

6. Les chaînes de caractères

MÉMO

► En langage Python, une **chaîne de caractères** est une suite de plusieurs caractères (chiffres, lettres, symboles) rangés dans un ordre donné. Une chaîne de caractères se note entre guillemets simples ' ' ou doubles " ".

Exemples : • '6Eg*F' est une chaîne de cinq caractères. • "" est une chaîne de caractères vide.

► Les opérations avec des chaînes de caractères

Concaténation : +	Répétition : *	Égalité : ==	Différence : !=	Relation d'ordre : ordre lexicographique
<pre>>>> x = "auto" >>> y = "bus" >>> x+y 'autobus'</pre>	<pre>>>> x = "bla" >>> 3*x 'blablaba'</pre>	<pre>>>> x = "chien" >>> x == "chien" True >>> x == "chine" False</pre>	<pre>>>> x = "chien" >>> x != "chine" True >>> x != "CHIEN" True</pre>	<pre>>>> x = "fiolle" >>> x = "foule" >>> y = "folie" >>> y = "fou" >>> x < y >>> x < y True False</pre>

► Pour déterminer la **longueur d'une chaîne de caractères**, on utilise la fonction `len()`. Les espaces sont comptés comme un caractère.

► On peut **extraire les caractères d'une chaîne** en notant le rang du ou des caractères à extraire entre crochets. Le premier caractère est le caractère de rang 0. Le deuxième est de rang 1, et ainsi de suite. Le rang du caractère à extraire doit être compris entre 0 et `len(chaîne) - 1`.

Exemple

```
>>> x = 'Bonjour à tous'
>>> len(x)
14
```

Exemple : Chaîne de caractères : ' B o n j o u r à t o u s '

Rang : 0 1 2 3 4 5 6 7 8 9 10 11 12 13

Caractère(s) à extraire	Le 4 ^e (ou de rang 3)	Du 1 ^{er} au 3 ^e (ou du rang 0 à 2)	Du 4 ^e au dernier (ou du rang 3 à 13)	Du 2 ^e au 5 ^e (ou du rang 1 à 4)
Instruction et affichage	<pre>>>> x[3] 'j'</pre>	<pre>>>> x[0:3] 'Bon' ou 'Bon'</pre>	<pre>>>> x[3:14] 'jour à tous' ou 'jour à tous'</pre>	<pre>>>> x[3:] 'onjo'</pre>

► On peut utiliser une **boucle bornée avec une chaîne de caractère**. Une instruction Python du type `for caractere in chaîne` fait prendre à la variable `caractere` toutes les valeurs des caractères de la chaîne.

1 On considère la chaîne de caractères `x = 'Ceci est un exercice que je fais avec plaisir !'`. Compléter le tableau ci-dessous.

Instruction		<code>x[13]</code>		<code>x[13:17]</code>		<code>x[7]+x[17:20]</code>
Affichage			'q'			
Question	Quelle est la longueur de la chaîne ?				Quel est le dernier caractère ?	

2 Compléter les affichages qui correspondent aux instructions ci-dessous.

```
>>> abc = 'trois lettres'
```

1. `>>> print(abc)`

2. `>>> print('abc')`

3. `>>> '1'+ '2'+ '3'`

4. `>>> 1 + 2 + 3`

5. `>>> abc + 'd'`



3 On souhaite dénombrer le nombre d'espaces dans une phrase. Pour cela, on propose la fonction suivante.

```
1 def nbre_espaces(phrase):
2     n=0
3     for caractere in phrase:
4         if caractere==" ":
5             n=n+1
6     return n
```

1. Traduire en langage naturel les lignes 4 et 5.

2. Quelle modification faut-il apporter au script pour qu'il affiche le nombre de lettres (y compris les apostrophes) qu'il y a dans la variable *phrase* ?

3. Contrôler le bon fonctionnement du script précédent avec la phrase : « Je n'ai pas peur de me tromper ». Le script doit afficher dans la console :

Le nombre de lettres de : "Je n'ai pas peur de me tromper" est 24.

4. On considère la fonction `compter_lettre` qui, pour une phrase et une lettre données par l'utilisateur, renvoie la fréquence d'apparition de cette lettre parmi les lettres de la phrase.

a. Compléter le script de cette fonction.

```
1 def compter_lettre(phrase, lettre) :
2     x, n, espace = lettre, 0, 0
3     for caractere in phrase :
4         if caractere == x :
5             n = .....
6         elif caractere == ' ' :
7             espace = .....
8     return .....
```

b. Expliquer le calcul de la ligne 8.

4 Les commandes `upper()`, `lower()`, `capitalize()`, `\t` et `\n`.

1. Saisir dans la console Python les instructions suivantes et compléter les affichages.

```
>>> phrase = 'La vie est belle'
```

a. >>> `phrase.upper()`

b. >>> `phrase.lower()`

c. >>> `phrase.capitalize()`

2. En déduire le rôle des commandes `upper()`, `lower()` et `capitalize()`.

3. Compléter le script ci-dessous qui demande à l'utilisateur son prénom, son nom et son âge et affiche, en respectant les majuscules et minuscules le résultat ci-contre :

Salut Prénom NOM
tu as age ans

```
1 def identite(nom, prenom, age):
2     nom=.....
3     prenom=.....
4     age=str(age)
5     print("Salut", ....., ....., "\ntu as",
6           ....., "ans")
```

4. Quelle est l'utilité des instructions `\n` et `\t` ?

5 Compléter la fonction pour qu'elle affiche tous les caractères d'une phrase séparés par des tirets « - ». Par exemple, en saisissant « vive les vacances », le script affiche `v-i-v-e- -l-e-s- -v-a-c-a-n-c-e-s-`.



```
1 def tirets(phrase):
2     for caractere in .....:
3         print(....., "-", end="")
```

7. Le graphe d'une fonction

MÉMO

► Les fonctions mathématiques se représentent graphiquement par des courbes. En langage Python, pour tracer la courbe représentative d'une fonction, on peut procéder de deux manières différentes selon les bibliothèques importées :

- la bibliothèque `pylab` que l'on peut renommer avec le raccourci (ou l'alias) `pb` ;
- les bibliothèques `numpy` et `matplotlib.pyplot` que l'on peut renommer avec les raccourcis respectifs `np` et `plt`.

Exemple : on veut tracer la courbe représentative de la fonction f définie sur $[-3 ; 4]$ par $f(x) = 2x - 3$, c'est-à-dire l'ensemble des points $M(x ; y)$ avec $y = f(x)$.

Avec la bibliothèque <code>pylab</code>	Avec les bibliothèques <code>numpy</code> et <code>matplotlib.pyplot</code>
<pre>1 import pylab as pb 2 pb.axis([-4,5,-10,10]) 3 pb.grid() 4 x = pb.linspace(-3,4,100) 5 y = 2*x-3 6 pb.plot(x,y) 7 pb.show()</pre>	<pre>1 import matplotlib.pyplot as plt 2 import numpy as np 3 plt.axis([-4,5,-10,10]) 4 plt.grid() 5 x = np.linspace(-3,4,100) 6 y = 2*x-3 7 plt.plot(x,y) 8 plt.show()</pre>

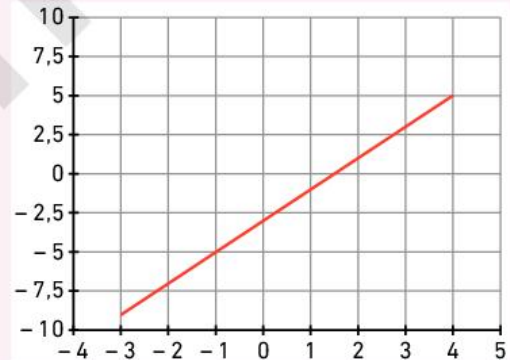
► L'instruction `axis([valeur1, valeur2, valeur3, valeur4])` définit les dimensions du repère. Dans l'exemple ci-dessus, l'axe des abscisses du repère va de -4 à 5 ; l'axe des ordonnées va de -10 à 10 .

► L'instruction `grid()` permet d'afficher un quadrillage.

► L'instruction `linspace(valeur1, valeur2, nombre_points)` définit l'intervalle dans lequel x varie et le nombre de points calculés pour tracer la courbe. x varie ici dans l'intervalle $[-3 ; 4]$ et la courbe sera tracée avec 100 points. La variable y est calculée en fonction de x aux lignes 5 et 6 des deux scripts ci-dessus.

► L'instruction `plot(x, y)` trace les points de coordonnées x et y .

► L'instruction `show()` permet l'affichage de la courbe.



1 Indiquer les dimensions du repère et l'intervalle dans lequel la fonction est définie et tracée.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 plt.axis([-10,8,-16,30])
4 plt.grid()
5 x = np.linspace(-8,8,100)
6 y = x+13
7 plt.plot(x,y)
8 plt.show()
```

2 Compléter le script pour qu'il trace la courbe représentative de la fonction f définie sur $[-4 ; 4]$ par $f(x) = x^2 + 2x - 1$. On se placera dans un repère de dimensions $[-5 ; 5]$ sur l'axe des abscisses et $[-10 ; 30]$ sur l'axe des ordonnées et on tracera 200 points.

```
1 import pylab as pb
2 pb.axis([ ..... ])
3 pb.grid()
4 x = pb.linspace( ..... )
5 y = .....
6 pb.plot(x,y)
7 pb.show()
```



3 On considère la fonction f définie sur l'intervalle $[-1 ; 6]$ par $f(x) = x^2 - 3x + 2$ et la fonction affine g définie sur \mathbb{R} par $g(x) = 3x - 1$.

1. Compléter le script pour qu'il affiche les courbes représentatives de ces fonctions dans un repère de dimensions $[-1 ; 7]$ sur l'axe des abscisses et $[-1 ; 18]$ sur l'axe des ordonnées.

```
1 import pylab as pb
2 pb.axis([ ..... ])
3 pb.grid()
4 x = pb.linspace( ..... , 100)
5 y = .....
6 z = .....
7 pb.plot(x, y)
8 .....
9 pb.show()
```

2. En utilisant le graphique, déterminer les abscisses des points d'intersection des deux courbes.

4 On considère les fonctions f et g définies sur $[0 ; 30]$ par $f(x) = 3x - 40$ et $g(x) = -2x + 50$.

1. Écrire un script qui trace les courbes représentatives des fonctions f et g dans un repère de dimensions $[-1 ; 30]$ sur l'axe des abscisses et $[-1 ; 15]$ sur l'axe des ordonnées, avec 100 points tracés. On importera les bibliothèques `numpy` et `matplotlib.pyplot`.

2. Exécuter le script et déterminer graphiquement une valeur approchée de l'abscisse du point d'intersection.

3. Déterminer par le calcul cette abscisse.

5 Un fabricant de moteurs d'avions fournit chaque année à une compagnie aérienne x moteurs. Le coût moyen de fabrication pour x moteurs (exprimé en millier d'euro) est déterminé par la fonction C définie par $C(x) = -2x^2 + 28x + 30$ sur $]0 ; 25]$.

1. Écrire un script qui trace la courbe représentative de la fonction C dans un repère. Veiller à ce que le repère permette de voir la courbe dans son ensemble pour répondre aux questions.

2. En utilisant le graphique obtenu, déterminer une valeur approchée du nombre de moteurs fabriqués qui rendent le coût moyen maximal.

3. En utilisant le graphique obtenu, déterminer les coûts fixes d'un seul moteur fabriqué.

4. a. En utilisant le graphique obtenu, déterminer le nombre x de moteurs fabriqués qui rendent le coût moyen égal à 30 000 €.

b. Démontrer le résultat conjecturé précédemment.