

Représentation des textes

Fiche de synthèse

① ENCODER UN TEXTE

Définition 1 (Encodage) Un **encodage** est une convention qui associe un nombre, donc une suite de bits, à chaque caractère : lettre, chiffre, signe de ponctuation, symbole.

② LE CODE ASCII

Le code **ASCII** (1963), conçu pour l'anglais, code 128 caractères de 0 à 127 : 95 caractères affichables et 33 caractères de commande. « A » vaut 65, soit 41_{16} ou 01000001_2 ; « a » vaut 97. Sept bits suffisent, mais chaque caractère occupe un octet dont le premier bit vaut 0. L'ASCII ne contient aucune lettre accentuée.

☞ **Décoder un texte en ASCII binaire.** Découper la suite de bits en octets, traduire chaque octet en décimal, puis chercher le caractère correspondant dans la table ASCII.

③ L'ASCII ÉTENDU

Le huitième bit, libre, a servi à ajouter 128 caractères, lettres accentuées et symboles. Mais chaque système a fait ses propres choix, appelés **pages de code** : ISO-8859-1, ou Latin-1, pour l'Europe occidentale, sa variante Windows-1252, ISO-8859-2 pour l'Europe centrale. Un même octet désigne ainsi des caractères différents d'une page à l'autre.

④ UNICODE ET UTF-8

Unicode donne à chaque caractère de toutes les écritures un numéro unique, son **point de code**, noté U+ suivi de chiffres hexadécimaux : U+0041 pour « A », U+00E9 pour « é », U+20AC pour « € ». Ses 128 premiers caractères sont ceux de l'ASCII. Trois formats écrivent un point de code en octets : UTF-32 sur 4 octets, UTF-16 sur 2 ou 4, et **UTF-8** sur 1 à 4 octets, le plus utilisé, car un texte ASCII y reste identique.

Point de code	Octets	Forme binaire
jusqu'à U+007F	1	0xxxxxxx
jusqu'à U+07FF	2	110xxxxx 10xxxxxx
jusqu'à U+FFFF	3	1110xxxx puis deux 10xxxxxx
jusqu'à U+10FFF	4	11110xxx puis trois 10xxxxxx

« é » a pour point de code 233, soit 11101001_2 : il faut deux octets. Ses onze bits, 000 1110 1001, remplissent les x :

11000011 10101001, soit C3 A9. En Latin-1, « é » est l'unique octet E9.

⚠ Attention. Un texte UTF-8 lu comme du Latin-1 affiche « Ã© » à la place de « é » ; un texte Latin-1 lu comme de l'UTF-8 affiche des caractères de remplacement, un point d'interrogation dans un losange. Ces défauts signalent toujours un mauvais choix d'encodage à la lecture.

⑤ EN PYTHON

```
print(ord('B'))           # 66
print(chr(67))            # C
print("é".encode("utf-8")) # b'\xc3\xa9'
print("é".encode("latin-1")) # b'\xe9'
print(b'\xc3\xa9'.decode("utf-8")) # é
```

`ord` donne le point de code d'un caractère, `chr` le caractère d'un point de code. `encode` donne les octets d'une chaîne dans un encodage, `decode` reconstitue la chaîne. `\xc3` désigne l'octet C3. Pour lire ou écrire un fichier, on précise toujours l'encodage : `open(nom, encoding="utf-8")`.

⑥ TEXTE SIMPLE ET TEXTE ENRICHI

Un **texte enrichi** ajoute au texte une mise en forme, police, gras, titres, liens, décrite par des **balises** : HTML, XML, RTF, les formats de bureautique `.odt` et `.docx`, ou **L^AT_EX**, qui excelle dans les formules mathématiques. Un texte simple n'est qu'une suite de caractères.

📌 À retenir.

- Un encodage associe un nombre à chaque caractère.
- ASCII : 128 caractères sur 7 bits, sans accents ; les pages de code de 8 bits sont incompatibles entre elles.
- Unicode numérote tous les caractères ; UTF-8 les écrit sur 1 à 4 octets.
- `ord` et `chr` passent du caractère au point de code et inversement.