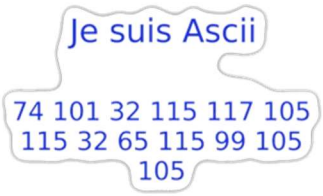
Cours
Codage des caractères et des textes

Objectifs :

- ⇨ Comprendre comment les ordinateurs peuvent stocker et manipuler des caractères
- ⇨ Découvrir les différents systèmes de codage des caractères, leurs avantages et leurs défauts
- ⇨ Comprendre les difficultés d'interopérabilité qu'il peut y avoir avec les caractères



Nous avons vu précédemment que l'ordinateur ne savait manipuler que des nombres binaires. Comment alors représenter des textes afin de communiquer avec les humains ?
Tout simplement en utilisant un **jeu de caractères codés**. C'est un code qui à chaque caractère (signe d'écriture d'un ou plusieurs systèmes d'écriture comme des alphabets) associe un nombre entier naturel.

I - Code ASCII

La première norme de codage de caractères à s'imposer vraiment et à devenir à peu près universelle est l'ASCII (American Standard Code for Information Interchange – *Code américain normalisé pour l'échange d'information*).

code	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10	LF	VT	NP	CR	SO	SI	DLE	DC1	DC2	DC3
20	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30	RS	US	SP	!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	DEL		

Remarques :

- Le codage se fait sur **7 bits** (valeurs de 0 à 127)
- Les codes 0 à 31 ne sont pas des caractères. On les appelle *caractères de contrôle* car ils permettent de faire des actions telles que:
  - retour à la ligne (CR)
  - bip sonore (BEL)
- Les codes 48 à 57 représentent les chiffres.
- Les codes 65 à 90 représentent les majuscules.
- Les codes 97 à 122 représentent les minuscules.

Le code ASCII a été mis au point pour la langue anglaise, il ne contient donc pas de caractères accentués, ni de caractères spécifiques à une langue. Son emploi est donc très limité.

On remarque que les chiffres et les lettres de l'alphabet sont placés avec des codes croissants. C'est ce qui permet de faire des comparaisons et donc des tris très facilement sur les caractères comme on le fait avec des nombres. Ainsi 'S' est bien après 'F' puisque 83 est bien supérieur à 70. Par contre 'Z' sera avant 'a' car 90 est plus petit que 97.

Q1 :
Classer les chaînes suivantes par ordre alphabétique selon le codage ASCII :
`"Turing","Il va la","bool","Ile-de-France","Encodage","3eme loi de Newton"`

II - Codage ASCII étendu

Le code ASCII de base ne permet pas l'affichage de caractères accentués (comme en français), ni les caractères spécifiques à certaines langues (β allemand, Ñ espagnol, д cyrillique, …).
Pour pallier à ce problème, on a imaginé différentes solutions, la première étant d'utiliser un **codage ASCII étendu** en utilisant 8 bits plutôt que 7 pour le codage d'un caractère. On accède ainsi à 128 possibilités supplémentaires.

1) Les pages de code

Dans un premier temps, on a utilisé les codes ASCII étendus en définissant un jeu étendu pour chaque langue, c'est ce que l'on appelle les **pages de code**.

En voici quelques exemples :

Page de code 437 (DOS Latin US) — mode standard

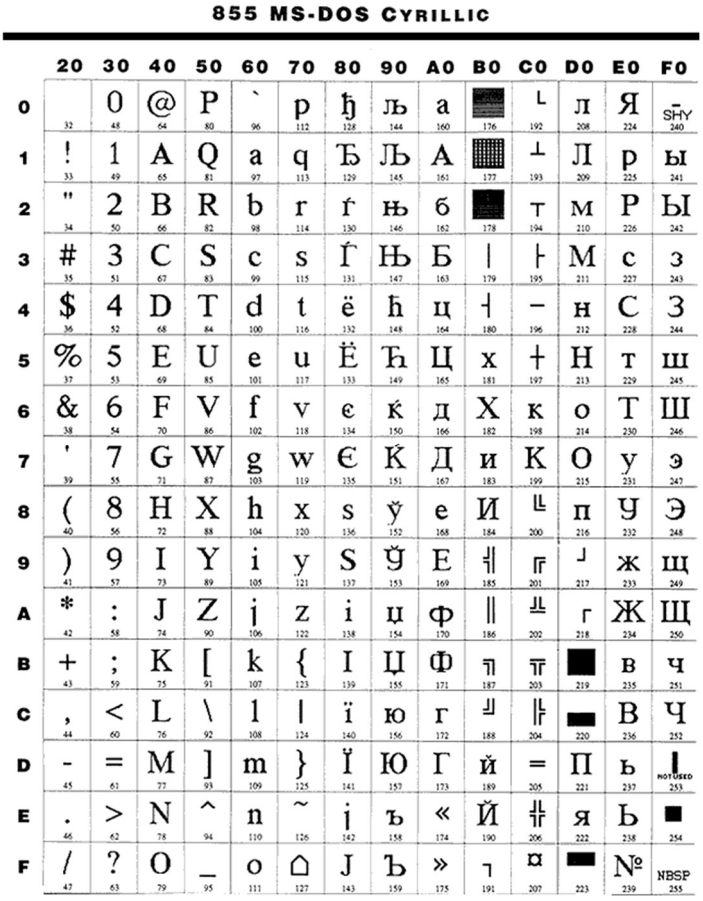
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9x	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
Ax	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
Bx	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
Cx	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
Dx	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
Ex	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
Fx	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	n	2	■	NBSP

Utilisé aux États-Unis

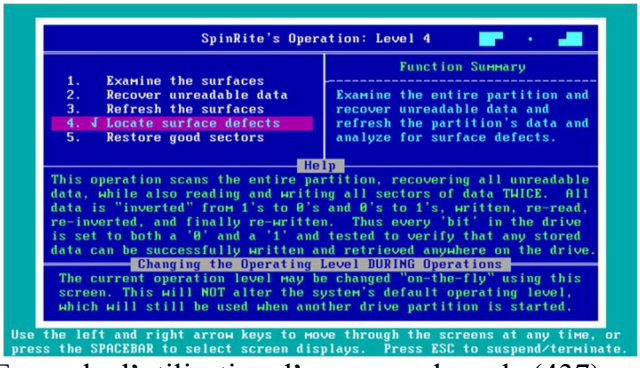
Page de code 852 (DOS Latin 2) — mode standard

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x	Ç	ü	é	â	ä	ů	ć	ç	ł	ë	Ő	ő	î	Ź	Ä	Ć
9x	É	Ĺ	Í	ô	ö	Ľ	ľ	Ś	ś	Ö	Ü	Ť	ť	Ł	×	č
Ax	á	í	ó	ú	Ą	ą	Ž	ž	Ę	ę	¬	ź	Č	ş	«	»
Bx	░	▒	▓	│	┤	Á	Â	Ě	Ş	╣	║	╗	╝	Ż	ż	┐
Cx	└	┴	┬	├	─	┼	Ă	ă	╚	╔	╩	╦	╠	═	╬	¤
Dx	đ	Đ	Ď	Ë	ď	Ň	Í	Î	ě	┘	┌	█	▄	Ţ	Ů	▀
Ex	Ó	ß	Ô	Ń	ń	ň	Š	š	Ŕ	Ú	ŕ	Ű	ý	Ý	ţ	´
Fx	SHY -	˝	˛	ˇ	˘	§	÷	¸	°	¨	˙	ű	Ř	ř	■	NBSP

Utilisé en Europe de l'ouest



Utilisé en Russie



Exemple d'utilisation d'une page de code (437)

On remarque l'introduction de caractère servant à dessiner des séparations ou des cadres (caractères ┐, └, ┴, ┬, …). Ils étaient très utilisés en mode console (c'est-à-dire un mode où l'écran n'affiche que du texte et aucun graphique).

Le problème est bien sûr qu'il faut absolument connaître la page de code dans laquelle est encodé un texte pour pouvoir l'afficher correctement. De même on ne pourra afficher simultanément deux langues ayant besoin d'une page de code différente (russe et français par exemple). Enfin, il est impossible de coder des alphabets ayant plus de 128 symboles spécifiques.

Q2 :
Pour saisir au clavier des caractères ascii qui ne sont pas accessibles au clavier, on peut utiliser la méthode des « Alt-codes ». Voir la page web : https://fr.wikipedia.org/wiki/Alt_codes

En exploitant ces informations, essayer de représenter dans un éditeur de texte (notepad ou notepad++) le texte suivant :

Enregistrer le résultat dans un fichier sous le nom « cadre.txt ».

NB : Sous windows, on peut également utiliser la table de caractère (Accessible par Démarrer>tous les prog>Accessoires>Outils Système>Table des caractères ou bien en tapant Windows+R (raccourci de Exécuter) et la commande « charmap »).

Remarque : L'introduction des caractères accentués à des positions dépassant 128, bouleverse l'ordre numéro-alphabétique précédemment établit. Ainsi `'été'` sera après `'hiver'` dans l'ordre alphabétique lorsque l'on comparera les 2 chaînes avec l'opérateur `>` ou `<`.

2) Normes courantes d'ASCII étendu

Il existe plusieurs normes de code ASCII étendue (et donc des difficultés d'interopérabilité). On donne l'utilisation la plus courante ci-dessous (ISO/Latin-1 ou ISO-8859-1) :

ISO/CEI 8859-1

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x	*positions inutilisées*															
9x																
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Les codes sont écrits en hexadécimal. Par exemple le caractère de code B5 est « µ », celui de code DB est « Û » et le caractère « Ã » a pour code C3.

Il manque cependant quelques lettres utiles comme le « œ » ou le « € ». Elles ont été ajoutée dans la norme ISO/Latin-9 ou ISO-8859-15 qui reprend pour l'essentiel la norme ISO-8859-1 :

ISO/CEI 8859-15

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	¡	¢	£	€	¥	Š	§	š	©	ª	«	¬		®	¯
Bx	°	±	²	³	Ž	µ	¶	·	ž	¹	º	»	Œ	œ	Ÿ	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Les différences entre les normes ISO-8859-1 et ISO-8859-15 sont indiquées en jaune

Enfin, Microsoft a introduit avec windows 3.0 son propre codage ASCII étendu, baptisé Windows-1252 (et souvent appelé improprement « ANSI »). Celui-ci reprend pour l'essentiel ISO-8859-1 :

Windows-1252

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
8x	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
9x		‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Les différences entre les normes Windows-1252 et ISO-8859-1 sont indiquées en jaune

3) Le codage de python

Trois fonctions autour du codage de caractère en python :

```
chr(x)   # Renvoi le caractère dont le code est x
ord(y)   # Renvoi le code du caractère y
hex(z)   # Renvoi la valeur hexadécimale z sous la forme d'une chaîne
         # de caractère (type str)
```

Q3 :

1) Ecrire un programme permettant d'afficher la table de codage utilisée par Python. On pourra écrire une ligne par caractère avec un affichage du type :

```
Caractère n°63 : ?
Caractère n°64 : @
Caractère n°65 : A
```

2) Que remarque-t-on pour les caractères 0 à 32 ?
3) Le codage utilisé par python est-il compatible avec ISO-Latin-1 ou ISO-Latin-9 ?
4) Quel est le code du caractère `'€'` en décimal et en hexadécimal ? De quelle famille d'ASCII étendu fait-il partie ?

Dans une chaîne python, les caractères peuvent également être directement saisis à l'aide de leur code ASCII en utilisant la notation `\xhh` où *hh* est le code hexadécimal du caractère.

Application :
Avec thonny, afficher la chaîne `"\x41ffi\x63hage de \x22 dans une chaîne\x2e"`.

Cette technique de saisie des caractères à l'aide du caractère `\` est appelée **caractère échappé** (ce nom vient du caractère `\` que l'on appelle également caractère d'échappement). D'autres raccourcis de caractères échappés sont également disponibles pour saisir des caractères spéciaux.

Q4 :
Compléter le tableau ci-contre qui donne les codes d'échappement les plus utilisés :

Code	Nom	Fonction	Fonctionne dans la console
`\a`	BEL (Bell)		
`\b`	BS (BackSpace)		
`\f`	FF (Form Feed)		
`\n`	LF (Line Feed)		
`\r`	CR (Carriage Return)		
`\t`	HT (Horizontal Tabulation)		
`\v`	VT (Vertical Tabulation)		
`\\`			

Retour à la ligne
Les caractères LF (Line Feed) et CR (Carriage Return) ont des effets très similaires, mais leur interprétation est différente selon les systèmes.
Sous Windows pour sauter une ligne et revenir au debut de la nouvelle ligne (on dit aussi *passer à la ligne*), il faut utiliser les deux caractères CR et LF. Sous Linux et macOS, le caractère LF suffit. Sur certains systèmes plus anciens, on utilise seulement CR.
Bref, selon la source du fichier le codage du passage à la ligne peut être différent, il peut donc être utile de faire des tests pour être certain de l'effet de ces caractères spéciaux.

III - Un codage universel ?

Les codages qui souhaitent représenter un grand nombre de caractère doivent donc être sur plus de 8 bits.
Pour solutionner cela, un nouveau standard a été introduit : **Unicode**. La table Unicode comporte la définition de près de cent quarante mille caractères couvrant la quasi-totalité des langues y compris celles disparues.
L'unicode fait correspondre un caractère à un point de code (nombre entier généralement noté en hexadécimal et variant pour l'unicode de 0_{hex} à $10FFFF_{hex}$[1]).
Voir https://fr.wikipedia.org/wiki/Table_des_caract%C3%A8res_Unicode_(0000-0FFF)#Contr%C3%B4les_C0_et_latin_de_base

Pour coder des caractères de la table unicode, il existe 3 codages normalisés : UTF-8, UTF-16 et UTF-32.
UTF signifiant **U***niversal character set* **T***ransformation* **F***ormat*, le nombre qui suit indiquant le nombre de bits sur lesquels sont codés les informations.

1) UTF-8

Le codage le plus couramment utilisé est UTF-8. Son principe est le suivant : une première série de caractères sont codés sur un octet. D'autres caractères sont codés sur deux octets, le premier octet débute par '110' pour

[1] Un point de code est noté U+xxxx où xxxx correspond à la valeur en hexadécimal du point de code.

l'indiquer, l'octet suivant débute par '10'. De même des codages sur 3 ou 4 octets sont utilisés pour d'autres caractères. (Cette rapide introduction à UTF-8 est volontairement simplifiée.)

Définition du nombre d'octets utilisés

Représentation binaire UTF-8	Signification
0xxxxxxx	1 octet codant 1 à 7 bits
110xxxxx **10**xxxxxx	2 octets codant 8 à 11 bits
1110xxxx **10**xxxxxx **10**xxxxxx	3 octets codant 12 à 16 bits
11110xxx **10**xxxxxx **10**xxxxxx **10**xxxxxx	4 octets codant 17 à 21 bits

Les 128 premiers caractères de la table UTF-8 sont compatibles avec le codage ASCII. Ainsi **le codage UTF-8 d'un texte ne comportant que des caractères présents dans la table ASCII sera le même que le codage ASCII de ce texte**.
Ce ne sera pas vrai pour un texte ISO-Latin-1.
Il importe donc, quand on veut décoder un texte, de savoir quel est le codage utilisé sous peine de décoder improprement les caractères.

Q5 :
On lit dans une table UTF-8 que le caractère 'é' est codé sur deux octets par les valeurs hexadécimales 'C3 A9' et le caractère '€' sur 3 octets par les valeurs hexadécimales 'E2 82 AC'.

Soit un fichier contenant le texte "J'écris € en UTF-8" codé en UTF-8. Quel sera le texte affiché si un logiciel décode ce texte en supposant que le codage utilisé est Latin-1 ?

Q6 :
Allez chercher dans votre répertoire de classe ou sur l'ENT le logiciel Notepad++ (npp). Copiez le répertoire « npp7.9.1 » dans votre répertoire personnel, puis lancez le logiciel (fichier « notepad++.exe »).
Dans Notepad++, ouvrez le fichier « Texte_utf-8.txt » qui se trouve dans le répertoire de l'activité (sur l'ENT ou votre répertoire de classe). Ce fichier est encodé en UTF-8.
Utilisez le pour afficher le texte en codage ANSI (Windows-1252)) avec le menu « Encodage ». Que remarquez-vous ? Votre réponse à l'exercice précédent était-elle correcte ?

Q7 :
Quels sont les avantages et les inconvénients du codage UTF-8 ?

2) UTF-16 et UTF-32

Pour résoudre une partie des inconvénients du codage UTF-8 (la difficulté d'indexer directement un texte codé en UTF-8), on peut utiliser l'UTF-16 qui code sur 16 bits les points de code et permet ainsi une taille fixe des caractères avec en contrepartie[2] une taille en mémoire plus importante et un nombre de symbole codé plus faible (16 bits ne peuvent coder que 65536 valeurs – on est loin d'accéder à tous les caractères unicode).
Pour pallier à ce dernier défaut – et au prix d'un doublement de la mémoire nécessaire – on peut utiliser UTF-32 qui lui utilise 32 bits pour les points de code.

[2] Il y a aussi avec UTF-8 un problème de boutisme (ou endianness en anglais), mais cela dépasse le cadre de ce cours. Voir https://fr.wikipedia.org/wiki/Boutisme et https://fr.wikipedia.org/wiki/UTF-16.

Q8 :
Utilisez maintenant le navigateur « Internet explorer » pour afficher une page en français (celle du lycée par exemple), puis essayez de changer le décodage des caractères de la page avec le menu « Affichage/Codage » en commençant par désactiver la sélection automatique et regarder ce que cela change.

Q9 :
En faisant un copier-coller directement dans la barre d'adresse, observer comment Firefox, convertit les adresses qui contiennent des caractères non définis dans le code ASCII :
http://fr.wikipedia.org/wiki/Salon_de_beauté
http://ps.wikipedia.org/wiki/فارسی
A quoi correspondent les codes de remplacement ?

… Maintenant vous comprenez pourquoi un informaticien averti évite les caractères non-ASCII (accentués, notamment) dans ses noms de variables, de fichier, répertoire et autres noms d'utilisateur dont le décodage correct est primordial pour le fonctionnement de l'ensemble.

TRAVAIL A FAIRE POUR LA PROCHAINE SEANCE :

Chercher sur internet l'encodage[3] utilisé par Python pour les `str` et pour les fichiers source (.py). Quelles sont les conséquences pour le programmeur ?
Et pour d'autres langages comme Java, quel est l'encodage utilisé ?

[3] En programmation le terme « encodage » désigne la norme utilisée pour coder les caractères.