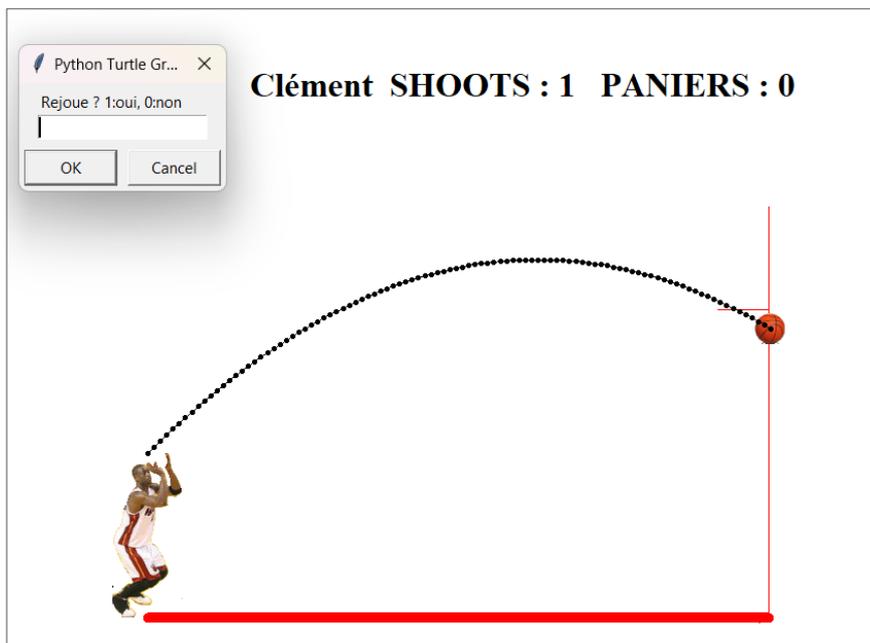


Chapitre 12 - Accès fichiers en python

Dans un code python, on est quelquefois amené à lire et à écrire dans un fichier. Les données écrites dans un fichier sont dites permanentes, elles pourront être utilisées ultérieurement.



Par exemple, dans le code qui permettait de simuler un shoot au basketball, une amélioration possible serait :

- De demander le nom du joueur en début de partie,
- De lire dans un fichier qui reprend l'historique de ce jeu, si ce joueur a déjà joué. Si oui, de lire le nombre de shoots déjà réalisés et parmi eux, ceux qui ont été réussis,
- De réécrire en fin de partie, les données actualisées dans ce fichier.

On voit tout cela dans ce chapitre.

1- ECRITURE DANS UN FICHIER :

Pour créer un fichier texte nommé « basket.txt » et y écrire les lignes ci-contre, on peut utiliser le code python ci-dessous :

```
1 Nom,nShoots,nPaniers
2 Clément,2,1
3 Anaïs,14,12
4 Malone,29,27
5
```

```
1 objBasket = open("basket.txt","w",encoding='utf-8')
2
3 objBasket.write("Nom,nShoots,nPaniers\n")
4 objBasket.write("Clément,2,1\n")
5 objBasket.write("Anaïs,14,12\n")
6 objBasket.write("Malone,29,27\n")
7
8 objBasket.close()
```

"r"	Accès en lecture seule. C'est le mode par défaut. Le fichier doit nécessairement exister
"w"	Accès en écriture seule. Si le fichier n'existe pas, il est créé et s'il existe il est vidé
"x"	Accès en écriture, mais exclusivement en création. Le fichier doit nécessairement ne pas exister
"a"	Accès en écriture seule, mais en fin de fichier (mode « append »). Si le fichier n'existe pas, il est créé
"r+"	Comme "r", mais autorise l'écriture
"w+"	Comme "w", mais autorise la lecture
"x+"	Comme "x", mais autorise la lecture
"a+"	Comme "a", mais autorise la lecture

2- LECTURE D'UN FICHIER EN 1 FOIS :

Pour lire le fichier texte nommé « basket.txt » qui a déjà été créé et qui contient le contenu ci-contre, on peut utiliser le code python ci-dessous :

```
1 Nom,nShoots,nPaniers
2 Clément,2,1
3 Anaïs,14,12
4 Malone,29,27
5
```

```
1 objBasket = open("basket.txt","r",encoding='utf-8')
2
3 contenu = objBasket.read()
4
5 objBasket.close()
```

La variable `contenu` est créée dans l'exécution de la seconde ligne : `contenu = objBasket.read()`

Cette variable est un string et contient la totalité du contenu du fichier :

```
>>> contenu
'Nom,nShoots,nPaniers\nClément,2,1\nAnaïs,14,12\nMalone,29,27\n'
```

3- METHODE SPLIT() :

Split veut dire *diviser* en anglais.

En informatique, c'est une fonction bien connue des développeurs. Elle permet de diviser un string en plusieurs morceaux et de renvoyer ces différents morceaux dans une liste.



En appliquant cette méthode sur le string `contenu` qui contient les caractères suivants :

```
'Nom,nShoots,nPaniers\nClément,2,1\nAnaïs,14,12\nMalone,29,27\n'
```

On obtient par exemple avec le séparateur `"\n"` :

```
>>> contenu.split("\n")
['Nom,nShoots,nPaniers', 'Clément,2,1', 'Anaïs,14,12', 'Malone,29,27', '']
```

En appliquant cette méthode sur le string `contenu` qui contient les caractères suivants :

```
'Nom,nShoots,nPaniers\nClément,2,1\nAnaïs,14,12\nMalone,29,27\n'
```

On obtient par exemple avec le séparateur : `" , "`

```
>>> contenu.split(",")
['Nom', 'nShoots', 'nPaniers\nClément', '2', '1\nAnaïs', '14', '12\nMalone', '29', '27\n']
```

Exercice : Que se passe-t-il en exécutant le code suivant ?

```
1 def lecture() :
2     objBasket = open("basket.txt","r",encoding='utf-8')
3
4     contenuString = objBasket.read()
5     l = contenuString.split("\n")
6     contenuListe = []
7     for i in range(1,len(l)-1) :
8         sousListe = l[i].split(",")
9         contenuListe.append(sousListe)
10
11     objBasket.close()
12     return contenuListe
13
14 # Programme principal
15 sortie = lecture()
16 print(sortie)
17 print(sortie[2])
18 print(sortie[2][0])
19 print(sortie[0][-1])
```

Etat des variables lors de l'exécution :

<code>contenuString</code>	<code>'Nom,nShoots,nPaniers\nClément,2,1\nAnaïs,14,12\nMalone,29,27\n'</code>
<code>l</code>	<code>['Nom,nShoots,nPaniers', 'Clément,2,1', 'Anaïs,14,12', 'Malone,29,27', '']</code>
<code>contenuListe</code>	<code>[]</code>
<code>contenuListe</code>	
<code>contenuListe</code>	
<code>contenuListe</code>	
<code>contenuListe[2]</code>	
<code>contenuListe[2][0]</code>	
<code>contenuListe[0][-1]</code>	

4- LECTURE D'UN FICHER LIGNE PAR LIGNE :

Pour lire le fichier texte nommé « basket.txt » qui a déjà été créé et qui contient le contenu ci-contre, on peut aussi utiliser le code python ci-dessous :

```
1 Nom,nShoots,nPaniers
2 Clément,2,1
3 Anaïs,14,12
4 Malone,29,27
5
```

```
1 objBasket = open("basket.txt","r",encoding='utf-8')
2
3 sortie = []
4 ligne = objBasket.readline()
5 while ligne != "" :
6     ligne = ligne[:-1]
7     ligneEnListe = ligne.split(",")
8     sortie.append(ligneEnListe)
9     ligne = objBasket.readline()
10
11 objBasket.close()
```

Etat des variables lors de l'exécution :

N° étape	Variable	
4	ligne	
6	ligne	
7	ligneEnListe	
8	sortie	
9	ligne	
8	sortie	
8	sortie	[['Nom', 'nShoots', 'nPaniers'], ['Clément', '2', '1'], ['Anaïs', '14', '12']]
8	sortie	[['Nom', 'nShoots', 'nPaniers'], ['Clément', '2', '1'], ['Anaïs', '14', '12'], ['Malone', '29', '27']]

Point Cours pour la lecture et l'écriture dans les fichiers :

- Pour ouvrir un fichier :

```
objFichier = open("fichier.txt","r",encoding='utf-8')
```

- Pour fermer un fichier : `objFichier.close()`

- Pour placer tout le contenu du fichier dans un string :

```
nomString = objFichier.read()
```

- Pour lire la ligne courante qui est placée dans un string :

```
nomString = objFichier.readline()
```


b. EXERCICE :

En exécutant la ligne `ecrire("bac.txt", sortie)` le contenu du fichier nommé bac.txt est modifié afin d'anonymiser les résultats scolaires. Son contenu devient celui de la figure ci-contre. La liste sortie est celle de l'exercice précédent.

1	Maths, Nsi, Physique
2	14, 17, 8
3	1, 7, 4
4	15, 20, 6
5	

⇒ Donner ci-dessous le code python de la fonction `ecrire()` qui permet de réécrire les données contenues dans la liste sortie sans prendre en compte la première valeur de chaque élément.

c. EXERCICE :

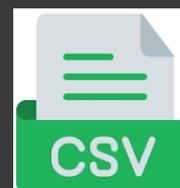
Ecrire le contenu de la variable « a » définie ci-dessous :

```
a = "Il faut manger pour vivre, et non pas vivre pour manger"[:-1].split(" ")
```

6- LECTURE D'UN FICHIER CSV :

Point Cours :

Un fichier au format csv (*Comma Separated Values*) est un fichier texte, composé de lignes qui contiennent des valeurs séparées généralement par une virgule.



De nombreuses données statistiques sont en libre accès pour le grand public (open-data). Les données brutes sont souvent contenues dans un fichier au format csv.