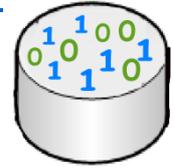


# Utilisation des fichiers

## Apprendre le langage python

Voir le document apprendre\_python3.pdf de Gérard Swinnen



Manipulation des fichiers chapitre 9

<b>9. Manipuler des fichiers.....</b>	<b>111</b>
Utilité des fichiers .....	111
Travailler avec des fichiers .....	113
Noms de fichiers – le répertoire courant .....	113
Les deux formes d'importation .....	114
Écriture séquentielle dans un fichier .....	115
Notes .....	116
Lecture séquentielle d'un fichier .....	116
Notes .....	117
L'instruction break pour sortir d'une boucle .....	118
Fichiers texte .....	119
Remarques .....	120
Enregistrement et restitution de variables diverses .....	120
Gestion des exceptions : les instructions try – except – else .....	122

## Sommaire

<b>1 Un fichier.....</b>	<b>2</b>
<i>C'est quoi un fichier :</i> .....	2
<i>Quels types de fichiers allons-nous utiliser ?</i> .....	3
<b>2 Le code ASCII.....</b>	<b>4</b>
<b>3 Les fichiers textes.....</b>	<b>5</b>
<i>Analyse</i> .....	5
<i>Utiliser les fichiers textes en python</i> .....	6
<b>4 Les fichiers textes structurés .....</b>	<b>7</b>
<i>Le format json</i> .....	7
<i>Utilisation du format json</i> .....	7
<b>5 Retour au calcul de score au scrabble.....</b>	<b>8</b>
<b>6 Pour approfondir .....</b>	<b>9</b>



# 1 Un fichier

## C'est quoi un fichier :

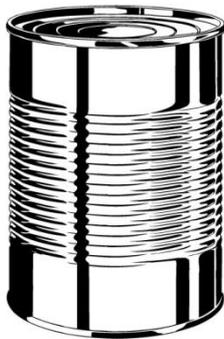
Un fichier est un ensemble de données binaires qui est géré par le système d'exploitation de l'ordinateur avec lequel vous travaillez, PC sous Windows, Linux, Mac. Le fichier est enregistré sur un dispositif de stockage.

Il est ainsi stocké de manière permanente contrairement à la mémoire vive de l'ordinateur qui est effacée à chaque extinction de la machine.

Le fichier peut être lu, écrit, modifié. Le contenu du fichier peut être travaillé uniquement avec un programme compatible avec l'organisation des données à l'intérieur.



Pour bien comprendre prenons un exemple : si une boîte de conserve est sans étiquette comment savoir ce qu'il y a à l'intérieur ? :



Dans le monde des ordinateurs qui peut savoir à quoi correspond ce contenu ?

```
55 6E 65 20 6C 69 67 6E 65 0D 0A 55 6E 65 20 61
75 74 72 65 20 6C 69 67 6E 65 0D 0A 23 20 55 6E
20 63 6F 6D 6D 65 6E 74 61 69 72 65 0D 0A 55 6E
65 20 74 72 6F 69 73 69 E8 6D 65 20 6C 69 67 6E
65 0D 0A 0D 0A 55 6E 65 20 63 69 6E 71 75 69 E8
6D 65 20 6C 69 67 6E 65 0D 0A 23 20 45 74 20 75
6E 20 61 75 74 72 65 20 63 6F 6D 6D 65 6E 74 61
69 72 65 0D 0A 44 65 72 6E 69 E8 72 65 20 6C 69
67 6E 65 0D 0A
```

### C'est quoi ??

- Un programme ?
- Un texte ?
- Une chanson au format mp3 ?
- Une vidéo ?

La seule solution pour le savoir c'est d'essayer d'ouvrir le fichier avec un logiciel comme Notepad++, windows media, visual c++, etc....

Une possibilité reste l'utilisation d'une étiquette permettant d'anticiper le contenu d'un fichier. C'est ce que réalisent les industriels des conserveries.



La gestion des noms de fichiers est simplifiée en utilisant, par convention, des extensions qui permettent de connaître le type de contenu.



## Un nom de fichier



Attention cependant le nom du fichier peut être modifié aussi il n'est pas une garantie absolue pour identifier le contenu.

## Quelques extensions



## Quels types de fichiers allons-nous utiliser ?

Nous allons utiliser des fichiers textes, qui peuvent donc être créés avec nos programmes mais également avec un éditeur de texte de type Notepad++.

Il y a alors deux grandes catégories, des fichiers textes qui possèdent une organisation quelconque, et des fichiers textes dont le contenu doit respecter une certaine organisation dépendant du format utilisé tel que json, csv, ...

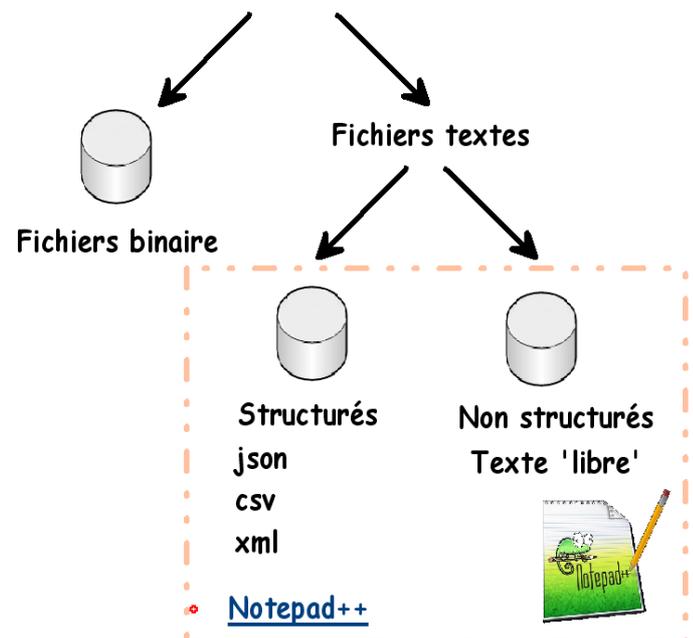
## Logiciel HexEdit 4.0



### HexEdit

Pour observer et éventuellement modifier directement au niveau des octets d'un fichier.

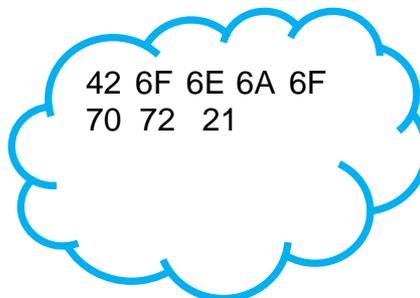
### Principaux types de fichiers de données



## 2 Le code ASCII

# American Standard Code for Information Interchange

Les ordinateurs représentent les lettres, chiffres et autres symboles avec des variables de type caractères. Quelque soit le langage de programmation utilisé dans nos projets la représentation utilise le code ASCII.



MSB \ LSB	0	1	2	3	4	5	6	7
	000	001	010	011	100	101	110	111
0	0000 NUL	001 DLE	010 SP	011 0	100 @	101 P	110 `	111 p
1	0001 SOH	001 DC1	010 !	011 1	100 A	101 Q	110 a	111 q
2	0010 STX	001 DC2	010 "	011 2	100 B	101 R	110 b	111 r
3	0011 ETX	001 DC3	010 #	011 3	100 C	101 S	110 c	111 s
4	0100 EOT	001 DC4	010 \$	011 4	100 D	101 T	110 d	111 t
5	0101 ENQ	001 NAK	010 %	011 5	100 E	101 U	110 e	111 u
6	0110 ACK	001 SYN	010 &	011 6	100 F	101 V	110 f	111 v
7	0111 BEL	001 ETB	010 '	011 7	100 G	101 W	110 g	111 w
8	1000 BS	001 CAN	010 (	011 8	100 H	101 X	110 h	111 x
9	1001 HT	001 EM	010 )	011 9	100 I	101 Y	110 i	111 y
A	1010 LF	001 SUB	010 *	011 :	100 J	101 Z	110 j	111 z
B	1011 VT	001 ESC	010 +	011 ;	100 K	101 [	110 k	111 }
C	1100 FF	001 FS	010 ,	011 <	100 L	101 \	110 l	111
D	1101 CR	001 GS	010 -	011 =	100 M	101 ]	110 m	111 {
E	1110 SO	001 RS	010 .	011 >	100 N	101 ^	110 n	111 ~
F	1111 SI	001 US	010 /	011 ?	100 O	101 _	110 o	111 DEL



Notre programmeur est distrait trouve et corrige son erreur !



Pour notre ordinateur voilà à quoi ressemble le caractère 'A'

\$41



Écrit ton nom en code ASCII



### 3 Les fichiers textes

#### Analyse

Voilà un exemple de fichier texte vu avec l'éditeur Notepad++ :

```

test.txt
1 Une ligne
2 Une autre ligne
3 # Un commentaire
4 Une troisième ligne
5
6 Une cinquième ligne
7 # Et un autre commentaire
8 Dernière ligne
9
    
```

```

55 6E 65 20 6C 69 67 6E 65 0D 0A 55 6E 65 20 61 Une ligne..Une a
75 74 72 65 20 6C 69 67 6E 65 0D 0A 23 20 55 6E utre ligne..# Un
20 63 6F 6D 6D 65 6E 74 61 69 72 65 0D 0A 55 6E commentaire..Un
65 20 74 72 6F 69 73 69 E8 6D 65 20 6C 69 67 6E e troisième lign
65 0D 0A 0D 0A 55 6E 65 20 63 69 6E 71 75 69 E8 e....Une cinquiè
6D 65 20 6C 69 67 6E 65 0D 0A 23 20 45 74 20 75 me ligne..# Et u
6E 20 61 75 74 72 65 20 63 6F 6D 6D 65 6E 74 61 n autre commenta
69 72 65 0D 0A 44 65 72 6E 69 E8 72 65 20 6C 69 ire..Dernière li
67 6E 65 0D 0A gne..|
    
```

Le même fichier vu directement au format binaire :

Les caractères accentués sont codés en ANSI, extension de la table ASCII.

1252 WINDOWS LATIN 1 (ANSI)

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	0	@	P	`	p	NOT USED	NOT USED	NBSP	°	À	Ð	à	ð	
1	!	1	A	Q	a	q	NOT USED	‘	ı	±	Á	Ñ	á	ñ
2	"	2	B	R	b	r	,	’	€	²	Â	Ò	â	ò
3	#	3	C	S	c	s	f	“	£	³	Ã	Ó	ã	ó
4	\$	4	D	T	d	t	„	”	¤	´	Ä	Ö	ä	ö
5	%	5	E	U	e	u	...	•	¥	µ	Å	Ö	å	õ
6	&	6	F	V	f	v	†	-	ı	¶	Æ	Ö	æ	ö
7	'	7	G	W	g	w	‡	-	§	·	Ç	×	ç	÷
8	(	8	H	X	h	x	^	~	¨	ˆ	È	Ø	è	ø
9	)	9	I	Y	i	y	%	™	©	ı	É	Û	é	ù
A	*	:	J	Z	j	z	Š	š	ª	º	Ê	Û	ê	ú
B	+	;	K	[	k	{	<	>	«	»	Ë	Û	ë	û
C	,	<	L	\	l		Œ	œ	-	¼	Ï	Û	ï	ü
D	-	=	M	]	m	}	NOT USED	NOT USED	SHY	½	Í	Ý	í	ý
E	.	>	N	^	n	~	NOT USED	NOT USED	®	¾	Î	Þ	î	þ
F	/	?	O	_	o	~	NOT USED	NOT USED	™	ˆ	Ï	ß	ï	ÿ



Comparer les deux vues du même fichier. Que peut-on observer à la fin de chacune des lignes ?



## Utiliser les fichiers textes en python

L'exemple de code ci-dessous utilise python pour créer un fichier texte, puis pour le relire :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from os import chdir
path = r'G:\GPRO\USB\ICN2nD\Preparation ICN ISN 2017 2018\Debuter en python'
chdir(path)

monFichier = open('MonFichier.txt','a')
monFichier.write('Bonjour, fichier !\n')
monFichier.write(" Aujourd'hui grand soleil")
monFichier.close()

monFichier = open('MonFichier.txt','r')
print(monFichier.read())
monFichier.close()

# Lecture des douze premiers caractères
monFichier = open('MonFichier.txt','r')
print(monFichier.read(12))
monFichier.close()
```

↑  
Modifier la chaîne de caractères pour l'adapter à votre dossier de travail.

↪ Coder et tester cet exemple.

Notez les étapes indispensables dans l'utilisation des fichiers :

1. **ouvrir le fichier** commande `monFichier = open('MonFichier.txt','a')`
  - a. 'a' en mode append réécriture du fichier depuis le début
  - b. 'r' en mode lecture seule
2. **utiliser le fichier**
  - a. pour écrire `monFichier.write`
  - b. pour lire `monFichier.read` `monFichier.read(x)` `monFichier.readLine()`
3. **fermer le fichier**
  - a. `monFichier.close()`

↪ Coder et tester l'exemple p.142 du livre de Swinnen.

```
test.txt
1 Une ligne
2 Une autre ligne
3 # Un commentaire
4 Une troisième ligne
5
6 Une cinquième ligne
7 # Et un autre commentaire
8 Dernière ligne
9

test_f.txt
1 Une ligne
2 Une autre ligne
3 Une troisième ligne
4
5 Une cinquième ligne
6 Dernière ligne
7
```



## 4 Les fichiers textes structurés

### Le format json<sup>1</sup>

Le format json est un format de texte structuré très utilisé sur internet. Ce format est traité par python. Mais les fichiers peuvent être également créés par un éditeur de type Notepad++.

Le format est simple c'est une collection de paire "nom":"valeur" écrites à la suite les unes des autres entre deux accolades. Exemple :

```
{
  "titre_album":"Abacab",
  "groupe":"Genesis",
  "annee":1981,
  "genre":"Rock"
}
```

Les valeurs possibles sont :

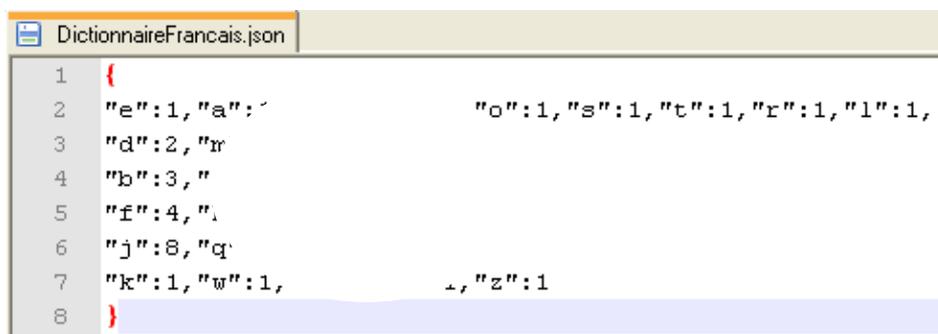
- Une chaîne de caractères: "titre": "Le format json", "description": "Le format <strong>simple</strong> et <strong>léger</strong>, "contenu": "<p>L'avantage de json est son incroyable simplicité d'apprentissage et de mise en oeuvre. C'est le \"Petit Poucet\" de l'échange de données.</p>"
- Un nombre (pas de guillemets requis dans ce cas): "pi": 3.14, "g": 9.81, "v\_son": 340
- Un tableau: [...]
- Un objet: {...}
- D'autres valeurs possibles: un booléen (true ou false), null, rien ("alsanaute": true, "autrenaute": null, "bisounaute": ""). Attention, ces valeurs doivent être écrites en minuscule.

Pour plus de détails consulter le site indiqué en référence.

### Utilisation du format json

Pour revenir à nos travaux sur le calcul du score au scrabble nous allons utiliser le format json pour enregistrer nos dictionnaires sous la forme d'un fichier.

➡ Créer le fichier DictionnaireFrancais.json avec Notepad++ :



```
DictionnaireFrancais.json
1 {
2   "e":1, "a":1, "o":1, "s":1, "t":1, "r":1, "l":1,
3   "d":2, "r
4   "b":3, "
5   "f":4, "i
6   "j":8, "q
7   "k":1, "w":1, "z":1
8 }
```



<sup>1</sup> <https://www.alsacreation.com/article/lire/1675-json-stockage-leger-pratique-donnees-multitypes.html>  
consulté le 13/11/17

## Coder et tester ensuite la lecture du fichier obtenu

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import json

from os import chdir
path = r'G:\GPRO\USB\ICN2nD\Preparation ICN ISN 2017 2018\Python json'
chdir(path)

json_data=open('DictionnaireFrancais.json')
data = json.load(json_data)
json_data.close()

print(data)

document = json.dumps(data, indent=4)

print(document)
print(type(data))
```

## 5 Retour au calcul de score au scrabble

---

Intégrons notre table de valeurs dans le calcul des scores. La lecture des informations au format json est gérée par python comme un dictionnaire. Le dictionnaire est constitué de champ dénommé clé 'keys ligne 42' et on peut alors connaître la valeur associée par 'get ligne 44'

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  from os import chdir
5  path = r'G:\GPRO\USB\ICN2nD\Preparation ICN ISN 2017 2018\Python json'
6  chdir(path)
7
8
9  # Saisie du mot à calculer
10 print('\n')
11 motSaisi=input("Entrez votre mot : ")
12 print('\n')
13
14 # Définition du tableau de valeurs des lettres
15 # Lecture du dictionnaire au format json
16 json_data=open('DictionnaireFrancais.json')
17 tableDesLettres = json.load(json_data)
18 json_data.close()
```



```

20 listeCaractereAutorises="abcdefghijklmnopqrstuvwxy"
21 |
22 # Mettre le mot saisi en minuscule comme
23 # le dictionnaire
24 motSaisi=motSaisi.lower()
25 |
26 # Tester si les caractères sont OK
27 testChaine = True
28 for c in motSaisi:
29     if c in listeCaractereAutorises:
30         print(c, ' ok')
31     else:
32         print(c, ' FAUX')
33         testChaine = False
34 |
35 print('\n')
36 |
37 # Si OK calcul du score, sinon affichage d'un message d'erreur
38 |
39 if testChaine:
40     scoreMot = 0
41     for c in motSaisi:
42         for cle in tableDesLettres.keys():
43             if c == cle:
44                 scoreMot = scoreMot + tableDesLettres.get(cle)
45 |
46 |
47     print('Mot saisi ',motSaisi)
48     print('Score : ', scoreMot)
49 |
50 else:
51     print('Mot saisi ',motSaisi)
52     print('Caractères interdits calcul impossible')
53 |
54 print('\n')

```

## 6 Pour approfondir

---

 Pratiquer les exemples du chapitre n°9 de Swinnen.

 Voir également ici, consulté le 13/11/17.

[https://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiOg4Xku7zXAhUJFOwKHaGrA3gQFggtMAA&url=https%3A%2F%2Feric.univ-lyon2.fr%2F~ricco%2Fcours%2Fslides%2FPE%2520-%2520fichiers%2520sous%2520python.pdf&usg=AOvVaw2tCK2l4joYqePGtQv7l\\_ok](https://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiOg4Xku7zXAhUJFOwKHaGrA3gQFggtMAA&url=https%3A%2F%2Feric.univ-lyon2.fr%2F~ricco%2Fcours%2Fslides%2FPE%2520-%2520fichiers%2520sous%2520python.pdf&usg=AOvVaw2tCK2l4joYqePGtQv7l_ok)

