

# Travail dirigé # 2

## Premiers pas avec Python

Entrées, sorties, types de données

### Cours 2

Ce deuxième laboratoire a pour objectif de vous faire commencer à vous amuser avec Python ! **Attention** : lorsque du code est donné dans les énoncés, **ne faites pas de copier-coller à partir du document PDF**. Un document PDF ne contient pas que du texte : certains caractères spéciaux pourraient se glisser dans votre copier-coller et vous pourriez obtenir des erreurs difficiles à comprendre dans votre programme Python. Réécrivez plutôt le code de votre côté.

## 1 Exercices sur les types

**Exercice 1.1** Lisez le code suivant (chaque ligne du tableau est une instruction) et expliquez ce qui se passe. Écrivez les valeurs des variables au fur et à mesure. Si vous ne savez pas, testez-le dans l'interpréteur Python ou VS Code !

	a	b	c	d
<b>a = 2</b>	2	-	-	-
<b>b = 3</b>	2	3	-	-
<b>c, d = 5, 9</b>	2	3	5	9
<b>a = b + c</b>	8	3	5	9
<b>b = a - (c * 2)</b>	8	-2	5	9
<b>c = c * b</b>	8	-2	-10	9
<b>a, b = b, a</b>	-2	8	-10	9
<b>a = -a</b>	2	8	-10	9
<b>d = b // c</b>	2	8	-10	-1

**Exercice 1.2** Décrivez le plus clairement et le plus complètement possible ce qui se passe à chacune des trois lignes de l'exemple ci-dessous :

```
>>> largeur = 20
>>> hauteur = 5 * 9.3
>>> largeur * hauteur
930.0
```

**Exercice 1.3** Testez les lignes d'instructions suivantes. Expliquez ce qui se passe :

```
>>> r, pi = 12, 3.14159
>>> s = pi * r**2
>>> print(s)
>>> print(type(r), type(pi), type(s))
```

**Exercice 1.4** Quel est, à votre avis, l'utilité de la *fonction* `type()`?

**Exercice 1.5** Lisez le code suivant et expliquez ce qui se passe. Écrivez les valeurs et les types au fur et à mesure. Si vous ne savez pas, testez dans l'interpréteur Python !

	a, type(a)	b, type(b)	c, type(c)	d, type(d)
a = "2"	'2', str	-	-	-
b = "3"	'2', str	'3', str	-	-
c, d = "4", "9"	'2', str	'3', str	'4', str	'9', str
a = b + c	'34', str	'3', str	'4', str	'9', str
b = int(a) - (int(c) * 2)	'34', str	'26', int	'4', str	'9', str
c = int(c) * b	'34', str	'26', int	'104', int	'9', str
a, b = b, a	'26', int	'34', str	'104', int	'9', str
a = -a	'-26', int	'34', str	'104', int	'9', str
d = b // c	'-26', int	'34', str	'104', int	'9', str

**Exercice 1.6** Que se passe-t-il à la dernière instruction ? Pourquoi ? Que manque-t-il ?

## 2 Exercices sur les entrées

**Exercice 2.1** Écrivez un programme qui permet de saisir le nom de l'utilisateur (avec la fonction `input()`) et de renvoyer "Bonjour", suivi de ce nom.

**Exercice 2.2** Ajoutez au programme le code qui demande à l'utilisateur son année de naissance et qui affiche son âge après lui avoir dit bonjour. L'année courante sera mise dans une variable.

**Exercice 2.3** Écrivez un programme qui demande à l'utilisateur la saisie de deux variables `a` et `b`, puis qui affiche la somme de `a` et `b`.

**Exercice 2.4** Écrivez un programme qui demande à l'utilisateur les coordonnées de deux points dans le plan (donc variables), et qui calcule puis affiche la distance entre ces deux points selon la formule

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Pour calculer un exposant en Python, utilisez l'opérateur `**`. Pour la racine carrée, utiliser l'opérateur de puissance `**` avec la valeur  $\frac{1}{2}$  (ou 0.5).

## 3 Exercices sur l'affichage

Nous avons utilisé jusqu'à présent la fonction `print()` sans en connaître réellement la portée. Les exercices (et tests) suivants ont pour but de vous familiariser avec certaines possibilités de celle-ci.

Par défaut, lorsqu'on appelle la fonction `print()` avec plusieurs paramètres, celle-ci les affiche tous en séquence en les séparant par des espaces puis en ajoutant un caractère de "retour chariot" (retour à la ligne).

Le *prototype* de la fonction `print` est le suivant :

```
print([exp1, exp2, ...][, sep=' '][, end='\n'])
```

Une expression entre crochets indique que celle-ci est facultative. On peut donc optionnellement afficher le résultat de plusieurs expressions à l'écran (`exp1`, `exp2`, etc.), choisir le séparateur (par défaut une espace), et le caractère de fin de ligne (par défaut un retour à la ligne, indiqué par le caractère spécial `\n`).

Pour s'en persuader, testez les lignes suivantes :

```
>>> print("Bonjour", "à", "tous", sep="*")
>>> print("Bonjour", "à", "tous", sep=" ")
>>> print("zut", end="\n\t ah ben ca alors !")
>>> print("tralala", end="\n\t ah ben ca alors !")
```

On remarquera que le '\n' induit un retour à la ligne et que le '\t' induit une tabulation.

Il est donc possible pour afficher plusieurs variables de manière *formatée* d'écrire le code suivant :

```
>>> a = 18
>>> b = 12.93333333
>>> c = "allo"
>>> d = 435.3333333
>>> print("a =", a, "\t b =", b, "\nc =", c, "\t d =", d)
```

Malheureusement si nous avions voulu aligner l'affichage, les tabulations n'auraient pas suffi puisqu'elles ne prennent pas en compte le contenu des variables. Il nous faut alors utiliser un formatage plus rigoureux :

Consulter ce site web pour avoir toute l'information sur les possibilités d'affichage :

[https://python.sdv.univ-paris-diderot.fr/03\\_affichage/](https://python.sdv.univ-paris-diderot.fr/03_affichage/)

Pour pratiquer un peu, écrivez un programme qui demande la saisie de 3 étudiants (nom et numéro d'étudiant) ainsi que leur note pour chacun des deux examens du cours.

Le programme affiche ensuite un tableau récapitulatif contenant les informations suivantes :

- Un rappel des informations saisies
- La moyenne (des deux notes) de chacun des étudiants
- Pour chaque examen, la moyenne des notes des 3 étudiants

Voici un exemple complet d'exécution du programme :

```

Run: exercise3
C:\Users\Dave\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Dave/PycharmProjects/td2_solutions/exercice3.py
Étudiant 1, nom : Alice
Étudiant 1, numéro : 1234
Étudiant 1, note intra : 76
Étudiant 1, note final : 54
Étudiant 2, nom : Bob
Étudiant 2, numéro : 5678
Étudiant 2, note intra : 87
Étudiant 2, note final : 65
Étudiant 3, nom : Carl
Étudiant 3, numéro : 9012
Étudiant 3, note intra : 88
Étudiant 3, note final : 63

Tableau récapitulatif
=====
Nom      | Numéro  | Note intra | Note final | Moyenne
-----|-----|-----|-----|-----
Alice    | 1234    | 76.00     | 54.00     | 65.00
Bob      | 5678    | 87.00     | 65.00     | 76.00
Carl     | 9012    | 88.00     | 65.00     | 76.50

Moyenne à l'intra : 83.67
Moyenne au final : 61.33

Process finished with exit code 0

```

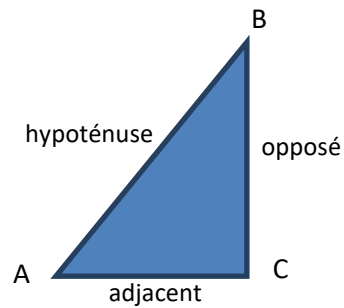
## 4 Programme – Conversions de températures

Créer un nouveau projet (fichier) nommé *exercice4.py*. Le but du programme est de demander à l'utilisateur d'entrer une valeur numérique qui représente la température en Celsius. Le programme doit par la suite saisir la donnée et effectuer les calculs pour convertir celle-ci en Fahrenheit. Finalement, le programme affiche la valeur convertie dans la console en indiquant l'unité de mesure. Afficher le résultat avec deux décimales.

Formule : **Température en °F = (Température en °C \* 1.8) + 32**

## 5 Programme – Triangle

Créer un nouveau projet (fichier) nommé *exercice5.py*. Le but du programme est de trouver les angles d'un triangle rectangle. On demande à l'utilisateur d'entrer la valeur de l'hypoténuse et du côté opposé à l'angle A. On utilise par la suite les formules suivantes pour trouver la valeur des angles en degrés. Affichez les résultats avec trois décimales.



Formules :

$$\text{angleA(radian)} = \text{asin}(\text{opposé} / \text{hypothénuse})$$

$$\text{angleA(degré)} = (180 / 3.1416) * \text{angleA(radian)}$$

$$\text{angleC} = 90$$

$$\text{angleB} = 180 - \text{AngleA} - \text{AngleC}$$