

DM 1 - LES BASES DU LANGAGE PYTHON

Types de base, variables et affectations, expressions, conditions...

Ce **DM** contient une première partie **sur papier**, à compléter directement **sur cette feuille**, et une deuxième partie **sur machine**, à réaliser **sur ordinateur**.

Partie 1 - Exercices sur papier**Exercice 1 - Vrai ou Faux ?**

Question	Votre réponse
1. Après les instructions <code>x = 3</code> , <code>y = 5</code> , <code>x = y</code> , <code>y = x</code> , la valeur de <code>x</code> est 5 et la valeur de <code>y</code> est 3 .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
2. Après les instructions <code>x = 3</code> , <code>y = 5</code> , <code>y == x</code> , <code>x = y</code> , la valeur de <code>x</code> est 5 et la valeur de <code>y</code> est 5 .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
3. L'instruction <code>print(bonjour)</code> va afficher "bonjour".	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
4. La fonction <code>input</code> renvoie une chaîne de caractères .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
5. L'instruction <code>if ...</code> est une instruction conditionnelle .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
6. Avec <code>for i in range(10)</code> , la variable <code>i</code> prend 9 valeurs , puisque la dernière est 9 .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
7. Avec <code>for i in range(9,0,-3)</code> , les valeurs successives prises par la variable <code>i</code> sont 9, 6, 3, 0 .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
8. Si l'on exécuté l'instruction <code>âge = int(input("Veuillez entrer votre âge ! "))</code> , la variable <code>âge</code> créée est de type <code>str</code> (chaîne de caractères).	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
9. Une variable booléenne ne peut prendre que deux valeurs .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux
10. Si <code>a = True</code> et <code>b = False</code> , alors le test <code>b or a and b</code> renvoie <code>True</code> .	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux

Exercice 2

1. En **quelle année** le langage **Python** a-t-il été créé (entourez la bonne réponse) :
- a. 1971
 - b. 1991
 - c. 2001
 - d. 2011

Exercice 2 suite

2. Voici 4 propositions, entourez celles qui sont **des expressions** et soulignez celles qui sont **des affectations** :

- a. $a < b$
- b. $a != b$
- c. $a = b$
- d. $a >= b$

3. On considère les **instructions** suivantes exécutées dans l'ordre : $a=8$, $b=5$, $a==b+1$, $b=b+1$, $a==b+1$, $b=b+1$, `print(a==b+1)`. Quel est le **résultat** affiché à l'issue de la dernière instruction ?

.....

4. Voici une séquence d'instructions :

```
1 n = 7
2 for i in range(2, n):
3     print(i)
```

Combien de fois la fonction `print` à l'intérieur du `for` est-elle exécutée ?

..... fois.

5. Voici une séquence d'instructions :

```
1 x = 4
2 while x > 0:
3     y = 0
4     while y < x:
5         y = y + 1
6         x = x - 1
```

Quelle seront les **valeurs finales** de x et de y ?

$x =$

$y =$

Exercice 3

Écrire une fonction `plus_grand` qui prend en paramètres **trois nombres entiers** et renvoie le **plus grand des deux nombres**.

```
def plus_grand(a, b, c):
```

.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 4

Écrire une fonction `temps` qui prend en paramètres **un nombre entier de secondes** et **affiche le nombre d'heures, de minutes et de secondes** qui correspond.

```
def temps(nbs):
```

.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 5

Écrire une fonction qui prend en arguments **deux mots** (type `str`) et renvoie `True` si les deux mots **commencent par la même lettre** et se **terminent par la même lettre** et `False` sinon.

```
def jeu(mot1, mot2):
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 6

Écrire une fonction `jeu` qui prend en paramètre un **nombre entier n strictement positif**, simule **n fois** le tirage d'un nombre choisi au hasard parmi les nombres **1, 2, 3, 4, 5, 6**, et **renvoie le pourcentage de 6 obtenus**.

On utilisera la fonction `randint` du module `random`.

```
from random import randint
def jeu(n):
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Partie 2 - Exercices sur machine

Pour cette partie, vous travaillerez **sur machine** avec un *IDE* comme **Thonny**. Vous pouvez installer **Thonny** chez vous depuis le site officiel : <https://thonny.org/>

Modalités de rendu : Pour chaque exercice ci-dessous, vous créez un fichier **Python** différent. L'exercice 1 sera rédigé dans un fichier `exercice1.py`, l'exercice 2 dans un fichier `exercice2.py`, etc. Vous créez ensuite une **archive ZIP** contenant tous vos fichiers, que vous nommerez : `<VOTRE_NOM>_DM1.zip`.

Vous pourrez rendre votre travail sur la **page des rendus** de mon site : <https://nsi.erwandemerville.fr/rendus/>.

Exercice 1 - Boucles for imbriquées, Turtle

Écrire un **programme** qui **trace** la **figure** ci-dessous en utilisant **deux boucles imbriquées**. Vous utiliserez le **module Python Turtle**.

Vous pouvez importer **toutes les fonctions** du module en plaçant cette ligne au début de votre programme : `from turtle import *`.



Figure 1: Figure exercice 1

Exercice 2 - Boucle for, utilisation d'un accumulateur

Écrire un **programme** qui **demande** à l'utilisateur de saisir un **entier naturel n** et qui **calcule** la **somme des carrés** des entiers de 1 à n.

Afin de vérifier votre programme, voici un exemple : $1^2 + 2^2 + 3^2 + \dots + 10^2 = 385$

Exercice 3 - Conditions imbriquées

Au jeu de *Mölkky*, chaque joueur marque à son tour de jeu entre **0** et **12 points**, qui viennent s'ajouter à son **score** précédent. Mais gare ! Quiconque **dépasse** le score cible de **51 points** revient immédiatement à **25 points**.

Écrire un **programme** qui **demande** un **score** et un **nombre de points marqués**, et qui **affiche** le **nouveau score** ou **signale** une éventuelle **victoire**.

Sources : *Numérique et Sciences Informatiques 1ère, Balabonski et al., Ed. Ellipses.*

Exercice 4 - Boucle for, conditions, opérateur modulo, filtrage

Écrire un **programme** qui **demande** à l'utilisateur de saisir un **entier naturel n** et qui **affiche** tous ses **diviseurs** les uns après les autres.

Les **diviseurs** devront être tous écrits sur une **même ligne** et **séparés par un tiret**.

Exemple avec les diviseurs de 15 : `1-3-5-15`.