Décrire et manipuler un document numérique

Cours 2 : Encodage et représentation des caractères

Loïc Grobol <lgrobol@parisnanterre.fr>

2024-01-22

Matériel complémentaire

- « *The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)* », un article de Joel Spolsky, simple et bien écrit (en anglais) qui reprend les points de ce cours avec une prose différente.
- « *Character Encodings* », un cours en vidéo et ses notes (en anglais) de Peter Larsson-Green, donné à l'université de Jönköping.
- « Tutoriel : comprendre les encodages », un tutoriel sur le site du zéro. En français, fait quelques références à HTML mais rien qui devrait vous empêcher de le lire dès maintenant.

Caractères, graphèmes, glyphes et encodage

Un caractère, c'est quoi ?

Donnez des exemples.

Est-ce que « a » est un caractère ?

- et « 1 » ?
- et « » ?
- et « é » ?
- et « @ » ?
- et « + » ?

- et « » ?
- et « ¨ » ?
- et « » ?
- et « » ?
- et « » ?
- et « » ?
- et « » ?
- et un bip d'ordinateur ?

Dans l'image suivante, combien y a-t-il de caractères différents ?

« A » et « a », est-ce le même caractère ?

- et « a » et « à » ?
- et « a » et « *a* » ?
- et « o » et « » ?
- Même si je vous dit que le deuxième est la lettre grecque omicron ?

Dans la mémoire de ma machine, la phrase « Je reconnais l'existence du kiwi. » est représentée par la suite de nombres

```
74, 101, 32, 114, 101, 99, 111, 110, 110, 97, 105, 115, 32, 108, 39, 101, 120, 105, 115, 116
```

Ou en binaire

```
1001010, 1100101, 100000, 1110010, 1100101, 1100011, 1101111, 1101110, 1101110, 1100001, 110
```

Ces nombres sont-ils des caractères ?

Définitions

La terminologie n'est pas standardisée, le concept de *caractère* est ambigüe et on distingue ces concepts :

- Un **graphème** est une unité élémentaire dans un système d'écriture : les lettres, les diacritiques (accents et autres), les logogrames simples…
- Un **glyphe** est une représentation *visuelle* d'un graphème, la seule notion ici qui correspond à une *perception*. Par exemple, le glyphe utilisé dans « a » est le glyphe :
  - Correspondant au graphème « lettre a minuscule » de l'alphabet latin.
  - Dans la police *Atkinson Hyperlegible*.
  - En roman (et non en italiques ou en gras).

- Un **caractère** est en général défini comme un graphème ou une série de graphème, perçue comme formant une unité.
- Un **encodage** est une représentation numérique d'un caractère dans un système informatique.
  - Elle peut ne pas être unique. Ainsi, en UTF-8, il y a deux représentations possibles pour « à ».

En résumé : dans la mémoire de l'ordinateur, un fichier texte correspond à une série de **caractères encodés**, qui sont traduits en **graphèmes**, eux-mêmes représentés à l'écran par des **glyphes** qui sont ce que vous voyez.

Ce qui nous intéresse dans la suite, c'est la partie **encodage** et on utilisera avec assez peu de précaution le mot *caractère*, sauf quand la distinction est importante.

Encodages

Principe

On l'a dit, un *encodage de caractère*, c'est une façon de représenter des caractères par des nombres ou des séries de nombres.

Le principe est simple : tout le monde se met d'accord pour décider d'une table de correspondance commune qui associe à chaque caractère un unique nombre et voilà.

En pratique, historiquement, ça n'a pas été si simple.

Pourquoi ? Deux raisons :

- Le manque de volonté :
  - Lié d'abord à l'absence de besoin d'interopérabilité.
  - Et parfois à une volonté délibérer de garder des clients captifs.
- Une contrainte technique : la longueur des messages.
  - Plus le nombre de caractères possibles est grand, plus chacun des caractères prend de place.
  - Il y a des astuces pour limiter cet effet, mais elles ne sont pas toujours optimal
  - *There is no free lunch.*

ASCII

L'***A**merican **S**tandard **C**ode for **I**nformation **I**nterchange* est une des premières conventions d'encodages à être devenue une norme (*de facto* et *de jure*) à l'usage international.

Principes

- Les caractères sont codés sur 7 bits, soit 128 caractères possibles
  - Sur des machines utilisant des mots de 8 bits, le dernier peut par exemple servir pour la détection d'erreurs.
- Les caractères disponibles sont :
  - L'espace
  - Les lettres de l'alphabet latin, sans diacritiques, majuscules et minuscules
  - Les chiffres de 0 à 9
  - `!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`
  - Des caractères dits « de contrôle » : fin de ligne, beep…

Table

	0	1	2	3	4	5	6	7	8	9
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
01	LF	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3
02	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
03	RS	US	SP	!	”	#	$	%	&	’
04	(	)	*	+	,	-	.	/	0	1
05	2	3	4	5	6	7	8	9	:	;
06	<	=	>	?	@	A	B	C	D	E
07	F	G	H	I	J	K	L	M	N	O
08	P	Q	R	S	T	U	V	W	X	Y
09	Z	[	\	]	^	_	‘	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

Exercice

1. À quelle série de nombres correspond la chaîne de caractères `Vive le TAL!` ?
2. À quelle chaîne de caractère correspond la suite de nombres `085, 080, 088, 060, 051` ?

Les extensions d'ASCII

Est-ce que vous voyez un problème avec ASCII ?

Et est-ce que vous voyez une solution ?

La solution qui a été retenue au problème « je peux pas écrire dans ma langue ! » c'est en général d'utiliser le 8ème bit tout en gardant les mêmes caractères pour les nombres de 0 à 127.

Ce qui fait combien de caractères en plus ?

---

128 caractères de plus ça laisse de la place pour pas mal de trucs. Comme des lignes pour dessiner des tableaux ou un soleil .

Évidemment ça n'est pas assez pour utiliser des systèmes d'ériture logographique ou même simplement pour tous les alphabets d'Europe.

La solution historique, c'est donc encore une fois que chaque constructeur définit ses propres extensions, évidemment incompatibles.

La norme ISO 8859 est une tentative de définir des jeux de caractères standardisés par région géographique. En particulier, ISO 8859-1, qui permet d'écrire la plupart des alphabets d'Europe de l'Ouest a connu un succès assez durable pour des raisons qu'on imagine aisément.

En particulier la version légèrement différente utilisée par Microsoft sous le nom de CP-1252 est encore utilisée par beaucoup de systèmes.

Il reste que la standardisation est difficile : en particulier il est nécessaire de transmettre l'encodage d'un document sous forme de métadonnée, ce qui :

- N'est pas très pratique.

- Déplace le problème : il faut des métadonnées standardisées

En conséquence, surtout avec la démocratisation du Web, les confusions sont nombreuses et mènent souvent à des « mojibake ». Par exemple pour un certain nombre de sites web, mon nom légal est LoÃ¯c.

Et ça ne résout pas le problème des systèmes d'écriture logographiques pour lesquels des encodages différents sont nécessaires.

Unicode

Unicode est une norme créée dans le but de résoudre ces problèmes : un système d'encodage unique pour tous les systèmes d'écritures.

- Initialement prévu sur 16 bits (UTF-16), puis sur 32 bits (UTF-32).
- Finalement représentation de **taille variable** : UTF-8
  - Les caractères ASCII sont codés sur un octet.
  - Les autres caractères sur plus, avec une organisation plus ou moins arbitraire.
- En évolution (croissance) constante depuis : à l'automne 2023, on en est à la version 15.1

---

https://unicode-table.com présente de façon agréable la liste des caractères Unicode.

Détails techniques

Notation hexadécimale

Au tableau !

Points de codes et représentations concrètes

On ajoute un niveau de représentation : le **point de code**, un nombre, souvent noté en hexadécimal, qui correspond à la position d'un caractère dans la table Unicode mais n'est pas nécessairement sa représentation concrète sur la machine.

Ainsi le caractère « », « *CYRILLIC CAPITAL LETTER YA* » correspond au point de code U+042F (1 071) et est représenté en UTF-8 par le nombre 208 175 (D0 AF).

Pour voir les représentations concrètes, on peut ouvrir les fichier textes dans un éditeur hexadécimal comme https://hexed.it.

Caractères combinants

Unicode contient beaucoup de caractères, pour des raisons plus ou moins bonnes. Entre autre il contient à la fois des caractères combinants (par exemple des accents qui s'ajoutent au caractère précédent), mais parfois aussi des caractères déjà combinés. Ainsi « ñ » peut se représenter comme U+0068 (n) suivi de U+0303 (le tilde) ou directement comme U+00F1. La normalisation n'est pas un procédé simple…

À votre avis, en plus des diacritiques, quels autres caractères sont combinants ?

Pour compliquer encore les choses, certains caractères servent à marquer une combinaison de caractères non-combinants. Ainsi l'emoji « » est représenté comme la suite des caractères , ZWJ et , ZWJ (***Z**ero **W**idth **J**oiner*) marquant une combinaison entre ces deux emojis.

Exercices

À faire par exemple à l'aide de https://unicode-table.com, rendre vos réponses dans un fichier txt nommé `utf.txt`.

1. Donner la représentation de votre nom complet sous la forme d'une suite de points de code Unicode.
2. Donner la représentation en UTF-8 des cinq (au plus) premiers caractères de votre prénom.
3. À quoi correspond la suite de points de code U+5317, U+4EAC, U+5E02 ?

2. Fichiers et encodage

- Dans un éditeur de texte (par exemple https://vscode.dev), créez un fichier texte contenant uniquement le mot `préfère`.
- Sauvegardez ce fichier dans une version encodée en UTF-8, à laquelle vous donnerez le nom de fichier `utf8.txt` et une encodée en windows-1252 sous le nom `windows.txt`.
- À l'aide ces fichiers et de https://hexed.it, déterminer comment sont représentés les caractères `é` et `è` dans ces deux formats d'encodage.

Rendre les deux fichier txt que vous avez créé, plus un fichier nommé `2.txt` qui contiendra votre réponse à la dernière question