Gilles Dowek
Jean-Pierre Archambault, Emmanuel Baccelli, Claudio Cimelli,
Albert Cohen, Christine Eisenbeis, Thierry Viéville et Benjamin Wack
Avec la contribution de Hugues Bersini et de Guillaume Le Blanc
Préface de Gérard Berry, professeur au Collège de France

Informatique et sciences du numérique

Édition spéciale Python !

Manuel de spécialité ISN en terminale

Avec des exercices corrigés
et des idées de projets

EYROLLES

, ISBN : 978-2-212-13676-0

8

Samuel Morse (1791-1872) est l'inventeur d'un code, dans lequel chaque lettre est exprimée par une alternance de sons brefs symbolisés par « . » et longs « – », utilisé pour télégraphier des textes. La lettre « a » y est exprimée par les sons « . – », la lettre « b » par les sons « – … », etc. Artiste peintre, Samuel Morse s'est intéressé aux télécommunications après qu'en 1925, un message lui annonçant que sa femme était malade ne lui est pas parvenu à temps. Comme nous le verrons au chapitre 12, le code morse est à références de longueurs variables, mais ce n'est pas un code préfixe.

Représenter des caractères et des textes

Les lettres ? Toutes des nombres !

Dans ce chapitre, nous voyons comment sont représentés les caractères et les textes de toutes les langues du monde.

Nous expliquons pourquoi il existe plusieurs codes tels *ASCII*, *latin-1*, *latin-2*, *UTF-32*, *UTF-8*. Nous présentons ensuite les formats enrichis qui permettent de décrire la forme des caractères et des textes, comme le font les logiciels de traitement de texte.

Un exemple de format enrichi est le langage HTML.

Nous nous intéressons, dans ce chapitre, à la représentation des textes, c'est-à-dire des suites de *caractères*, éventuellement enrichies d'informations typographiques.

La représentation des caractères

Puisqu'un texte est une suite de caractères, on commence par s'intéresser à la représentation des caractères, c'est-à-dire entre autres choses aux lettres minuscules et majuscules, aux chiffres, aux signes de ponctuation et aux symboles mathématiques. Pour représenter ces caractères, on attribue un nombre à chacun.

Le *code ASCII*, par exemple, attribue le nombre 65 à la lettre « A », le nombre 66 à la lettre « B », le nombre 97 à la lettre « a » et le nombre 98 à la lettre « b ». Il représente 95 caractères : les 26 lettres minuscules, les 26 lettres majuscules, les 10 chiffres, les 32 symboles ! " # $ % &' ( ) * + , – . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ et 1 signe d'espace. Il représente aussi 33 autres *symboles de mise en page*, par exemple le retour chariot qui signale la fin de la ligne et le saut de page qui signale le passage à la page suivante. Le code ASCII représente donc $95 + 33 = 128 = 2^7$ caractères, par des nombres qui peuvent eux-mêmes être représentés en binaire par des mots de sept bits. Ils sont en fait représentés par des mots de huit bits, le premier étant toujours un zéro.

Le code ASCII était à l'origine conçu pour des textes écrits en anglais, comme l'indique son nom, *American Standard Code for Information Interchange*. Il n'est pas adapté pour représenter des textes écrits dans d'autres langues, même celles qui, comme le français, utilisent l'alphabet latin, car ces langues utilisent des accents, des cédilles et autres signes diacritiques. C'est pourquoi on a tout d'abord conçu une extension du code ASCII, le code *latin-1*, qui contient 191 caractères. Aux 128 caractères du code ASCII, qui sont représentés comme en ASCII, s'ajoutent les lettres « é », « É », « è », « ç », « æ », « ñ », « ö », etc. qui permettent de représenter les textes écrits dans la plupart des langues d'Europe de l'Ouest, même si, pour le français, le « œ » a été oublié. Il manque toutefois des lettres utilisées par les langues d'Europe de l'Est, si bien qu'un autre format, le code *latin-2*, a été proposé pour ces langues. Ensuite, pour représenter les textes écrits en grec, russe, chinois, japonais, coréen, etc., il a fallu proposer un format universel : *Unicode*. Unicode recense près de 110 000 caractères et associe un nom et un numéro à chacun. *A priori*, ce numéro se code sur 32 bits. Cependant, Unicode existe en plusieurs déclinaisons, parmi lesquelles *UTF-32*, dans laquelle chaque caractère est ainsi exprimé sur 32 bits, et *UTF-8*, dans laquelle les caractères les plus courants sont exprimés sur 8 bits et les moins courants sur 16, 32 ou 64 bits, utilisant une idée discutée en détail au chapitre 12 à propos de la notion de compression.

Le format UTF-8 a vocation à devenir le standard, mais il ne l'est pas encore : malgré les efforts des comités de normalisation, l'humanité n'a pas encore réussi à se doter d'un format universellement accepté, si bien qu'il est parfois nécessaire de traduire un texte d'UTF-8 en latin-1 ou de latin-2 en UTF-8. Quand cette traduction n'est pas bien faite, les caractères accentués sont remplacés par des caractères bizarres.

Cependant, tous ces formats reposent sur une même idée : associer un nombre, c'est-à-dire un mot binaire, à chaque caractère. Tous ces formats sont accessibles sur le Web.

La représentation des textes simples

Un texte étant une suite de caractères, on peut le représenter en écrivant les caractères les uns après les autres.

SAVOIR-FAIRE Trouver la représentation en ASCII binaire d'un texte

En utilisant une table, on cherche le code ASCII de chaque caractère. Puis on traduit chacun de ces nombres en représentation binaire.

Exercice 8.1 (avec corrigé)

Trouver la représentation binaire en ASCII du texte « Je pense, donc je suis. »

On cherche la table des codes ASCII sur le Web de manière à traduire le texte, caractère par caractère : 74, 101, 32, 112, 101, 110, 115, 101, 44, 32, 100, 111, 110, 99, 32, 106, 101, 32, 115, 117, 105, 115, 46. On exprime ensuite chacun de ces nombres en binaire sur huit bits :
01001010 01100101 00100000 01110000 01100101 01101110 01110011
01100101 00101100 00100000 01100100 01101111 01101110 01100011
00100000 01101010 01100101 00100000 01110011 01110101 01101001
01110011 00101110.

Exercice 8.2

Trouver la représentation binaire en ASCII du texte « Cet exercice est un peu fastidieux. »

SAVOIR-FAIRE Décoder un texte représenté en ASCII binaire

On découpe la suite de bits en octets, on traduit chaque octet en décimal, puis on cherche en utilisant une table, le caractère exprimé par chacun de ces nombres.

Exercice 8.3 (avec corrigé)

Trouver le texte représenté en ASCII binaire par la suite de bits 01000011001001110110010101110011011101000010000001100110011000010110001101101001011010001100101.

On commence par découper la suite de bits en octets : 01000011 00100111 01100101 01110011 01110100 00100000 01100110 01100001 01100011 01101001 01101100 01100101. Chaque octet représente un nombre entier : 67, 39, 101, 115, 116, 32, 102, 97, 99, 105, 108, 101. On cherche ensuite la table des codes ASCII en ligne de manière à traduire chacun de ces nombres en une lettre : « C'est facile ».

Exercice 8.4

Trouver le texte représenté en ASCII binaire par la suite de bits 00110000011101000110010101110100011101000010001.

Exercice 8.5

Traduire en ASCII binaire votre phrase préférée, par exemple : « Le commencement de toutes les sciences, c'est l'étonnement. » en oubliant les accents. Traduire ensuite cette phrase en UTF-8 avec les accents.

Exercice 8.6

Traduire une phrase en ASCII binaire, puis la passer à son voisin qui la décode.

Exercice 8.7

On suppose que les seize lettres qui suivent sont codées ainsi :

ء	آ	أ	ؤ	إ	ئ	ا	ث	د	ش	ف	ك	ل	ن	و	ي
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Décoder le message suivant : 1011 1100 1001 1111 0000 1010 1111 0111 1101 0110 0101 1111 0110 1100 1011 1110 1000, puis en se faisant éventuellement aider d'une personne qui lit l'arabe ou en utilisant le mécanisme de traduction de Google, déterminer s'il correspond à la phrase « tout en code binaire » ou « les lettres deviennent des nombres ».

Exercice 8.8

On considère l'alphabet de 32 signes constitué des 26 lettres de l'alphabet, de l'espace et de cinq symboles de ponctuation : la virgule, le point, le point-virgule, le point d'interrogation et le point d'exclamation. On représente les caractères de cet alphabet par un code binaire de cinq bits présenté dans le tableau suivant. Sur la dernière ligne, on a fait figurer la traduction décimale de chaque code binaire.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		,	.	;	?	!
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**

1. Quel est le troisième bit du code de la lettre « P » ?
2. Décoder le message suivant : 00001011100110101001011101010010001.
3. Écrire son prénom avec ce code.
4. Quels sont les caractères qui commencent par deux 1 ?
5. Pourquoi n'y a t-il pas de caractères en minuscule dans cet alphabet ?
6. On considère l'opération qui consiste à transformer les 0 en 1 et vice-versa dans chaque caractère. Recopier et remplir le tableau ci-après qui, à chaque caractère, associe le caractère obtenu par cette transformation.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		,	.	;	?	!
!	?																														

7. On considère l'opération qui consiste à échanger le premier bit avec le dernier, et le deuxième bit avec l'avant-dernier ; recopier et remplir le tableau ci-après qui, à chaque caractère, associe le caractère obtenu par cette transformation.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		,	.	;	?	!
A	Q																														

ATTENTION **Utiliser un code standard**

Pour les exercices 8.7 et 8.8, on a inventé un code valable exclusivement pour la durée de l'exercice. Pour partager des informations, il ne faut jamais faire cela, mais utiliser un code standard, connu de tous. Sauf, bien sûr, si on veut garder le texte secret (voir le chapitre 12).

Exercice 8.9

Selon Pierre Lecomte du Noüy, 01001100 01100101 00100000 01100010 01110101 01110100 00100000 01100100 01100101 00100000 01101100 01100001 00100000 01110011 01100011 01101001 01100101 01101110 01100011 01100101 00100000 01100101 01110011 01110100 00100000 01100100 01100101 00100000 01110000 01110010 01100101 01110110 01101111 01101001 01110010 00101100 00100000 00100000 01101110 01101111 01101110 00100000 01100100 01100101 00100000 01100011 01101111 01101101 01110000 01110010 01100101 01101110 01100100 01110010 01100101. Êtes-vous d'accord avec lui ?

La représentation des textes enrichis

Les textes en ASCII ou en Unicode sont simplement des suites de caractères. Les *éditeurs de texte* sont les logiciels qui manipulent ces suites de caractères. Toutefois, quand on écrit un texte, on peut souhaiter lui donner une forme spéciale, plus jolie, plus lisible, comme le fait un imprimeur. On peut jouer sur la police de caractères – Times, Courier, etc. –, sur la taille des caractères – 11 points, 12 points, etc. –, sur leur forme – romain, italique, etc. –, leur graisse – maigre, gras, etc. On peut aussi souhaiter découper un texte en chapitres et mettre en valeur les titres des chapitres, etc. Or, les seules caractéristiques que l'on puisse exprimer avec un code comme le code ASCII, par exemple, sont la casse d'une lettre – minuscule ou majuscule – et le découpage en paragraphes, grâce au symbole retour chariot. Les *traitements de texte* sont les logiciels qui permettent ces mises en pages plus élaborées.

Ceci a amené à enrichir ces formats, de manière à :

1. *qualifier* certaines parties du texte, par exemple en mettant certaines parties en gras ou en italique,
2. structurer le texte en *divisions* : un texte n'est pas uniquement une suite de paragraphes, mais est hiérarchisé en parties, chapitres, sections, sous-sections, etc.
3. présenter certaines informations sous forme de listes et de tables,
4. permettre de faire *référence* à d'autres textes,
5. donner des informations sur le texte : son titre, son ou ses auteur(s), sa date de création, sa langue, des mots-clés utilisés pour le rechercher parmi plusieurs textes, etc. Ces informations **sur** le texte, et non **du** texte, sont appelées des *métadonnées*.

Toutes ces considérations sont, bien entendu, valables aussi bien pour les textes manuscrits ou imprimés que pour les textes traités par les ordinateurs.

L'un de ces formats enrichis, qui est utilisé en particulier pour écrire des pages web est appelé le format HTML. En HTML, pour mettre un passage en gras, on le délimite par les *balises* `<b>` et `</b>` et pour le mettre en italique, on le délimite par les balises `<i>` et `</i>`. Ainsi le texte :

```
Ma <i>première</i> page <b>web</b>
```

s'affiche dans un navigateur :

> Ma *première* page **web**

Comme les parenthèses, les balises vont par deux : on ouvre le passage à mettre en gras avec la balise `<b>` et on le ferme avec la balise `</b>`.

Une division du texte est délimitée par les balises `<div>` et `</div>`, ainsi le texte :

```
<div>Ma première page web
<div> comporte une première sous-division pour dire « Bonjour tout le monde ! »</
div>
<div> et une seconde qui finit par « À bientôt ! »</div>
</div>
```

s'affichera dans le navigateur en rendant ces divisions explicites.

Comme les parenthèses, les balises peuvent s'emboîter les unes dans les autres, mais pas se chevaucher.

On indique qu'un passage est un titre en le délimitant par les balises `<h1>` et `</h1>` et que c'est un sous-titre en le délimitant par les balises `<h2>` et `</h2>`.

De même, les autres structurations du texte comme les énumérations ou les tableaux sont exprimées par d'autres balises.

Quand on écrit un texte, il est fréquent de mentionner d'autres textes : par exemple, de parler dans une lettre d'un livre que l'on a lu. Dans le cas d'un texte manuscrit ou imprimé, on donne en général une référence du texte cité, par exemple le titre du livre et son auteur, afin que le lecteur puisse s'y référer s'il le souhaite. Quand on veut exprimer, dans une page web, une référence à une autre page, on peut faire mieux que simplement indiquer l'adresse de la page web en question (par exemple l'adresse http://fr.wikipedia.org/wiki/Hypertext_Markup_Language) ; on peut changer l'apparence du passage où l'on fait la référence, pour indiquer au lecteur que s'il clique sur ce passage, le navigateur affichera la page demandée. On utilise pour cela les balises `<a>` et `</a>` : on encadre la partie du texte à qualifier par ces deux balises et on indique à l'intérieur de la balise `<a>` l'adresse de la page référencée. Par exemple le texte :

```
Pour les détails sur le langage HTML, on pourra consulter <a href = "http://
fr.wikipedia.org/wiki/Hypertext_Markup_Language">la page <i>Hypertext Markup
Language de Wikipédia</i></a>.
```

qui affiche dans un navigateur :

> Pour les détails sur le langage HTML, on pourra consulter la page *Hypertext MarkupLanguage de Wikipédia*.

Si l'on clique sur le passage en bleu et souligné, le navigateur affiche la page dont l'adresse est http://fr.wikipedia.org/wiki/Hypertext_Markup_Language. Un tel passage sur lequel on peut cliquer pour accéder à une autre page s'appelle un *lien*, et un texte qui contient au moins un lien est un *hypertexte*.

Les balises `<body>` et `</body>` délimitent le texte à afficher dans le navigateur. On indique avant ces informations les méta-données relatives à la page : son titre, le format utilisé pour les lettres accentuées, etc.

Voici, au bout du compte, un exemple de texte au format HTML :

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta>
  <title>Un exemple</title>
</head>

<body>
  <h1>Un titre</h1>
  <h2>Un sous-titre</h2>
  <div><a href="http://www.wikipedia.org/">Un lien</a></div>
  <div><b>Un passage important</b></div>
</body>
</html>
```

L'en-tête situé entre les deux balises `<head>` et `</head>` indique d'une part que le texte est exprimé en UTF-8, c'est l'objet de la ligne :

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

et d'autre part que le titre de la page est `Un exemple`.

Le contenu est situé entre les balises `<body>` et `</body>`. On y retrouve les balises `<b>`, `</b>`, `<i>`, `</i>`, `<h1>`, `</h1>`, `<h2>`, `</h2>`, `<div>`, `</div>`, `<a>` et `</a>` que l'on a décrites. Dans un navigateur, le texte s'affiche ainsi.

SAVOIR-FAIRE Écrire une page en HTML

Écrire le texte contenu dans cette page. Structurer ce texte en divisions. Identifier les titres de parties, les passages à mettre en gras, en italique, etc. et les références vers d'autres pages. Ajouter les balises `<body>` et `</body>` autour du corps du texte, l'en-tête qui contient les méta-données et terminer avec les balises `<html>` et `</html>`.

Exercice 8.10 (avec corrigé)

Écrire une page HTML qui présente les différents projets informatiques des élèves d'une classe.

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta>
  <title>Projets Ada Lovelace</title>
</head>

<body>
  <h1>Les projets de la classe de TS1 du Lycée <a href = "http://
www.adalovelace.fr">Ada Lovelace</a></h1>
  <div>
   <a href="http://www.adalovelace.fr/informatique/ts1/projets/backgammon/
index.html">Un programme qui <b>joue aux Backgammon</b></a>
  </div>
  <div>
   <a href="http://www.adalovelace.fr/informatique/ts1/projets/compression/
index.html">Un programme qui <b>compresse des images</b></a>
  </div>
  <div><a href="http://www.adalovelace.fr/informatique/ts1/projets/montecarlo/
index.html">Un programme qui <b>calcule des intégrales</b> sans peine</a>
  </div>
</body>
</html>
```

Exercice 8.11

Écrire une page HTML qui présente la liste des concerts et des spectacles présentés dans un théâtre.

Exercice 8.12

Changer le texte HTML `Ma <i>première</i> page web` pour que le mot « première » apparaisse non en italique, mais en gras.

Exercice 8.13

Ce texte HTML est incorrect. Comment le corriger ?

```
Il faut <i>comprendre/i> le codage des objets numériques pour les maîtriser.
```

Exercice 8.14

Dans ce texte, vers quel site web pointe le lien ?

```
Votre compte bancaire présente une anomalie. Cliquer <a href="http://grosse-
arnaque.com">ici</a> pour avoir de l'aide.
```

Comment ce texte s'affiche-t-il dans un navigateur ? Quel est l'intérêt de regarder le source HTML de cette page avant de cliquer ?

Exercice 8.15

Donner le source HTML du texte suivant sachant que le texte en bleu et souligné est un lien vers la page http://www.monlivre.fr/page2 :

> On pourra consulter la page suivante.

Ai-je bien compris ?

- Comment représente-t-on un caractère ?
- Quelle est la différence entre le code ASCII et le format Unicode ?
- Quelle est la différence entre le format Unicode et le format HTML ?