

Exercice 1

Après l'affectation suivante, la variable `toto` est de type :

- float                       int                       str                       bool

```
toto = 3 * 5.                      toto = "3 x 5"                      toto = 8 / 3                      toto = 3 > 0
```

N.B. Dans l'énoncé initial l'expression `3*5` était notée par erreur `3 x 5` qui n'est pas syntaxiquement bien formée.

..... Corrigé .....

```
toto = 3 * 5.                      toto = "3 x 5"                      toto = 8 / 3                      toto = 3 > 0
```

float

str

float

bool

Le comportement de python 2 est différent, et produit une variable de type int.

Exercice 2

Après l'instruction suivante, donnez le type et la valeur de la variable `x`.

```
x = str(3*5)                      x = int(3.5)                      x = str("3x5")                      x = int(False)
```

N.B. Dans l'énoncé initial l'expression `3*5` était notée par erreur `3x5` qui n'est pas syntaxiquement bien formée.

..... Corrigé .....

```
x = str(3*5)                      x = int(3.5)                      x = str("3x5")                      x = int(False)
```

Type	str
Valeur	"15"

Type	int
Valeur	3

Type	str
Valeur	"3x5"

Type	int
Valeur	0

L'expression est d'abord calculée, ce qui donne l'entier 15, qui est ensuite converti en `str`.

La conversion d'un réel vers un entier arrondit la valeur.

La conversion ne change rien : la valeur "convertie" est déjà une string.

La constante `False` est associée à la valeur 0, par convention.

Exercice 3

Ecrire un programme qui divise un nombre par deux autant de fois que nécessaire pour arriver en dessous de 1.

On utilisera une variable `val` initialisée à 342, et le programme affichera à la fin de son calcul le nombre de tours qu'il aura fallu.

..... Corrigé .....

```
val = 342
c = 1
while val > 1:
    val /= 2
    c += 1
print ("il a fallu %d tours" % c)
```