

17-10-2016

QCM 1

PTSI

Question 1

```
x = 3
y = x/2
t = (x<1)
```

Le type de **y** est :☐ *int* ☐ *bool* ☒ *float*

Le résultat de la division de 3 par 2 est 1.5 qui est de type *float*. Notons que **4/2** donne également un *float* (qui est 2.0).

Question 2 Le type de **t** est :☐ *int* ☐ *float* ☒ *bool*

Le résultat d'une comparaison est de type *bool*.

Question 3

```
x = 2
y = a+1
print(y)
```

Le résultat affiché est :

☐ **a+1** ☐ **1** ☐ **2** ☒ Une erreur

Il y a une erreur car la variable **a** n'a pas été initialisée.

Question 4

```
x = 3+7/2
print(x)
```

Le résultat affiché est :

☐ **7** ☐ **6** ☐ **5** ☒ **6.5**

PYTHON respecte la priorité des opérations : la division est faite avant l'addition.

CORRECTION

Question 5

```
x = 2
y = x+1
x = 4
print(y)
```

Le résultat affiché est :

☐ 4 ☐ 1 ☐ 2 ☒ 3

Lorsque **x** change de valeur, la valeur contenue dans **y** n'est pas mise à jour et reste égale à 3.

Question 6

```
x = 2
y = 4
x = y+1
y = x+1
print(x, y)
```

Le résultat affiché est :

☒ 5 6 ☐ 3 5 ☐ 5 3 ☐ 6 5

Question 7

```
x = 2
y = 5
print(x>2 and y>2)
```

Le résultat affiché est :

☐ True ☐ True False ☒ False ☐ False True

Le résultat du test **x>2** est le booléen **False**, le résultat du test **y>2** est le booléen **True** et le **and** opère sur les booléens : **False and True** donne **False**.

Question 8

```
x = 2
y = 5
if x>2 or (not y>2):
    print("a")
else:
    print("b")
```

Le résultat affiché est :

☐ a ☐ Rien du tout ☒ b

Le test **x>2** donne **False**, le test **y>2** donne **True** donc **not y>2** donne **False**. Finalementn, **False or False** donne **False**. La condition du **if** est fausse, donc c'est la partie **else** qui est effectuée.

Question 9

```
x = 5
if x>2:
    print("a")
elif x>3:
    print("b")
else:
    print("c")
```

Le résultat affiché est :

☐ b puis a ☐ a puis b ☒ a ☐ c

Dans une séquence **if**, **elif**,..., **else** il n'y a qu'une seule branche qui est réalisée. Comme la condition du **if** est vraie, la partie **elif** n'est pas effectuée. Il y a donc seulement affichage de **a**.

Question 10

```
x = 5
if x>2:
    print("a")
if x>3:
    print("b")
else:
    print("c")
```

Le résultat affiché est :

☐ b ☐ c ☒ a puis b ☐ a

Ici ce n'est pas **if**, **elif**, **else** mais **if**, **if**, **else**. La condition du premier **if** est vraie donc on affiche **a** puis on passe au deuxième **if** dont la condition est vraie, on affiche alors **b**.

Question 11

```
x = 10
if x>1:
    x = -4
    print("a")
else:
    x = 5
    print("b")
```

Le résultat affiché est :

☐ b ☒ a ☐ a puis b

Le fait que la condition **x>1** soit vraie fait que la première branche du **if** est choisie. La deuxième branche n'est pas exécutée, même si la valeur de **x** est changée dans la première branche.

Question 12

```
x = 10
if x>1:
    x = -4
    print("a")
if x<=1:
    x = 5
    print("b")
```

Le résultat affiché est :

☐ b ☐ a ☒ a puis b

Comme la condition `x>1` est vraie, les instructions du premier `if` sont exécutées. À la sortie du premier `if`, `x` vaut `-4` de sorte que la condition du second `if` est également vraie et les instructions du second `if` sont donc exécutées.

Question 13

```
x = 10
if x%2==0:
    print("a")
print("b")
```

Le résultat affiché est :

☐ a ☒ a suivi de b ☐ b

La condition `x%2==0` signifie que `x` est pair, ce qui est vrai. Par conséquent `a` est affiché puis, après le `if`, on rencontre l'instruction qui fait afficher `b`.

Question 14

```
for k in range(0,3):
    print(k)
```

Le résultat affiché est :

☐ 1 2 ☐ 0 1 2 3 ☒ 0 1 2 ☐ 1 2 3

On rappelle que `range(0,3)` représente les nombres 0, 1, 2.

CORRECTION

Question 15 On définit une suite (u_n) en posant $u_0 = 1$ et pour tout $n \in \mathbb{N}$: $u_{n+1} = u_n^2 + 1$. On donne $u_1 = 2$, $u_2 = 5$, $u_3 = 26$ et $u_4 = 677$.

```
u = 1
for k in range(0, 3):
    u = u**2+1
print(u)
```

Le résultat affiché est :

☐ 2 ☒ 26 ☐ 677 ☐ 5

On donne à u la valeur u_0 puis on réalise 3 fois (pour $k = 0$, $k = 1$ puis $k = 2$) l'instruction permettant de passer de u_n à u_{n+1} . C'est donc la valeur de u_3 qui est affichée.

Question 16

```
u = 1
for k in range(1, 3):
    u = u**2+1
print(u)
```

Le résultat affiché est :

☐ 677 ☐ 2 ☒ 5 ☐ 26

On donne à u la valeur u_0 puis on réalise 2 fois (pour $k = 1$ puis $k = 2$) l'instruction permettant de passer de u_n à u_{n+1} . C'est donc la valeur de u_2 qui est affichée.

Question 17

```
L = [0, 1, 2]
print(len(L))
```

Le résultat affiché est :

☒ 3 ☐ 4 ☐ 2

On rappelle que `len(L)` représente le nombre d'éléments contenus dans la liste **L**, donc ici 3.

Question 18

```
L = []
L.append(12)
L.append(6)
print(L)
```

Le résultat affiché est :

☒ [12, 6] ☐ [6, 12] ☐ [6] ☐ [12]

On rappelle que l'instruction **L.append** ajoute un nouvel élément à la fin de la liste **L**.

Question 19

```
L = [1, 3, 7]
for k in range(0, len(L)) :
    print(L[k])
```

Le résultat affiché est :

☐ 1 3 ☐ 1 3 7 puis une erreur ☒ 1 3 7 ☐ 3 7

Ici `len(L)` est égal à 3 donc `range(0, len(L))` représente les nombres 0,1,2. La variable `k` va donc prendre successivement ces trois valeurs et on va afficher `L[0]`, `L[1]` puis `L[2]`.

Question 20 L'écriture du nombre 43 en binaire est :

☒ 101011 ☐ 101001 ☐ 110011 ☐ 110101

Voir le cours sur la représentation des nombres.

CORRECTION