Exercice 1, feuille n°6 (LYOU008, 20/21, groupe M)

Créer trois variables `a`, `b`, `c` et initialisez-les de manière à ce qu'elles contiennent une valeur entière.

[1]:
```
a = 12
b = 8
c = 5
```

[2]:
```
type(a)
```

[2]:
```
int
```

Créer une variable `x` réelle en lui affectant la valeur 7.

[3]:
```
x = 7.
type(x)
```

[3]:
```
float
```

Afficher le type de la variable `a`.

[4]:
```
type(a)
```

[4]:
```
int
```

[5]:
```
print(type(a))
```

```
<class 'int'>
```

Affecter à `a` le résultat de la division de `b` par `x`.

[6]:
```
a = b / x
```

Montrer que `a` a «silencieusement» changé de type.

[7]:
```
type(a)
```

[7]:
```
float
```

Afficher successivement le quotient et le reste de la division entière de `b` par `c`.

[8]:
```
print("Quotient", b // c)
print("Reste", b % c)
```

```
Quotient 1
Reste 3
```

[9]:
```
print("Division entière: %d = %d * %d + %d" % (b,c,b//c,b%c))
```

```
Division entière: 8 = 5 * 1 + 3
```

Mettre `a`, `b`, `c` à zéro en une seule instruction.

[10]:
```
a = b = c = 0
print(a)
```

```
0
```

Initialiser la variable booléenne `pos` de sorte qu'elle soit vraie si `x` est négative.

[11]:
```
pos = x < 0
print(pos)
```

```
False
```

Vérifier que la variable est bien booléenne en affichant son type.

[12]:
```
print(type(pos))
```

```
<class 'bool'>
```

Ajouter à `pos` la valeur 4 et afficher le résultat. Comment se fait-il que cette opération soit permise ?

[13]:
```
pos = pos + 4
# L'opération est possible mais pos a été silencieusement convertie
```

[14]:
```
print(pos)
print(type(pos))
```

```
4
<class 'int'>
```

Créer une variable `rep` de type chaîne en lui affectant la chaîne `"un"`.

[15]:
```
rep = "un"
```

Que se passe-t-il quand on tente d'ajouter 4 à la variable `rep` ?

[16]:
```
titi = rep + 4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-16-fe1f7b8a0a32> in <module>
----> 1 titi = rep + 4

TypeError: can only concatenate str (not "int") to str
```

Créer une variable `suite` de type chaîne et lui affecter la chaîne `"deux"`.

[17]:
```
suite = "deux"
```

Afficher le résultat de l'opération `rep + suite`.

[18]:
```
print(rep + suite)
```

```
undeux
```