par Philippe Poulard

Internationalisation avec XML

INRIA SOPHIA ANTIPOLIS
Philippe.Poulard@sophia.inria.fr

Unicode

- i18n dans le WWW
- Un bug dans la specification !
- Unicode et ISO/IEC 10646
- Composition des caractères
- Adressage UCS
- Jeux de caractères
- Encodages UTF

XML

- Entités caractères
- Unicode et XML
- Formats de données XML
- Langues dans XML
- Transmission XML avec HTTP

L'idée d'un World Wide Web :

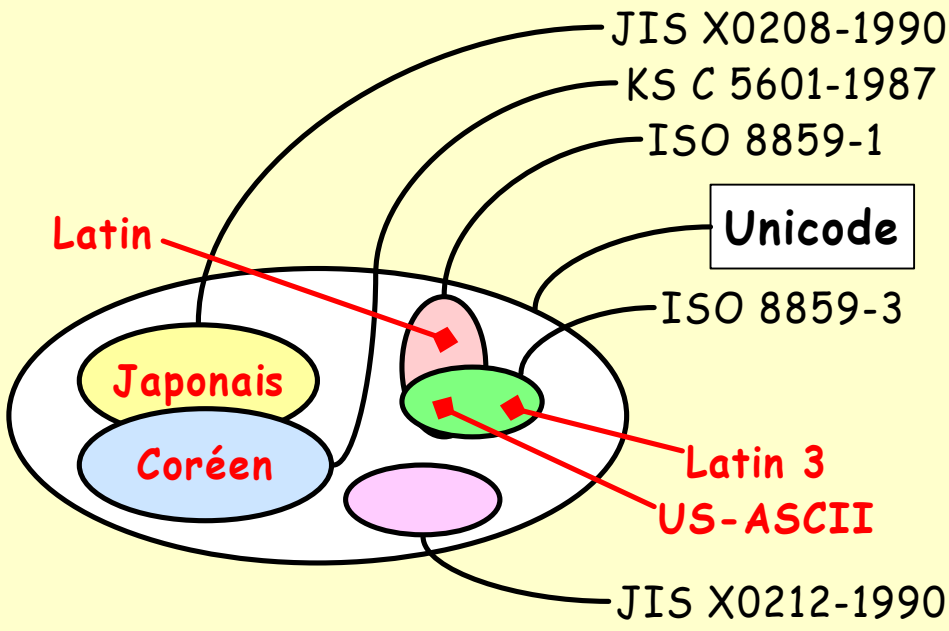
Vue des clients du WWW pour recevoir des réponses compréhensibles d'un serveur Web : qu'importe d'où ils y accèdent dans le monde, et qu'importe dans quel langage et quel encodage sont les données auxquelles ils accèdent.

Intéropérabilité et communication globale.

Le WWW est une application
- Les différentes parties doivent coopérer

La représentation doit être indépendante de la localisation
- Les données devraient être visible par n'importe qui n'importe où

Doit s'appuyer sur des standards (XML, Unicode)



Un bug dans la spec

http://www.w3.org/TR/NOTE-sgml-xml-971215

Déclaration SGML de XML 1.0 (caractères)

```
CHARSET
  DESCSET
    0         9  UNUSED
    9         2         9
   11         2  UNUSED
   13         1        13
   14        18  UNUSED
   32        95        32
  127         1  UNUSED
  128        32  UNUSED
  160     55136       160
55296      2048  UNUSED   -- surrogates --
57344      8190   57344
65534         2  UNUSED   -- FFFE and FFFF --
65536   1048576   65536
```

XML : un standard dérivé de SGML
(utilise une déclaration SGML fixe)

???

Définition [2] dans XML 1.0 (Second Edition)

```
Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
/* any Unicode character, excluding the surrogate blocks, FFFE, and FFFF. */
```

(seizets d'indirection)

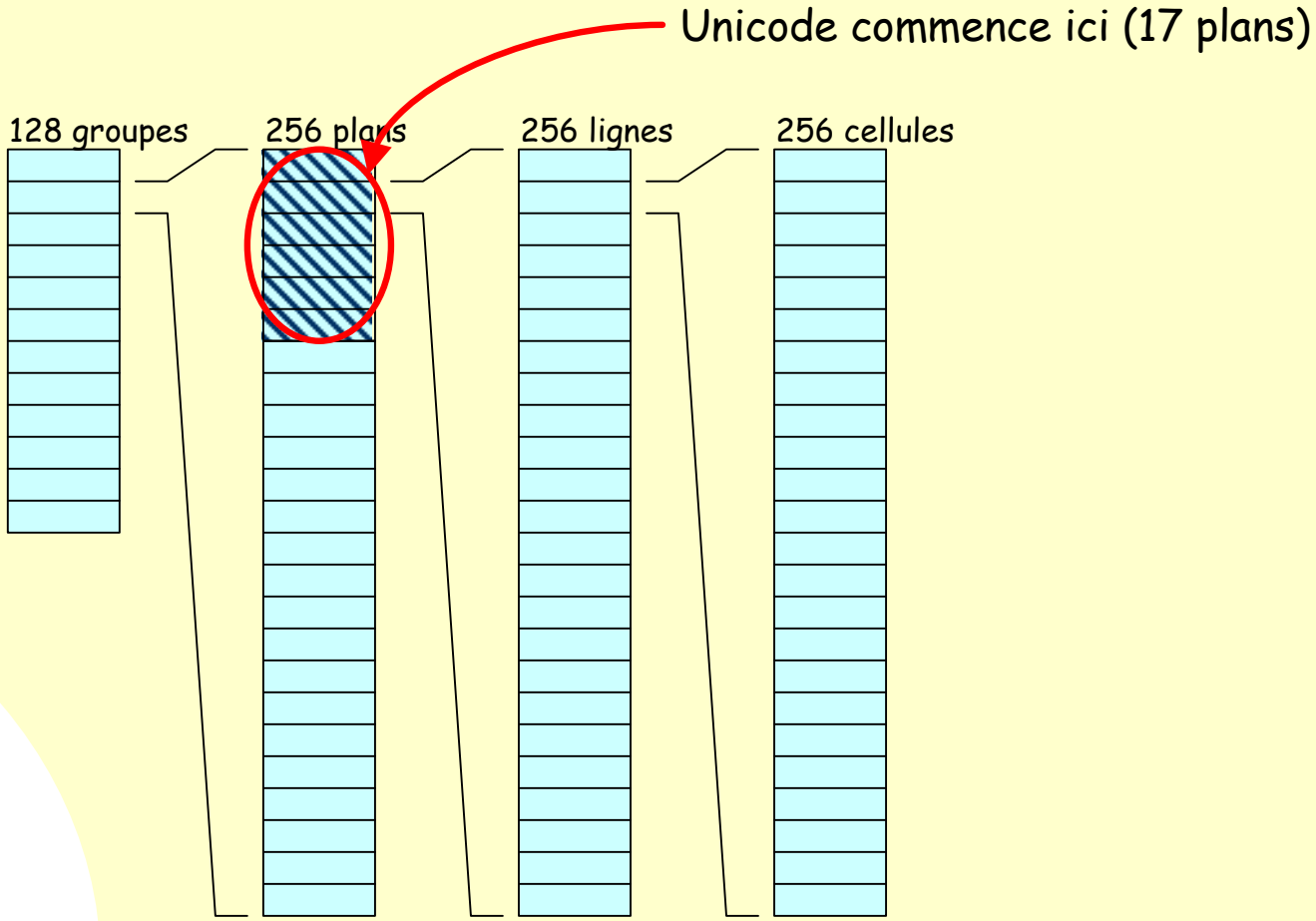
Adressage avec ISO/IEC 10646 et Unicode

ISO / IEC 10646 = 2 milliards de caractères

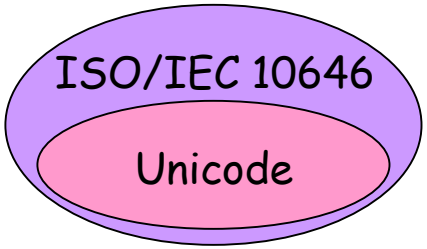
2^{31} codes positions = 128 groupes x 256 plans x 256 lignes x 256 cellules

Unicode = 1 million de caractères

2^{20} codes positions = 17 plans x 256 lignes x 256 cellules

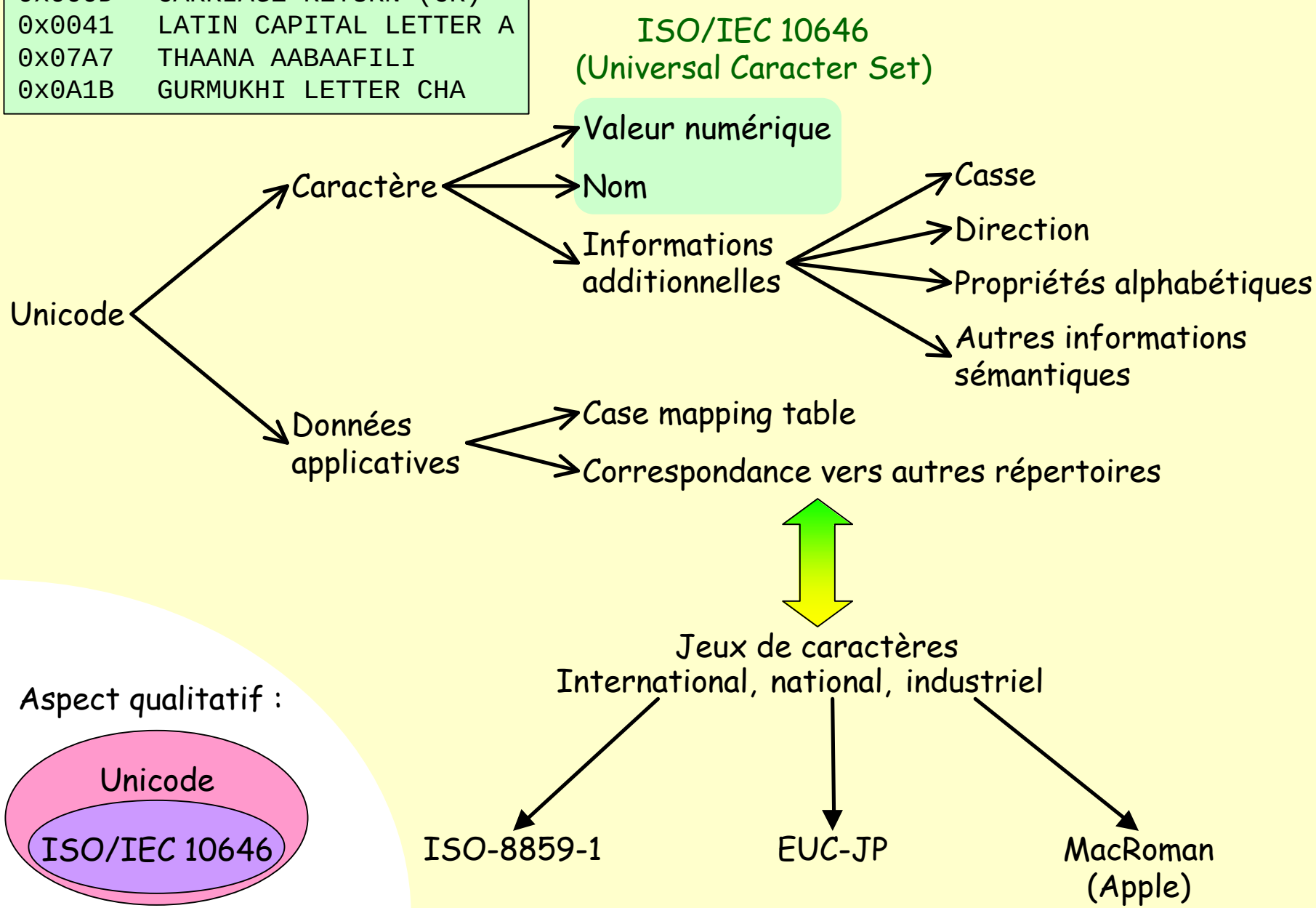


Aspect quantitatif :



Relations entre Unicode et ISO/IEC 10646

Valeur	Nom
0x000D	CARRIAGE RETURN (CR)
0x0041	LATIN CAPITAL LETTER A
0x07A7	THAANA AABAAFILI
0x0A1B	GURMUKHI LETTER CHA



Aspect qualitatif :

Composition des caractères

Caractères composites

â ← a + ◌̂

Peut poser des problèmes pour comparer des chaînes de caractères ou des documents

Caractères précomposés

ü ← 0x00FC

ü ← u + ◌̈ ← 0x0075 + 0x0308

Base + diacritique

u + modification

Ordre alphabétique respecté

String matching avec XML

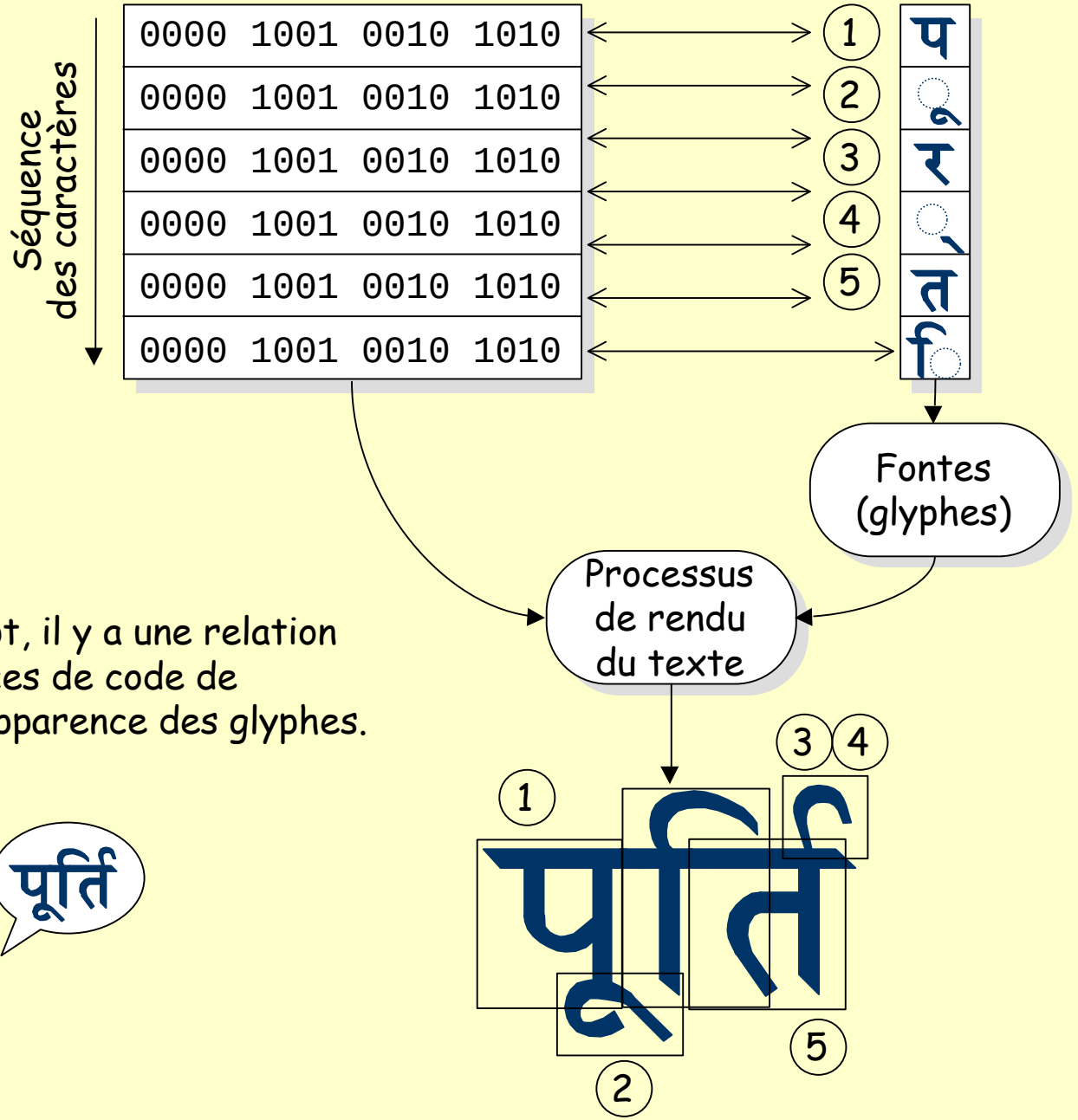
Correspondance de chaînes de caractères ou de noms dans XML :

Caractères à représentation multiple
(forme précomposée et base+diacritique)

Correspondent s'ils ont la même représentation dans les 2 chaînes
Les processeurs peuvent **normaliser** les correspondances entre chaînes en ayant recours à leur **forme canonique**

Composition des caractères

Unicode ne spécifie pas le rendu des caractères



Pour chaque script, il y a une relation entre les séquences de code de caractères et l'apparence des glyphes.

पूर्ति

ISO / IEC 10646 = 2 milliards de caractères

2^{31} codes positions = 128 groupes x 256 plans x 256 lignes x 256 cellules

Unicode = 1 million de caractères

2^{20} codes positions = 17 plans x 256 lignes x 256 cellules

Découpage Unicode :

- BMP : Basic Multilingual Plane (premiers 64k)
- 16 plans suppléants (surrogate)

Unicode 3.1
94140 caractères alloués
dont
70207 idéogrammes Han unifiés

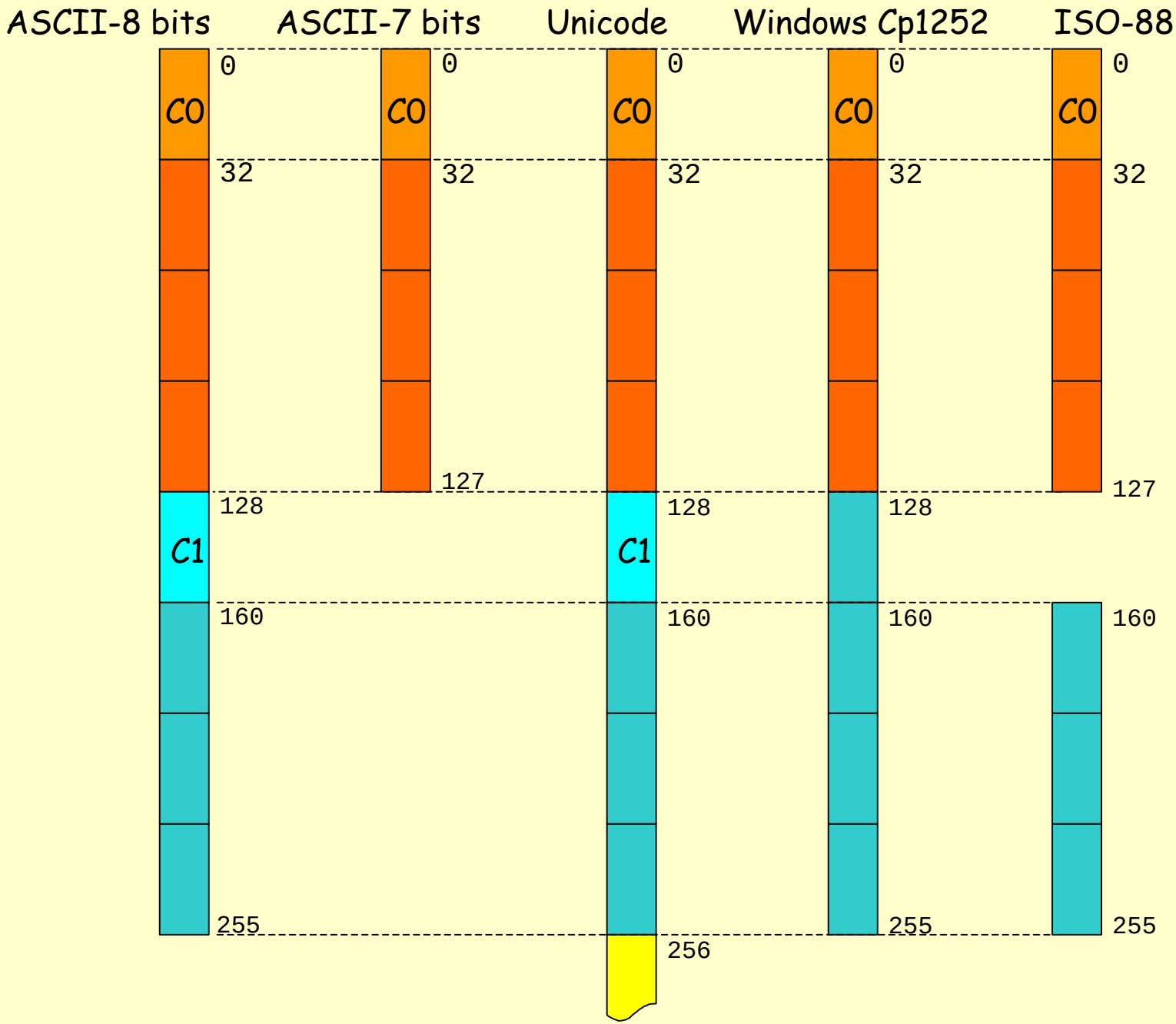
ISO / IEC 10646-1 : BMP	Unicode BMP	UCS-2
ISO / IEC 10646-1 : 127 autres plans	Unicode surrogate : 16 autres plans	UCS-4

Adressages

UCS : Universal Caracter Set
UCS-2 : 2 octets (adressage BMP)
UCS-4 : 4 octets (adressage complet)

Extension des zeros (encodage de longueur fixe)

Bits	Adressage	Hex	Dec	Car	Binaire
7	US-ASCII	41	65	A	1000001
8	ASCII 8bits	41	65	A	01000001
16	UCS-2	41	65	A	00000000 01000001
32	UCS-4	41	65	A	00000000 00000000 00000000 01000001


ASCII-8 bits
ASCII-7 bits
Unicode
Windows Cp1252
ISO-8859-1
C0
C1
0
32
127
128
160
255
256

Jeux de caractères propriétaire : le cas Windows

ASCII	7 bits	caractères de 32 à 127 + caractères de contrôles C0 (de 0 à 31) + caractère 127 (DELETE)
ASCII	8 bits	ASCII + caractères de 160 à 255 + caractères de contrôles C1 (de 128 à 159)
Windows Cp1252		ASCII 8 bits + substitution des contrôles C1 par des caractères
ISO-8859-1		ASCII 8 bits sans les caractères de contrôles C1

Attention : en enregistrant sous un outil trop intégré à windows, on peut obtenir l'erreur suivante :

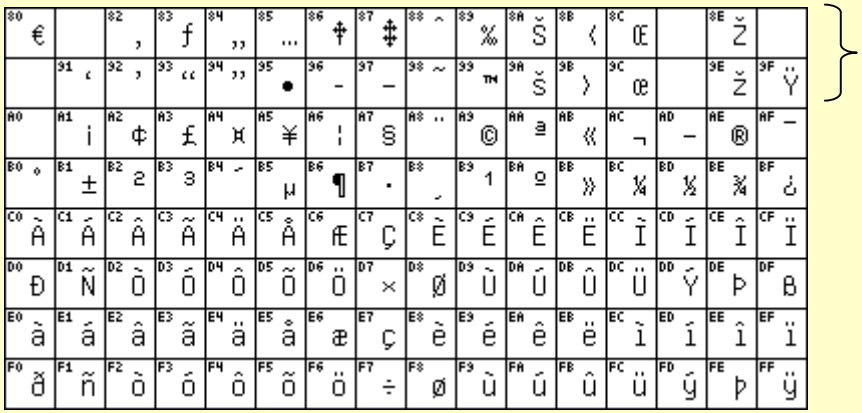
œ

Exemple avec "oe" ligaturé

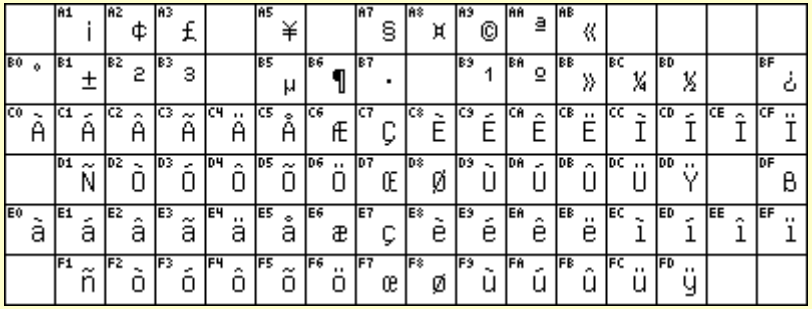
code 156 avec Windows Cp1252 → L'outil sous Windows pourra utiliser ce code…

code 339 dans Unicode non représenté dans ISO-8859-1 → …ce qui provoquera une erreur si l'encodage ISO-8859-1 est utilisé

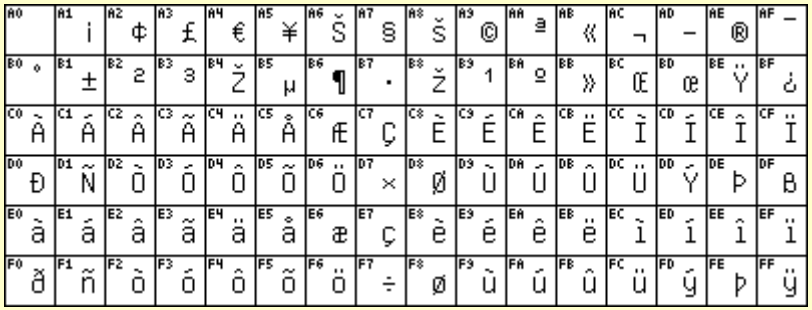
Windows Cp1252



ISO-8859-1



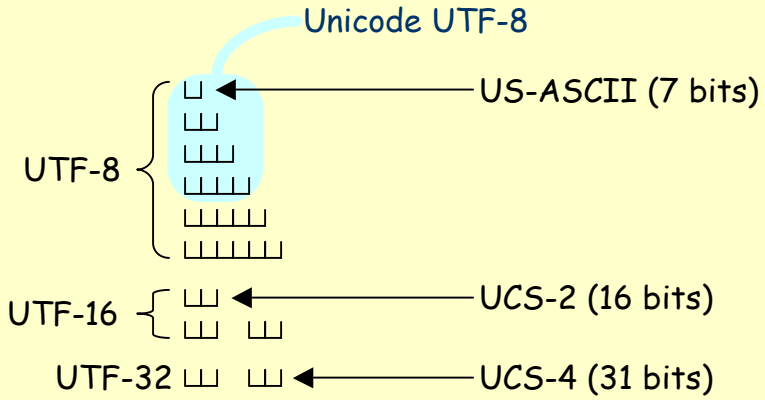
ISO-8859-1



UTF : UCS Transformation Format

UTF-16 : encodage des caractères de 0 à 1048575 (20bits)
sur 2 ou 4 octets

UTF-8 : encodage des caractères de 0 à 2147483647 (31 bits)
sur 1 a 6 octets
compatible US-ASCII



Encodages UTF-8 et UTF-16

Plage UCS-4 (hexa)	Séquence d'octet UTF-8 (binaire)	
de 0000 0000 à 0000 007F	0xxxxxxx	BMP
de 0000 0080 à 0000 07FF	110xxxxx 10xxxxxx	
de 0000 0800 à 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx	
de 0001 0000 à 001F FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	Surrogate
de 0020 0000 à 03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	
de 0400 0000 à 7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	

Unicode :
de 0000 0000 à 000F FFFF

Plage UCS-4 (hexa)	Bits UCS-4	Séquence d'octet UTF-16 (binaire)
de 0000 0000 à 0000 FFFF	xxxxxxxx xxxxxxxx	xxxxxxxx xxxxxxxx
de 0001 0000 à 000F FFFF	yyyy yyyyyyxx xxxxxxxx	110110yy yyyyyyyy 110111xx xxxxxxxx

Les 2048 valeurs entre 0xD800 et 0xDFFF sont spécialement réservées pour une utilisation avec UTF-16 et n'ont pas de caractères assignés.

Sur la plage d'adressage de 0x0000 à 0xFFFF (65536 valeurs) il y a en fait :
63486 caractères assignés dans le BMP
+ 2048 qui servent d'indirection pour l'adressage des 16 plans suppléants
+ 2 caractères non assignés (0xFFFE et 0xFFFF)

Caractère de contrôle BOM

Valeur	Nom
0xFEFF	ZERO WIDTH NON-BREAKING SPACE

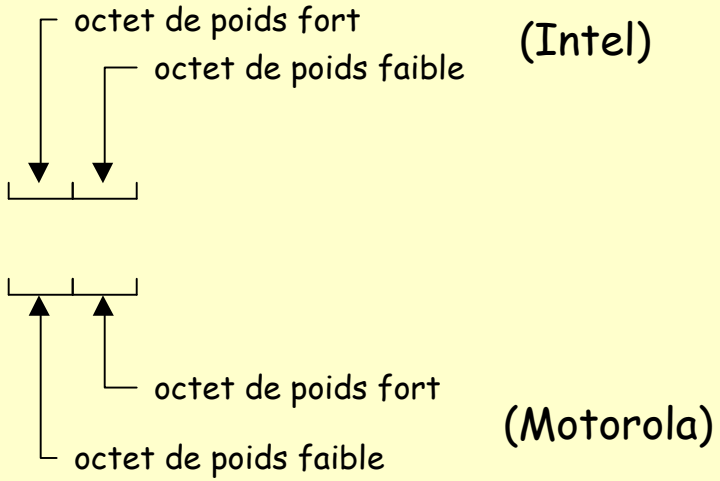
Première paire d'octet en UTF-16 :

Valeur	Nom
0xFEFF	BYTE ORDER MARK

Ordre d'apparition :

FE FF → grand-boutiste (big-endian)

FF FE → petit-boutiste (little-endian)



Adressage ou encodage ?

Ne pas confondre adressage et encodage…

Unicode permet d'**adresser** les caractères :
de `0x000000` à `0x00D7FF`
de `0x00E000` à `0x00FFFD`
de `0x010000` à `0x10FFFF`

Les caractères `0xFE` et `0xFF` n'apparaissent jamais dans un document **encodé** en UTF-8, mais il est possible de les encoder (puisqu'ils sont adressables).

Un document XML n'adresse pas tout Unicode, puisqu'il exclue les caractères
de `0x00` à `0x08`
de `0x0A` à `0x0C`
de `0x0E` à `0x1F`

Un adressage spécifie une (des) plage(s) de valeurs.
Un encodage est une représentation binaire, éventuellement de longueur variable, de tout ou partie de ces adresses.

Exemples d'encodage

…vus par octet

```
<html>
&agrave; la p&ecirc;che on demande : « &ccedil;a mord ? »
</html>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<document>
à la pêche on demande : « ça mord ? »
</document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
+á la p+¬che on demande : -½-á+ºa mord ?-á-+
</document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
&#224; la p&#234;che on demande : « &#231a mord ? »
</document>
```

Unicode big-endian

```
   < ? x m l   v e r s i o n = " 1. 0" ? >
 < d o c u m e n t >
 Ó   l a   p Û c h e   o n   d e m a n d e   :   ½ á þ a   m o r d   ? á +
 < / d o c u m e n t >
```

Appels de caractères

Permet d'utiliser n'importe quel caractère Unicode, même hors de la portée de l'encodage utilisé dans le document
Appel avec la valeur numérique décimale ou hexadécimale

Syntaxe

```
&#1114;
&#x45A;
```

Glyphe	Valeur	Nom
?	0x045A	CYRILLIC SMALL LETTER NJE

Restrictions sur les appels de caractère
Ne s'utilise que dans :

- les contenus de balises
- les valeurs d'attributs
- les commentaires

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE document [
        <!ENTITY nje "&#x45A;">
]>
<document>
        &nje;
</document>
```

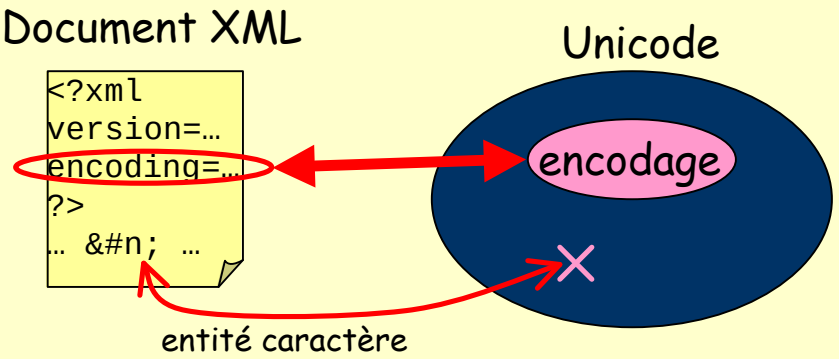
Spec. XML : UTF-8 et UTF-16
Par defaut : UTF-8

```
<?xml version="1.0" encoding="encoding" ?>
<monDocument>
            …
</monDocument>
```

	encoding
7-bit ASCII	US-ASCII
8-bit UCS Transformation Format	UTF-8
16-bit UCS Transformation Format	UTF-16
Unicode	ISO-10646-UCS-2
UCS	ISO-10646-UCS-4
Latin-1, Europe occidentale	ISO-8859-1

IANA
Internet Assigned Numbers Authority
http://www.iana.org/assignments/character-sets

Les applications ne voient que des caractères Unicodes



Il n'y a pas de données binaires dans un document XML.
(Cela rend les fichiers lisibles dans un bloc-note)

Tous les caractères sont des caractères Unicode

```
<dauphin date-naissance="1/4/1997">
          <nom>Flipper</nom>
          <sexe>M</sexe>
          <taille unité="cm">215</taille>
          <poids unité="kg">105</poids>
</dauphin>
```

Ceci n'est pas une date
C'est une chaîne de caractères
contenant **1/4/1997**

Ceci n'est pas un nombre
C'est une chaîne de caractères
contenant **105**

Les nombres de type "octet" sont représentés par 1 à 3 caractères Unicodes

Attribut xml:lang

```
<p xml:lang="la">alea jacta est</p>
<p xml:lang="fr">aller à la gare de l'est</p>
```

```
<p xml:lang="fr-CA">Gare ton char, Ben-hur</p>
```

Codes de langue ISO-639 ou IANA (plus complet)

IANA
http://www.isi.edu/in-notes/iana.org/assignments/languages/tags

ISO-639	no-bok	Norvégien littéraire
	no-nyn	Norvégien nouveau
IANA	i-navajo	Navajo
	i-mingo	Mingo
personnels	x-verlan	Verlan
	x-klingon	Klingon

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<èmepo xml:lang="x-verlan">
    <treti>Le beaucor et le narre</treti>
    <p>Tremaî beaucor sur un brear chéper, naite un gemafro en son bec</p>
    <p>Tremaî narre par l'deuro chéléal …/…</p>
</èmepo>
```

Transmission XML avec HTTP

Les en-têtes HTTP sont décrits dans la spécification MIME

La RFC 3023 "XML media types" décrit les types de médias pour XML

```
Content-type: text/xml; charset="utf-8"
<?xml version="1.0" encoding="utf-8"?>
```

Le jeu de caractères est **utf-8**

```
Content-type: text/xml
<?xml version="1.0" encoding="utf-8"?>
```

Le jeu de caractères est **us-ascii**, malgré l'encodage spécifié dans le document xml

```
Content-type: application/xml; charset="utf-16"
{BOM}<?xml version="1.0" encoding="utf-16"?>
```

Le jeu de caractères est **utf-16**

```
Content-type: application/xml
<?xml version="1.0" encoding="un-encodage-connu-de-votre-parser"?>
```

Le jeu de caractères est **un-encodage-connu-de-votre-parser**