

5. Tractament de formats

5.1. Introducció

El traductor sovint s'ha d'enfrontar amb problemes derivats de la gestió dels formats i la codificació de caràcters dels fitxers que ha de traduir. Per aquest motiu és imprescindible tenir unes nocions sobre el funcionament bàsic dels formats i les codificacions. Amb aquests coneixements evitarem produir errors que sovint fan perdre una quantitat de temps considerable.

El capítol comença amb un repàs a una sèrie de conceptes molt bàsics i probablement coneguts per al lector, però que són del tot imprescindibles per a comprendre la resta del capítol, especialment al que fa al tema de les codificacions de caràcters.

En aquest capítol també presentem a fons el format XML, cada cop més utilitzat, i per tant, cada vegada més traduït. Aprendrem a traduir documents XML amb les eines de traducció més habituals. També dedicarem un espai a repassar els formats basats en XML que s'utilitzen al món de la traducció: TMX, TBX, SRX i XLIFF i que hem anat veien en els capítols anteriors.

5. 2. Representació de la informació textual: codi de caràcters

5.2.a. Sistemes de numeració

Les persones estem acostumades a fer servir un codi de numeració decimal, és a dir, un sistema que té un total de 10 símbols: 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9. Si volem representar números més grans afegim un o més dígits a l'esquerra, per exemple, 10, 34, 234, 1234, etc.

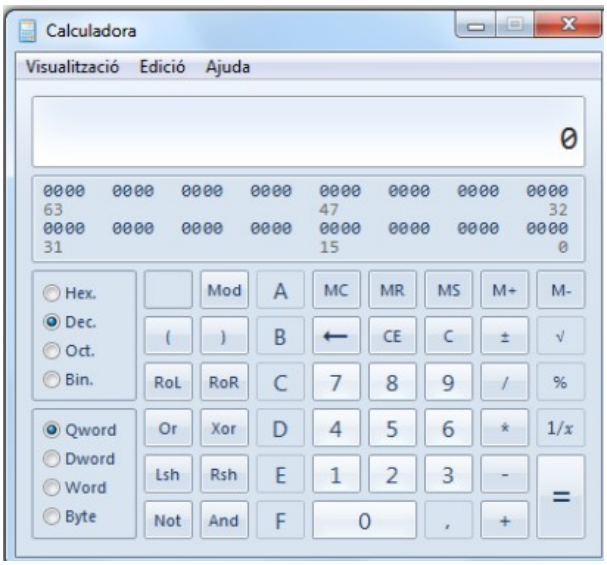
Aquest sistema de numeració no és l'únic, existeixen d'altre, dos dels quals són molt emprats en informàtica. Un d'ells és el *sistema binari*. El sistema binari només compta amb dos símbols, el 0 i l'1. De la mateixa manera que en el sistema decimal, si necessitem representar números més grans afegirem un o més dígits a l'esquerra, per exemple, 10, 11, 101, 11001011.

Un altre sistema molt emprat és el *sistema hexadecimal*, que compta amb 16 símbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E i F. De la mateixa manera que amb el sistema decimal i el sistema binari, si volem representar números més grans afegirem xifres a l'esquerra: 10, 23, 2F, A10, BE3. A la taula següent podeu veure alguns exemples de conversió entre els tres sistemes de numeració.

Decimal	Binari	Hexadecimal
0	0	0
1	01	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	0
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

Altres exemples de conversions: el número decimal 2003 és 11111010011 en binari i 7D3 en hexadecimal. El número hexadecimal F03A correspon al decimal 61498 i al binari 1111000000111010.

Hi ha una sèrie operacions matemàtiques no gaire complicades per passar d'un sistema de numeració a un altre, però no les estudiarem en aquest capítol. Per si algun dia necessiteu transformar d'un sistema a un altre (mireu els exercicis d'aquest mateix capítol), recordeu que hi ha moltes calculadores científiques que realitzen aquestes operacions. Les calculadores integrades a la majoria de sistemes operatius són capaces de dur a terme aquestes conversions. Obriu la calculadora de Windows, la que apareix per defecte és una calculadora simple. Per canviar de tipus de calculadora es pot anar a *Ver* i escoollir la calculadora de *Programador* (si la teva versió de Windows no disposa d'aquest mode de calculadora, llavors hauràs de seleccionar la *Científica*):



Quan s'obre la calculadora normalment està en mode Dec (decimal). Podeu introduir un número en decimal i per passar-lo a binari, per exemple, només caldrà que seleccioneu el mode Bin (binari).

En aquest mode podreu escriure números en binari. També estan disponibles els modes Hex (hexadecimal) i Oct (Octal, no l'hem explicat, és un sistema de numeració que té 8 símbols). Fixeu-vos que quan teniu seleccionat el mode Dec. les tecles numèriques del 0 al 9 estan activades, ja que totes aquestes són xifres vàlides en aquest sistema de numeració. En canvi, quan estem en mode Bin. només queden activades les tecles 0 i 1, que són les úniques vàlides. Quan active el mode Hex. a més de les tecles numèriques del 0 al 9 s'activen també les tecles de la A a la F, que recordeu que en hexadecimal són xifres.

Ara podeu fer els següent exercici:

Passeu els següents números:
De decimal a hexadecimal: 23, 269, 62165
De decimal a binari: 3, 15, 56, 258, 1645
De hexadecimal a decimal: 4, 1A, FE0, 10C0

5.3. El llenguatge intern de l'ordinador i les unitats de mesura en informàtica

L'ordinador internament només treballa amb 0 i 1 (pas o no de corrent elèctric), és a dir treballa amb un sistema binari. A la informació donada per un únic dígit amb sistema binari (0 o 1) l'anomenem bit (el nom prové de *binary digit*). Per tal de poder expressar una major quantitat d'informació els bits s'agrupen en grups de 8 que anomenen *byte*. Un byte pot prendre 256 valors diferents (28).

5.3.a. Els múltiples del byte

Com que la base dels càlculs en informàtica és el bit i aquest només admet dos valors, totes les mesures es realitzen amb nombres que són potències de 2. Un *quilobyte* (KB) són 1000 bytes (de fet són 1024 ja que aquesta és la potència de 2 més propera al 1000). Un megabyte (MB) són 1.000.000 de bytes (en realitat 1024x1024=1.048.576 bytes). Un gigabyte (GB) són 1.000.000.000 de bytes (en realitat 1024x1024x1024= 1.073.741.824 bytes).

Per tenir una idea de les capacitats d'emmagatzematge de diferents unitats, teniu en compte un disc dur estàndard té actualment entre 500GB i 1 TB en un CD_ROM caben fins a 700 MB, en un DVD 4,7 GB.

5.4. Representació d'informació no numèrica

Com hem vist els ordinadors treballen amb un codi binari, que és capaç de representar números. Quan treballem amb ordinadors no únicament volem emmagatzemar i treballar amb números, sinó que també hem de ser capaços de processar text, so, imatge, etc. Com podem fer servir un codi numèric per representar un altre tipus de dades? A continuació presentem la representació de text.

5.4.a. Representació de text

La idea bàsica per representar text és assignar a cada caràcter del conjunt que volem representar un valor numèric. Si treballem amb bytes de 8 bits podrem treballar amb 256 caràcters diferents (8 bits, 2^8=128). Tot i que veurem els codis de caràcters en detall més endavant en aquest mateix capítol, veurem el codi de caràcters ASCII (American Standard Code for Information Interchange) (7 bits, 2^7=128) (font Wikipedia http://en.wikipedia.org/wiki/Ascii):

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	(space)
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

Podem representar el codi ASCII d'una manera molt més compacta. Per exemple, el caràcter 'z' correspon al codi hexadecimal 7A, que en decimal equival a 122.

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

Exercici:

Mirant aquesta taula (no la de la plana anterior, feu-la servir per verificar la resposta) digueu el codi hexadecimal i decimal del caràcters següents:

@:
q:
M:
~:

Per cert, si alguna vegada teniu dificultat per teclejar algun caràcter (passa sovint amb ~) en Windows podeu fer Alt + codi decimal del caràcter. Per exemple, per fer la ~ podeu picar 126 mantenint pitjada la tecla Alt.

5.5. Representació de la informació textual: codi de caràcters

A la secció anterior hem vist que podem representar caràcters assignant un codi numèric a cada caràcter del conjunt que volem representar. En aquesta unitat veurem a fons tots els aspectes relacionats amb els diferents codis de caràcters. Dedicarem també una especial atenció a Unicode. Aprendrem també a determinar en quin codi de caràcters està escrit un document i a transformar codis de caràcters.

5.5.a. Conceptes bàsics

En informàtica com a norma general les dades estan representades com a octets. Un octet és una unitat d'informació formada per 8 bits i que pot representar un valor numèric comprès entre 0 i 255 (2^8=256). El concepte d'octet està molt relacionat amb el concepte de byte.

Es poden establir diferents convencions sobre com un octet o una seqüència d'octets representa una dada en concret. Per exemple, sota certs estàndards, quatre octets consecutius sovint representen una unitat que presenta un número real. En aquesta assignatura estem interessats en la representació de caràcters. En el cas més senzill, i que es fa servir molt sovint, és el que un octet representa un caràcter segons una taula de correspondència. La interpretació correcta suposa que es coneix el codi de caràcters que es fa servir. Més endavant veurem quines tècniques hi ha per poder determinar la codificació de caràcters d'un document.

5.5.b. Algunes definicions importants

En aquest apartat intentarem definir alguns conceptes importants. La denominació que es fa servir no és universal i sovint condueix a errors:

- **Repertori de caràcters** (*character repertoire*): és el conjunt de caràcters diferents a representar.
- **Codi de caràcters** (*character code*): és una correspondència, normalment presentada en forma tabular, entre els caràcters d'un repertori de caràcters i un conjunt de números enters positius. És a dir, s'assigna un codi numèric únic a cada caràcter del repertori.
- **Codificació de caràcters** (*character encoding*): És un mètode (algorisme) per presentar els caràcters digitalment fent una correspondència entre les seqüències de codis de caràcters i les seqüències d'octets.

En el cas més simple, a cada caràcter li correspon un número enter entre 0 i 255 i aquest es fa servir com a octet. Naturalment, aquesta possibilitat només funciona per a repertoris de caràcters de com a màxim 256 caràcter (quantitat que no és suficient per a totes les llengües, pensem, per exemple, en el xinès).

5.5.c. Els codis de caràcters més habituals

En aquest apartat descriurem els codis de caràcters més emprats. Deixarem per al següent apartat tot el que fa referència a l'Unicode. Així, aquí exposarem els següents codis:

- ASCII
- La família ISO 8859
- Codis de caràcters de Windows
- Codis de caràcters de DOS
- Codis de caràcters de Machintosh
- La família KOI de codis de caràcters ciríl·lics

Hi ha més codis de caràcters. No cal exposar-los tots, sinó entendre bé el mecanisme de funcionament.

Aprendrem a reconèixer altres codis de caràcters i a transformar-los en propers apartats.

ASCII

Ja hem vist aquest codi de caràcters en la secció anterior. Com vam veure, el codi de caràcters ANSI (*American Standard Code for Information Interchange*) és un codi de 7 bits (128 posicions). Ara estem parlant d'octets o bytes de 8 bits, per tant ens sobra un bit (el primer). Aquest primer bit es pot fer servir com a bit de paritat o bé per disposar de 128 posicions addicionals (128-255).

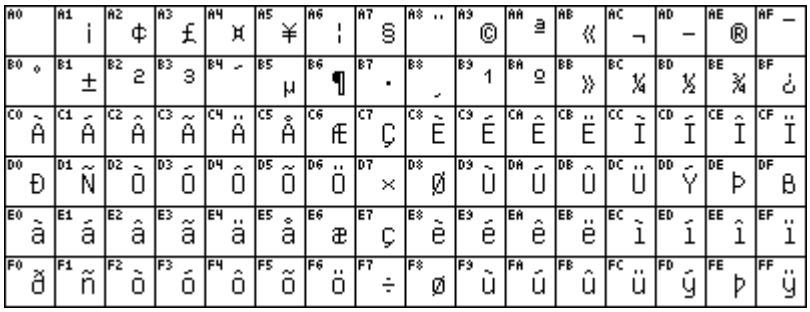
La taula de caràcters corresponents a l'ASCII és la següent:

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

La familia ISO 8859

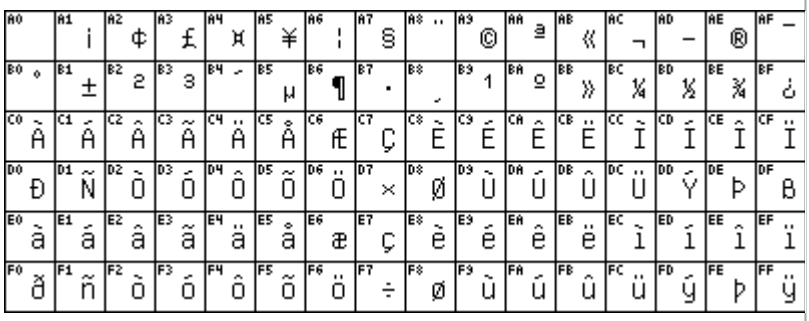
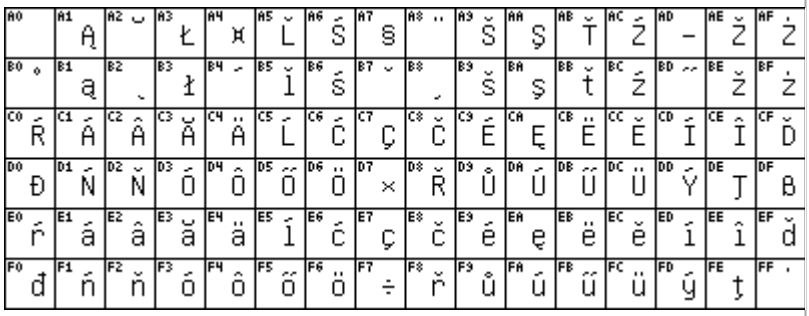
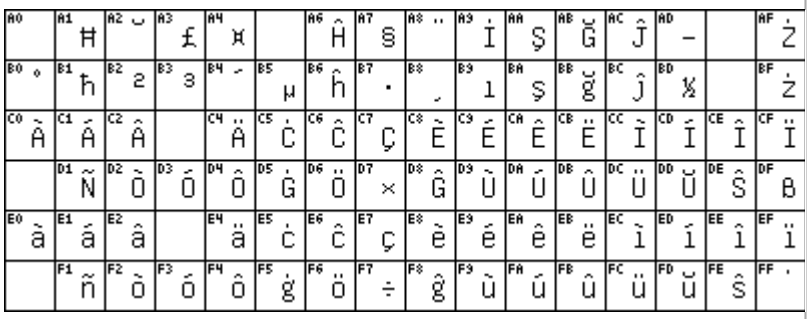
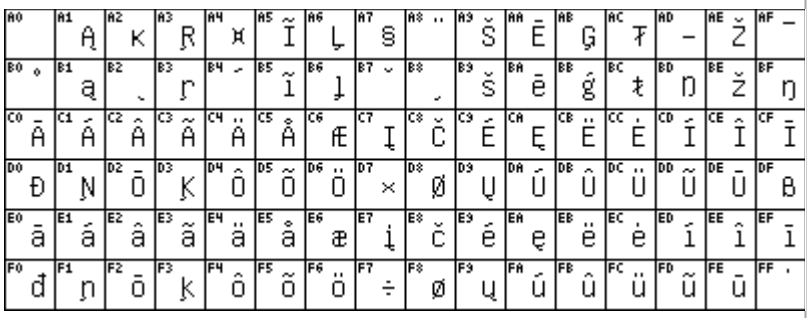
Aquesta família de codis de caràcters està formada per diverses parts, i cada una cobreix els caràcters necessaris per a algunes llengües. Són codis de caràcters de 8 bits (per tant poden codificar 256 caràcters). La part baixa (els 7 primers bits, és a dir, els 128 primers caràcters, de la posició 0 a la 127) de les taules de la família IS0 8859 és exactament igual a la de l'ASCII. La part alta es fa servir per codificar els caràcters no inclusos en el llatí bàsic.

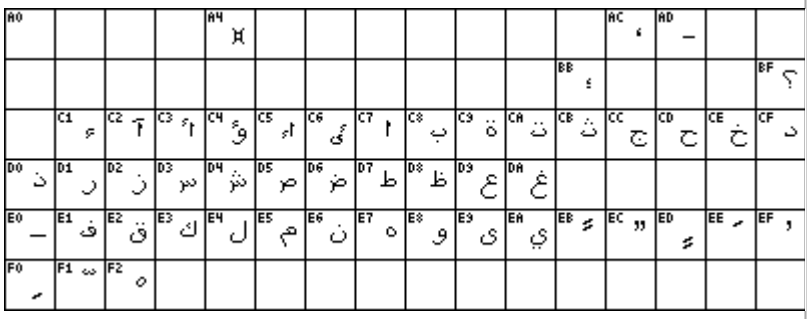
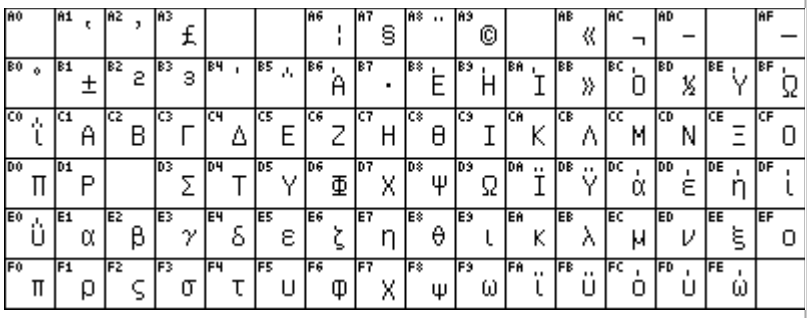
Veiem com a exemple la taula corresponent a la ISO-8859-1 (extreta http://czyborra.com/charsets/), que representa els caràcters de les següents llengües: afrikaans, albanès, eusquera, bretó, català, cors, danès, anglès, feroès, gallec, alemany, islandès, indonesi, irlandès (nova ortografia), italià, llatí (ortografia clàssica bàsica), leonese, luxemburguès (ortografia clàssica bàsica), malai, gaèlic manx, noruec (Bokmål and Nynorsk), occità, portuguès, retoromànic, gèlic escocès, castellà, swahili, suec i való.

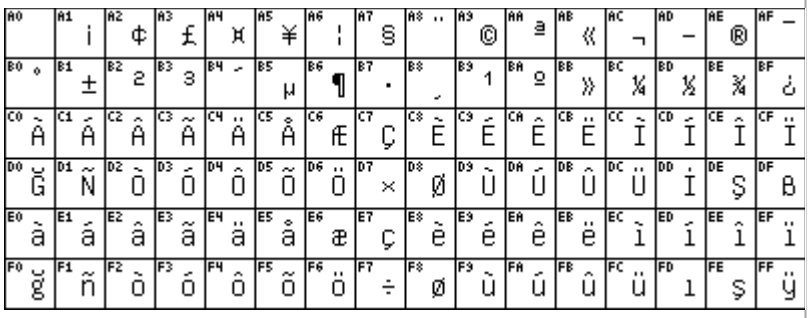
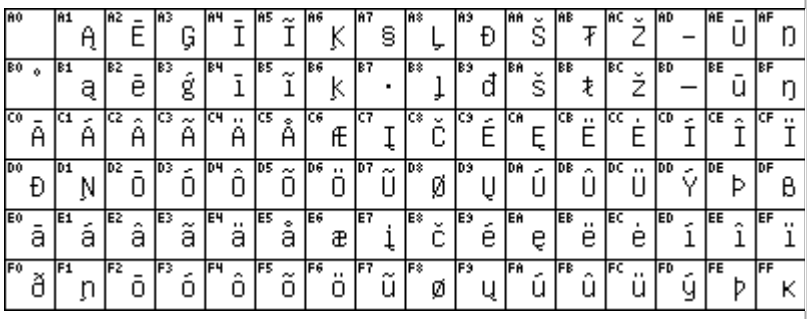


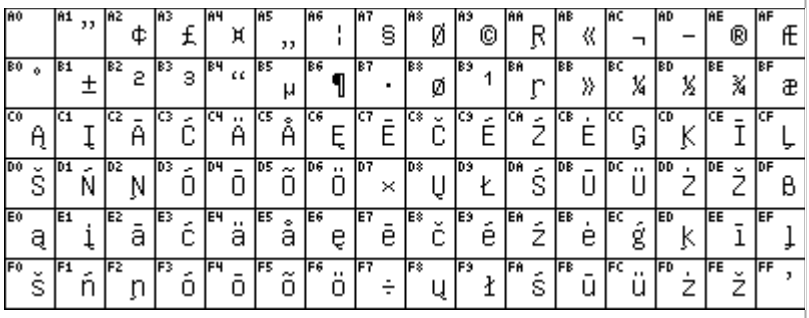
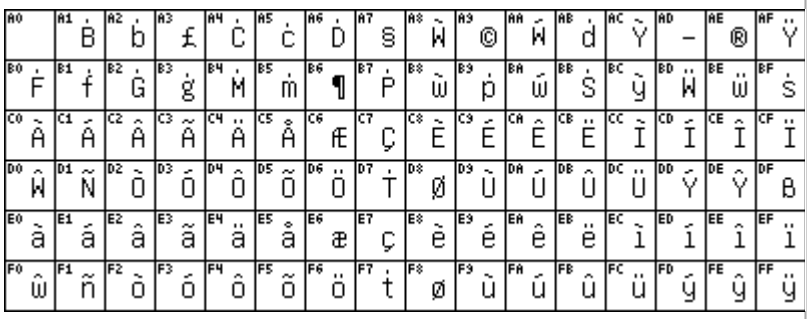
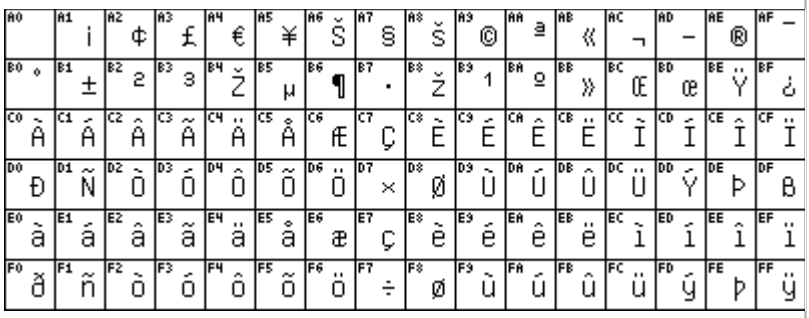
Fixem-nos que la primera posició d'aquesta taula és la A0 (hexadecimal, en decimal és la 160, tot i que aquesta primera posició està buida). Hem comentat que de la posició 0 a la 127 la taula és exactament igual que l'ASCII. Per tant de la posició 128 a la 159 tenim un espai buit que no està ocupat per caràcters imprimibles.

A la taula següent (adaptada de la Wikipedia http://en.wikipedia.org/wiki/ISO_8859 amb les taules de http://czyborra.com/charsets/) podem observar les diferents parts de la ISO 8859

Part 1	*Latin-1 Western European*	
Part 2	*Latin-2 Central European*	
Part 3	*Latin-3 South European*	
Part 4	*Latin-4 North European*	

Part 5	Latin/Cyrillic	
Part 6	Latin/Arabic	
Part 7	Latin/Greek	
Part 8	Latin/Hebrew	
Part 9	*Latin-5*	

	Turkish	
Part 10	*Latin-6 Nordic*	
Part 11	Latin/Thai	
Part 12	Latin/Devanagari	The work in making a part of 8859 for Devanagari was officially abandoned in 1997. ISCII and Unicode/ISO/IEC 10646 cover Devanagari.
Part 13	*Latin-7 Baltic Rim*	

		
Part 14	*Latin-8* *Celtic*	
Part 15	Latin-9	
Part 16	*Latin-10* *South-Eastern European*	Intended for Albanian, Croatian, Hungarian, Italian, Polish, Romanian and Slovene, but also Finnish, French, German and Irish Gaelic (new orthography). The focus lies more on letters than symbols. The currency sign is replaced with the euro sign.

Codis de caràcters de Windows

En ISO 8859-1 les posicions 128-159 estan reservades per caràcters de control, és a dir, corresponen a combinacions que no representen caràcters gràfics. Els codis de caràcters de Windows (WinLatin1 o Windows code page 1251) fa servir algunes d'aquestes posicions per caràcters imprimibles. Veiem aquí la taula corresponent (extreta de http://czyborra.com/charsets/codepages.html):

80 €		82 ‚	83 ƒ	84 „	85 …	86 †	87 ‡	88 ˆ	89 ‰	8A Š	8B ‹	8C Œ		8E Ž	
	91 ‘	92 ’	93 “	94 ”	95 •	96 –	97 —	98 ˜	99 ™	9A š	9B ›	9C œ		9E ž	9F Ÿ
A0	A1 ¡	A2 ¢	A3 £	A4 ¤	A5 ¥	A6 ¦	A7 §	A8 ¨	A9 ©	AA ª	AB «	AC ¬	AD -	AE ®	AF ¯
B0 °	B1 ±	B2 ²	B3 ³	B4 ´	B5 µ	B6 ¶	B7 ·	B8 ¸	B9 ¹	BA º	BB »	BC ¼	BD ½	BE ¾	BF ¿
C0 À	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Ç	C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
D0 Ð	D1 Ñ	D2 Ò	D3 Ó	D4 Ô	D5 Õ	D6 Ö	D7 ×	D8 Ø	D9 Ù	DA Ú	DB Û	DC Ü	DD Ý	DE Þ	DF ß
E0 à	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 ç	E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
F0 ð	F1 ñ	F2 ò	F3 ó	F4 ô	F5 õ	F6 ö	F7 ÷	F8 ø	F9 ù	FA ú	FB û	FC ü	FD ý	FE þ	FF ÿ

Compareu les posicions en aquesta taula i en la corresponent de la ISO-8859-1 d'alguns caràcters, per exemple: A, à, ç, ñ, 6. Són les mateixes?

Hi ha codis de caràcters de Windows per a altres alfabets, concretament:

- 1250 (Central Europe)
- 1251 (Cyrillic)
- 1252 (Latin I)
- 1253 (Greek)
- 1254 (Turkish)
- 1255 (Hebrew)
- 1256 (Arabic)
- 1257 (Baltic)
- 1258 (Vietnam)
- 874 (Thai)

La família KOI de codis de caràcters ciríl·lics

Posem com a exemple aquesta família per destacar que des dels inicis de la informàtica existien codis de caràcters per a altres alfabets, com per exemple els caràcters ciríl·lics. KOI significa Код Обмена Информацией ("Kod Obmena Informatsiey" - Codi per a l'Intercanvi d'Informació). El primer KOI, anomenat KOI-7 era un codi de caràctes de 7 bits que tenia només lletres en majúscules:

KOI-7

20	21 !	22 "	23 #	24 ¤	25 %	26 &	27 '	28 (	29)	2A *	2B +	2C ,	2D -	2E .	2F /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7	38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G	48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W	58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 Ю	61 А	62 Б	63 Ц	64 Д	65 Е	66 Ф	67 Г	68 Х	69 И	6A Й	6B К	6C Л	6D М	6E Н	6F О
70 П	71 Я	72 Р	73 С	74 Т	75 У	76 Ж	77 В	78 Ь	79 Ы	7A З	7B Ш	7C Э	7D Щ	7E Ч	

El KOI-8

Aquesta és una versió de 8 bits, i inclou tant caràcters en majúscules com en minúscules. En la següent figura podem veure la porció superior:

C0 Ю	C1 а	C2 б	C3 ц	C4 д	C5 е	C6 ф	C7 г	C8 х	C9 и	CA й	CB к	CC л	CD м	CE н	CF о
D0 п	D1 я	D2 р	D3 с	D4 т	D5 у	D6 ж	D7 в	D8 ь	D9 ы	DA з	DB ш	DC э	DD щ	DE ч	DF ъ
E0 Ю	E1 А	E2 Б	E3 Ц	E4 Д	E5 Е	E6 Ф	E7 Г	E8 Х	E9 И	EA Й	EB К	EC Л	ED М	EE Н	EF О
F0 П	F1 Я	F2 Р	F3 С	F4 Т	F5 У	F6 Ж	F7 В	F8 Ь	F9 Ы	FA З	FB Ш	FC Э	FD Щ	FE Ч	

KOI-8 amb ë (KOI8-R)

			A3 ё												
			B3 Ё												
C0 Ю	C1 а	C2 б	C3 ц	C4 д	C5 е	C6 ф	C7 г	C8 х	C9 и	CA й	CB к	CC л	CD м	CE н	CF о
D0 п	D1 я	D2 р	D3 с	D4 т	D5 у	D6 ж	D7 в	D8 ь	D9 ы	DA з	DB ш	DC э	DD щ	DE ч	DF ъ
E0 Ю	E1 А	E2 Б	E3 Ц	E4 Д	E5 Е	E6 Ф	E7 Г	E8 Х	E9 И	EA Й	EB К	EC Л	ED М	EE Н	EF О
F0 П	F1 Я	F2 Р	F3 С	F4 Т	F5 У	F6 Ж	F7 В	F8 Ь	F9 Ы	FA З	FB Ш	FC Э	FD Щ	FE Ч	FF Ъ

5.5.d. Unicode

Introducció

A l'apartat anterior hem presentat una sèrie de codis de caràcters que fan servir 8 bits. Això dóna la possibilitat de codificar fins a 256 caràcters. Per molts idiomes això és suficient, però no per tots (penseu, per exemple en els caràcters xinesos). Tot i que pugui ser suficient per a molts idiomes, fa que sigui impossible guardar en un únic arxiu de text (compte! de text, fent servir altres formats sí que és possible) documents multilingües per a certes combinacions d'idiomes (per exemple, barrejar en un únic document català i rus). També s'ha de tenir en compte que de tant en tant apareixen nous símbols (pensem per exemple en el € de l'euro) que s'han d'anar incorporant al codi de caràcters.

En l'apartat anterior hem vist uns quants codis de caràcters dels molts existents. Aquesta gran quantitat de codis de caràcters implica la dificultat d'obrir un document correctament (aspecte que tractarem més endavant en aquest mateix capítol) ja que la detecció del codi de caràcters no és totalment automàtica.

Per aquests motius s'intenta adoptar un codi de caràcters universal. Aquest codi de caràcters és l'Unicode. Unicode fa servir més de 8 bits, de manera que pot codificar molts més caràcters.

Originàriament es pensava fer servir simplement una codificació de16 bits que proporciona la possibilitat de codificar més de 65.000 caràcters (2^{16}=65.536). Tot i que aquesta xifra és suficient per codificar la majoria dels milers de caràcters que es fan servir a les diferents llengües del món, l'estàndard Unicode ISO/IEC 10646 permet tres formes de codificació que fan servir un repertori de caràcters comú però que permeten codificar al voltant d'un milió més de caràcters. Aquesta xifra és suficient per cobrir totes les necessitats de codificació conegudes, incloent totes les escriptures històriques del món i altres sistemes de notació.

Codificacions de caràcters amb Unicode

Existeixen diferents maneres de codificar els caràcters amb Unicode. La majoria d'ordinadors fan servir unitats mínimes de 8 bits. Si fem servir més de 8 bits haurem d'organitzar la codificació de manera que fem servir múltiples de 8 bits, és a dir, més d'un byte. L'estàndard Unicode defineix tres tipus de codificacions que permet representar la informació en un byte, dos bytes o quatre bytes. Les tres codificacions codifiquen el mateix repertori de caràcters comú i es pot passar d'una codificació a una altre sense pèrdua de dades:

- **UTF-8**: la codificació en bytes és d'una longitud variable, des d'1 byte per als caràcters coincidents amb l'ASCII.
- **UTF-16**: la codificació també és variable, però o bé en dos bytes o bé en quatre.
- **UTF-32**: tots els caràcters es codifiquen amb quatre bytes.

UTF-8

En el següent esquema podem observar com es codifiquen els caràcters en UTF-8

Bits	Darrera posició (HEX)	Nombre de caràcters	Byte 1	Byte 2	Byte 3	Byte 4
7	U+007F	127	0xxxxxxx			
11	U+07FF	2.048	110xxxxx	10xxxxxx		
16	U+FFFF	65.536	1110xxxx	10xxxxxx	10xxxxxx	
21	U+1FFFFF	2.097.152	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Fixem-nos en primer lloc, que no tots els bits dels diferents bytes es fan servir per contenir informació, sinó que alguns d'ells es fan servir per indicar com s'han de llegir aquests bytes. Així, si el byte comença per 0, indica que només s'ha de llegir un byte (i per tant queden 7 bits per a contenir informació i el nombre de caràcters possibles és de 127). La part més baixa de l'Unicode coincideix plenament amb l'ASCII. D'aquesta manera s'assegura la retrocompatibilitat amb l'ASCII. Això també significa que un arxiu que contingui només caràcters llatins bàsics i xifres i puntuacions (per exemple un text en anglès) i que estigui en codificació ASCII és exactament igual que aquest mateix arxiu en Unicode UTF-8.

Si s'han de llegir dos bytes, el primer comença per 110 (i per tant té disponibles 5 bits per informació) i el segon comença per 10 (i en té disponibles 6). Fixeu-vos que tots els bytes que no són el primer tenen l'estructura 10xxxxxx. Generalitzant, si un byte comença per 1, vol dir que s'haurà de llegir més d'un byte i el nombre de bytes que cal llegir (incloent aquest primer) vindrà donat pel nombre d'1 del primer.

Fixeu-vos també que en UTF-8, si disposem de 4 bytes tenim disponibles 21 bits per informació i no 32 (8x4).

UTF-16

L'UTF-16 té una longitud variable d'1 o 2 paraules de 16 bits, és a dir, de 2 o 4 bytes. Està optimitzat per a representar els caràcters del *pla bàsic multilingüe* (BMP o *Basic Multilingual Plane*), és a dir, els caràcters que es fan servir més freqüentment en totes les llengües (inclosos els caràcters xinesos, japonesos i coreans). Aquests caràcters estan inclosos en el rang U+0000 a U+FFFF (0 a 65535) i quan es limita a representar caràcters d'aquest pla fa servir una longitud fixa de 16 bits.

L'UTF-16 produeix una seqüència d'unitats de 16 bits. Com que aquestes unitats tenen 2 bytes de 8 bits, l'ordre d'aquests bits pot dependre de l'*endianess* (ordre dels bytes) de l'arquitectura de l'ordinador. Fixem-nos en la següent taula (extreta de la Wikipedia http://en.wikipedia.org/wiki/Utf-16) per explicar aquest concepte.

Posició	Glif[1]	Caràcter	Unitats del codi UTF-16 (hex)	Unitats del codi UTF-16BE (hex)	Unitats del codi UTF-16LE (hex)
U+007A	z	LATIN SMALL LETTER Z	007A	00, 7A	7A, 00
U+6C34	□	CJK UNIFIED IDEOGRAPH-6C34 (water)	6C34	6C, 34	34, 6C

La z minúscula, que està en la posició 007A, si es representa com a *Big Endian* queda de la mateixa manera (007A), però si es representa com a *Little Endian* queda com a 7A00. Si el sistema falla en determinar l'*endianess* podrà confondre la z amb el símbol 稀.

Per ajudar a reconèixer l'ordre dels bytes (l'*endianess*) l'UTF-16 permet una Marca d'Ordre de Byte (BOM – *Byte Order Mark*), que un valor de U+FEFF i que precedeix el valor codificat real (U+FEFF és l'espai invisible d'ample zero / caràcter ZWNBSP - *invisible zero-width non-breaking space*). Si l'*endianess* del decodificador coincideix amb la del codificador, el decodificador detecta el valor correcte de 0xFEFF; però si no és així es detecta el valor U+FFFE que està reservat per a aquesta funció. Aquest error permet corregir *l'endianess* per a la resta de valors. Si no hi ha BOM la norma RFC 2781 diu que s'ha de suposar una codificació *big-endian*, però com que Windows fa servir *little-endian* per defecte, moltes aplicacions suposen aquesta opció per defecte. En absència de BOM, per determinar l'*endianess* s'acostuma a buscar el caràcter espai (U+0020) que és molt freqüent en els textos de la majoria de llengües.

UTF-32

L'UTF-32 fa servir 32 bits (4 bytes) per a tots els caràcters. En UTF-32 els caràcters es representen directament segons la seva posició Unicode. L'avantatge principal és que l'algorisme per llegir els fitxers és molt simple i ràpid, ja que simplement ha d'anar llegint 4 bytes en 4 bytes. El principal inconvenient és que és ineficient pel que fa a espai, ja que fa servir el doble que l'UTF-16 i fins a 4 vegades més que l'UTF-8 (tot i que això dependrà del caràcter a representar).

A la següent taula (extreta de **http://en.wikipedia.org/wiki/Comparison_of_Unicode_encodings**) podem observar el nombre de bytes emprat per cada codificació segons els rangs dels codis de caràcters.

1 Un *glif* (en anglès *glyph*) és una representació gràfica d'un caràcter. Un caràcter és una unitat textual i en canvi un glif és una unitat gràfica.

Rang de codis (hexadecimal)	UTF-8	UTF-16	UTF-32
000000 – 00007F	1	2	4
000080 – 00009F	2		
0000A0 – 0003FF			
000400 – 0007FF			
000800 – 003FFF	3		
004000 – 00FFFF			
010000 – 03FFFF	4	4	
040000 – 10FFFF			

5.5.e. Detecció de la codificació de caràcters

Per llegir adequadament un fitxer de text és imprescindible conèixer la codificació de caràcters utilitzada. Passa sovint que no coneixem exactament en quina codificació de caràcters està el fitxer que volem llegir i per aquest motiu hi ha una sèrie d'algorismes que intenten detectar la codificació automàticament. Aquests algorismes funcionen d'una manera heurística, a partir d'anàlisis estadístiques de diversos textos en diferents llengües i codificacions de caràcters. Normalment es treballa amb trigrams de caràcters d'una manera semblant a la que es fa per la detecció automàtica de llengua[2].

Quan el fitxer està en Unicode UTF-8, la detecció automàtica de la codificació de caràcters acostuma a funcionar molt bé. Això s'explica pel gran percentatge de seqüències de bytes invàlides en UTF-8, de manera que un arxiu escrit en una altra codificació molt difícilment es detectarà com a UTF-8. La detecció de la codificació UTF-16 també és força fiable donat l'alt número de caràctes de nova línia (U+000A) i espais (U+0020) que es troben quan es divideixen les dades en paraules de 16 bits (2 bytes). Quan treballem amb codificacions ISO-8859 la detecció de la codificació exacta resulta més complexa, ja que totes les codificacions d'aquesta família comparteixen la part baixa de la taula (que coincideix totalment amb l'ASCII).

Per evitar aquests problemes, molts formats, com per exemple l'HTML permeten especificar en quina codificació de caràcters està escrit el fitxer.

```
<html>
<meta charset="UTF-8">
<body>
...
</body>
</html>
```

A continuació veurem un parell d'eines que ens permetran detectar la codificació de caràcters dels nostres arxius.

2 Mireu la secció *Per ampliar coneixements*

Chardet

Chardet (**https://pypi.python.org/pypi/chardet**) és un paquet per al llenguatge de programació Python que permet detectar les següents codificacions de caràcters:

- ASCII, UTF-8, UTF-16 (2 variants), UTF-32 (4 variants)
- Big5, GB2312, EUC-TW, HZ-GB-2312, ISO-2022-CN (Traditional and Simplified Chinese)
- EUC-JP, SHIFT_JIS, ISO-2022-JP (Japanese)
- EUC-KR, ISO-2022-KR (Korean)
- KOI8-R, MacCyrillic, IBM855, IBM866, ISO-8859-5, windows-1251 (Cyrillic)
- ISO-8859-2, windows-1250 (Hungarian)
- ISO-8859-5, windows-1251 (Bulgarian)
- windows-1252 (English)
- ISO-8859-7, windows-1253 (Greek)
- ISO-8859-8, windows-1255 (Visual and Logical Hebrew)
- TIS-620 (Thai)

Funciona sota línia de comandes:

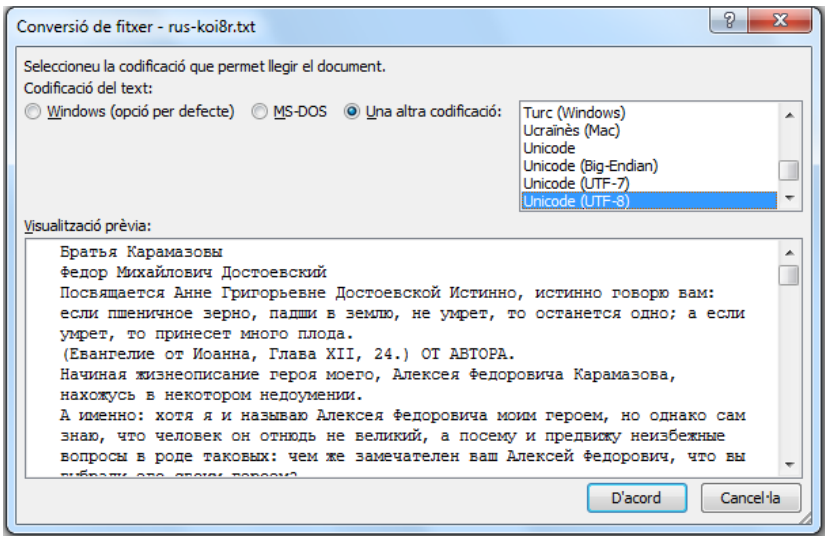
```
chardet doc4.txt
doc4.txt: KOI8-R (confidence: 1.00)
```

També pot detectar la codificació d'un conjunt de fitxers:

```
chardet *
doc10.txt: UTF-16BE (confidence: 1.00)
doc1.txt: ISO-2022-JP (confidence: 0.99)
doc2.txt: UTF-16BE (confidence: 1.00)
doc3.txt: ISO-8859-2 (confidence: 0.87)
doc4.txt: KOI8-R (confidence: 1.00)
doc5.txt: ISO-2022-KR (confidence: 0.99)
doc6.txt: GB2312 (confidence: 0.99)
doc7.txt: UTF-8 (confidence: 1.00)
doc8.txt: UTF-8 (confidence: 1.00)
doc9.txt: ascii (confidence: 1.00)
```


Microsoft Word

Microsoft Word disposa d'un bon algorisme de detecció de codificacions de caràcter. L'únic que hem de fer és anar a *File* > *Open* i seleccionar com a format *Encoded Text*. Llavors obrir l'arxiu de què volem detectar la codificació i apareixerà una pantalla com la següent:



En aquesta pantalla ens indica la codificació de caràcters més probable i ens mostra un fragment de text perquè puguem veure si la detecció és correcta. En la majoria dels casos ho és, però si no, podem seleccionar una altra i veurem com es visualitzaria el text. Un cop seleccionem la codificació correcta podrem obrir el fitxer.

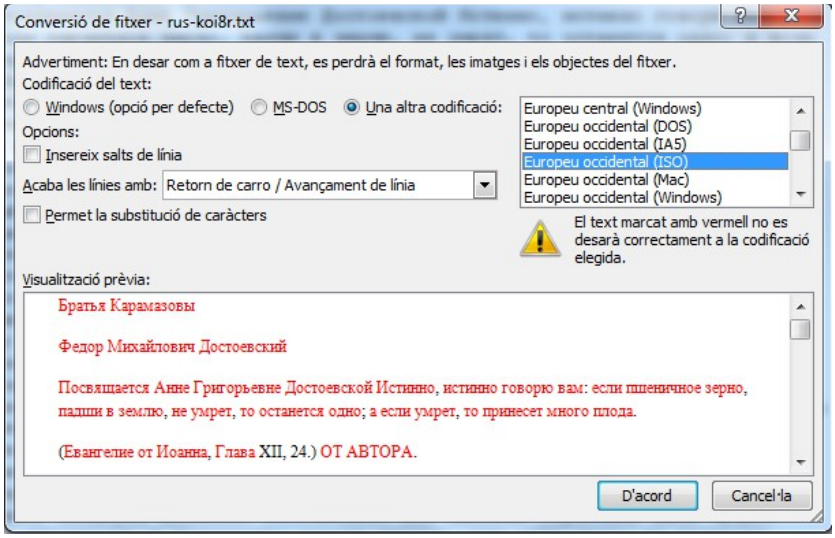
5.5.f. Canvi de la codificació de caràcters

En algunes situacions és possible que necessitem canviar la codificació de caràcters d'un determinat arxiu. Si hem estat capaços de detectar-la (veieu apartat anterior) la manera més senzilla de convertir-la és fer servir un editor de textos amb un bon suport pel que fa a la codificació de caràcters. Llavors, l'únic que hem de fer és obrir l'arxiu seleccionant la codificació adequada i un cop obert guardar-lo fent Save as i indicant la codificació desitjada.

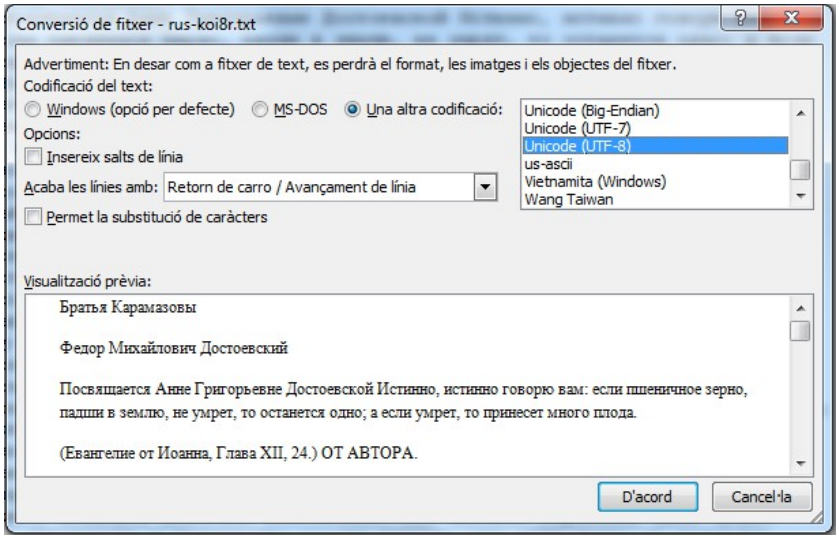
Veurem ara com fer aquesta operació en Microsoft Word i després veurem un programa específic per fer aquestes conversions: l'*iconv*.

Microsoft Word

Un cop hem obert l'arxiu en la codificació adequada (que molt probablement haurà detectat automàticament), l'únic que hem de fer és Save as i seleccionar una altra codificació. L'avantatge de fer servir aquesta aplicació és que és capaç d'avisar-nos si estem seleccionant una codificació que no és capaç de guardar correctament l'arxiu. Seguint l'exemple anterior, un cop hem obert l'arxiu que està en codificació KOI8-R, si ara l'intentem guardar amb codificació ISO-8859-1, en la pantalla ens mostrarà els caràcters que no és capaç de guardar.



Per tant, serà necessari seleccionar una codificació que sigui capaç de guardar l'arxiu, per exemple Unicode UTF-8. En aquest cas fixem-nos que no ens apareix cap caràcter en vermell.



Un cop guardat, l'arxiu tindrà la nova codificació.

iconv

És un programa i un conjunt de llibreries per a diversos llenguatges de programació que permeten canviar un arxiu d'una codificació a una altra. Està disponible per a Linux i Mac i també per Windows sota Cygwin (http://cygwin.com/) i GNUWin32 (http://gnuwin32.sourceforge.net/).

El seu funcionament és molt senzill:

```
iconv -f koi8-r -t utf-8 < doc4.txt > doc4b.txt
```

Ara el document doc4b.txt estarà en codificació utf-8, i ho podem verificar fent:

```
chardet doc4b.txt
doc4b.txt: utf-8 (confidence: 0.99)
```

5.6. La representació de la informació no textual

En la unitat anterior hem après com es codifica la informació bàsicament textual (incloent salts de línia i alguns altres caràcters de control) en un document de text. Els documents amb què haurem de treballar contenen molta més informació, de caràcter no textual, com poden ser qüestions de format (negretes, tipus de lletra, colors, etc.) o bé referències a altres objectes (com poden ser imatges o gràfics). Tota aquesta informació es codifica també de diferents maneres. En aquesta unitat estudiarem les maneres més habituals de codificar aquest tipus d'informació en els documents.

El traductor sovint han de traduir, a més a més, fitxers que no són documents en sentit estricte: planes web, bases de dades, codi de programes o fitxers d'imatges.

Aquest tema no pot organitzar-se com a una enumeració dels formats d'arxiu més habituals, ja que la llista seria interminable. El que procurarem serà exposar algunes idees bàsiques i estratègies per enfrontar-nos a formats d'arxiu desconeguts.

5.6.a. Noms d'arxiu i extensions. Relació amb el format i l'aplicació

Hi acostuma a haver una relació entre el format de l'arxiu i la extensió que té però aquesta relació no és inequívoca, és a dir, una mateixa extensió pot emprar-se per diferents tipus d'arxius.

Recordeu que l'*extensió* d'un arxiu ve donada pels caràcters que venen darrera del punt (.) del seu nom. L'extensió acostuma a tenir dues o tres lletres. Per exemple, a l'arxiu:

```
resum.txt
```

el nom de l'arxiu és "resum" i l'extensió "txt". Al següent arxiu:

```
resum de resultats de vendes.doc
```

el nom és "resum de resultats de vendes" i l'extensió és "doc".

En alguns sistemes operatius per defecte s'amaga l'extensió dels tipus d'arxius coneguts. Això vol dir, per exemple, que en els exemples anteriors, al nostre explorador d'arxius veuríem només "resum" i "resum de resultats de vendes", ja que les extensions txt i doc molt probablement serien conegudes per al nostre sistema.

El fet que el nostre sistema conegui una extensió o no ve donada pels programes que tenim instal·lats. Per exemple, si no tenim instal·lat l'Open Office o el Libre Office al nostre sistema, probablement el sistema no conegui que l'extensió "odt" correspon als documents d'aquest processador de textos. Així, si tenim activada l'opció d'amagar les extensions conegudes al nostre explorador d'arxius, molt probablement aquest mostrarà les extensions dels fitxers "odt". Si instal·lem el Libre Office o l'Open Office, en el moment de la instal·lació s'indicarà al sistema operatiu que aquesta extensió està associada a aquesta aplicació i per tant l'extensió passarà a ser coneguda per al sistema, i s'amagarà si tenim activada l'opció corresponent.

En moltes situacions pot ser recomanable desactivar l'opció d'amagar l'extensió de tipus de fitxers coneguts del nostre sistema operatiu.

El sistema operatiu, doncs, té associades algunes aplicacions a algunes extensions. Per a aquestes extensions, si fem doble clic en un arxiu des de l'explorador d'arxius, el sistema operatiu posarà en marxa l'aplicació associada i obrirà l'arxiu amb aquesta aplicació. Si el sistema operatiu no té associada cap aplicació a una

extensió determinada, si fem doble clic en un fitxer d'aquest tipus el sistema no sabrà què fer i no podrà obrir-lo.

També és possible indicar manualment quin programa està associat a cada extensió.

El fet que un arxiu no s'obri fent doble clic en l'explorador d'arxius no vol dir que no puguem obrir-lo. Una situació habitual és rebre un arxiu de text amb una extensió no estàndard (per exemple un arxiu resum.rsq). Si qui ens envia l'arxiu ens diu que és de text, podrem anar a un editor de textos i obrir-lo des de el menú *File* o similar. Si no ens diuen res de què és l'arxiu, podem fer les següents accions:

- Buscar en alguna base de dades d'extensions si hi apareix. Algunes webs que ofereixen aquestes cerques són http://filext.com/ o http://www.fileinfo.com/
- Suposar que és un fitxer de text i intentar obrir-lo directament amb un editor de textos
- Si treballem amb Linux o Mac podem fer servir la instrucció *file* que ens intentarà determinar el tipus de fitxer

Si fem:

```
file resum.rsq
```

Obtindrem la informació sobre el tipus d'arxiu:

```
resum.rsq: UTF-8 Unicode text
```

En aquest cas podríem obrir aquest fitxer amb algun editor de textos. Si rebéssim sovint arxius d'aquests tipus podríem associar aquesta extensió a l'editor de textos del nostre sistema operatiu, de manera que a partir d'aquest moment al fer doble clic sobre arxius "rsq" s'obririen amb l'editor seleccionat.

5.6.b. El format HTML

L'*Hyper Text Markup Language* (HTML) és un llenguatge de marcatge derivat de l'SGML (*Standard Generalized Markup Language*), dissenyat per per visualitzar textos i relacionar-los en forma d'hipertext.

En HTML es fan servir una sèrie d'etiquetes que defineixen com es visualitzarà el text en un navegador. Les etiquetes més habituals són (font Vikipèdia http://ca.wikipedia.org/wiki/Hyper_Text_Markup_Language):

- **<html>**: És l'etiqueta arrel de qualsevol document HTML.
- **<head>**: Defineix la capçalera del document HTML.
- **<body>**: Defineix el cos del document. Aquesta és la part del document HTML que es mostra en el navegador.

Dintre de la capçalera **<HEAD>** hi podem trobar:

- **<title>**: Permet definir el títol de la pàgina. En navegadors gràfics el contingut del title apareix a la barra del títol a sobre de la finestra.
- **<meta>**: Permet definir metainformacions del document tals com l'autor, la data de realització, la codificació del document (UTF, ISO, etc.), les paraules clau i la descripció del mateix

- **<LINK>**: Permet definir metadades complementàries a les del meta tals com el document anterior, el següent, el capítol al qual pertany el document, la pàgina, glossari, etc.

Dintre del cos **<BODY>** hi podem trobar:

- **<a>**: Etiqueta àncora. Crea un enllaç a un altre document o a una altra zona del mateix, segons els atributs.
- **<h1>**, **<h2>**,… **<h6>**: capçaleres o títols del document, acostumen a distingir-se per mida.
- ** :** Divisió estructural de la pàgina.
- **<p>**: Paràgraf.
- **
**: Salt de línia.
- **<table>**: Indica el començament d'una taula, després s'haurà de definir les files amb **<tr>** i les cel·les dintre de les files amb **<td>**.
- **<ul>**: Llista desordenada (sense numerar). Els ítems es defineixen amb <li>.
- **<ol>**: Llista ordenada (numerada). Els ítems es defineixen amb <li>.
- **<dl>**: Llista de definició. Hi ha dos tipus d'ítem; el dt i el dd.
  - **<dt>**: Terme a definir.
  - **<dd>**: Definició del terme.

Per regla general les etiquetes que s'obren s'han de tancar (tot i que la majoria de navegadors permeten ometre el tancament d'algunes etiquetes). Veiem ara com a exemple un document HTML molt senzill:

```
<html>
<body>
<p>Aix&ograve; &eacute;s un <b>document</b> d'<i>exemple</i>?</p>
</body>
</html>
```

En un navegador d'Intenet aquest html es visualitzaria de la següent manera:

Això és un **document** d'*exemple*?

Fixem-nos que l' “ò” s'ha expressat com ò i le “é” com a é. Aquestes combinacions de

caràcters per expressar caràcters especials no inclosos en l'alfabet llatí bàsic s'anomenen *entitats d'html*

Aquestes entitats també es poden expressar de manera numèrica:

```
<html>
<body>
<p>Aix&#242; &#201;s un <b>document</b> d'<i>exemple</i>?</p>
</body>
</html>
```

A l'Annex I d'aquest capítol presentem una llista força completa de les entitats d'html.

Només és imprescindible fer servir entitats d'html per representar els caràcters reservats, que són: **<**, **>** i **&**. Si volem escriure en html una cosa com:

Si a > b & b < c i ho fem:

```
<html>
<body>
<p>Si a > b & b < c</p>
</body>
</html>
```

El navegador podria confondre's (tot i que la majoria de navegadors moderns són capaços de representar aquest document sense problemes, Estrictament, hauríem de representar els document de la següent manera:

```
<html>
<body>
<p>Si a &gt; b &amp; b &lt; c</p>
</body>
</html>
```

El nostre exemple anterior, el podem escriure sense problemes fent servir els caràcters accentuats normals:

```
<html>
<body>
<p>Això és un <b>document</b> d'<i>exemple</i>?</p>
</body>
</html>
```

En aquest cas, si el html no especifica la codificació de caràcters, depenent de la configuració del nostre navegador podem veure el document de manera incorrecta:

AixÃ² Ã©s un **document** d'*exemple*?

En la configuració del nostre navegador podem indicar una altra codificació de caràcters per fer que el document es visualitzi correctament. Ara bé, per evitar aquests problemes, és útil indicar la codificació de caràcters emprada dins del propi html, de manera que el navegador farà servir aquesta informació per visualitzar el document de manera correcta.

```
<html>
<meta charset="UTF-8">
<body>
<p>Això és un <b>document</b> d'<i>exemple</i>?</p>
</body>
</html>
```

I ara el navegador farà servir aquesta informació i visualitzarà correctament els caràcters accentuats:

Això és un **document** d'*exemple*?

5.6.c. L'XHTML

L'XHTML és com l'HTML però escrit com a un XML (veurem amb detall l'XML una mica més endavant en aquest mateix capítol). L'XHTML es por definir com a una versió d'HTML més estricta i neta. En HTML s'ha permès escriure documents amb etiquetes que s'obren i no es tanquen, problemes d'aniuament d'etiquetes, etc. I els navegadors han estat capaços de visualitzar les planes igualment. És a dir, s'ha primat la robustesa a la correcció en la sintaxi. En XHTML es verifica que el document sigui correcte de manera estricta i si no ho és el navegador impedeix la seva visualització.

Si tornem al nostre exemple anterior, però introduïm un error:

```
<html>
<meta charset="UTF-8">
<body>
<p>Això és un <b>document</b> d'<i>exemple</i>?
</body>
</html>
```

(fixeu-vos que l'etiqueta <p> no es tanca). Si guardem aquest arxiu com a exemple.htm (és a dir, un html normal) i l'intentem visualitzar en un navegador, el visualitzarem sense problemes. En canvi, si el guardem com a exemple.xhtml (és a dir, com a XHTML), a l'intentar obrir-lo en un navegador, ens mostrarà un missatge com el següent:

This page contains the following errors:

error on line 5 at column 8: Opening and ending tag mismatch: p line 0 and body

Below is a rendering of the page up to the first error.

Això és un document d'exemple

5.6.d. Open Document

L'ODF (*Open Document Format for Office Applications – Document Obert per a Aplicacions Informàtiques)* és un format basat en XML per a la representació de documents, fulls de càlcul, gràfics i presentacions. L'estàndard va ser desenvolupat per un comitè tècnic del consorci OASIS (*Organisation for the Advancement of Structured Information Standards*). Es basa en l'especificació XML d'OpenOffice.org de Sun Microsystems.

A més de ser un estàndard OASIS, la versió 1.1 és també un estàndard internacional ISO/IEC (ISO/IEC 26300:2006/Amd 1:2012 — Open Document Format for Office Applications (OpenDocument) v1.1.).

Les extensions associades als Open Document són:

- Text: .odt
- Full de càlcul: .ods
- Presentació: .odp
- Dibuix: .odg
- Gràfic: .odc
- Fórmula matemàtica: .odf
- Base de dades: .odb
- Imatge: .odi
- Document mestre: .odm

I les associades a les plantilles són:

- Text: .ott
- Full de càlcul: .ots
- Presentació: .otp
- Dibuix: .otg

Els arxius Open Document són arxius comprimits ZIP que contenen diversos arxius i directoris:

- Directoris
  - META-INF
  - Thumbnails
  - Pictures
  - Configurations2
- Arxius XML
  - content.xml
  - meta.xml
  - settings.xml
  - styles.xml
- Altres arxius
  - mimetype
  - layout-cache

Per veure per dins el contingut d'aquests arxius hem creat un document que conté:

Això és un **document** d'*exemple*?

I l'hem guardat com a document.odt. Canviem l'extensió a document.zip i el descomprimim.

L'arxiu **content.xml** emmagatzema el contingut real del document, exceptuant les dades binàries com ara imatges. Si obrim aquest fitxer amb un editor de textos obtindrem una cosa similar a la següent (aquí hem simplificat el contingut):

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"  [....]
<style:style style:name="T1" style:family="text"><style:text-properties fo:font-weight="bold"
style:font-weight-asian="bold" style:font-weight-complex="bold"/></style:style><style:style
style:name="T2" style:family="text"><style:text-properties fo:font-style="italic" style:font-style-
asian="italic" style:font-style-complex="italic"/> [...]
</text:sequence-decls><text:h text:style-name="Standard" text:outline-level="10">Això és un
<text:span text:style-name="T1">document</text:span> d&apos;<text:span text:style-
name="T2">exemple</text:span>?</text:h></office:text></office:body></office:document-content>
```

Bona part de la informació sobre els estils de format i disposició del document s'emmagatzema en l'arxiu styles.xml. No tota la informació sobre els estils s'emmagatzema en aquest fitxer, també hi ha informació sobre estils a content.xml. Fixeu-vos com es defineixen els estils T1 i T2 en l'arxiu content-xml anterior. A continuació mostrem alguns fragments de l'arxiu styles.xml:

```
[...]
<style:font-face style:name="Courier 10 Pitch" svg:font-family="&apos;Courier 10 Pitch&apos;"
style:font-pitch="fixed"/>
[...]
<style:style style:name="Text_20_body" style:display-name="Text body" style:family="paragraph"
style:parent-style-name="Standard" style:class="text"><style:paragraph-properties fo:margin-
top="0in" fo:margin-bottom="0.0835in"/></style:style>
[...]
<style:list-level-label-alignment text:label-followed-by="listtab" text:list-tab-stop-
position="0.7in" fo:text-indent="-0.7in" fo:margin-left="0.7in"/>
```

L'arxiu **meta.xml** conté les metadades del document, com ara l'autor, l'usuari que va fer la darrera modificació, la data de creació i de la darrera modificació, així com algunes estadístiques del document com el nombre de taules, imatges, planes, paràgrafs, paraules, etc. A continuació podem observar el contingut d'aquest arxiu:

```
<?xml version="1.0" encoding="UTF-8"?>
[...]
<office:meta>
<meta:initial-creator>Antoni Oliver</meta:initial-creator>
<meta:creation-date>2014-08-14T18:20:54</meta:creation-date>
<dc:date>2014-08-14T18:22:52</dc:date>
<dc:creator>Antoni Oliver</dc:creator>
<meta:editing-duration>P0D</meta:editing-duration>
<meta:editing-cycles>1</meta:editing-cycles>
<meta:document-statistic meta:table-count="0" meta:image-count="0" meta:object-count="0" meta:page-
count="1" meta:paragraph-count="1" meta:word-count="5" meta:character-count="30" meta:non-
whitespace-character-count="26"/>
<meta:generator>LibreOffice/3.5$Linux_x86 LibreOffice_project/350m1$Build-
2</meta:generator></office:meta></office:document-meta>
```

L'arxiu **settings.xml** conté informació que no fa referència ni a contingut ni a disposició, sinó informació que fa referència a aspectes de la visualització del document dins de l'aplicació en el moment en què s'obre el document. Aquesta informació pot fer referència a les àrees de visualització, la posició del cursor o el factor de zoom. A continuació podem observar un fragment d'aquest arxiu:

```
[...]
<config:config-item config:name="ViewAreaTop" config:type="int">0</config:config-item>
<config:config-item config:name="ViewAreaLeft" config:type="int">0</config:config-item>
<config:config-item config:name="ViewAreaWidth" config:type="int">21527</config:config-item>
<config:config-item config:name="ViewAreaHeight" config:type="int">9289</config:config-item>
<config:config-item config:name="ShowRedlineChanges" config:type="boolean">true</config:config-item>
<config:config-item config:name="InBrowseMode" config:type="boolean">false</config:config-item>
[...]
<config:config-item config:name="ZoomFactor" config:type="short">150</config:config-item>
<config:config-item config:name="IsSelectedFrame" config:type="boolean">false
```

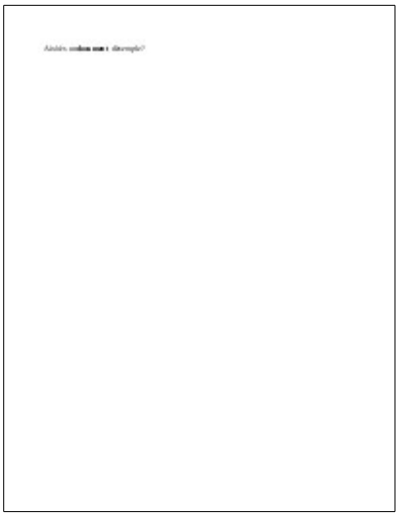


```
[...]
```

L'arxiu **mimetype** té una sola línia i conté informació sobre el tipus d'arxiu. Això fa que de fet l'extensió de l'arxiu sigui irrellevant i només serveixi per a que l'usuari pugui identificar més fàcilment el tipus de document. En el nostre exemple l'arxiu mimetype conté la següent línia:

```
application/vnd.oasis.opendocument.text
```

La carpeta **Thumbnails** conté una imatge en miniatura de la primera plana del document i es genera per defecte quan es guarda l'arxiu. La imatge està en format png i té una mida de 128x128 píxels. En el nostre exemple tindria el següent aspecte:



La carpeta **META-INF** conté un arxiu **manifest.xml** que conté informació sobre els arxius continguts en el fitxer comprimit OpenDocument. A continuació veiem el contingut d'aquest arxiu corresponent al nostre exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0"
manifest:version="1.2">
<manifest:file-entry manifest:full-path="/" manifest:version="1.2" manifest:media-
type="application/vnd.oasis.opendocument.text"/>
<manifest:file-entry manifest:full-path="meta.xml" manifest:media-type="text/xml"/>
<manifest:file-entry manifest:full-path="settings.xml" manifest:media-type="text/xml"/>
<manifest:file-entry manifest:full-path="content.xml" manifest:media-type="text/xml"/>
<manifest:file-entry manifest:full-path="Thumbnails/thumbnail.png" manifest:media-type="image/png"/>
<manifest:file-entry manifest:full-path="manifest.rdf" manifest:media-type="application/rdf+xml"/>
<manifest:file-entry manifest:full-path="Configurations2/accelerator/current.xml" manifest:media-
type=""/>
<manifest:file-entry manifest:full-path="Configurations2/" manifest:media-
type="application/vnd.sun.xml.ui.configuration"/>
<manifest:file-entry manifest:full-path="styles.xml" manifest:media-type="text/xml"/>
</manifest:manifest>
```

La carpeta **Pictures** conté totes les imatges del document. Aquestes imatges s'emmagatzemen en el seu format original, a excepció de les imatges en mapa de bits que es transformen a PNG per qüestió d'espai. Aquesta carpeta només apareix si el document conté imatges. La informació sobre la imatge i com apareix en el document apareixerà en l'arxiu content.xml. Veiem aquí un exemple:

```
<draw:frame draw
:style-name="fr1" draw:name="graphics1" text:anchor-type="paragraph" svg:width=
"6.2575in" svg:height="1.8043in" draw:z-index="0"><draw:image xlink:href="Pictu
```

res/100000000000025F000000AFF9CA7EDD.png" xlink:type="simple" xlink:show="embed
" xlink:actuate="onLoad"/></draw:frame>

5.6.e. Els formats de documents de Microsoft Word

Microsoft Word DOC (.doc)

El format **doc** corresponent al Microsoft Word (versions del 97 al 2003) és un format binari. Tot i que Microsoft va publicar les especificacions d'aquest format, sempre hi ha hagut la queixa de que aquestes especificacions no són completes. Per aquest motiu és difícil obrir documents en aquest format en processadors de textos diferents del Microsoft Word. Tot i que processadors de textos lliures com el LibreOffice i el Open Office puguin obrir aquest format, no es pot garantir una compatibilitat al 100 % i és possible que algunes característiques dels documents es perdin.

Les eines de Traducció Assistida que inclouen la possibilitat d'importar documents Word tipus doc normalment requereixen tenir el Microsoft Word instal·lat a l'ordinador. En realitat l'eina no obre l'arxiu word, sinó que fa obrir el document al Microsoft Word i es comunica amb aquesta aplicació per a obtenir els segments traduïbles. Aquesta comunicació es fa mitjançant una API (*Application Programming Interface*). Un cop feta la traducció amb l'eina de traducció assistida, en el moment de crear el document traduït es torna a establir una comunicació amb el Microsoft Word que reemplaça els segments originals pels traduïts i crea d'aquesta manera un document traduït amb el mateix format que l'original.

Microsoft Word 2003 XML (.xml)

Microsoft va introduir un format XML en el Office. A diferència de l'Open Document aquests documents estan format per un únic fitxer XML. Si continuem amb el mateix document d'exemple, en aquest format tindria el següent aspecte (mostrem únicament un fragment del document):

```
<w:pStyle w:val="Standard"/></w:pPr><w:r><w:t>Això és un </w:t></w:r><w:r><w:rPr><w:rStyle
w:val="T1"/></w:rPr><w:t>document</w:t></w:r><w:r><w:t> d'</w:t></w:r><w:r><w:rPr><w:rStyle
w:val="T2"/></w:rPr><w:t>exemple</w:t></w:r><w:r><w:t>?</w:t></w:r></w:p>....
```

Si el document conté imatges o altres objectes, aquests queden també representats en el document XML, per exemple:

```
<w:pict><w:binData
w:name="wordml://graphics1">iVBORw0KGgoAAAANSUhEUgAAAlEAAAC5CAIAAACHlv1PAAAAA3NCSVQICAjb4U/gAAAgAElE
QVR4nO3dd1gU1/o48Bd26ZaNrjZnrjUaNdEYrz0mQSVCNBbUWLAXIBIQGxE1iiJaTYEBekgvbP4[...
]u+ADxAAAAAElFTkSuQmCC</w:binData>
```

Microsoft Word 2007/2010 XML (.docx)

Posteriorment, a partir de la versió 2007, Microsoft va introduir un format basat en XML consistent en diversos arxius en un arxiu zip, anomenat Office Open XML (i també conegut de manera informal com a OOXML o OpenXML). El format va assolir la qualificació d'estàndard primer per l'Ecma (as ECMA-376) i en versions posteriors per ISO i IEC (com a ISO/IEC 29500).

A partir de la versió d'Office 2007 aquest format és el format per defecte en aquesta aplicació ofimàtica.

L'estructura dels arxius docx és la següent:

- Arxiu [Content_Types].xml

- Carpeta docProps
- Carpeta _rels
- Carpeta word

No entrarem en detalls, només direm que el contingut del document està en un arxiu document.xml que està dins de la carpeta word. Mostrem el corresponent a l'arxiu d'exemple:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships" xmlns:v="urn:schemas-
microsoft-com:vml" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"><w:body><w:p><w:pP
r><w:pStyle w:val="style0"/><w:numPr><w:ilvl w:val="8"/><w:numId w:val="1"/></w:numPr><w:spacing
w:line="100" w:lineRule="atLeast"/></w:pPr><w:r><w:rPr><w:lang w:val="ca-ES"/></w:rPr><w:t
xml:space="preserve">Això és un </w:t></w:r><w:r><w:rPr><w:b/><w:bCs/><w:lang w:val="ca-
ES"/></w:rPr><w:t>document</w:t></w:r><w:r><w:rPr><w:lang w:val="ca-ES"/></w:rPr><w:t
xml:space="preserve"> d&apos;</w:t></w:r><w:r><w:rPr><w:i/><w:iCs/><w:lang w:val="ca-
ES"/></w:rPr><w:t>exemple</w:t></w:r><w:r><w:rPr><w:lang w:val="ca-
ES"/></w:rPr><w:t>?</w:t></w:r></w:p><w:sectPr><w:type w:val="nextPage"/><w:pgSz w:h="15840"
w:w="12240"/><w:pgMar w:bottom="1134" w:footer="0" w:gutter="0" w:header="0" w:left="1134"
w:right="1134" w:top="1134"/><w:pgNumType w:fmt="decimal"/><w:formProt
w:val="false"/><w:textDirection w:val="lrTb"/></w:sectPr></w:body></w:document>
```

Si el document conté imatges, aquestes s'emmagatzemen en una subcarpeta de la carpeta word, anomenada media.

5.6.f. El format LaTeX

LaTeX és un llenguatge de marcatge de documents i un sistema d'edició de documents que es fa servir molt per a la publicació de documents i llibres científics. Els documents LaTeX es poden escriure en qualsevol editor de textos, tot i que hi ha alguns específics que contenen algunes funcions d'ajuda (com per exemple Kile (http://kile.sourceforge.net/) o TexMaker (http://www.xm1math.net/texmaker/)). Els fitxers LaTeX acostument a tenir la extensió tex. A més de l'editor es necessita tenir instal·lats els programes i macros que permeten la transformació dels arxius tex en altres formats com el dvi, ps o pdf. Aquestes macros es poden instal·lar fàcilment en sistemes operatius com Linux i Mac. Per Windows es pot fer servir l'entorn MiKTex (http://www.miktex.org/) i que es pot configurar per funcionar de manera integrada a l'editor TEXnicCenter (http://www.texniccenter.org/).

A continuació veiem un exemple de document LaTeX mínim:

```
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\begin{document}
Això és un \bf{document} d'\emph{exemple}.
\end{document}
```

Per processar aquest document farem servir les següents instruccions (si el document es diu document.tex):

```
latex document.tex
```

I es mostrarà per pantalla:

```
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009/Debian)
entering extended mode
(./latex.tex
LaTeX2e <2009/09/24>
Babel <v3.81> and hyphenation patterns for english, usenglishmax, dumylang, noh
```



```
yphenation, loaded.
(/usr/share/texmf-texlive/tex/latex/base/article.cls
Document Class: article 2007/10/19 v1.4h Standard LaTeX document class
(/usr/share/texmf-texlive/tex/latex/base/size12.clo))
(/usr/share/texmf-texlive/tex/latex/base/inputenc.sty
(/usr/share/texmf-texlive/tex/latex/base/utf8.def
(/usr/share/texmf-texlive/tex/latex/base/t1enc.dfu)
(/usr/share/texmf-texlive/tex/latex/base/ot1enc.dfu)
(/usr/share/texmf-texlive/tex/latex/base/omsenc.dfu))) (./latex.aux) [1]
(./latex.aux) )
Output written on document.dvi (1 page, 364 bytes).
Transcript written on document.log.
```

Aquí s'haurà creat un arxiu xdvi que tindrà el següent aspecte:

Això és un **document** d'*exemple.*

Si ara volem convertir aquest arxiu xdvi en pdf farem servir les instruccions:

```
dvips document.dvi
ps2pdf document.ps
```

Alternativament,es pot convertir un document LaTeX en pdf amb una única instrucció:

```
pdflatex document.tex
```

Molts entorns gràfics permeten fer aquestes operacions amb un sol clic en un botó.

El LaTeX es fa servir molt en el món acadèmic perquè permet escriure fàcilment fórmules matemàtiques i perquè es pot gestionar la bibliografía de manera molt fàcil i eficient amb l'entorn Bibtex

5.6.g. El format DocBook

DocBook és un llenguatge de marcatge semàntic basat en XML que serveix per representar qualsevol tipus de document, com ara llibres, manuals i articles acadèmics. De la mateixa manera que passava amb LaTeX, els usuaris poden crear el contingut de la publicació sense preocupar-se de la seva presentació ni del format final. Un cop acabat el document en format DocBook es poden crear fàcilment els documents finals en diversos formats (entre ells PDF, EPUB, HTML, XHTML, etc.) sense haver de fer cap canvi al document.

Les etiquetes dels documents DocBook es poden dividir en tres grans categories: estructurals, de bloc i de línia.

Algunes de les etiquetes estructurals són:

`set`: És un conjunt d'un o més `book`. L'avantatge de fer servir `set` és que es poden fer servir els enllaços entre tots els llibres

`book`: s'estructuren de la següent manera:

```
book
      meta information
      chapter
```

```
    sect1
    sect1
  chapter
    sect1
  appendix
    sect1
  appendix
    sect1
   ...
  glossary
```

article: S'estructura de la següent manera:

```
article
     meta information
     sect1
     sect1
       sect2
     sect1
     ...
```

Les etiquetes de bloc són elements com ara paràgrafs, llistes, barres laterals, tables i cites. Les etiquetes de línia representen elements com ara èmfasis, hyper-enllaços, etc. que fan que s'apliqui algun tipus de distinció tipogràfica al text, com canvis de la mida de la font, font en cursiva o negreta, etc. A continuació podem observar un fragment d'un llibre en format DocBook:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book>
<title>THE ADVENTURES OF SHERLOCK HOLMES</title>
<bookinfo>
<author>
<firstname>Arthur Conan</firstname>
<surname>Doyle</surname>
</author>
</bookinfo>
<chapter>
<title>ADVENTURE I. A SCANDAL IN BOHEMIA</title>
<section>
<title>I.</title>
<para>To Sherlock Holmes she is always <emphasis>the</emphasis> woman. I have seldom heard him
mention her under any other name. In his eyes she eclipses and predominates the whole of her
sex. ... And yet there was but one woman to him, and that woman was the late Irene Adler, of dubious
and questionable memory.</para>
...
</section>
...
</chapter>
...
</book>
```

Per editar documents DocBook, com que són documents XML, només necessitem un editor de textos. Podem fer servir editors de textos d'XML que ens facilitaran la feina d'edició.

Donat que DocBook és un format XML, podem fer servir eines estàndard per validar i processar documents DocBook i transformar-los en altres formats. Es pot trobar informació detallada sobre com processar documents DocBook a Stayton (2007).

5.6.h. El format PDF

PDF (acrònim en anglès de Portable Document Format, Format de Document Portable) és un format desenvolupat per l'empresa Adobe amb la idea que el document es pugui visualitzar exactament igual amb independència del programari, maquinari o sistema operatiu utilitzat

Característiques
Les principals característiques del format PDF són les següents:

- És multiplataforma, es pot visualitzar en els principals sistemes operatius com GNU/Linux, Windows o Mac, respectant l'aspecte original.
- Pot guardar una combinació de text, gràfics, imatges i fins i tot àudio.
- És un dels formats més estesos a Internet i és emprat tant per governs com per empreses.
- Té l'especificació oberta, permet fins i tot distribuir eines per a crear, visualitzar o modificar documents en format PDF com programari lliure.
- Pot xifrar-se per protegir el seu contingut i fins i tot signar-se electrònicament.

Visualització
Per poder visualitzar arxius PDF cal disposar d'un programari específic, però que són gratuïts i alguns fins i tot de programari lliure. En podem destacar els següents:

- Adobe Acrobat reader (http://get.adobe.com/reader): disponible per a Windows, Linux i Mac, entre d'altres. És gratuït, tot i que no de programari lliure.
- Okular (http://okular.kde.org/): és de programari lliure i funciona sota Linux, tot i que també és possible instal·lar-lo en Windows.
- XPDF (http://www.foolabs.com/xpdf/): és de programari lliure i funciona sota Linux, Windows i Mac.

Creació
Hi ha moltes opcions per crear arxius PDF:

- En Linux i Mac s'inclouen utilitats per imprimir arxius en format PDF. En Windows es poden afegir impressores virtuals, com per exemple PDF Creator (http://sourceforge.net/projects/pdfcreator/).
- LibreOffice i OpenOffice, així com les darreres versions de Microsoft Office, permeten crear arxius PDF
- Google Docs (https://docs.google.com) i Google Drive (https://drive.google.com/) permeten també crear i carregar arxius i després guardar-los en PDF

Edició

Per poder editar i modificar arxius PDF cal disposar d'un editor. Adobe ofereix solucions propietàries i de pagament (ja sigui de compra o de subscripció mensual). Hi ha també alguns programes lliures que permeten editar PDF, entre els que podem destacar:

- PDFedit (http://pdfedit.cz/en/index.html): és una completa llibreria per a la manipulació de documents PDF. Disposa d'una interfície gràfica per facilitar el seu ús.
- pdftk (https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/): Disposa de versions gratuïta i de pagament. La versió gratuïta permet fer diverses operacions, com ara partir o ajuntar documents pdf, fer rotacions, etc. Funciona sota Windows i Linux.
- Nitro PDF (http://www.nitropdf.com/): és un programa propietari i que és de pagament en la seva versió PRO. Ofereix un Reader gratuït però que té més funcionalitats que un simple lector: creació de PDF's,

conversió de PDF a text, captures de seccions del PDF, extreure les imatges del PDF, etc.

Transformar PDFs a formats editables

Sovint és necessari transformar un arxiu PDF en un format editable, com ara text, odf o word. Com a traductors és possible que reben arxius PDFs per traduir i que els vulguem tractar amb la nostra eina de traducció assistida. Per fer això serà necessari convertir el PDF en un format editable. Cal tenir en compte, però, que aquesta conversió no serà sempre perfecta i que en molts casos perdrem tot el format del document. Cal recordar que els arxius PDF es generen sempre a partir d'arxius editables. En cas de rebre per traduir un PDF, caldria demanar al client si disposa del document editable original, ja que segurament serà més fàcil de tractar. En cas que no sigui possible disposar del document en el format original podrem transformar-lo a text d'alguna de les següents maneres:

- Amb algun programa d'edició de PDFs, com els exposats una mica més amunt.
- En alguns casos, serà possible fer servir un programa de visualització i seleccionar el text, copiar-lo al porta-retalls i enganxar-lo en un document. Alguns documents PDF, com explicarem una mica més endavant, no permeten la selecció de text.
- En Linux disposem de programes que funcionen sota terminal que permeten fer la conversió, com ara pdftotext o pdf2txt. Per Windows disposem d'algunes aplicacions com pdf2textpilot (http://sourceforge.net/projects/pdf2textpilot/). XPDF (http://www.foolabs.com/xpdf/), disponible per Linux, Mac i Windows, disposa, entre altres funcionalitats d'una implementació de pdftotext.
- Cal tenir en compte que els fitxers PDF creats a partir de l'escannejat de documents en paper que contenen text no tenen la mateixa estructura que el mateix PDF que s'hagués creat directament des de l'aplicació corresponent. El document PDF provinent de l'escannejat internament conté una imatge del document, sense cap informació respecte al text. Per poder passar a text aquest document caldrà fer servir tècniques d'OCR (*Optical Character Recognition*). Alguns editors de PDF ja implementen aquestes tècniques per poder tractar aquest tipus de document. Sinó, podrem fer servir algun programa específic d'OCR (com Tesseract OCR – http://code.google.com/p/tesseract-ocr/, que té llicència lliure). Algunes versions de Microsoft Windows disposen de la funcionalitat d'OCR, que es troba en el menú d'accessoris. També és possible fer servir algun servei gratuït d'OCR on-line, com per exemple http://www.free-online-ocr.com/ o http://www.onlineocr.net/.
- Google Drive (https://drive.google.com/) permet pujar arxius PDF. En el moment de pujar-los permet seleccionar si desitgem convertir el document PDF en un document de Drive (que és editable). Si seleccionem aquesta opció es durà a terme la conversió, i si el PDF prové d'un document escannejat, es durà a terme un OCR.

Fixeu-vos que si feu servir un OCR, ja sigui un programa, com un servei on-line, com el propi Google Drive, cal indicar la llengua del document. Aquesta informació és important, ja que la precisió de l'OCR augmentarà, ja que en cas de dubtes, triarà una paraula de la llengua.

5.7. XML

5.7.a. Introducció

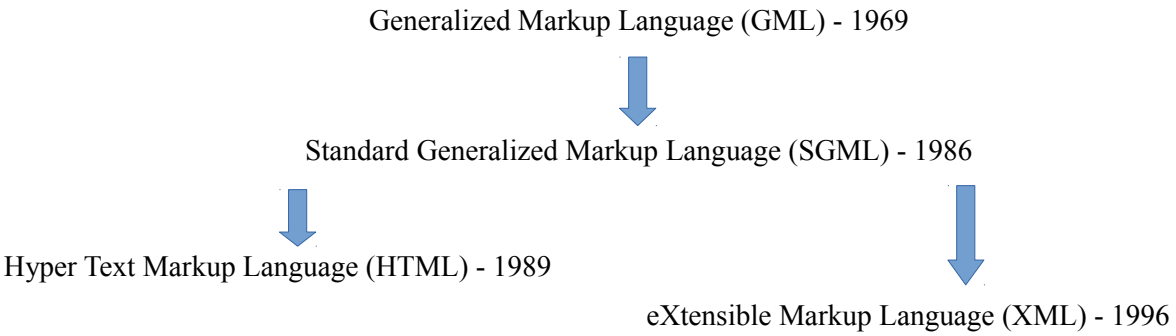
L'*eXtensible Markup Language* (XML) és un llenguatge de marcatge que defineix una sèrie de regles per representar informació estructurada d'una manera que és fàcilment llegible tant per als humans com per a les màquines. L'XML no és, per tant, un llenguatge en particular, sinó una manera de definir llenguatges per a diferents necessitats: representar documents, bases de dades, en serveis web com a contenidor de la informació, etc. XML proporciona una manera senzill per representar i transmetre informació i hi ha tota una

sèrie de tecnologies associades que permeten un tractament senzill i eficient d'aquests tipus de documents.

En aquesta secció farem una introducció general, centrant-nos en aquells aspectes que poden ser més interessants per a un traductor. Qui vulgui una introducció més completa i general pot consultar el tutorial d'XML de W3Schools a http://www.w3schools.com/xml/

El primer que hem de tenir en compte és que l'XML és un format de text, i que per tant, el arxius XML es podran obrir per visualitzar i modificar en qualsevol editor de textos. Si visualitzem un fitxer XML el primer que segurament ens vindrà al cap és que s'assembla molt a l'HTML. Això té una explicació, l'HTML és un llenguatge que es va derivar de l'SGML (*Standard Generalized Markup Language*) i de fet l'XML és una simplificació de l'SGML, i per tant, es pot considerar també un llenguatge derivat. Tot i això cal recordar dues coses: que l'XML no és un substitut de l'HTML i que tots dos llenguatges es van dissenyar amb dos objectius ben diferents: l'XML està dissenyat per descriure dades i l'HTML està dissenyat per a visualitzar dades.

Podem establir doncs una genealogia i una temporització de tots aquests llenguatges de marcatge:



Així doncs, l'XML és una versió abreujada i simplificada de l'SGML optimitzada per Internet. Si mesurem la complexitat de les especificacions pel seu nombre de planes, les de l'SGML té unes 500 planes mentre que les de l'XML n'ocupen només 80. L'XML ofereix el 80% dels avantatges de l'SGML amb només el 20 % de las seva complexitat.

5.7.b. Exemples senzills de documents XML

A continuació podem observar un exemple de document XML. Com ja hem comentat, una de les característiques d'aquest llenguatge és que és molt clar de llegir. Segurament no us costarà gens deduir quin tipus d'informació representa aquest fitxer:

```
<?xml version="1.0" standalone="yes"?>
<diccionari>
    <entrada id='1'>
        <cat>casa</cat>
        <eng>house</eng>
    </entrada>
    <entrada id='2'>
        <cat>cotxe</cat>
        <spa>coche</spa>
        <eng>car</eng>
    </entrada>
</diccionari>
```

Tampoc us costarà entendre quin tipus d'informació conté el següent exemple:

```
<?xml version="1.0" standalone="yes"?>
<agenda-telefons>
    <contacte id='1'>
        <nom>Maria Gil</nom>
        <telefon>456783909</telefon>
    </contacte>
    <contacte id='2'>
        <nom>Ernesto Villalba</nom>
        <telefon>768436543</telefon>
    </contacte>
</agenda-telefons>
```

Els dos exemples que hem presentat són arxius XML ben formats, però no es basen en cap estàndard per a representar bases de dades terminològiques o bé agendes de telèfon. Com veurem més endavant, de l'XML es deriven una sèrie d'estàndards que defineixen com han de ser els documents XML per a una determinada aplicació.

5.7.c. Estructura dels documents XML

Ja hem comentat més a dalt que un XML és un format de text. El text pot tenir dues funcions diferenciades: o marcar o ser una dada. Les marques de l'XML serveixen per estructurar d'una manera lògica el document XML. Observa el següent exemple:

```
<?xml version="1.0" standalone="yes"?>
<llibre id='143'>
    <titol>Don Quijote de la Mancha</titol>
    <autor>Miguel de Cervantes</autor>
</llibre>
```

En aquest document les marques són: **`xml`** – **`version`** – **`standalone`** – **`llibre`** – **`titol`** –**`autor`**
i les dades són: 1.0 – **`yes`** – **`Don Quijote de la Mancha`** – **`Miguel de Cervantes`**

Elements o tags

Tot el que està entre els símbols **<** i **>** es considera *element* (o *tag*) (excepte si està dins d'una secció **CDATA**). Els noms dels elements han de complir les següents normes:

- No pot començar per xifres o caràcters de puntuació
- Pot contenir lletres, xifres i caràcters de puntuació (tot i que és recomanable no fer servir el guió (-), ni els punts (.) ni els dos punts (:))
- No pot començar amb xml (o XML, o Xml, etc)
- Els noms no poden contenir espais

Cal recordar que els noms són sensibles a les majúscules i minúscules. Per tant **<Nom>** és diferent a **<nom>**.

Un element pot estar buit, és a dir, que no contingui cap dada. Llavors es pot tancar amb **/>**. Per exemple: **<entrada></entrada>** és equivalent a **<entrada/>**.

Atributs

En l'exemple anterior teníem **<llibre id='143'>**. **id** és un atribut i el valor d'aquest atribut és 143. Tot el que hem dit per als noms dels elements també és vàlid pels atributs. Els valors dels atributs van entre cometes, tant poden ser cometes simples com dobles.

Comentaris

Els comentaris en XML s'escriuen igual que en HTML.

```
<!-- Això és un comentari en XML -->
```

Entitats

Són marques que es reemplacen per caràcters quan s'analitza el document. En XML només hi ha 5 entitats predefinides:

&amp;	&
&lt;	<
&gt;	>
&apos;	" (cometes dobles)
&quot;	' (cometes simples)

Blocs CDATA

Fins ara hem dit que tot el que està entre els símbols **<>** són marques (o valors d'atribut) i tot el que està fora són dades. Hi ha, però, una excepció les seccions o blocs CDATA. El que està dins d'una secció CDATA no és interpretat pel parser d'XML. Els únics caràcters no permesos dins d'una secció CDATA són: **]]>** (ja que és la marca de tancament de les CDATA). La utilitat d'aquestes seccions és fer el document més llegible, veiem-ho amb un exemple:

Exemple: si haguéssim de posar en un document XML:

```
x < c2 > y
```

ho hauríem de fer amb les entitats:

```
x &lt; c2 &gt; y
```

el que resulta una mica difícil de llegir. Podem facilitar la lectura escrivint:

```
<!CDATA[x < c2 > y]]>
```

5.7.d. Els documents XML ben formats

Hi ha 6 regles que cal respectar per assegurar-nos de que un document XML està ben format.

- Tot element que contingui dades ha de tenir un tag per obrir-lo i un tag per tancar-lo: **<autor>Miguel de Cervantes</autor>**
- Tot element que no contingui dades ha de tenir un tag únic terminat amb **/>**: **
**
- Hi ha d'haver un únic element que contingui a tots els altres (arrel). A l'exemple del diccionari aquest element era **<llibre>**
- Els elements han d'estar aniuats, no superposats:

```
<B>Aquest XML <I>no està</B> ben format.</I>
<B>Aquest XML <I>si està</I> ben format.</B>
```

- Els valors dels atributs van entre cometes (simples o dobles)
- Les úniques referències a entitats permeses són: **& < > '** i **"**

Definició dels tipus de documents

L'XML és un llenguatge que permet intercanviar informació. En el moment de l'intercanvi sorgeix la necessitat de validar els documents, és a dir, verificar que els documents estiguin ben formats i que siguin vàlids. La diferència entre ben format i vàlid és la següent:

- Un document XML ben format és aquell que té una sintaxi correcta, és a dir, que compleix les 6 regles bàsiques que hem exposat més a dalt.
- Un document està ben format si té l'estructura que esperàvem, és a dir si és el tipus de document esperat.

Existeixen dos mecanismes per validar els documents XML:

- Els DTD (*Document Type Definition*)
- XML Schema

Els DTD i els XML Schema especifiquen les regles que defineixen l'estructura d'un document XML. No entrarem en detalls, simplement mostrarem un exemple de cada, corresponent al nostre XML d'exemple:

DTD (Document Type Definition)

El DTD pot estar inclòs dins del propi XML:

```
<?xml version=”1.0”?>
<DOCTYPE note  [
      <!ELEMENT llibre (titol, autor)>
      <!ELEMENT titol (#PCDATA)>
      <!ELEMENT autor (#PCDATA)>
            <!ATTLIST llibre
            src    CDATA          #REQUIRED
      >
]>
<llibre id='143'>
<titol>Don Quijote de la Mancha</titol>
<autor>Miguel de Cervantes</autor>
</llibre>
```

El DTD també pot estar en un arxiu extern i posant una referència a l'arxiu XML:

L’arxiu XML:

```
<?xml version=”1.0”?>
<!DOCTYPE note SYSTEM “llibre.dtd”>
<llibre id='143'>
<titol>Don Quijote de la Mancha</titol>
<autor>Miguel de Cervantes</autor>
</llibre>
```

I el DTD “llibre.dtd”

```
<?xml version=”1.0”?>
<DOCTYPE note  [
      <!ELEMENT llibre (titol, autor)>
      <!ELEMENT titol (#PCDATA)>
      <!ELEMENT autor (#PCDATA)>
      <!ATTLIST llibre
            src    CDATA          #REQUIRED
      >
]>
```

XML Schema

El XML Schema és a la vegada un XML, que si us fixeu en el següent exemple, té un XML Schema que serveix per validar-lo (està definit a la segona línia).

```
<?xml version="1.0" encoding="utf-16"?>
<xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="llibre" type="llibreType" />
  <xsd:complexType name="llibreType">
    <xsd:sequence>
      <xsd:element name="titol" type="xsd:string" />
      <xsd:element name="autor" type="xsd:string" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:int" />
  </xsd:complexType>
</xsd:schema>
```

El XML Schema es referencia dins del propi XML

```
<?xml version=”1.0”?>
<llibre id='143' xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="llibre.xsd">
<titol>Don Quijote de la Mancha</titol>
<autor>Miguel de Cervantes</autor>
</llibre>
```

Com veiem, per definir el tipus de document es pot fer servir tant un DTD com un XML Schema. En general, però, es considera que l'XML Schema és molt més potent i és l'opció recomanada.

5.7.e. Tecnologies associades: XSLT i XPATH

En aquesta secció parlarem de dues importants tecnologies associades a l'XML: les XSLT o Transformacions XSL (*Extensible Stylesheet Language Transformations*) i XPath (*XML Path Language*).

XSLT

És un sofisticat llenguatge que permet transformar un XML en un altre XML diferent, seleccionant quina informació de l'XML original ha d'aparèixer i de quina manera en l'XML transformat. Una aplicació molt habitual de l'XSLT és transformar un XML en un XHTML per poder-lo transmetre a través d'Internet i visualitzar-lo en qualsevol navegador. Veiem-ho en un exemple:

Disposem del fitxer to.xml, que conté un glossari terminològic en el format de Terminologia Oberta del TermCat:

```
<?xml version="1.0" encoding="UTF-8"?>
<terminologiaoberta>
<autor>TERMCAT, Centre de Terminologia</autor>
<titol>TO Termes normalitzats (2007)</titol>
<fitxes>
<fitxa num="1">
<areatematica>Protecció civil: Policia</areatematica>
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="loc adj">a boca de canó</denominacio>
<denominacio llengua="es" tipus="equivalent" jerarquia="terme pral." categoria="">a bocajarro </denominacio>
</fitxa>
<fitxa num="2">
<areatematica>Protecció civil: Policia</areatematica>
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="loc adj">a curta distància</denominacio>
<denominacio llengua="es" tipus="equivalent" jerarquia="terme pral." categoria="">a corta distancia</denominacio>
```

```
<denominacio llengua="en" tipus="equivalent" jerarquia="terme pral." categoria=""> near </denominacio>
</fitxa>
....
<fitxa num="5880">
<areatematica>Esports: Esquí artístic i acrobàtic</areatematica>
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="m"> zúdnic </denominacio>
<denominacio llengua="es" tipus="equivalent" jerarquia="terme pral." categoria=""> zudnik </denominacio>
<denominacio llengua="fr" tipus="equivalent" jerarquia="terme pral." categoria=""> zudnik </denominacio>
<denominacio llengua="en" tipus="equivalent" jerarquia="terme pral." categoria=""> zudnik </denominacio>
</fitxa>
</fitxes>
</terminologiaoberta>
```

El següent fitxer XSLT (stylesheetTO.xsl) transformarà aquest XML en un HTML que mostrarà una taula amb els termes en les diferents llengües:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Terminologia</h2>
    <table border="1">
    <tr bgcolor="#9acd32">
      <th align="left">ca</th>
      <th align="left">en</th>
      <th align="left">es</th>
      <th align="right">Area temàtica</th>
    </tr>
    <xsl:for-each select="terminologiaoberta/fitxes/fitxa">

    <tr>
      <td><xsl:value-of select="denominacio [@llengua='ca']"/></td>
      <td><xsl:value-of select="denominacio [@llengua='en']"/></td>
      <td><xsl:value-of select="denominacio [@llengua='es']"/></td>
      <td><xsl:value-of select="areatematica"/></td>
    </tr>
    </xsl:for-each>
    </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Per aplicar aquest full d'estil XSLT al fitxer XML podem fer servir diverses eines, com per exemple xsltproc disponible en Linux sota terminal:

```
xsltproc stylesheetTO.xsl to.xml > sortida.html
```

El fitxer sortida.xhtml tindrà el següent aspecte:

```
<html>
<body>
<h2>Terminologia</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th align="left">ca</th>
<th align="left">en</th>
<th align="left">es</th>
<th align="right">Area temàtica</th>
</tr>
<tr>
<td>a boca de canó</td>
<td></td>
```

```
<td>a bocajarro</td>
<td>Protecció civil: Policia</td>
</tr>
<tr>
<td>a curta distància</td>
<td>near</td>
<td>a corta distancia</td>
<td>Protecció civil: Policia</td>
</tr>
<tr>
<td>a frec de roba</td>
<td></td>
<td>a quemarropa</td>
<td>Protecció civil: Policia</td>
</tr>
...
<tr>
<td>zúdnic</td>
<td>zudnik</td>
<td>zudnik</td>
<td>Esports: Esquí artístic i acrobàtic</td>
</tr>
</table>
</body>
</html>
```

Si visualitzem aquest document en un navegador d'Internet veurem el següent:

Terminologia

ca	en	es	Area temàtica
a boca de canó		a bocajarro	Protecció civil: Policia
a curta distància	near	a corta distancia	Protecció civil: Policia
a frec de roba		a quemarropa	Protecció civil: Policia
a l'unison	in unison		Esports: Patinatge artístic sobre gel
a llarga distància	distant	a larga distancia	Protecció civil: Policia
a peu de fàbrica	ex works	en fábrica	Comerç internacional
a portell	quincunx	a tresbolillo	Construcció
abadejo	pollack	abadejo	Peixos
abandonament	drop out	abandono	Farmacologia
abatre un obstacle	knock down an obstacle, to	derribar un obstáculo	Esports: Atletisme
abonament	ski-pass	abono	Esports d'hivern
aborrallonament	pilling	pildeo	Indústria tèxtil: Teixits
abrusament	kindling	activación propagada	Psiquiatria
absortància	absorptance	absortancia	Química física
accident en el trajecte	commuting accident	accidente en el trayecto	Assegurances
accidentogen -ògena	accident prone	accidentógeno	Transport per carretera
acció de cessació		acción de cesación	Dret civil
acció de primera	blue chip	acción de primera clase	Borsa
acció del mos	action of the loin	acción del bocado	Esports: Hípica
acció negatòria		acción negatoria	Dret civil
acer maràging	maraging steel	acero maraging	Indústria metal·lúrgica

Per aprendre més sobre XSLT pots consultar el tutorial de W3Schools a: http://www.w3schools.com/xsl/

Xpath

Xpath és un llenguatge que permet seleccionar nodes i conjunt de nodes d'un document XML. És un llenguatge senzill, que recorda a les instruccions de línia de comandes de molts sistemes operatius, però que és molt potent i permet fer consultes a un document XML com si es tractés d'una mena de base de dades.

Seguint amb l'exemple de l'XML de la base de dades terminològica del TermCat, podem veure les següents expressions i els seus resultats. Per avaluar les expressions farem servir xmllint, una aplicació en línia de comandes per a Linux.

Per seleccionar totes les denominacions podem fer:

```
xmllint -xpath '/terminologiaoberta/fitxes/fitxa/denominacio' to.xml
```

O bé:

```
xmllint -xpath '//denominacio' to.xml
```

El resultat que obtenim són totes les denominacions:

```
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="
loc adj">a boca de canó</denominacio><denominacio llengua="es" tipus="equi
valent" jerarquia="terme pral." categoria="">a bocajarro</denominacio><denomina
cio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="loc adj">
a curta distància</denominacio><denominacio llengua="es" tipus="equivalent
" jerarquia="terme pral." categoria="">a corta distancia</denominacio><denomina
cio llengua="en" tipus="equivalent" jerarquia="terme pral." categoria="">near</
denominacio><denominacio llengua="ca" tipus="principal" jerarquia="terme pral."
 categoria="loc adj">a frec de roba</denominacio>.....
```

Si ara volem només les denominacions en català podem fer:

```
xmllint -xpath '/terminologiaoberta/fitxes/fitxa/denominacio[@llengua="ca"]' to.xml
```

O bé:

```
xmllint -xpath '//denominacio[@llengua="ca"]' to.xml
```

El resultat que obtenim en tots dos casos són totes les denominacions en català:

```
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="
loc adj">a boca de canó</denominacio><denominacio llengua="ca" tipus="prin
cipal" jerarquia="terme pral." categoria="loc adj">a curta distància</deno
minacio><denominacio llengua="ca" tipus="principal" jerarquia="terme pral." cat
egoria="loc adj">a frec de roba</denominacio>...
```

Si ara volem la informació només del segon element podem escriure:

```
xmllint -xpath '/terminologiaoberta/fitxes/fitxa[2]/denominacio[@llengua="ca"]' to.xml
```

I obtenim:

```
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="
loc adj">a curta distància</denominacio>
```

També podem combinar dues expressions, per per exemple obtenir la denominació catalana i l'anglesa, com per exemple:

```
xmllint -xpath '//fitxa[2]/denominacio[@llengua="ca"] |
//fitxa[2]/denominacio[@llengua="en"]' to.xml
```

On obtenim:

```
<denominacio llengua="ca" tipus="principal" jerarquia="terme pral." categoria="
loc adj">a curta distància</denominacio><denominacio llengua="en" tipus="e
quivalent" jerarquia="terme pral." categoria="">near</denominacio>
```

Una altra molt bona opció d'eina en línia de comandes, disponible per Linux i Windows, per avaluar XPATH és xmlstarlet (http://xmlstar.sourceforge.net/). Si fem:

```
xmlstarlet sel  -t -v '//denominacio' to.xml
```

obtenim:

```
a boca de canó
a bocajarro
a curta distància
a corta distancia
near
a frec de roba
a crema-roba
a quemarropa
a l'uníson
```

És a dir, sense els tags d'XML.

Podeu aprendre més sobre les expressions Xpath al tutoral del W3Schools (http://www.w3schools.com/xpath)

5.7.f. Editors d'XML

Els arxius XML són arxius de text, i poden editar-se en qualsevol editor de text. Tot i això, és recomanable fer servir editors específics per a XML. Aquests editors específics tenen diversos avantatges:

- Acoloreixen el text diferenciant elements, atributs, etc., cosa que ajuda visualment a escriure i llegir els fitxers
- Avisen si deixem de tancar alguna etiqueta
- Autocompleció d'etiquetes
- Poden verificar si el document està ben format i si és vàlid.
- Alguns d'ells permeten fer també transformacions XSLT
- Alguns d'ells permeten també avaluar expressions XPATH

Una bona opció, i de programari lliure, disponible tant per a Linux com per a Windows, és l'XML Copy Editor, que es pot descarregar de http://xml-copy-editor.sourceforge.net/

5.7.g. Traducció de documents XML

Els arxius XML són arxius de text i per tant no ofereixen massa dificultat en la seva traducció. La traducció es pot fer directament amb qualsevol editor de text. Si fem servir un editor específic per a XML ens podrà avisar si en algun moment modifiquem algun aspecte de l'estructura de l'XML.

Els arxius XML també es poden traduir fàcilment amb qualsevol eina de traducció assistida. Imaginem-nos que tenim el següent arxiu XML per traduir (mostrem només la informació corresponent a un dels llibres i a més de forma escurçada, però imaginem-nos que hi ha centenars d'aquests):

```
<bd_llibres>
<llibre id="1">
<autor>John Steinbeck</autor>
<titol>El raïm de la ira</titol>
<traductor>Mercè López Arrabat</traductor>
<editorial>Edicions 62</editorial>
<colleccio>Les millors obres de la literatura universal segle XX</colleccio>
<numero>83</numero>
<any_publicacio>1993</any_publicacio>
<biografia_autor>John Steinbeck (1902-1968), escriptor nordamericà, nascut a
Califòrnia. ...</biografia_autor>
<resum>Quan es va publicar El raïm de la ira l'any 1939, va conmoure l'opinió
nord-americana, que s'estava tot just recuperant de la Gran Depressió.
....</resum>
</llibre>
...
</bd_llibres>
```

Si creem un arxiu XLIFF sense crear un filtre específic (en aquesta prova hem fet servir l'eina Rainbow d'Okapi) obtenim el següent:

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:okp="okapi-framework:xliff-
extensions" xmlns:its="http://www.w3.org/2005/11/its" xmlns:itsxlf="http://www.w3.org/ns/its-xliff/"
its:version="2.0">
<file original="bdllibres.xml" source-language="ca" target-language="es" datatype="xml">
<body>
<trans-unit id="1">
```



```
<source xml:lang="ca">John Steinbeck</source>
<target xml:lang="es">John Steinbeck</target>
</trans-unit>
<trans-unit id="2">
<source xml:lang="ca">El raïm de la ira</source>
<target xml:lang="es">El raïm de la ira</target>
</trans-unit>
<trans-unit id="3">
<source xml:lang="ca">Mercè López Arrabat</source>
<target xml:lang="es">Mercè López Arrabat</target>
</trans-unit>
<trans-unit id="4">
<source xml:lang="ca">Edicions 62</source>
<target xml:lang="es">Edicions 62</target>
</trans-unit>
<trans-unit id="5">
<source xml:lang="ca">Les millors obres de la literatura universal segle XX</source>
<target xml:lang="es">Les millors obres de la literatura universal segle XX</target>
</trans-unit>
<trans-unit id="6">
<source xml:lang="ca">83</source>
<target xml:lang="es">83</target>
</trans-unit>
<trans-unit id="7">
<source xml:lang="ca">1993</source>
<target xml:lang="es">1993</target>
</trans-unit>
<trans-unit id="8">
<source xml:lang="ca">John Steinbeck (1902-1968), escriptor nordamericà, nascut a
Califòrnia.</source>
<target xml:lang="es">John Steinbeck (1902-1968), escriptor nordamericà, nascut a
Califòrnia.</target>
</trans-unit>
<trans-unit id="9">
<source xml:lang="ca">Quan es va publicar El raïm de la ira l'any 1939, va conmoure l'opinió nord-
americana, que s'estava tot just recuperant de la Gran Depressió. </source>
<target xml:lang="es">Quan es va publicar El raïm de la ira l'any 1939, va conmoure l'opinió nord-
americana, que s'estava tot just recuperant de la Gran Depressió. </target>
</trans-unit>
</body>
</file>
</xliff>
```

Per poder traduir de manera eficient aquest arxiu haurem de crear un filtre específic per a que el programa seleccioni només la informació que és traduïble, i evitar haver de traduir per exemple, el nom de l'autor, del traductor, l'editorial, la col·lecció i l'any de publicació. Potser per uns pocs llibres no té importància, però si el fitxer XML té una gran quantitat d'entrades pot ser interessant eliminar aquesta informació de l'arxiu a traduir. Així doncs, només voldrem traduir el títol, la biografia_autor i el resum.

Per fer això podem crear un filtre específic per a aquest fitxer XML. Rainbow utilitza la recomanació ITS (*Internationalization Tag Set*) per crear els filtres. Podeu consultar els detalls a: http://www.opentag.com/okapi/wiki/index.php?title=XML_Filter. El filtre resultant seria:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><its:rules
xmlns:its="http://www.w3.org/2005/11/its" xmlns:itsx="http://www.w3.org/2008/12/its-extensions"
xmlns:okp="okapi-framework:xmlfilter-options" xmlns:xlink="http://www.w3.org/1999/xlink"
version="1.0">
 <its:translateRule selector="//autor" translate="no"/>
 <its:translateRule selector="//traductor" translate="no"/>
 <its:translateRule selector="//editorial" translate="no"/>
 <its:translateRule selector="//colleccio" translate="no"/>
 <its:translateRule selector="//numero" translate="no"/>
 <its:translateRule selector="//any_publicacio" translate="no"/>
</its:rules>
```

Que si l'apliquem a la creació de l'arxiu XLIFF, ens resulta:

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:okp="okapi-framework:xliff-
```

```
extensions" xmlns:its="http://www.w3.org/2005/11/its" xmlns:itsxlf="http://www.w3.org/ns/its-xliff/"
its:version="2.0">
<file original="bdllibres-mod.xml" source-language="ca" target-language="es" datatype="xml">
<body>
<trans-unit id="1">
<source xml:lang="ca">El raïm de la ira</source>
<target xml:lang="es">El raïm de la ira</target>
</trans-unit>
<trans-unit id="2">
<source xml:lang="ca">John Steinbeck (1902-1968), escriptor nordamericà, nascut a
Califòrnia.</source>
<target xml:lang="es">John Steinbeck (1902-1968), escriptor nordamericà, nascut a
Califòrnia.</target>
</trans-unit>
<trans-unit id="3">
<source xml:lang="ca">Quan es va publicar El raïm de la ira l'any 1939, va conmoure l'opinió nord-
americana, que s'estava tot just recuperant de la Gran Depressió. </source>
<target xml:lang="es">Quan es va publicar El raïm de la ira l'any 1939, va conmoure l'opinió nord-
americana, que s'estava tot just recuperant de la Gran Depressió. </target>
</trans-unit>
</body>
</file>
</xliff>
```

Que conté, efectivament la informació que volem traduir. Cada eina de traducció assistida pot tenir un mecanisme diferent per a la creació de filtres XML, però en tot cas sempre es tracta d'especificar quina informació de l'XML és traduïble, o bé, quina informació no ho és.

5.8. Els formats XML emprats en el món de la traducció

En capítols anteriors ja hem vist alguns del formats d'intercanvi basats en XML que es fan servir en el món de la traducció. En aquesta secció revisarem aquests formats i en veurem també unes mètriques especials (GMX), que tot i no ser un format XML, tenen relació amb els nous estàndards que estan apareixen els darrers anys.

5.8.a. Intercanvi de memòries de traducció: TMX

El TMX (*Translation Memory eXchange*) és un format XML estàndard per a l'intercanvi de memòries de traducció (el següent exemple no correspon a un document sencer, únicament a una entrada).

```
<tu tuid="1" datatype="Text" srclang="ca">
   <prop type="x-Client">001</prop>
   <prop type="x-Domain">0049</prop>
   <prop type="x-Project">2053797</prop>
   <prop type="FileID">1</prop>
   <prop type="RowID">0000009</prop>
   <tuv xml:lang="ca" creationdate ="20030601T08:21:33Z"
      creationid="Antoni">
      <prop type="IsSource">True</prop>
      <seg>Tema 4.- </seg>
   </tuv>
   <tuv xml:lang="es" creationdate="20030601T08:21:33Z"
      creationid="Antoni">
      <prop type="IsSource">False</prop>
      <seg>Tema 4.-</seg>
   </tuv>
   <tuv xml:lang="en-gb" creationdate="18991229T23:00:00Z"
      creationid="Antoni">
      <prop type="IsSource">False</prop>
      <seg>Unit 4.-</seg>
   </tuv>
</tu>
```

Aquesta memòria de traducció és multilingüe: català, castellà i anglès. La llengua original del segment que presentem és el català. Les dades que es guarden son client, especialitat, id. del projecte d'on prové i del segment dins del projecte, data de creació, usuari que l'ha creat (en aquest cas Antoni) i el segment en cada una de les llengües.

5.8.b. Intercanvi de bases de dades terminològiques: TBX

El TBX (*TermBase eXchange*) és un format estàndard per a l'intercanvi de bases de dades terminològiques basat en XML. A continuació podem observar un exemple senzill:

```
<?xml version="1.0" ?>
<martif type="TBX" xml:lang="en">
<text>
<body>
   <termEntry id="1">
      <descrip type="subjectField">Linguistics</descrip>
      <langSet xml:lang="ru">
         <tig>
            <term>Компьютерная лингвистика</term>
         </tig>
      </langSet>
      <langSet xml:lang="en">
         <tig>
            <term>Computational linguistics</term>
         </tig>
      </langSet>
      </termEntry>
```


5.8.c. Intercanvi de projectes de traducció: XLIFF

L'XLIFF (*XML Localization Interchange File Format*) és un format basat en XML per a l'intercanvi de projectes de traducció i localització. Amb aquest format és possible crear un projecte amb una eina de traducció assistida (A) i traduir-lo amb una altra de diferent (B), i un cop traduït, exportar el projecte (és a dir crear els arxius traduïts en el seu format original) amb l'eina A.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
<file datatype="x-test" original="manual"
 source-language="EN-US" target-language="CA-ES">
<body>
   <trans-unit id="1">
      <source xml:lang="EN-US">Untranslated text.</source>
   </trans-unit>
   <trans-unit id="2">
      <source xml:lang="EN-US">Translated but un-approved text.</source>
      <target xml:lang="CA-ES">Text traduït però que encara no està aprovat.</target>
   </trans-unit>
   <trans-unit id="3" approved="yes">
      <source xml:lang="EN-US">Translated and approved text.</source>
      <target xml:lang="CA-ES">Text traduït i aprovat.</target>
   </trans-unit>
   <trans-unit id="4">
      <source xml:lang="EN-US">Some other text.</source>
      <alt-trans>
         <source xml:lang="EN-US">Other text.</source>
         <target xml:lang="CA-ES">Un altre text.</target>
      </alt-trans>
   </trans-unit>
</body>
</file>
</xliff>
```

5.8d. Intercanvi de regles de segmentació: SRX

SRX (*Segmentation Rule eXchange*) és un format estàndard basat en XML per a l'intercanvi de regles de segmentació, és a dir, les regles que es fan servir per dividir el text a traduir en segments que es presenten d'un a un al traductor. Si volem aprofitar una determinada memòria de traducció és interessant que les regles de segmentació que faci servir el nostre programa de traducció assistida siguin iguals a les regles de segmentació emprades en la creació de la memòria de traducció. Si no és així, no és gaire greu però es possible que perdem alguna coincidència interessant per diferències en la segmentació. Per aquest motiu s'ha creat aquest format, de manera que quan compartim les nostres memòries de traducció puguem també compartir les regles de segmentació que fem servir. La situació seria la següent. Un col·lega nostre treballa amb l'eina de traducció assistida A (que fa servir unes regles de segmentació determinades) i té una gran memòria de traducció que pot compartir amb nosaltres. Ara nosaltres volem crear un projecte de traducció amb una eina de traducció assistida B (que fa servir unes regles de segmentació que poden ser diferents que les de l'eina A) però volem fer servir la gran memòria de traducció del nostre col·lega. Per maximitzar el nombre de coincidències, serà interessant demanar també l'arxiu de regles de segmentació en format SRX per poder crear el projecte fent servir aquestes regles.

Un arxiu SRX té el següent aspecte:

```
<?xml version="1.0" encoding="UTF-8"?>
<srx       xmlns="http://www.lisa.org/srx20"       xmlns:okpsrx="http://okapi.sf.net/srx-extensions"
version="2.0">
<body>
<languagerules>
<languagerule languagerulename="default">
<rule break="no">
<beforebreak>\b(pp|e\.?\s*g|i\.?\s*e|no|[Vv]ol|[Rr]col|maj|Lt|[Ff]ig|[Ff]igs|[Vv]iz|[Vv]ols|
[Aa]pprox|[Ii]ncl|Pres|Prof|[Dd]ept|min|max|[Gg]ovt|c\.?\s*f|vs)\.</beforebreak>
<afterbreak>\s[^\p{Lu}]</afterbreak>
</rule>
<rule break="no">
<beforebreak>\b(St|Gen|Hon|Dr|Mr|Ms|Mrs|Col|Maj|Brig|Sgt|Capt|Cmnd|Sen|Rev|Rep|Revd)\.</beforebreak>
<afterbreak>\s\p{Lu}</afterbreak>
</rule>
<rule break="no">
<beforebreak>([A-Z]\.){2,}</beforebreak>
<afterbreak>\s[\p{Lu}]</afterbreak>
</rule>
<rule break="yes">
<beforebreak>\w+[\p{Pe}\p{Po}"]*[\.?!]+[\p{Pe}\p{Po}"]*</beforebreak>
<afterbreak>\s</afterbreak>
</rule>
</languagerule>
</languagerules>
<maprules>
<languagemap languagepattern=".*" languagerulename="default"></languagemap>
</maprules>
</body>
</srx>
```

5.8.e. Mètriques GILT: GMX

El GMX (*Global information management Metrics eXchange*) és una col·lecció d'estàndars, alguns en fase de proposta encara, orientats principalment a les necessitats de la indústria de la traducció i que tenen a veure amb la mesura dels aspectes quantitatius d'un document, especialment aquells que tenen una rellevància especial per al procés de traducció (per exemple, comptatge de paraules, complexitat, etc.). No es tracta, doncs, d'un format, sinó d'una sèrie de procediments per poder mesurar d'una manera unificada alguns aspectes dels documents.

En podem distingir tres:

- GMX-V (que té a veure amb el volum) i estableix una manera verificable de calcular els comptatges de paraules. Podem veure el següent exemple: si tenim el següent fragment:

  ```
  <source>In this <g id="g1">exa<x id="x1"/>mple</g> the in-line codes do not form part of the
  word or character counts but are counted separately.</source>
  ```

  es comptaria com a:

  ```
  <source>In this example the in-line codes do not form part of the word or character counts
  but are counted separately.</source>
  ```

  i les mesures GMX- serien les següents:

  ```
  words: 20, characters: 91, inline elements: 3, punctuation characters: 1, white space
  characters: 19
  ```


Les especificacions del GMX-V es poden trobar a:

http://www.gala-global.org/oscarStandards/gmx-v/gmx-v.html

- GMX-Q (que té a veure amb la qualitat) i que representa el nivell de qualitat requerit per a una determinada tasca de traducció. En el moment d'escriure aquest capítol encara no estaven disponibles les especificacions.
- GMX-C (que té a veure amb la complexitat) i que tindrà en compte el document original, el seu format. En el moment d'escriure aquest capítol encara no estaven disponibles les especificacions.

6. Conclusions

En aquest capítol hem vist els conceptes bàsics necessaris per a evitar problemes amb les codificacions de caràcters i els formats d'arxiu. Un cop assimilats aquests conceptes serem capaços de traduir arxius en una gran quantitat de formats i assegurant-nos de que un cop traduïts els documents es podran obrir i visualitzar correctament.

Hem donat una importància especial al format XML, ja que aquest format s'està estenent cada dia més en tot tipus d'aplicacions. Amb els conceptes adquirits, serem capaços de traduir arxius XML amb la nostra eina de traducció assistida preferida.

Per ampliar coneixements

Tutorials d'W3Schools

En aquest capítol he fet referència a un tutorial d'XML de W3Schools (http://www.w3schools.com/). En aquesta web s'ofereixen tutorials gratuïts sobre diverses tecnologies relacionades amb Internet. Val molt la pena donar-li una ullada i seguir els tutorials dels temes que t'interessin. Aquests tutorials estan molt ben desenvolupats, son clars i directes, i et permeten tenir una bona idea d'aquestes tecnologies en molt poc temps.

Detecció automàtica de llengua

Hem comentat que la detecció automàtica de la codificació de caràcters es du a terme d'una manera heurística a partir d'estadístiques de trigrames de caràcters. Una tècnica semblant es pot fer servir per a determinar la llengua en la que està escrito un document. D'aquesta manera es poden crear detectors automàtics de llengua, que a partir de fragments d'un text són capaços de determinar amb força precisió la llengua en la que està escrit.

Hi ha diverses web que permeten detectar la llengua, entre les que podem destacar:

- Language Identifier de Xerox (https://open.xerox.com/Services/LanguageIdentifier)
- TextCat (http://odur.let.rug.nl/~vannoord/TextCat/). Aquest permet descarregar el programa i els models de llengua.

Si alguna vegada us arriba un text i no esteu segurs de en quina llengua està escrit podreu fer servir una d'aquestes eines.

Taules d'Unicode

En http://www.unicode.org/charts/ es poden consultar totes les taules de caràcters. Mostrem ara algunes de les posicions de l'Unicode, corresponents al sil·labari japonès hiragana i a alguns dels caràctes CJK (xinesos, japonesos i coreans unificats):

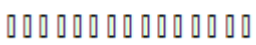
12353-->ぁ-->HIRAGANA LETTER SMALL A
12354-->あ-->HIRAGANA LETTER A
12355-->ぃ-->HIRAGANA LETTER SMALL I
12356-->い-->HIRAGANA LETTER I
12357-->ぅ-->HIRAGANA LETTER SMALL U
12358-->う-->HIRAGANA LETTER U
12359-->ぇ-->HIRAGANA LETTER SMALL E
12360-->え-->HIRAGANA LETTER E
12361-->ぉ-->HIRAGANA LETTER SMALL O
12362-->お-->HIRAGANA LETTER O
12363-->か-->HIRAGANA LETTER KA
12364-->が-->HIRAGANA LETTER GA
12365-->き-->HIRAGANA LETTER KI
12366-->ぎ-->HIRAGANA LETTER GI
12367-->く-->HIRAGANA LETTER KU
12368-->ぐ-->HIRAGANA LETTER GU
12369-->け-->HIRAGANA LETTER KE
12370-->げ-->HIRAGANA LETTER GE
12371-->こ-->HIRAGANA LETTER KO
12372-->ご-->HIRAGANA LETTER GO
12373-->さ-->HIRAGANA LETTER SA
12374-->ざ-->HIRAGANA LETTER ZA
12375-->し-->HIRAGANA LETTER SI
12376-->じ-->HIRAGANA LETTER ZI
12377-->す-->HIRAGANA LETTER SU
12378-->ず-->HIRAGANA LETTER ZU
12379-->せ-->HIRAGANA LETTER SE
12380-->ぜ-->HIRAGANA LETTER ZE
12381-->そ-->HIRAGANA LETTER SO
12382-->ぞ-->HIRAGANA LETTER ZO
12383-->た-->HIRAGANA LETTER TA
12384-->だ-->HIRAGANA LETTER DA
12385-->ち-->HIRAGANA LETTER TI
12386-->ぢ-->HIRAGANA LETTER DI
12387-->っ-->HIRAGANA LETTER SMALL TU
12388-->つ-->HIRAGANA LETTER TU
12389-->づ-->HIRAGANA LETTER DU
12390-->て-->HIRAGANA LETTER TE
12391-->で-->HIRAGANA LETTER DE
12392-->と-->HIRAGANA LETTER TO
12393-->ど-->HIRAGANA LETTER DO
12394-->な-->HIRAGANA LETTER NA

12395-->に-->HIRAGANA LETTER NI
12396-->ぬ-->HIRAGANA LETTER NU
12397-->ね-->HIRAGANA LETTER NE
12398-->の-->HIRAGANA LETTER NO
12399-->は-->HIRAGANA LETTER HA
12400-->ば-->HIRAGANA LETTER BA
12401-->ぱ-->HIRAGANA LETTER PA
12402-->ひ-->HIRAGANA LETTER HI
12403-->び-->HIRAGANA LETTER BI
12404-->ぴ-->HIRAGANA LETTER PI
12405-->ふ-->HIRAGANA LETTER HU
12406-->ぶ-->HIRAGANA LETTER BU
12407-->ぷ-->HIRAGANA LETTER PU
12408-->へ-->HIRAGANA LETTER HE
12409-->べ-->HIRAGANA LETTER BE
12410-->ぺ-->HIRAGANA LETTER PE
12411-->ほ-->HIRAGANA LETTER HO
12412-->ぼ-->HIRAGANA LETTER BO
12413-->ぽ-->HIRAGANA LETTER PO
12414-->ま-->HIRAGANA LETTER MA
12415-->み-->HIRAGANA LETTER MI
12416-->む-->HIRAGANA LETTER MU
12417-->め-->HIRAGANA LETTER ME
12418-->も-->HIRAGANA LETTER MO
12419-->ゃ-->HIRAGANA LETTER SMALL YA
12420-->や-->HIRAGANA LETTER YA
12421-->ゅ-->HIRAGANA LETTER SMALL YU
12422-->ゆ-->HIRAGANA LETTER YU
12423-->ょ-->HIRAGANA LETTER SMALL YO
12424-->よ-->HIRAGANA LETTER YO
12425-->ら-->HIRAGANA LETTER RA
12426-->り-->HIRAGANA LETTER RI
12427-->る-->HIRAGANA LETTER RU
12428-->れ-->HIRAGANA LETTER RE
12429-->ろ-->HIRAGANA LETTER RO
12430-->ゎ-->HIRAGANA LETTER SMALL WA
12431-->わ-->HIRAGANA LETTER WA
12432-->ゐ-->HIRAGANA LETTER WI
12433-->ゑ-->HIRAGANA LETTER WE
12434-->を-->HIRAGANA LETTER WO
12435-->ん-->HIRAGANA LETTER N
...
....
23473-->宿-->CJK UNIFIED IDEOGRAPH-5BB1
23474-->寂-->CJK UNIFIED IDEOGRAPH-5BB2
23475-->害-->CJK UNIFIED IDEOGRAPH-5BB3
23476-->宴-->CJK UNIFIED IDEOGRAPH-5BB4
23477-->宵-->CJK UNIFIED IDEOGRAPH-5BB5

23478-->家-->CJK UNIFIED IDEOGRAPH-5BB6
23479-->寒-->CJK UNIFIED IDEOGRAPH-5BB7
23480-->宸-->CJK UNIFIED IDEOGRAPH-5BB8
23481-->容-->CJK UNIFIED IDEOGRAPH-5BB9
23482-->寔-->CJK UNIFIED IDEOGRAPH-5BBA

Problemes de visualització de documents relacionats amb les fonts

L'Unicode és capaç de representar la majoria de caràcters existents, però això no vol dir que els visualitzem correctament. Per poder visualitzar-los hem de tenir una font que tingui el glif (la representació del caràcter). Fixeu-vos en aquest exemple, d'un text en glagolític.



Observem que podem visualitzar els caràcters glagolítics i es visualitzen uns quadrats. Des del punt de vista informàtic, no s'ha perdut cap informació (ja que en aquella posició hi ha els bytes que hi ha d'haver), però el editor no és capaç de visualitzar el contingut. Si seleccionem una font que contingui els caràcters a representar, llavors podrem veure el text correctament:

També, si explorem algunes de les zones de l'Unicode, com en el següent exemple, veurem que no tenim els caràcters corresponents.

18558-->-->CJK UNIFIED IDEOGRAPH-487E

que correspon a:

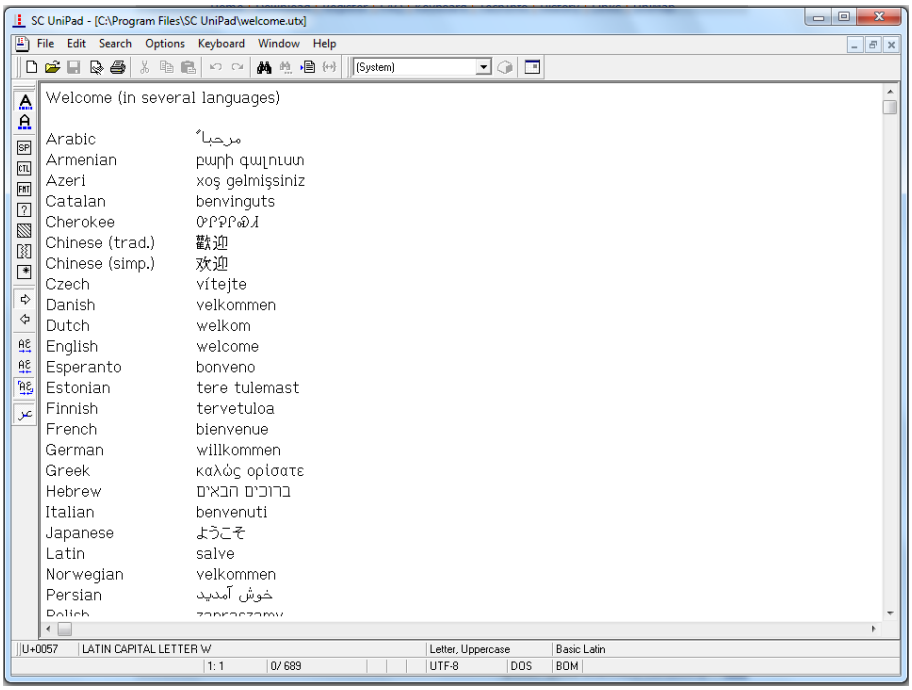
Així que recordem que el fet de veure en el nostre editor un quadradet en lloc del caràcter adequat suposa que la font que estem fent servir no té la imatge del caràcter corresponent. Podrem seleccionar una altra font i fins i tot buscar noves fonts per instal·lar al nostre sistema.

SC Unipad: Editor d'Unicode

Els arxius Unicode són arxius de text que es poden obrir en qualsevol editor de textos. Hi ha però, uns editors especials per Unicode que simplement ofereixen les següents funcionalitats:

- Treballen amb una font completa de tot l'Unicode, pel que poden visualitzar la totalitat dels caràcters representats per l'Unicode.
- Ofereixen teclats virtuals per poder entrar el text fàcilment en diversos alfabets
- Permeten seleccionar qualsevol caràcter del repertorit Unicode.

SCP-Unipad és un bon exemple d'aquest tipus d'editors. Està disponible per Windows, és gratuït tot i que no de programari lliure i es pot descarregar de http://www.unipad.org/.



Bibliografia

Sayton, Bob (2007) *DocBook XSL: The Complete Guide* Sagehill Enterprises. Es pot accedir al contingut d'aquest llibre a http://sagehill.net/docbookxsl/index.html

Annex I. Entitats d'html

Caràcters ASCII

	Entity Name	Entity Number
		&#32;
!		&#33;
"		&#34;
#		&#35;
$		&#36;
%		&#37;
&	&amp;	&#38;
'		&#39;
(		&#40;
)		&#41;
*		&#42;
+		&#43;
,		&#44;
-		&#45;
.		&#46;
/		&#47;
0		&#48;
1		&#49;
2		&#50;
3		&#51;
4		&#52;
5		&#53;
6		&#54;
7		&#55;
8		&#56;
9		&#57;
:		&#58;
;		&#59;
<	&lt;	&#60;
=		&#61;
>	&gt;	&#62;
?		&#63;
@		&#64;
A		&#65;

	Entity Name	Entity Number
B		&#66;
C		&#67;
D		&#68;
E		&#69;
F		&#70;
G		&#71;
H		&#72;
I		&#73;
J		&#74;
K		&#75;
L		&#76;
M		&#77;
N		&#78;
O		&#79;
P		&#80;
Q		&#81;
R		&#82;
S		&#83;
T		&#84;
U		&#85;
V		&#86;
W		&#87;
X		&#88;
Y		&#89;
Z		&#90;
[		&#91;
****		&#92;
]		&#93;
^		&#94;
_		&#95;
`		&#96;
a		&#97;
b		&#98;
c		&#99;
d		&#100;
e		&#101;
f		&#102;
g		&#103;
h		&#104;
i		&#105;
j		&#106;
k		&#107;
l		&#108;
m		&#109;
n		&#110;
o		&#111;
p		&#112;
q		&#113;
r		&#114;

	Entity Name	Entity Number
s		&#115;
t		&#116;
u		&#117;
v		&#118;
w		&#119;
x		&#120;
y		&#121;
z		&#122;
{		&#123;
|		&#124;
}		&#125;
~		&#126;

Caràcters ISO-8859-1

	Entity Name	Entity Number
À	&Agrave;	&#192;
Á	&Aacute;	&#193;
Â	&Acirc;	&#194;
Ã	&Atilde;	&#195;
Ä	&Auml;	&#196;
Å	&Aring;	&#197;
Æ	&AElig;	&#198;
Ç	&Ccedil;	&#199;
È	&Egrave;	&#200;
É	&Eacute;	&#201;
Ê	&Ecirc;	&#202;
Ë	&Euml;	&#203;
Ì	&Igrave;	&#204;
Í	&Iacute;	&#205;
Î	&Icirc;	&#206;
Ï	&Iuml;	&#207;
Ð	&ETH;	&#208;
Ñ	&Ntilde;	&#209;
Ò	&Ograve;	&#210;
Ó	&Oacute;	&#211;
Ô	&Ocirc;	&#212;
Õ	&Otilde;	&#213;
Ö	&Ouml;	&#214;
Ø	&Oslash;	&#216;
Ù	&Ugrave;	&#217;
Ú	&Uacute;	&#218;
Û	&Ucirc;	&#219;
Ü	&Uuml;	&#220;
Ý	&Yacute;	&#221;
Þ	&THORN;	&#222;
ß	&szlig;	&#223;
à	&agrave;	&#224;

	Entity Name	Entity Number
á	á	á
â	â	â
ã	ã	ã
ä	ä	ä
å	å	å
æ	æ	æ
ç	ç	ç
è	è	è
é	é	é
ê	ê	ê
ë	ë	ë
ì	ì	ì
í	í	í
î	î	î
ï	ï	ï
ð	ð	ð
ñ	ñ	ñ
ò	ò	ò
ó	ó	ó
ô	ô	ô
õ	õ	õ
ö	ö	ö
ø	ø	ø
ù	ù	ù
ú	ú	ú
û	û	û
ü	ü	ü
ý	ý	ý
þ	þ	þ
ÿ	ÿ	ÿ

Símbols ISO-8859-1

	Entity Name	Entity Number
¡	¡	¡
¢	¢	¢
£	£	£
¤	¤	¤
¥	¥	¥
¦	¦	¦
§	§	§
¨	¨	¨
©	©	©
ª	ª	ª
«	«	«
¬	¬	¬
®	®	®

	Entity Name	Entity Number
¯	¯	¯
°	°	°
±	±	±
²	²	²
³	³	³
´	´	´
µ	µ	µ
¶	¶	¶
¸	¸	¸
¹	¹	¹
º	º	º
»	»	»
¼	¼	¼
½	½	½
¾	¾	¾
¿	¿	¿
×	×	×
÷	÷	÷

Símbols matemàtics

	Entity Name	Entity Number
∀	∀	∀
∂	∂	∂
∃	∃	∃
∅	∅	∅
∇	∇	∇
∈	∈	∈
∉	∉	∉
∋	∋	∋
∏	∏	∏
∑	∑	∑
−	−	−
∗	∗	∗
√	√	√
∝	∝	∝
∞	∞	∞
∠	∠	∠
∧	∧	∧
∨	∨	∨
∩	∩	∩
∪	∪	∪
∫	∫	∫
∴	∴	∴

	Entity Name	Entity Number
~	∼	∼
≅	≅	≅
≈	≈	≈
≠	≠	≠
≡	≡	≡
≤	≤	≤
≥	≥	≥
⊂	⊂	⊂
⊃	⊃	⊃
⊄	⊄	⊄
⊆	⊆	⊆
⊇	⊇	⊇
⊕	⊕	⊕
⊗	⊗	⊗
⊥	⊥	⊥
·	⋅	⋅

Lletres gregues

Letter	Entity Name	Entity Number
Α	Α	Α
Β	Β	Β
Γ	Γ	Γ
Δ	Δ	Δ
Ε	Ε	Ε
Ζ	Ζ	Ζ
Η	Η	Η
Θ	Θ	Θ
Ι	Ι	Ι
Κ	Κ	Κ
Λ	Λ	Λ
Μ	Μ	Μ
Ν	Ν	Ν
Ξ	Ξ	Ξ
Ο	Ο	Ο
Π	Π	Π
Ρ	Ρ	Ρ
Σ	Σ	Σ
Τ	Τ	Τ
Υ	Υ	Υ
Φ	Φ	Φ
Χ	Χ	Χ
Ψ	Ψ	Ψ
Ω	Ω	Ω
α	α	α
β	β	β
γ	γ	γ
δ	δ	δ
ε	ε	ε

Letter	Entity Name	Entity Number
ζ	ζ	ζ
η	η	η
θ	θ	θ
ι	ι	ι
κ	κ	κ
λ	λ	λ
μ	μ	μ
ν	ν	ν
ξ	ξ	ξ
ο	ο	ο
π	π	π
ρ	ρ	ρ
ς	ς	ς
σ	σ	σ
σ	σ	σ
τ	τ	τ
υ	υ	υ
φ	φ	φ
χ	χ	χ
ψ	ψ	ψ
ω	ω	ω
ϑ	ϑ	ϑ
ϒ	ϒ	ϒ
ϖ	ϖ	ϖ

Altres símbols

Symbol	Entity Name	Entity Number
Œ	Œ	Œ
œ	œ	œ
Š	Š	Š
š	š	š
Ÿ	Ÿ	Ÿ
ƒ	ƒ	ƒ
ˆ	ˆ	ˆ
˜	˜	˜
	‌	‌
	‍	‍
	‎	‎

Symbol	Entity Name	Entity Number
	‏	‏
–	–	–
—	—	—
‘	‘	‘
’	’	’
‚	‚	‚
“	“	“
”	”	”
„	„	„
†	†	†
‡	‡	‡
•	•	•
…	…	…
‰	‰	‰
′	′	′
″	″	″
‹	‹	‹
›	›	‹
‾	‾	‾
€	€	€
™	™	™
←	←	←
↑	↑	↑
→	→	→
↓	↓	↓
↔	↔	↔
↵	↵	↵
⌈	⌈	⌈
⌉	⌉	⌉
⌊	⌊	⌊
⌋	⌋	⌋
◊	◊	◊
♠	♠	♠
♣	♣	♣
♥	♥	♥
♦	♦	♦