

Représentation de texte - Plan de travail

1NSI – janvier 2023

1 Tables d'encodage

- Exercice 1 : Table ASCII ☆☆☆☆☆
- Exercice 2 : Table ASCII et Python ☆☆☆☆☆
- Exercice 3 : Norme ISO-8859-1 ☆☆☆☆☆

2 Unicode

- Exercice 4 : Unicode table ☆☆☆☆☆
- Exercice 5 : Exploration de l'Unicode ☆☆☆☆☆
- Exercice 6 : Les strings en Python et cryptographie ☆☆☆☆☆

3 Les strings en Python

4 QCM

- Exercice 7 : QCM ☆☆☆☆☆

Exercice 1

Table ASCII

Vous trouverez la table ASCII à la fin de l'exercice.

1. Combien de caractères sont décrits par cette table ?
2. Quels sont les caractères d'écriture absents de cette table ?
3. Combien de bits sont nécessaires pour encoder un caractère avec la table ascii ? Combien d'octets ?
4. Encoder en binaire puis en hexadécimal grâce à la table ASCII le message `NSI <3`.
5. Décoder grace à la table ASCII le message

```
01001101 01100001 01110100 01101000 00100000 00101011 00100000 01010011
01010110 01010100 00100000 00101011 00100000 01010000 01000011 00100000
00111101 00100000 00111100 00110011
```

6. Décoder grace à la table ASCII le message

```
4D 61 74 68 3A 29
```

7. Le programme de première NSI contient 25 107 caractères. Quelle sera son poids s'il est numérisé en ASCII ? Vous exprimerez le résultat en octet et en bit.

Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char
0	0000 0000	00	[NUL]	32	0010 0000	20	space	64	0100 0000	40	@	96	0110 0000	60	`
1	0000 0001	01	[SOH]	33	0010 0001	21	!	65	0100 0001	41	A	97	0110 0001	61	a
2	0000 0010	02	[STX]	34	0010 0010	22	"	66	0100 0010	42	B	98	0110 0010	62	b
3	0000 0011	03	[ETX]	35	0010 0011	23	#	67	0100 0011	43	C	99	0110 0011	63	c
4	0000 0100	04	[EOT]	36	0010 0100	24	\$	68	0100 0100	44	D	100	0110 0100	64	d
5	0000 0101	05	[ENQ]	37	0010 0101	25	%	69	0100 0101	45	E	101	0110 0101	65	e
6	0000 0110	06	[ACK]	38	0010 0110	26	&	70	0100 0110	46	F	102	0110 0110	66	f
7	0000 0111	07	[BEL]	39	0010 0111	27	'	71	0100 0111	47	G	103	0110 0111	67	g
8	0000 1000	08	[BS]	40	0010 1000	28	(72	0100 1000	48	H	104	0110 1000	68	h
9	0000 1001	09	[TAB]	41	0010 1001	29)	73	0100 1001	49	I	105	0110 1001	69	i
10	0000 1010	0A	[LF]	42	0010 1010	2A	*	74	0100 1010	4A	J	106	0110 1010	6A	j
11	0000 1011	0B	[VT]	43	0010 1011	2B	+	75	0100 1011	4B	K	107	0110 1011	6B	k
12	0000 1100	0C	[FF]	44	0010 1100	2C	,	76	0100 1100	4C	L	108	0110 1100	6C	l
13	0000 1101	0D	[CR]	45	0010 1101	2D	-	77	0100 1101	4D	M	109	0110 1101	6D	m
14	0000 1110	0E	[SO]	46	0010 1110	2E	.	78	0100 1110	4E	N	110	0110 1110	6E	n
15	0000 1111	0F	[SI]	47	0010 1111	2F	/	79	0100 1111	4F	O	111	0110 1111	6F	o
16	0001 0000	10	[DLE]	48	0011 0000	30	0	80	0101 0000	50	P	112	0111 0000	70	p
17	0001 0001	11	[DC1]	49	0011 0001	31	1	81	0101 0001	51	Q	113	0111 0001	71	q
18	0001 0010	12	[DC2]	50	0011 0010	32	2	82	0101 0010	52	R	114	0111 0010	72	r
19	0001 0011	13	[DC3]	51	0011 0011	33	3	83	0101 0011	53	S	115	0111 0011	73	s
20	0001 0100	14	[DC4]	52	0011 0100	34	4	84	0101 0100	54	T	116	0111 0100	74	t
21	0001 0101	15	[NAK]	53	0011 0101	35	5	85	0101 0101	55	U	117	0111 0101	75	u
22	0001 0110	16	[SYN]	54	0011 0110	36	6	86	0101 0110	56	V	118	0111 0110	76	v
23	0001 0111	17	[ETB]	55	0011 0111	37	7	87	0101 0111	57	W	119	0111 0111	77	w
24	0001 1000	18	[CAN]	56	0011 1000	38	8	88	0101 1000	58	X	120	0111 1000	78	x
25	0001 1001	19	[EM]	57	0011 1001	39	9	89	0101 1001	59	Y	121	0111 1001	79	y
26	0001 1010	1A	[SUB]	58	0011 1010	3A	:	90	0101 1010	5A	Z	122	0111 1010	7A	z
27	0001 1011	1B	[ESC]	59	0011 1011	3B	;	91	0101 1011	5B	[123	0111 1011	7B	{
28	0001 1100	1C	[FS]	60	0011 1100	3C	<	92	0101 1100	5C	\	124	0111 1100	7C	
29	0001 1101	1D	[GS]	61	0011 1101	3D	=	93	0101 1101	5D]	125	0111 1101	7D	}
30	0001 1110	1E	[RS]	62	0011 1110	3E	>	94	0101 1110	5E	^	126	0111 1110	7E	~
31	0001 1111	1F	[US]	63	0011 1111	3F	?	95	0101 1111	5F	_	127	0111 1111	7F	[DEL]

Exercice 2

Table ASCII et Python

En Python, il existe deux fonctions qui permettent de convertir un caractère en son point de code (lisez place dans la table ASCII pour le moment) en décimal et inversement.

- `ord` : caractère vers point de code
- `chr` : point de code vers le caractère

1. Avec python afficher les 128 caractères de la table ascii (Bonus : faites les apparaitres sur 4 colonnes comme dans la table au dessus).
2. Comparer le résultat avec la table obtenu.

Exercice 3

Norme ISO-8859-1

1. Chercher la table d'encodage ISO-8859-1 (ou latin-1).
2. Comparer cette table avec la table ASCII.
3. Encoder le mot a«`€éø`

Exercice 4

Unicode table

La norme Unicode découpe le processus d'encodage en deux parties : le point de code et l'encodage.

Vous trouverez les points de code et les encodages correspondant sur le moteur de recherche <https://unicode-table.com/fr>.

1. Compléter le tableau ci-dessous avec les points de code et les encodage (représentée sous la forme qui vous semble la plus familière).

Caractère	Point de code	UTF-8	UTF-16BE	UTF-32BE
l				
€				
ℵ				
	U+20B3			
	U+1F606			
	U+FE9A			

2. Quels sont les caractères qui sont inclus dans la norme Unicode ?
3. Quels sont les encodages possibles et quelles sont leur caractéristiques ?
4. Comment est placé la table ASCII dans cette norme ?

Exercice 5

Exploration de l'Unicode

Notebook 02_unicode

Exercice 6

Les strings en Python et cryptographie

Notebook 03_str

Exercice 7

QCM

Question 1 : Quel est un avantage du codage UTF8 par rapport au codage ASCII ?

- a) il permet de coder un caractère sur un octet au lieu de deux
- b) il permet de coder différentes polices de caractères
- c) il permet de coder les majuscules
- d) il permet de coder tous les caractères

Question 2 : On considère les codes ASCII en écriture hexadécimale (en base 16).

Le code ASCII de la lettre A est 0x41, celui de la lettre B est 0x42, celui de la lettre C est 0x43, etc. Quelle est le code ASCII, en hexadécimale, de la lettre X (c'est la 24ième lettre de l'alphabet usuel)

- a) 0x58
- b) 0x64
- c) 0x7A
- d) 0x88

Question 3 : Laquelle de ces affirmations concernant le codage UTF-8 des caractères est vraie ?

- a) le codage UTF-8 est sur 7bits
- b) le codage UTF-8 est sur 8bits
- c) le codage UTF-8 est sur 1 à 4 octets
- d) le codage UTF-8 est sur 8 octets

Question 4 : Parmi les caractères ci-dessous, lequel ne fait pas partie du code ASCII ?

- a) a
- b) A
- c) @
- d) é

Question 5 : Dans le bloc <head> d'un fichier HTML, afin d'encoder les caractères avec le standard Unicode/UTF-8 on insère la ligne :

```
<meta http-equiv="Content -Type" content="text/html; charset=UTF-8">
```

Pourquoi cela ?

- a) UTF-8 est l'encodage Linux
- b) ASCII est une vieille norme, il est temps d'en changer
- c) UTF-8 est une norme conçue pour permettre un affichage correct des caractères spéciaux sur tout système
- d) UTF-8 est un encodage qui protège mieux contre le piratage informatique.