

## 1 Présentation de Pyzo

### 1.1 Le shell Python

#### Exercice 1. Opérations de base

Taper les instructions suivantes et observer :

```
>>> 3+5
8
>>> 3-5
-2
>>> 3*5
15
>>> 2^5
7
>>> 2**5
32
>>> 2e3
2000.0
>>> 60/7
8.571428571428571
```

1. Que représente l'opérateur `**`? C'est l'opérateur puissance.
2. Quel symbole représente le séparateur décimal? C'est le point `.`
3. Comment est notée l'écriture scientifique de  $a \cdot 10^n$ ? On note *aen*.

### 1.2 Enrichir la calculatrice

#### Exercice 2. Différents passages d'un réel à un entier

En utilisant l'aide et en faisant des essais, expliquer l'usage des fonctions :

```
>>> from math import *
>>> floor(2.3)
2
>>> floor(-2.3)
-3
```

Donc, `floor` est la fonction partie entière.

```
>>> from math import *
>>> ceil(2.3)
3
>>> ceil(-2.3)
-2
```

Donc, `ceil` est la fonction qui à un réel  $x$  associe le plus petit entier supérieur à  $x$ .

```
>>> from math import *
>>> round(2.3)
2
>>> round(-2.3)
-2
>>> round(-2.8)
-3
>>> round(2.8)
3
```

Donc, `round` est la fonction qui à un réel  $x$  associe l'entier le plus proche de  $x$ .

```
>>> from math import *
>>> trunc(2.3)
2
>>> trunc(-2.3)
-2
```

Donc, `trunc` est la fonction qui tronque l'écriture de  $x$  pour ne conserver que la partie avant la virgule.

#### Exercice 3. Aide à la frappe

En utilisant l'aide à la frappe (quand on tape le début du nom d'une fonction connue de Python, il propose le nom complet), déterminer :

1. quelle fonction permet de calculer la racine carrée d'un réel?

```
>>> sqrt(2)
1.4142135623730951
>>> sqrt(4)
2.0
```

C'est la fonction `sqrt` du module `math`.

2. que devez-vous taper pour calculer  $(42)!$ ?

```
>>> factorial(42)
1405006117752879898543142606244511569936384000000000
```

C'est la fonction `factorial` du module `math`.

3. quelle est la commande qui correspond au logarithme népérien? et au logarithme décimal?

```
>>> ln(e)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'ln' is not defined

>>> log(e)
1.0
```

La fonction `log` du module `math` calcule le logarithme népérien.

```
>>> log10(10)
1.0
>>> log10(e)
0.4342944819032518
```

La fonction `log10` du module `math` calcule le logarithme en base 10.

## 2 Manipulation des variables

## 3 Les différents types de variables

### 3.1 Les entiers

#### Exercice 4. Chiffre des centaines

1. Proposer une instruction qui donne la valeur du nombre s'écrivant en base dix sous la forme  $abc$  où  $a$ ,  $b$  et  $c$  sont trois entiers strictement inférieurs à 10.  
 $n = 100 * a + 10 * b + c$ .
2. Proposer un algorithme (suite d'instructions) qui permet de déterminer le chiffre des centaines d'un entier  $n$ . Le taper en une ligne dans le shell.

- On calcule le quotient dans la division euclidienne de  $n$  par 100 :  $x = n//100$ .
- On calcule le reste dans la division euclidienne de ce nouvel entier par 10 :  $y = x\%10 = (n//100)\%10$ .

### 3.2 Les flottants

### 3.3 Les booléens

### 3.4 Autres types de variables

## 4 Fonctions

### 4.1 Éditeur et environnement de développement intégré

### 4.2 Syntaxe

#### Exercice 5. Fonction inconnue

```
1 def fonction_inconnue(m,n):  
2     a=(m+n)/2  
3     return a
```

1. Que va rendre l'instruction `fonction_inconnue(3,5)` ?  
La fonction renvoie la moyenne des deux nombres en entrée. Le résultat sera donc ici 4.
2. Proposer une fonction qui calcule l'aire d'un rectangle de côtés  $m$  et  $n$ .

```
1 def aire_rectangle(m,n):  
2     return m*n
```

3. Et pour un carré ?

```
1 def aire_carre(m):  
2     return m**2
```

#### Exercice 6. Produits

1. Écrire une fonction `prod2` qui reçoit en entrée  $a$  et  $b$  et qui renvoie le produit  $ab$ .

```
1 def prod2(a,b):  
2     return a*b
```

2. En utilisant la fonction `prod2`, écrire une fonction `prod3` qui reçoit en entrée  $a$ ,  $b$  et  $c$  et qui renvoie le produit  $abc$ .

```
1 def prod3(a,b,c):  
2     return prod2(prod2(a,b),c)
```