnel. C'est une habitude que vous devez acquérir le plus vite possible !

7 En résumé ...

Principaux points de matière du TD

Voici les principaux points abordés lors de ce TD. Vous devez absolument être à l'aise avec ceux-ci avant d'aborder la prochaine séance d'exercice.

1. Utiliser Python en mode interactif pour exécuter des commandes.
2. Détecter une commande erronée.
3. Effectuer des manipulations de base portant sur des nombres, du texte et des listes d'objets.
4. Obtenir de l'aide sur Python en utilisant l'aide en ligne ou la fonction `help()`.