- Codage du texte -

Pour commencer, précisons ce qu'est un **fichier de texte pur** :

- c'est un fichier dans lequel seuls les caractères sont enregistrés,
- sans aucune mise en forme comme une taille ou une couleur de police.

Pour créer un fichier texte, il faut utiliser un **éditeur de texte** :

- le bloc-notes, Notepad++ ou Sublime Text sous Windows ;
- Vi ou Nano sous Linux.

Un fichier texte peut être enregistré sous de nombreux **formats de fichiers**, correspondants chacun à une extension particulière sous Windows :

- .txt pour des fichiers textes ;
- .htm ou .html pour des pages web ;
- .py pour des fichiers de programmes en Python ;
- .js pour des fichiers de programmes en Javascript ;
- .php pour des fichiers de programmes en Php.

Cette extension permet uniquement de dire au système d'exploitation Windows quel programme utiliser pour ouvrir et lire le fichier au moment du double-clic.

Sous linux, on tape le nom du programme puis celui du fichier en ligne de commande :

python3 essai.py, pour lancer le script
vi essai.py, pour éditer le contenu du fichier

Exercice :

- Créer un dossier **exemples** dans votre dossier de travail.
- Créer un fichier texte **essai.txt** dans ce dossier.
- Enregistrer une phrase dans ce fichier puis fermer-le.
- Enregistrer une copie de ce fichier nommée **essai**, sans l'extension .txt !
- Ouvrir le fichier et modifier la phrase.
- Enregistrer une copie de ce fichier nommée **essai.html** puis fermer-le.
- Ouvrir ce nouveau fichier.
- Enregistrer le code source d'une page web dans un fichier **page.txt**.
- Renommer ce fichier en **page.html**.
- Ouvrir ce nouveau fichier dans un navigateur et observer ce qui ne fonctionne pas.

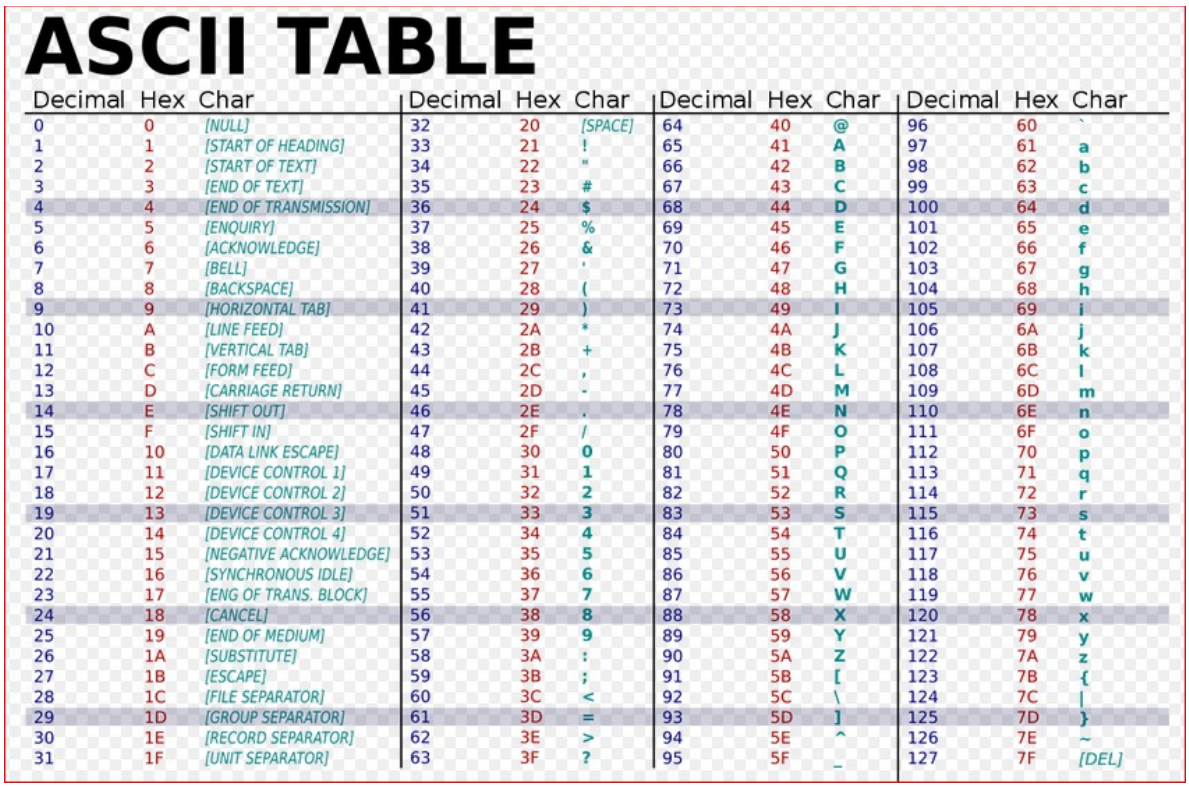
La table de caractères ASCII :

https://fr.wikibooks.org/wiki/Les_ASCII_de_0_%C3%A0_127/La_table_ASCII

C'est un standard de codage américain des années 60 comportant :

- des caractères de contrôle (saut de ligne, tabulation, etc.) ;
- les lettres majuscules et minuscules non accentuées ;
- les 10 chiffres arabes ;
- des signes de ponctuation ;
- des symboles mathématique.

Le **codage** des caractères s'effectue **sur 1 octet**
Le bit de poids fort étant utilisé comme bit de parité (contrôle des erreurs), il reste **7 bits** pour coder les caractères, donc **128 caractères** possibles, dont 95 imprimables :



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Exemples :

A est codé 65_{10} en décimal et $100\ 0001_2$ en binaire
a est codé 97_{10} en décimal et $110\ 0001_2$ en binaire

Comme il faut ajouter $32 = 2^5$ pour passer du codage des majuscules aux minuscules, cela revient à mettre le bit de poids 5 à 1 dans le codage binaire d'une majuscule pour obtenir celui de la lettre minuscule correspondante : B est codé $100\ 0010_2$, et b est codé $110\ 0010_2$.

On peut donc retenir que les caractères de la table **ASCII** sont codés sur **1 octet**.

Les normes ISO-8859-1 (Latin-1) et ISO-8859-15 (Latin-9) :

Cet encodage a été fait pour l'alphabet d'Europe Occidental et comporte 191 caractères.
L'encodage Latin-9 a plus tard intégré l'**œ** et le symbole **€** par exemples.
Le problème est qu'il est incompatible avec les systèmes d'encodages étrangers, car des mêmes points de code vont correspondre à des caractères différents.

L'unicode et l'encodage utf-8 : https://unicode-table.com/fr/

L'unicode est un standard qui vise à établir une table de caractères (définition des caratères + point de code) regroupant le plus de langues possible.

L'encodage **utf-8** se fait sur **1 à 4 octets**.

Les 128 caractères de la table ASCII ont le même codage en utf-8 :

on met le bit de poids fort à 0 pour obtenir leur codage utf-8 sur 1 octet :
la lettre A sera donc codée par $100\ 0001_2$ en ASCII, et $0100\ 0001_2$ en utf-8.

Les bits de poids fort du premier octet forment une suite de 1 indiquant le nombre d'octets pour coder le caractère, suivi d'un zéro pour indiquer la fin des 1.
Les octets suivants commencent tous par le bloc binaire 10.

Représentation binaire en utf-8	Signification
0 *xxx xxxx*	1 octet codant 1 à 7 bits (ASCII)
110 *x xxxx* 10 *xx xxxx*	2 octets codant 8 à 11 bits
1110 *xxxx* 10 *xx xxxx* 10 *xx xxxx*	3 octets codant 12 à 16 bits
1111 0 *xxx* 10 *xx xxxx* 10 *xx xxxx* 10 *xx xxxx*	4 octets codant 17 à 21 bits

Les octets ne sont donc pas utilisés complètement.

Exercice : Le symbole œ a pour valeur décimale 339. Donner son codage en utf-8.
Donner le codage utf-8 du caractère de valeur décimale 2845.