

**Exercice 1**

Donner le type (nom ou syntaxe Python) de chacune des variables Python ci-contre :

| Variable                        | Type  |
|---------------------------------|-------|
| <code>a = 0.333</code>          | ..... |
| <code>b = "1/3"</code>          | ..... |
| <code>c = [0,333]</code>        | ..... |
| <code>d = (1/3 == 0.333)</code> | ..... |
| <code>e = 0.33//1</code>        | ..... |

**Exercice 2**

Que valent `a` et `b` après la suite de commandes suivantes ?

|                        |          |
|------------------------|----------|
| <code>a = 2</code>     | Réponse: |
| <code>b = a - 3</code> |          |
| <code>a = a + 2</code> | .....    |
| <code>b = b</code>     |          |

Que valent `x`, `y` et `z` après la suite de commandes suivantes?

|                           |          |
|---------------------------|----------|
| <code>x = 5</code>        | Réponse: |
| <code>x /= 2</code>       |          |
| <code>x += x</code>       | .....    |
| <code>y, z = -1, 1</code> |          |
| <code>y == z</code>       |          |

**Exercice 3**

Que renvoie la commande `3*[3,14]` dans la console ?

.....

**Exercice 4**

Que faut-il écrire en Python pour faire afficher la valeur numérique de  $\sqrt{3}$  en l'exécutant ?

.....

**Correction de l'exercice 1**

```
a = 0.333          # flottant
b = "1/3"          # chaîne
c = [0,333]         # liste
d = (1/3 == 0.333)  # booléen
e = 0.33//1         # entier
```

**Correction de l'exercice 2**

```
a = 2
b = a - 3
a = a + 2          # a = 4
b = b              # b = -1

x = 5
x /= 2
x += x             # x = 5.0

y, z = -1, 1
y == z
# y = -1, z = 1
```

**Correction de l'exercice 3**

La commande `3*[3,14]` renvoie la liste

```
[3,14,3,14,3,14]
```

**Correction de l'exercice 4**

Pn peut très simplement écrire:

```
print(3**(1/2))
```

ou alors en important puis en utilisant la fonction `sqrt`:

```
import numpy as np
print(np.sqrt(3))
```