

Exercice 1 (Comme une calculatrice)

Taper dans la console (ou l'interpréteur) l'instruction `10+3` puis
Entrée, vous devriez obtenir ceci :

```
>>> 10+3
13
```

1. Tester chacune des instructions suivantes dans la console et noter quelle opération est effectuée :

`3-10``abs(3-10)``10*3``10**3``10/3``round(10/3)``10//3``10%3`

2. Que renvoie l'instruction `type(5)` ? et `type(5.0)` ? et `type(5+5.0)` ?

3. Quel est le résultat donné par Python de l'opération $3 \times 0,1$?

Exercice 2 (Avec des chaînes de caractères)

1. Dans la console, taper l'instruction `type("Bonjour")`. Que renvoie-t-elle ?

2. Tester les instructions suivantes et donner le résultat obtenu :

`"Bon"+"jour"``2*"bon"``'Bon'*3`

3. Comparer le résultat de chacune des expressions suivantes `"2 baguettes à 1.1 euro coûtent 2*1.1 euros."` et `"2 baguettes à 1.1 euro coûtent "+str(2*1.1)+" euros."`. A quoi sert l'expression `str(2*1.1)` ?

4. Comparer le résultat obtenu avec `"5"+"6"` et `int("5")+int("6")`. A quoi sert l'expression `int("5")` ?

Exercice 3 (Avec des listes)

Les listes sont des tableaux avec une seule ligne qui permettent de stocker des données composites dans une même structure et qui nous seront très utiles par la suite. Pour définir une liste, on place les éléments entre des crochets en les séparant par des virgules.

1. Dans la console, tester chacune des instructions suivantes et noter le résultat obtenu.

`type([1,2,3])``[1,2,3]==[1,3,2]``[1,"B",2<3]``[1,2,3]+[4,5]``[]``3*[1]``2*[0]+[7]``len([1,-2,3,10])`

2. Une liste particulière.

`range(0,10)``list(range(0,10))``list(range(10))``list(range(2,15))``list(range(1,15,2))``list(range(5,-1,-1))`**Exercice 4 (Avec des variables)**

1. Dans la console, taper les instructions suivantes :

```

>>> a
>>> a=3; a; type(a)
>>> b=a+1; b
>>> a="Bonjour"; a; type(a)
>>> b=a+1
>>> b=a+"1"; b
>>> del(a)
>>> a
>>> b
>>> a="Python"; a; type(a); len(a)
>>> a[1]
>>> a[0]; a[4]; a[5]
>>> a=["P","y","t","h","o","n"]; a; type(a); len(a)
>>> a[1]
>>> a[0]; a[4]; a[6]
>>> b

```

2. A quelle instruction algorithmique le symbole = correspond-t-il ?

3. A quoi sert le point-virgule ?

4. Que fait la fonction del ?

5. A quoi servent les instructions a[4] ?

6. Pourquoi l'instruction a[6] produit-elle une erreur ?

Exercice 5 (Pyhon tutor) Saisir l'adresse suivante

<http://www.pythontutor.com/visualize.html#mode=edit>

pour accéder en ligne à Python Tutor et saisir les instructions ci-dessous (à gauche). Cliquer ensuite sur *Lancer l'exécution* puis sur le bouton *Avant >* autant de fois que nécessaire pour remplir le tableau de variables (à droite).

Ligne 1 `a = 7`
 Ligne 2 `b = 8`
 Ligne 3 `c = a + b`
 Ligne 4 `a = c + 1`
 Ligne 5 `b = a - 3`

Contenu des variables	a	b	c
Ligne 1			
Ligne 2			
Ligne 3			
Ligne 4			
Ligne 5			

Exercice 6 On donne les algorithmes suivants :

Algorithme A	Algorithme B
$a \leftarrow 4$	$x \leftarrow 4$
$b \leftarrow a + 8$	$y \leftarrow 3$
$c \leftarrow a - b$	$x \leftarrow y + 3$
$a \leftarrow c + 1$	$y \leftarrow x + 1$
$d \leftarrow c + 2a$	$y \leftarrow y - 8$
$c \leftarrow d - b$	$x \leftarrow x - 2y$
$a \leftarrow c + 3$	$x \leftarrow x / 2$

1. Quelles sont les variables utilisés dans l'algorithme A ? dans l'algorithme B ?
2. Construire et remplir un tableau de variables comme dans l'exercice précédent pour chacun des deux algorithmes.

Exercice 7 (Différentes formes d'affectation)

Affectation augmentée 1. Comparer les instructions suivantes :

```
a = 7
a = a+1
```

```
a = 7
a += 1
```

2. Tester les opérateurs `-=`, `*=`, `/=`, `%=` et noter ce qu'ils font avec des variables de différents types.

Affectations multiples et affectations parallèles Les instructions de la forme `var1 = var2 = val` permettent d'affecter à la fois aux variables `var1` et `var2` la valeur `val`.

On peut également procéder à une affectation parallèle avec des instructions de la forme `var1, var2 = val1, val2` où la variable `var1` prend la valeur `val1` et la variable `var2` prend la variable `val2`.

1. Que contiennent les variables `x` et `y` après l'instruction `x = y = 7` ?
2. Que contiennent les variables `a` et `b` après l'instruction `a, b = 7, True` ? Quelle est le type de chacune d'elle ?
3. Que produit l'instruction `a, b = 7, a - 1` ? Pourquoi ?
4. Testez les trois séquences d'instructions suivantes après avoir initialisé la variable `a` à 7 et la variable `b` à 12.
A quoi servent-elles ?

```
tmp = a
a = b
b = tmp
```

```
a = a + b
b = a - b
a = a - b
```

```
a, b = b, a
```

Exercice 8 (Une calculatrice plus puissante)1. *Bibliothèque math.*

- (a) Saisir l'expression `sqrt(2)`. Qu'obtient-on ? Pourquoi ?
- (b) Charger la bibliothèque `math` en tapant dans la console l'instruction `from math import *` puis "Entrée".
- (c) Que renvoie à présent l'expression `sqrt(2)` ?

2. *Bibliothèque random.*

- (a) Charger la bibliothèque `random` en tapant dans la console l'instruction `import random` puis "Entrée".
- (b) Saisir l'expression `help(random)`. Qu'obtient-on ?
- (c) A l'aide de la question précédente, quelle instruction faut-il saisir pour simuler le lancer aléatoire d'un dé à 6 faces ?

Exercice 9 (Hello World) On donne le script suivant :

```
def nom() :  
    N=input("Veuillez-vous identifier : ")  
    print("Bonjour "+N+". Comment allez-vous ?")
```

- 1. Tapez ce script dans l'éditeur et enregistrez le dans un fichier `HelloWorld.py`.
- 2. Cliquez sur la flèche verte afin de faire interpréter votre code puis testez-le dans la console en tapant `nom()`. A quoi sert l'instruction `nom()` ?
- 3. Expliquer à quoi servent les fonctions `input` et `print`.
- 4. Que contient la variable `N` et quel est son type ?
- 5. Modifier la fonction `nom` en une fonction `age` qui demande l'année de naissance et qui renvoie l'âge de la personne.

Exercice 10 (Définir ses propres fonctions)

Les instructions ci-contre permettent de calculer l'aire d'un rectangle de largeur `larg` et de longueur `long` données.

```
larg = 4
long = 6
airerect = larg * long
```

On peut modifier les valeurs des variables `larg` et `long` puis exécuter à nouveau l'instruction `airerect = larg * long` pour un nouveau calcul mais cela nous oblige à resaisir 3 lignes à chaque fois.

Avec Python, comme avec de nombreux langages de programmation, on peut définir ses propres fonctions. Le mot clé `def` indique que l'on définit une fonction, il faut ensuite donner un nom à cette fonction et, entre parenthèses, donner la liste des paramètres (valeurs dont la fonction a besoin pour fonctionner), le mot clé `return` indique la ou les valeurs que renvoie la fonction.

Nous allons définir une nouvelle fonction qui s'applique à deux nombres, la largeur et la longueur du rectangle, et calcule l'aire du rectangle.

On définit dans l'éditeur de script la fonction ci-contre.
Elle est nommée `airerect` et s'applique à deux arguments `larg` et `long`.

```
def airerect(larg, long) :
    '''Calcule l'aire d'un rectangle
    de dimension larg × long'''
    return larg * long
```

1. Saisir ce script dans l'éditeur sous le nom `Rectangle.py`.
2. A quoi sert le texte écrit entre `'''` ?
3. Quelles sont les étapes à réaliser afin de tester ce script pour un rectangle de dimension 5×7 ?
4. Écrire une fonction `perirect` calculant le périmètre du rectangle. Combien de paramètres lui faut-elle ?

Exercice 11 (Fonctions affines) On donne le script suivant :

```
def f(x) :
    return 3*x + 1
```

1. Quel est le nom de cette fonction ?
2. Combien de paramètre(s) prend-t-elle ? De quel type ?
3. Quelle est la valeur renvoyée par `f(1)` ?
4. On souhaite écrire une fonction plus générique permettant de modéliser les fonctions affines en général.
On rappelle qu'une fonction affine est une fonction f définie sur \mathbb{R} dont l'expression peut s'écrire sous la forme $f(x) = a * x + b$ où a et b désignent deux réels. Par exemple, la fonction f proposée dans la question 1 est affine avec $a = 3$ et $b = 1$.
Définir une fonction nommée `affine` répondant à la question. Ne pas oublier de la documenter.
5. Quelle est la valeur renvoyée par `affine(3, 1, 1)` ? Etait-ce prévisible ?

Exercice 12 (Conversions de température) Le **degré celsius** (symbole $^{\circ}C$) est l'unité de l'échelle de température Celsius. Le **kelvin** (symbole K) est l'unité de température thermodynamique.

La formule suivante $C = K - 273,15$, où C est la température en degré Celsius et K la température en degré kelvin, permet de passer d'une unité à l'autre.

1. Lorsque $K = 12$, quelle est la valeur de C ?
2. Lorsque $C = 25$, quelle est la valeur de K ?
3. Lorsque $C = 0$, quelle est la valeur de K ?
4. Écrire deux fonctions de conversion : l'une permettant de convertir des degrés Celsius en degré Kelvin et l'autre convertissant des degrés Kelvin en degré Celsius. On n'oubliera pas de les documenter.