

## TD 1 (programmation) - Affectations et conditions.

### Exercice 1 (Types et opérations)

Donnez pour chacune des instructions suivantes le résultat et le type du résultat.

<code>2 * 5</code>	<code>2 + 1.5</code>	<code>2.0 * 4</code>	<code>int(2.0) * 4</code>
<code>4 / 3</code>	<code>4.0 / 3.0</code>	<code>5 % 2</code>	<code>5 // 2</code>
<code>float(4) * 2</code>	<code>str(4) * 2</code>	<code>float(4) / 2</code>	<code>str(4) / 2</code>
<code>'toto' + str(4)</code>	<code>'toto' * str(4)</code>	<code>'toto' + 'titi'</code>	<code>int('toto') + 'titi'</code>
<code>4 &gt; 3.4</code>	<code>str(4) != 3.4</code>	<code>str(4) == 4</code>	<code>float(4) == float('4')</code>

### Exercice 2 (Nommage)

Rayez parmi les noms suivants ceux qui ne sont pas valides pour une variable ou une fonction.

bonjour	au revoir	Ciao	byebye7
6hello6	good_morning	good-afternoon	Hi!
oui	NON	def	_iut_

### Exercice 3 (Affectations)

Dessinez l'état des cases mémoires pour les trois variables après l'exécution de chaque ligne du programme ci-dessous.

```

1  if __name__ == "__main__":
2      toto = 4
3      toto = 2 * 5
4      toto = toto + 1
5      tata = toto
6      tata = toto + tata * 2
7      toto = 10 + tata
8      toto = titi

```

### Exercice 4 (Échange de variables)

Voici un petit programme:

```

1  if __name__ == "__main__":
2      a = ... # initialisation de a
3      b = ... # initialisation de b
4      a = b + a
5      b = a - b
6      a = a - b
7      print(a)
8      print(b)

```

1. Quel est le comportement des lignes de code ci-dessus quand `a` et `b` contiennent des valeurs de type `int` ? Pour vous aider, essayez avec `a = 1` et `b = 2`.
2. Que se passe-t-il si `a` et `b` contiennent des chaînes de caractères ?

- Proposez une autre méthode pour obtenir le même comportement quels que soient les types des valeurs contenues dans a et b.

### Indices

Utilisez une variable temporaire pour copier et restaurer le contenu d'une des deux variables à échanger.

### Exercice 5 (= ou == ?)

Voici un petit programme:

```
1 if __name__ == "__main__":
2     a ... int(input("Entrez une valeur pour a: "))
3     b ... int(input("Entrez une valeur pour b: "))
4     c ... b - a
5     if c ... 0:
6         print('a et b sont identiques')
```

- Rappelez la différence entre = et ==.
- Remplacez chaque ... par = et == dans le programme ci-dessus afin qu'il ait du sens.

### Exercice 6 (Conditions)

Qu'affiche le programme ci-dessous dans les différents cas suivants ?

- quand a vaut 1 et b vaut 0;
- quand a vaut 2 et b vaut 2;
- quand a vaut 3 et b vaut 0;
- quand a vaut 4 et b vaut 4;

```
1 if __name__ == "__main__":
2     print('1')
3     if a > 2:
4         print('2')
5         if b >= a :
6             print('3')
7         print('4')
8     print('5')
```

### Exercice 7 (Tri)

Complétez le code ci-dessous pour qu'il affiche les valeurs saisies au clavier dans l'ordre croissant. Par exemple, si l'on rentre 4, 0 et 7, on voudra que le programme affiche 0 4 7.

```
1 if __name__ == "__main__":
2     a = int(input("Entrez une valeur pour a: "))
3     b = int(input("Entrez une valeur pour b: "))
4     c = int(input("Entrez une valeur pour c: "))
5     ...
6     print(a, b, c)
```

Remarque: on ne suppose pas que cet objectif est réalisable en une seule ligne.

**Indices**

Procédez à des échanges de valeurs pour que vos variables soient ordonnées correctement.