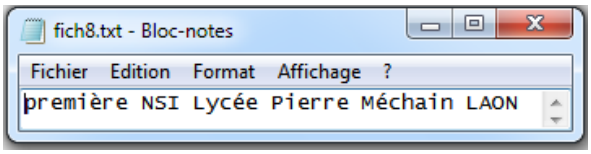
Encodage des caractères

Cours

⌦ Pas la peine de refaire le cours, tout est là : www.infoforall.fr/act/python/encodage-des-textes/

Une chaine de caractères n'est pas stockée dans la machine sous forme de caractères mais sous forme de nombres.




```
01110000 01110010 01100101 01101101 01101001 11000011
10101000 01110010 01100101 00100000 01001110 01010011
01001001 00100000 01001100 01111001 01100011 11000011
10101001 01100101 00100000 01010000 01101001 01100101
01110010 01110010 01100101 00100000 01001101 11000011
10101001 01100011 01101000 01100001 01101001 01101110
00100000 01001100 01000001 01001111 01001110
```

⇨ il faut donc établir une correspondance entre un caractère et un nombre.

⇨ la **table ASCII** a été inventée (en 1963) pour établir une correspondance **standardisée**.

Par exemple :

H -> 72
e -> 101
l -> 108
l -> 108
o -> 111

Voici un extrait de la table ASCII (0 - 127)

Décimal	Hexadécimal	Binaire	Caractère
⋮	⋮	⋮	⋮
032	20	00100000	Espace
⋮	⋮	⋮	⋮
065	41	01000001	A
066	42	01000010	B
067	43	01000011	C
⋮	⋮	⋮	⋮
097	61	01100001	a
098	62	01100010	b
099	63	01100011	c
⋮	⋮	⋮	⋮

On stocke alors les valeurs numériques sous forme binaire (→ enregistrées sur un disque ou envoyées sur le réseau).
Exemple : le texte 'Hello' correspond à la suite d'octets 01001000 01100101 01101100 01101100 01101111

⇨ La table ASCII a été conçue à l'origine pour contenir uniquement les caractères de bases de la langue anglaise.
Il a fallu trouver d'autres solutions pour donner une correspondance numérique aux caractères manquants.
On a créé différentes extensions de la table ASCII : latin-1 (iso8859-1), cp1252 (ANSI), latin-9 (iso8859-15) …
⇨ Un **standard** a finalement été trouvé : **la table UNICODE**.

La table UNICODE est une liste des caractères avec
- un "point de code" (*codepoint*) ≫ un nombre entier unique donné en général en hexadécimal
- un caractère
- un nom

Voici quelques exemples : https://www.unicode.org/charts/fr/

Point de code	caractère	nom
00E7	ç	LETTRE MINUSCULE LATINE C CÉDILLE
00E8	è	LETTRE MINUSCULE LATINE E ACCENT GRAVE
00E9	é	LETTRE MINUSCULE LATINE E ACCENT AIGU

2658	♘	CAVALIER BLANC DU JEU D'ÉCHECS
266C	♬	DEUX DOUBLES CROCHES RAMÉES
2684	⚄	FACE DE DÉ-5

La version Unicode 15.0 de septembre 2022 contient 149 186 caractères (plus d'un million de caractères possibles).

⚠⚠ **ATTENTION** ⚠⚠

La table UNICODE ne dit pas du tout comment sont stockés les caractères dans la machine.
Il faut ensuite utiliser un encodage pour transformer le point de code en valeur binaire.

Passer du "point de code" au nombre binaire enregistré sur la machine : l'encodage

Il y a de nombreux encodages possibles :

- ASCII : caractères codés sur 7 bits, 127 caractères possibles
- latin-1 (iso8859-1) : caractères codés sur 8 bits, compatible avec ASCII sur les 128 premiers caractères
- latin-9 (iso8859-15) : mise à jour de latin-1 (ajout de quelques symboles : €, œ, …)
- cp1252 (ANSI) : extension de latin-1, historiquement utilisé sur les systèmes Microsoft Windows
- UTF-8 : caractères codés sur 1 à 4 octets, standard actuel de l'encodage
- UTF-16 : caractères codés sur 2 ou 4 octets
- UTF-32 : taille fixe, 4 octets par caractère

latin-1, latin-9 et UTF8 sont compatibles avec ASCII sur les 128 premiers caractères

Pour voir les tables de caractères des différentes normes : http://www.kostis.net/charsets/

Table d'encodage UTF8 (Universal Transformation Format)

Tableau permettant de passer du point de code UNICODE à l'encodage en UTF-8 (découpage des bits par quartets)

Point de code			Encodage du point de code en UTF8
Compris entre (*décimal*)	Compris entre (*hexadécimal*)	écriture binaire *(séparée par quartets)*	
0 et 127	00 et 7F	xxxxxxx (7 bits)	0xxxxxxx
128 et 2047	080 et 7FF	xxxxxxxxxxx (11 bits)	110xxxxx 10xxxxxx
2048 et 65535	0800 et FFFF	xxxxxxxxxxxxxxxx (16 bits)	1110xxxx 10xxxxxx 10xxxxxx
65536 et 1114111	010 000 et 10 FFFF	xxxxxxxxxxxxxxxxxxxxx (21 bits)	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

---

Exemples :

Voici l'encodage décimal et binaire en UTF8 de la chaine de caractère : "ABCDE"

```
[65, 66, 67, 68, 69]
['01000001', '01000010', '01000011', '01000100', '01000101']
```

⇨ Il n'y aucun caractère spécial ni accentué. Chaque caractère est codé sur un octet.
⇨ L'encodage UTF8 est identique à l'encodage ASCII pour les caractères dont le point de code est inférieur à 127.
On reconnait que le point de code est inférieur à 127 car les MSB sur un octet est égal à 0.

Voici l'encodage décimal et binaire en UTF8 de la chaine de caractère : "C'est écrit♫"

```
[67, 39, 101, 115, 116, 32, 195, 169, 99, 114, 105, 116, 226, 153, 172]
['01000011', '00100111', '01100101', '01110011', '01110100', '00100000',
 '11000011', '10101001',
 '01100011', '01110010', '01101001', '01110100', '11100010', '10011001', '10101100']
```

⇨ Le é est encodé avec deux octets :

```
'11000011', '10101001'                          [195, 169]
```

⇨ Le ♫ est encodé avec trois octets :

```
'11100010', '10011001', '10101100'              [226, 153, 172]
```