

TD 1 : CONSTANTES, VARIABLES, EXPRESSIONS ET TYPES

Une fois activé, l'interpréteur Python répète inlassablement le cycle suivant :

- Python affiche une invite (le signe `>>>`),
- l'utilisateur compose au clavier une expression Python correcte,
- Python évalue l'expression et affiche la valeur.

La plus grande partie de ce TD consiste à taper les expressions proposées ci-après, constater que les valeurs affichées par Python sont bien celles qui étaient attendues, ou bien découvrir quelles sont ces valeurs, en comprendre la raison et, le cas échéant, répondre aux questions posées par le sujet (signalées par \rightarrow).

Interpréteur et versions du langage

Quand vous tapez la commande `python`, vous exécutez un programme appelé interpréteur. Celui-ci vous indique la version du langage qu'il comprend.

```
alexis@alexis-XPS13:~$ python
Python 3.7.0 (default, Jun 28 2018, 13:15:42)
[GCC 7.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Pour nos TP, nous utiliserons la version 3.7 du langage.

Exercice 1. Valeurs et expressions

- (1) Manipulation des nombres entiers :

```
>>> 2 + 3
>>> 2 + 3 * 100
```

\rightarrow Pourquoi n'obtient-on pas 500 ? Comment aurait-il fallu écrire l'expression précédente pour obtenir 500 ?

- (2) Manipulation de texte : les chaînes de caractères (en anglais : strings)

```
>>> 'aaa'
>>> len('aaa')

→ Expliquez.

>>> len('aaa') + len('ttttt')
>>> 'aaa' + 'ttttt'
```

\rightarrow Expliquez.

```
>>> 'aaa' + 3
```

\rightarrow Lisez attentivement le message obtenu. Où se trouve l'information la plus utile pour vous ?

```
>>> type('aaa')
>>> type(3)
```

\rightarrow Connaissez-vous d'autres types ?

```
>>> type(1.5)
>>> 1 == 1
>>> 1 == 2
>>> not 1 == 2
>>> 1 == 1 and 2 > 3
>>> 1 == 1 or 2 > 3
>>> False or True
>>> False and True
>>> type(1 == 1)
```

Un booléen (cf M. Boole) ne peut prendre que deux valeurs : soit vrai, soit faux. En python c'est **True** ou **False**.

Opérations « mixtes » entre nombres :

```
>>> 1 / 3
>>> 1.0 / 3
>>> 1 / 3.0
>>> 1 // 3
>>> 1.0 // 3
>>> 1 // 3.0
>>> 10 % 3
```

/ est l'opérateur de division décimale. Pour obtenir le quotient d'une division entière, utilisez //. Pour obtenir le reste de la division entière, utilisez %. Un nombre flottant permet de représenter une valeur approchée d'un nombre réel.

Il est souvent impossible de représenter exactement les nombres rationnels :

```
>>> 1 / 3
```

Exercice 2. Variables

On peut associer un nom à une valeur :

```
>>> a = 3
>>> a
```

L'interpréteur affiche 3 au lieu de a. On dit que a est une **variable** ayant la **valeur** 3. Cette valeur peut changer :

```
>>> a = 2
>>> a
>>> a * 5
>>> b = a * 5
>>> b
>>> a = 1
>>> b
```

→ Pourquoi b est-il resté inchangé ?

La valeur d'une variable change au cours du temps :

```
>>> combien = 0
>>> combien
>>> combien = combien + 1
>>> combien
>>> combien = combien + 1
>>> combien
```

→ Expliquez le fonctionnement de l'expression **combien = combien + 1**

Un nom de variable n'est pas la même chose qu'une chaîne de caractères :

```
>>> 'onze'
>>> onze
>>> onze = 11
>>> onze
>>> 'onze'
```

→ Expliquez.

D'autres jeux avec des variables

```
>>> a = 123
>>> type(a)
>>> a + a
>>> a = '123'
>>> type(a)
>>> a + a
>>> a = 13579 % 3 == 0
>>> type(a)
```

→ Vous semble-t-il correct de parler du « type de la variable a » ?

Exercice 3. Syntaxe et erreurs

→ Quelles sont les erreurs produites par les entrées suivantes et comment les expliquez vous ?

```
>>> 2.1.2
>>> 2.1
>>> 2 1
>>> 21
>>> "a
>>> "a"
>>> +
>>> a
>>> a +
>>> a + 1
>>> a = 2
>>> 2 = a
>>> 'a' = 11
>>> a = 0
>>> 10 / a
```