

<code>f = open('nom_fichier', 'r')</code>	Ouvre le fichier en lecture. Le fichier doit exister et seule la lecture est autorisée.
<code>f = open('nom_fichier', 'w')</code>	Ouvre le fichier en écriture. Si le fichier n'existe pas, il est créé. S'il existe, il est effacé.
<code>f = open('nom_fichier', 'a')</code>	Ouvre le fichier en ajout (append). Si le fichier n'existe pas, il est créé.
<code>f.close()</code>	S'il n'existe pas, ce qu'on écrit est ajouté à la fin du fichier. Ferme le fichier ouvert précédemment. Ligne impérative pour les fichiers ouverts en écriture, car le fichier n'est écrit complètement qu'à la fermeture.
<code>f.read()</code>	Lit tout le fichier d'un coup et le renvoie sous forme d'une seule chaîne de caractères. À réserver aux fichiers de taille raisonnable.
<code>f.readlines()</code>	Lit tout le fichier d'un coup et le renvoie sous forme d'une liste de chaîne de caractères. Chaque élément de la liste correspond à une ligne. Attention, le saut de ligne /n est présent à la fin de chaque chaîne.
<code>f.readline()</code>	Lit une unique ligne du fichier et la renvoie sous forme de chaîne (avec /n au bout). Le curseur de lecture (virtuel) est alors placé au début de la ligne suivante. Si un appel à cette méthode renvoie une chaîne de caractères vide, c'est qu'on est arrivé à la fin du fichier.
<code>f.write(s)</code>	Écrit la chaîne <i>s</i> à la suite du fichier.

On peut parcourir le fichier : `for ligne in f.readlines():`

Enfin, signalons

— la méthode `split()` très utile pour séparer des données contenues dans une chaîne de caractères.

```
chaine = '0,45 ; 0,52'
liste = chaine.split(';')
```

`print(liste)` donne `['0,45 ', ' 0,52']`.

— la méthode `replace` qui permet de remplacer une chaîne de caractères par une autre.

— la méthode `strip` qui permet de supprimer un caractère donné.

### (★) Exercice 1

Suite à un travail de TIPE avec ARDUINO, on récupère le fichier texte suivant :

C:\Users\dsaud\Documents\Lycee\mp\into\Le l

Fichier Édition Recherche Affichage Encod

donnees.txt

	temps (s)	vitesse (m/s)
1		
2	0	0,976046951
3	0,1	0,246953631
4	0,2	0,521714315
5	0,3	1,297157786
6	0,4	1,628274653
7	0,5	1,18319406
8	0,6	1,73148951
9	0,7	2,2093624
10	0,8	2,050888026
11	0,9	2,102517135
12	1	2,121023231
13	1,1	2,524622097
14	1,2	3,057259097
15	1,3	2,831657028
16	1,4	3,124454027
17	1,5	3,343885711
18	1,6	4,072670469
19	1,7	4,03082126
20	1,8	3,752277047
21	1,9	3,839691858
22		

On souhaite visualiser les données en Python.

```

1 fichier = open('donnees.txt', 'r')
2 fichier.readline() # la première ligne ne nous intéresse pas
3 lignes = fichier.readlines() # récupération des lignes suivantes
4 print(lignes)

```

```

['0\t0,976046951\n', '0,1\t0,246953631\n', '0,2\t0,521714315\n',
'0,3\t1,297157786\n', '0,4\t1,628274653\n', '0,5\t1,18319406\n',
'0,6\t1,73148951\n', '0,7\t2,2093624\n', '0,8\t2,050888026\n',
'0,9\t2,102517135\n', '1\t2,121023231\n', '1,1\t2,524622097\n',
'1,2\t3,057259097\n', '1,3\t2,831657028\n', '1,4\t3,124454027\n',
'1,5\t3,343885711\n', '1,6\t4,072670469\n', '1,7\t4,03082126\n',
'1,8\t3,752277047\n', '1,9\t3,839691858\n']

```

1. Compléter.

```

1 L = []
2 for ligne in lignes:
3
4 print(L)

```

```

[['0', '0,976046951\n'], ['0,1', '0,246953631\n'], ['0,2', '0,521714315\n'],
['0,3', '1,297157786\n'], ['0,4', '1,628274653\n'], ['0,5', '1,18319406\n'],
['0,6', '1,73148951\n'], ['0,7', '2,2093624\n'], ['0,8', '2,050888026\n'],
['0,9', '2,102517135\n'], ['1', '2,121023231\n'], ['1,1', '2,524622097\n'],
['1,2', '3,057259097\n'], ['1,3', '2,831657028\n'], ['1,4', '3,124454027\n'],
['1,5', '3,343885711\n'], ['1,6', '4,072670469\n'], ['1,7', '4,03082126\n'],
['1,8', '3,752277047\n'], ['1,9', '3,839691858\n']]

```

2. Enregistrer tous les temps dans une liste `temps`. Attention à bien récupérer des nombres et pas des caractères. Même question pour les vitesses, dans une liste `vitesses`.

3. Tracer les données.