

LE JEU DU PENDU

Crédit : Didier Müller, nymphoMath.ch

Matériel : Un dossier pendu.zip contenant le jeu du pendu et un fichier contenant une liste de mot.

A quoi sert cette mystérieuse ligne dans le code d'une page web ?

```
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"/>
```

Vous avez déjà croisé des mots comme **charset**, **ASCII**, **UTF-8**, **ISO-8859**, **latin-1**... et vous vous demandez ce que sont ces mots un peu barbare ?

Comment fait-on alors pour écrire du texte ?

Tout simplement, on associe à chaque caractère (une lettre, un signe de ponctuation, une espace...) un nombre. Un texte est alors une séquence de tels nombres, ce qu'on appelle une « chaîne de caractères ».

4A27616920767520756E652061726169
676EC3A96520736F7274697220647520
766964C3A96F70726F6A656374657572
2E



Pouvez-vous repérer dans cette « chaîne de caractères » des codages déjà vu ?

.....

Quelle phrase est « cachée » dans cette chaîne de caractères ?

Pour vous aider, utiliser la page 2 de cette activité !

.....

.....

Saisissez votre phrase décodée dans notepad++, sauvegardez votre texte sous différents encodages. Qu'observez-vous ?

.....

.....

Au Commencement ...

Le code **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) est une norme d'encodage informatique des caractères alphanumériques de l'alphabet latin. La norme **ASCII** établit une correspondance entre une représentation numérique des caractères de l'alphabet latin ainsi que les symboles et les signes. Par exemple, le caractère « A » est associé à 65 et « a » à 97. Les nombres du code ASCII vont de 0 à 127.

Le codage **ASCII** (voir tableau ci-dessous) est souvent complété par des correspondances Supplémentaires afin de permettre l'encodage informatique d'autres caractères, comme les caractères accentués par exemple. Cette norme s'appelle **ISO-8859** et se décline par exemple en **ISO-8859-1** lorsqu'elle étend l'**ASCII** avec les caractères accentués d'Europe occidentale.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	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	Neg. 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	End 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	Substitution	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	□

Triez par ordre croissant les mots "Avant", "après" et "Pendant", qu'obtenez-vous ?

.....

Pourquoi ?

.....

Et en python...

En Python, la fonction qui donne de code ASCII d'un caractère s'appelle `ord`.

Ainsi, `ord('A')` donnera comme résultat 65.

Inversement, pour écrire le caractère correspondant à un code ASCII, on utilisera la fonction `chr` :

`chr(65)` donnera comme résultat 'A'.

Le pendu - règles du jeu

Le jeu du pendu consiste à retrouver un mot le plus vite possible (avant que le dessin du pendu soit terminé) en proposant des lettres. Si la lettre appartient au mot, elle est écrite aux bons emplacements, sinon on continue de dessiner le pendu.

Code :

```
# Le jeu du pendu
```

```
from random import choice
```

```
fichier = open("liste_mots.txt", "r")
liste_mots = fichier.readlines()
mot = choice(liste_mots)
mot = mot.rstrip()
fichier.close()
```

```
mot_devine = "-" * len(mot)
print(mot_devine)
nbr_essais = 0
```

```
while mot_devine != mot:
    lettre = input("Entrez une lettre ou '?' pour abandonner : ")
    lettre = lettre[0] # Evite des erreurs si un mot est entré au lieu d'une lettre
    if lettre == '?':
        print('Le mot était', mot)
        break
    lettre = lettre.upper()
    for i in range(len(mot)):
        if lettre == mot[i]:
            mot_devine = mot_devine[:i] + lettre + mot_devine[i+1:]
    print(mot_devine)
    nbr_essais += 1

if mot == mot_devine:
    print('Bravo ! Le mot', mot, 'a été trouvé en', nbr_essais, 'coups')
```

Vous veillerez à placer le fichier contenant la liste des mots dans le même répertoire que votre programme python.

Mission

Vous veillerez à toujours conserver un programme en état de fonctionnement en enregistrant successivement vos programmes avec une version v1, v2, v3 ...

Mission 1 :

Et si on jouait en hexadécimal ?

Sans modifier votre liste de mots, faites en sorte que le joueur puisse saisir le code hexadécimal d'une lettre. Attention, le code hexadécimal est sur plusieurs caractères !

Mission 2 :

Et si on jouait en UTF-8 ?

Reprenez le code d'origine du jeu du pendu, prenez en compte que le joueur peut à présent jouer avec des lettres accentuées.

Pensez à modifier votre liste de mot pour créer un *jeu de test* cohérent.

Définition :

- **répertoire (ou jeu) de caractères** (*character set* en anglais, abrégé *charset*) : désigne l'ensemble des **caractères** dont on a besoin pour coder un texte en mémoire
- Les caractères sont regroupés dans un **registre de caractères** (également appelé **répertoire**), appelé ensuite **registre de caractères codés** lorsqu'un chiffre précis est attribué à chaque caractère, nommé **point de code**. Ces points de code sont représentés dans l'ordinateur par un octet ou plus.
- L'**encodage de caractères** est la clé qui structure les points de code en octets dans la mémoire de l'ordinateur, puis lit les octets à nouveau en points de code.

Documentation :

<http://sdz.tdct.org/sdz/comprendre-les-encodages.html>

<https://www.w3.org/International/questions/qa-what-is-encoding.fr>

<https://zestedesavoir.com/tutoriels/1114/comprendre-les-encodages/>

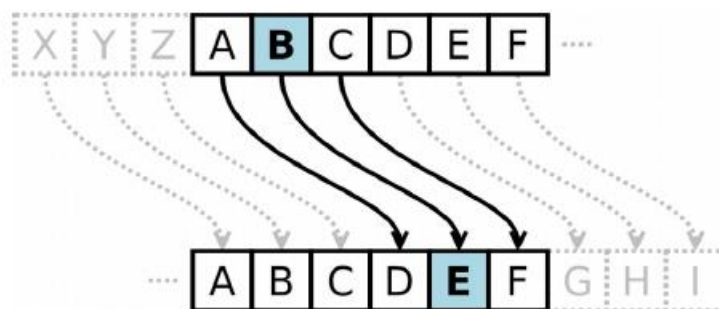
https://youtu.be/y2e7RuFd5gU?list=PLOapGKeH_KhGI1CsCvu_nK9otou8_KsCZ

Pour aller plus loin :

Le chiffre de César



Le **chiffre de César** est une méthode de chiffrement consistant à décaler les lettres de l'alphabet de quelques crans vers la gauche. Par exemple, si on décale les lettres de 3 rangs (comme le faisait Jules César), on obtient la table de chiffrement suivante :



Cette méthode a été utilisée par Jules César dans ses correspondances secrètes (ce qui explique le nom « chiffre de César »).

En utilisant cette table, le message Ave Caesar morituri te salutant devient DYH FDHVDU PRULWXUL WH VDOXWDQW.

Écrivez un programme Python qui implémente le chiffre de César. Utilisez pour cela les fonctions `ord()` et `chr()`. On doit pouvoir choisir le décalage des lettres entre 1 et 25. L'entrée sera un texte quelconque sans accents et sans ponctuation ; par contre, le texte d'entrée sera composé de majuscules et de minuscules. Le texte de sortie sera écrit tout en majuscules. Les espaces du texte d'entrée se retrouveront aux mêmes endroits dans le texte chiffré.

Faites en sorte que l'on puisse crypter, mais aussi décrypter un message