Corso di Biblioteche Digitali

- Vittore Casarosa
  - Mail: casarosa@isti.cnr.it
  - Tel: 050 621 3115 (office) 348 397 2168 (mobile)
  - Skype: vittore1201
- “Ricevimento” at the end of the lessons or by appointment
- Final assessment
  - 70% oral examination
  - 30% project (development of a small digital library))
- Reference material:
  - Ian Witten, David Bainbridge, David Nichols, How to build a Digital Library, Morgan Kaufmann, 2010, ISBN 978-0-12-374857-7 (Second edition)
  - Material provided by the teacher
- **http://cloudone.isti.cnr.it/casarosa/BDG/**

Modules

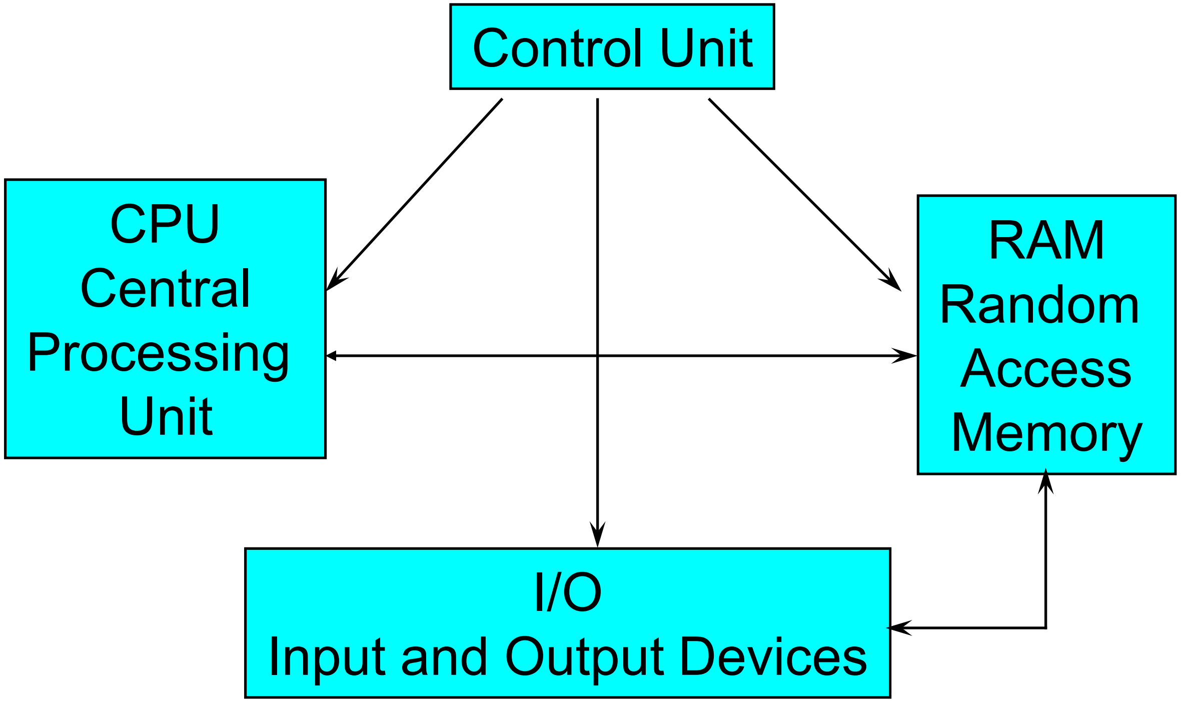
- Computer Fundamentals and Networking
- A conceptual model for Digital Libraries
- Bibliographic records and metadata
- Information Retrieval and Search Engines
- Knowledge representation
- Digital Libraries and the Web
- Hands-on laboratory: the Greenstone system

Refresher

Refresher on Computer Fundamentals and Networking

- History of computers
- Architecture of a computer
- Computer networks and the Internet
- Data representation within a computer

Basic components of a computer

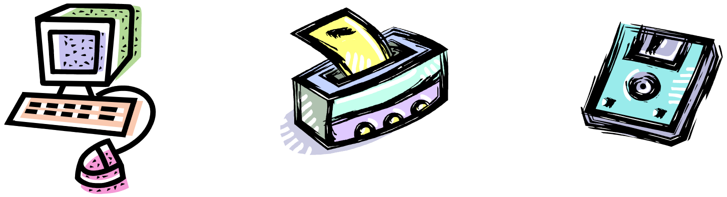


Control Unit

CPU
Central
Processing
Unit

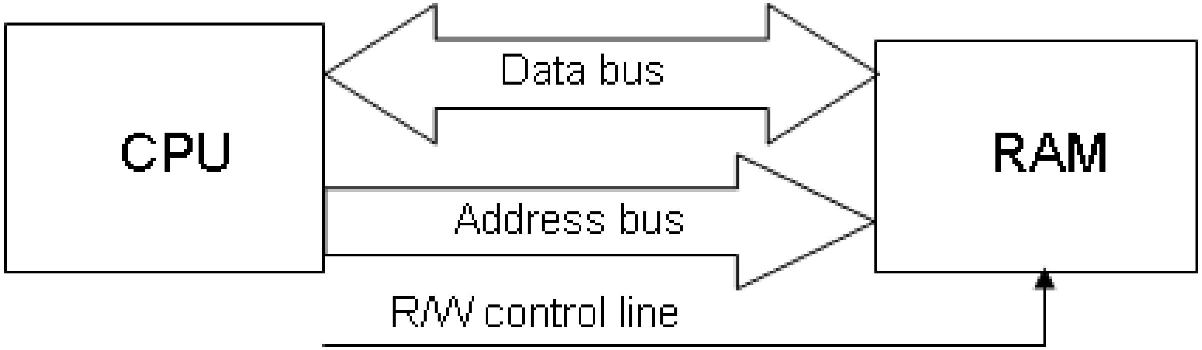
RAM
Random
Access
Memory

I/O
Input and Output Devices



Random Access Memory

- The RAM is a linear array of “cells”, usually called “words”
- The words are numbered from 0 to N, and this number is the “address” of the word
- In order to read/write a word from/into a memory cell, the CPU has to provide its address on the “address bus”
- A “control line” tells the memory whether it is a read or write operation
- In a read operation the memory will provide on the “data bus” the content of the memory cell at the address provided on the “address bus”
- In a write operation the memory will store the data provided on the “data bus” into the memory cell at the address provided on the “address bus”



Data within a computer

- The Control Unit, the RAM, the CPU and all the physical components in a computer act on electrical signals and on devices that (basically) can be in only one of two possible states
- The two states are conventionally indicated as “zero” and “one” (0 and 1), and usually correspond to two voltage levels
- The consequence is that all the data within a computer (or in order to be processed by a computer) has to be represented with 0s and 1s, i.e. in “binary notation”

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio

Positional notation base 10

Positional notation in base 10

Ten different symbols are needed for the digits (0,1,2,3,4,5,6,7,8,9)

The “weight” of each digit is a power of 10 (the base) and depends on its position in the number

$10^0=1$

$10^1=10$

$10^2=100$

$10^3=1000$

$10^4=10000$

$$\boxed{3} \quad \boxed{4} \quad \boxed{7}$$

$$3x10^2 + 4x10^1 + 7x10^0 = 347$$

Roman numbers

Roman numbers are not positional

They are the sum of the values, unless a smaller value precedes a larger one; in that case the smaller value is subtracted from the larger one

I=1	XXVII = 27
V=5	XXXIV = 34
X=10	XLV = 45
L=50	MCMXCIX = 1999
C=100	MMVIII = 2008
D=500	MMIX = 2009
M=1000	MMX = 2010

Positional notation base 8

Positional notation in base 8

Eight different symbols are needed for the digits (0,1,2,3,4,5,6,7)

The “weight” of each digit is a power of 8 (the base) and depends on its position in the number

$8^0=1$

$8^1=8$

$8^2=64$

$8^3=512$

$8^4=4096$

$$\boxed{\mathbf{3}} \qquad \boxed{\mathbf{4}} \qquad \boxed{\mathbf{7}}$$

$$3x8^2 \; + \; 4x8^1 \; + \; 7x8^0$$

$$192 \; + \; 32 \; + \; 7 = 231$$

Positional notation base 16

Positional notation in base 16

Sixteen different symbols are needed for the digits (0,1,2,3,4,5,6,7, 8,9,A,B,C,D,E,F)

The “weight” of each digit is a power of 16 (the base) and depends on its position in the number

$16^0=1$

$16^1=16$

$16^2=256$

$16^3=4096$

$16^4=65536$

$\boxed{\mathbf{3}} \quad \boxed{\mathbf{B}} \quad \boxed{\mathbf{F}}$

$$3\text{x}16^2 + \text{B}\text{x}16^1 + \text{F}\text{x}16^0$$

$$3\text{x}256 + 11\text{x}16 + 15\text{x}1$$

$$768 + 176 + 15 = 959$$

Positional notation base 2

Positional notation in base 2

Two different symbols are needed for the digits (0,1)

The “weight” of each digit is a power of 2 (the base) and depends on its position in the number

$2^0=1$

$2^1=2$

$2^2=4$

$2^3=8$

$2^4=16$

$2^5=32$

$2^6=64$

$2^7=128$

$2^8=256$

$$\boxed{1} \quad \boxed{0} \quad \boxed{1} \quad \boxed{1}$$

$$1x2^3 + 0x2^2 + 1x2^1 + 1x2^0$$

$$1x8 + 0x4 + 1x2 + 1x1$$

$$8 + 0 + 2 + 1 = 11$$

Powers of 2

$2^0=1$

$2^1=2$

$2^2=4$

$2^3=8$

$2^4=16$

$2^5=32$

$2^6=64$

$2^7=128$

$2^8=256$

$2^9=512$		
$2^{10}=1024$	**1K**	
$2^{11}=2048$	2K	
$2^{12}=4096$	4K	
$2^{13}=8192$	8K	
$2^{14}=16384$	16K	
$2^{15}=32768$	32K	
$2^{16}=65356$	**64K**	
. . . . . .		
$2^{20}=1.048.576$	**1024K**	**1M**
$2^{30}=1.073.741.824$	**1024M**	**1G**
$2^{32}=4.271.406.736$	**4096M**	**4G**

Binary and hexadecimal numbers

$2^0=1$
$2^1=2$
$2^2=4$
$2^3=8$
$2^4=16$
$2^5=32$
$2^6=64$
$2^7=128$
$2^8=256$

0000=0	1000=8	
0001=1	1001=9	
0010=2	1010=10	A
0011=3	1011=11	B
0100=4	1100=12	C
0101=5	1101=13	D
0110=6	1110=14	E
0111=7	1111=15	F

10000=16 10

decimal and exadecimal
decimal
hexadecimal

01011011 si può rappresentare in esadecimale come 5B

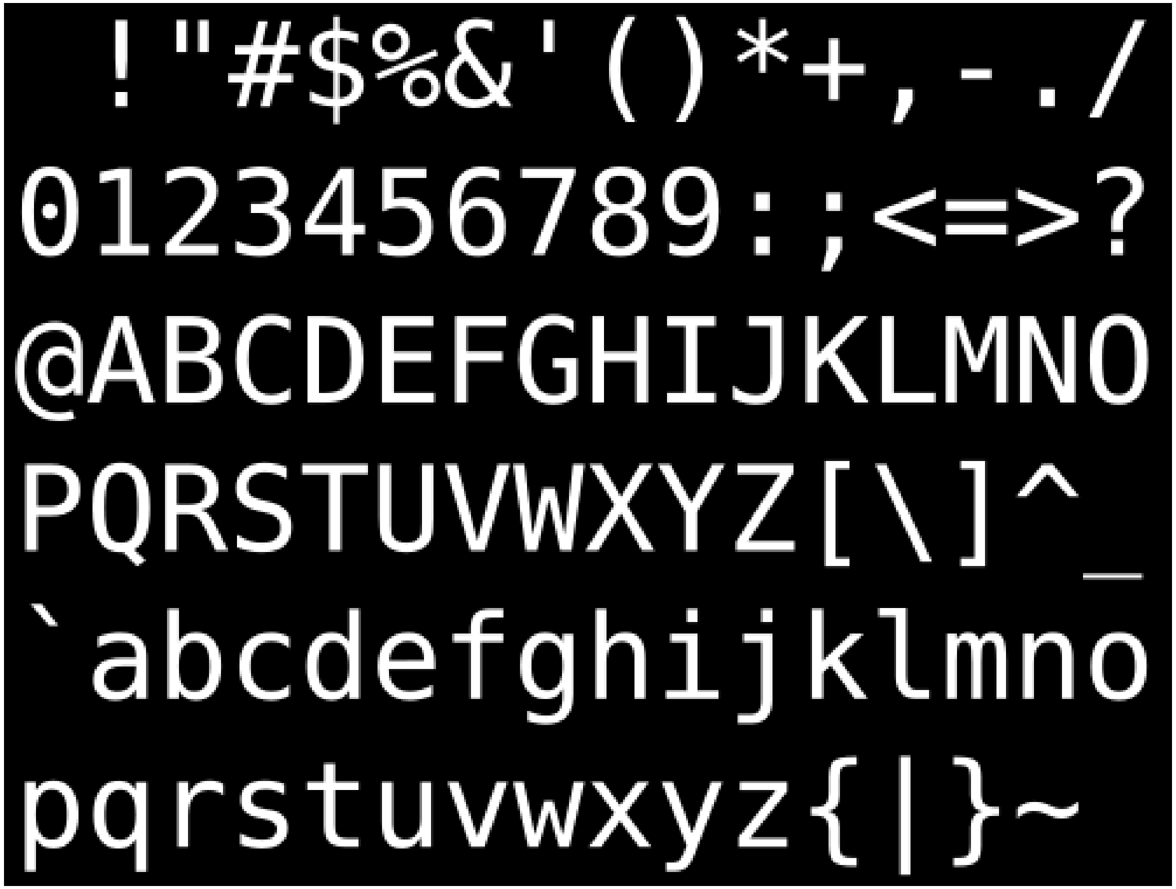
Representation of information within a computer

- Numbers
- Text (characters and ideograms) ⬅
- Documents
- Images
- Video
- Audio

Representation of characters

- The “natural” way to represent (alphanumeric) characters (and symbols) within a computer is to associate a character with a number, defining a “coding table”
- How many bits are needed to represent the Latin alphabet ?

The ASCII characters

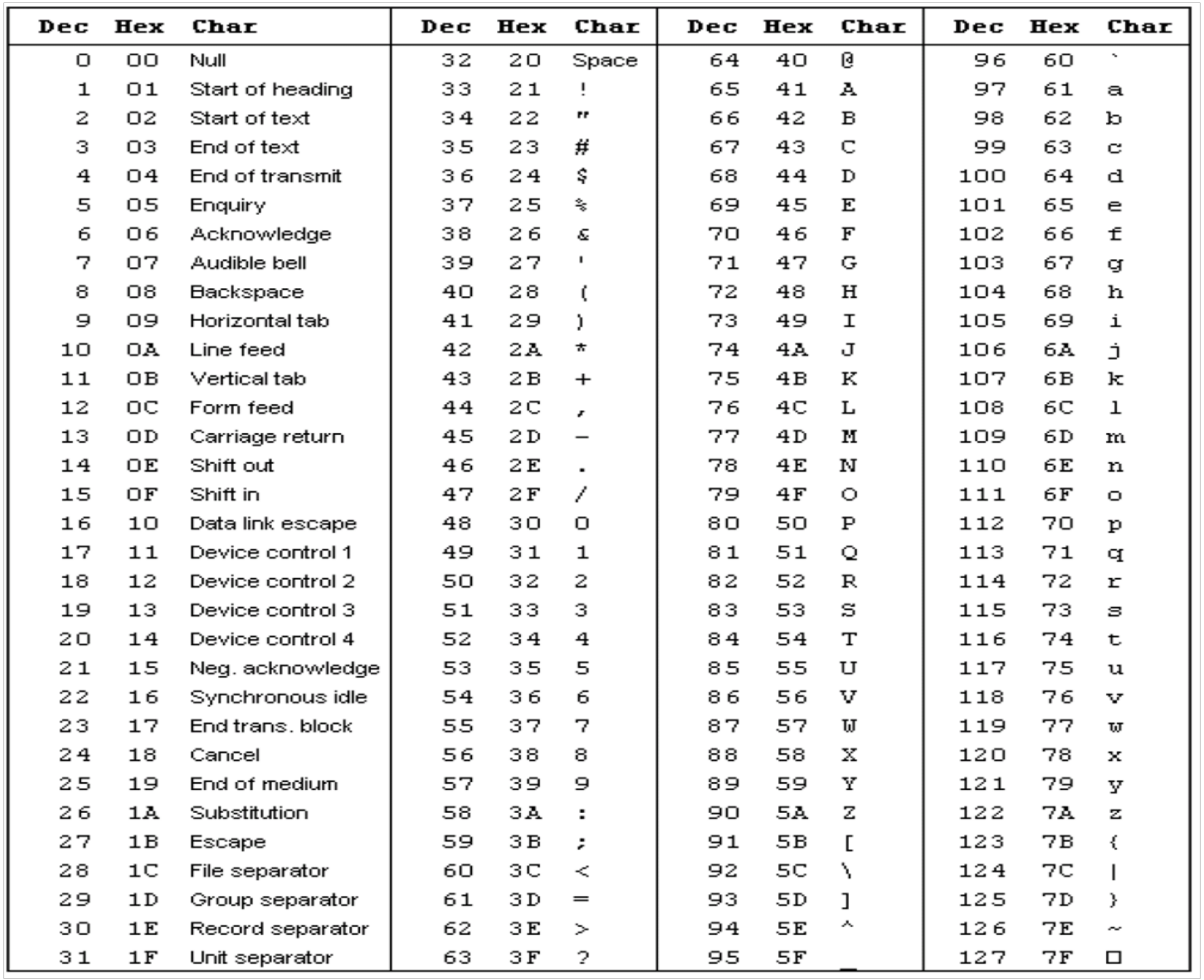


```
 !"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~
```

The 95 printable ASCII characters, numbered from 32 to 126 (decimal)

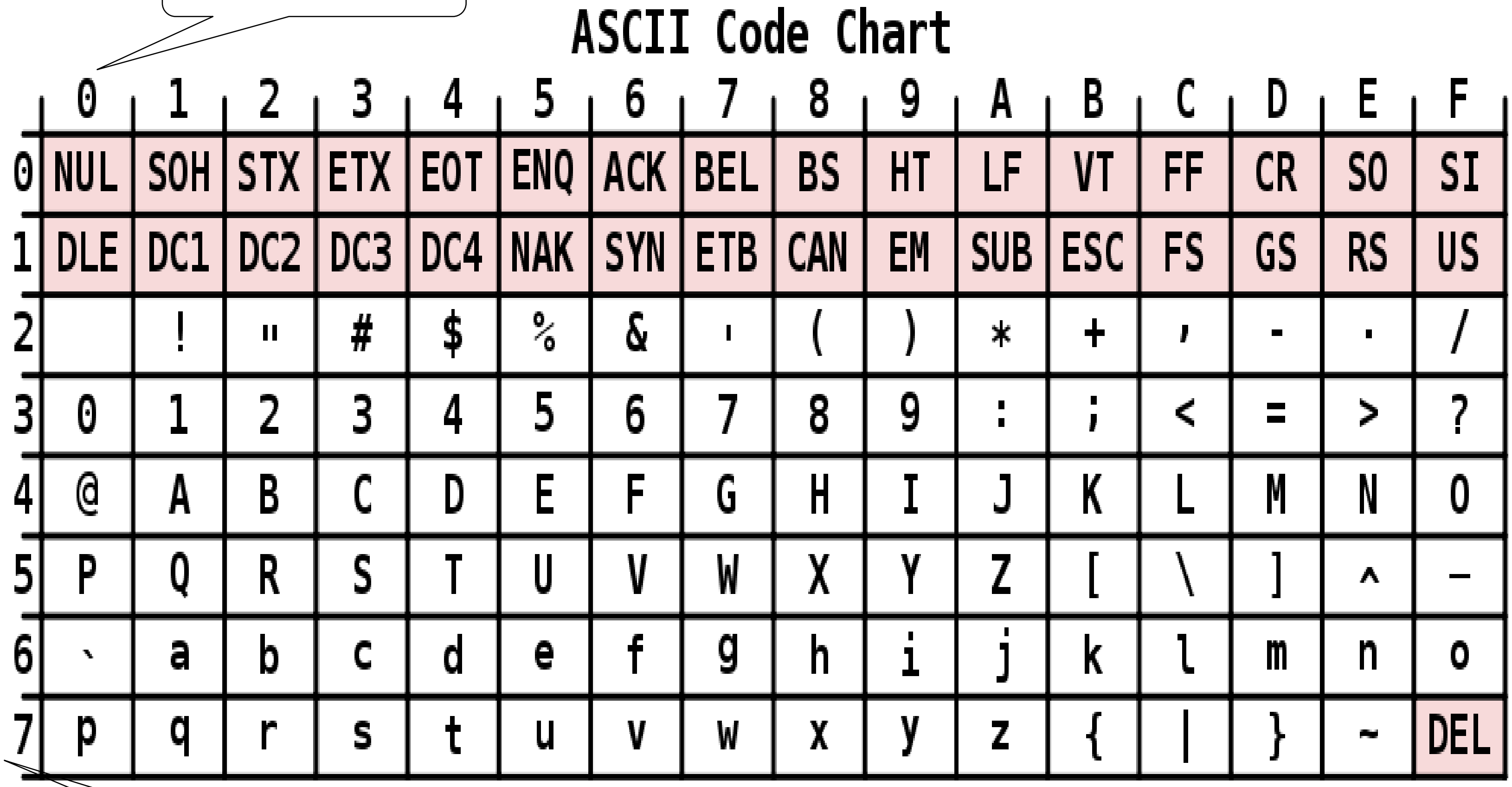
33 control characters

ASCII table (7 bits)



Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	–	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

ASCII 7-bits character set



ASCII Code Chart

	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		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
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

Last 4 bits

First 3 bits

Representation standards

- ASCII 7 bits (late fifties)
  - American Standard Code for Information Interchange
  - 7 bits for 128 characters (Latin alphabet, numbers, punctuation, control characters)
- EBCDIC (early sixties)
  - Extended Binary Code Decimal Interchange Code
  - 8 bits; defined by IBM in early sixties, still in use on (very) old IBM computers
- ASCII 8 bits (ISO 8859-xx) extends original ASCII to 8 bits to include accented letters and non Latin alphabets (e.g. Greek, Russian)
- UNICODE or ISO-10646 (1993)
  - Merged efforts of the Unicode Consortium and ISO
  - UNIversal CODE still evolving
  - It incorporates all(?) the pre-existing representation standards
  - Basic rule: round trip compatibility
    - Side effect is multiple representations for the same character

ISO-8859-xx (ASCII 8-bits)

- Developed by ISO (International Organization for Standardization)
- There are 16 different tables coding characters with 8 bit
- Each table includes ASCII (7 bits) in the lower part and other characters in the upper part for a total of 191 characters and 32 control codes
- It is also known as ISO-Latin–xx (includes all the characters of the “Latin alphabet”)

ISO-8859-xx code pages

8859-1	Latin-1	Western European languages
8859-2	Latin-2	Central European languages
8859-3	Latin-3	South European languages
8859-4	Latin-4	North European languages
8859-5	Latin/Cyrillic	Slavic languages
8859-6	Latin/Arabic	Arabic language
8859-7	Latin/Greek	modern Greek alphabet
8859-8	Latin/Hebrew	modern Hebrew alphabet
8859-9	Latin-5	Turkish language (similar to 8859-1)
8859-10	Latin-6	Nordic languages (rearrangement of Latin-4)
8859-11	Latin/Thai	Thai language
8859-12	Latin/Devanagari	Devanagari language (abandoned in 1997)
8859-13	Latin-7	Baltic Rim languages
8859-14	Latin-8	Celtic languages
8859-15	Latin-9	Revision of 8859-1
8859-16	Latin-10	South-Eastern European languages

Representation standards

- ASCII (late fifties)
  - American Standard Code for Information Interchange
  - 7 bits for 128 characters (Latin alphabet, numbers, punctuation, control characters)
- EBCDIC (early sixties)
  - Extended Binary Code Decimal Interchange Code
  - 8 bits; defined by IBM in early sixties, still used and supported on many computers
- ISO 8859-1 extends ASCII to 8 bits (accented letters, non Latin characters)
- UNICODE or ISO-10646 (1993)
  - Merged efforts of the Unicode Consortium and ISO
  - UNIversal CODE still evolving
  - It incorporates all(?) the pre-existing representation standards
  - Basic rule: round trip compatibility
    - Side effect is multiple representations for the same character

UNICODE

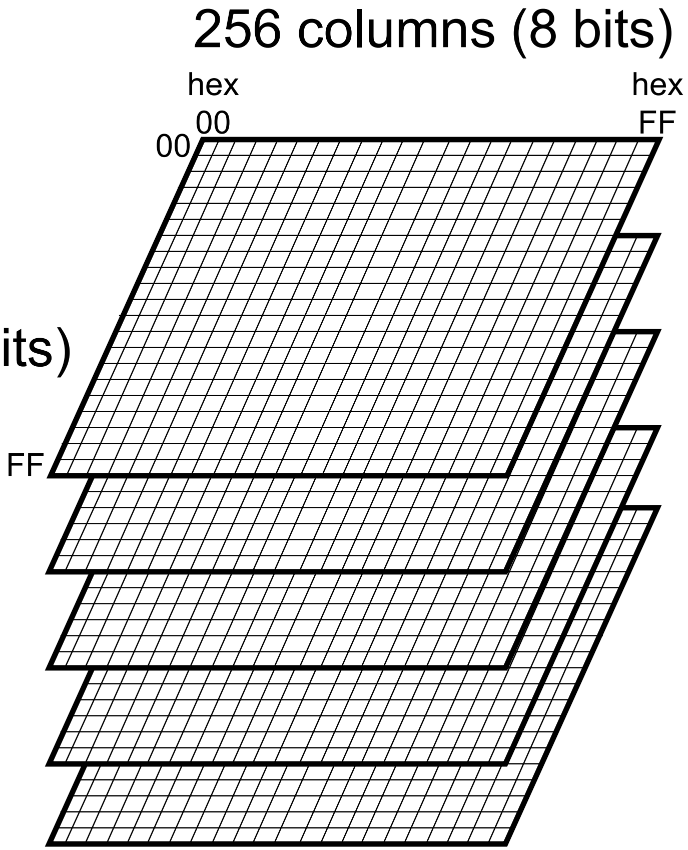
- In Unicode, the word “character” refers to the notion of the abstract form of a “letter”, in a very broad sense
  - a letter of an alphabet
  - a mark on a page
  - a symbol (in a language)
- A “glyph” is a particular rendition of a character (or composite character). The same Unicode character can be rendered by many glyphs
  - Character “a” in 12-point Helvetica, or
  - Character “a” in 16-point Times
- In Unicode each “character” has a name and a numeric value (called “code point”), indicated by U+hex value.
  For example, the letter “G” has:
  - Unicode name: “LATIN CAPITAL LETTER G”
  - Unicode value: U+0047 (see ASCII codes)

Unicode representation

- The Unicode standard has specified (and assigned values to) about 96.000 characters
- Representing Unicode characters (code points)
  - 32 bits in ISO-10646
  - 21 bits in the Unicode Consortium
- In the 21 bit address space, we can take the last 16 bits to address a “plane” of 64K characters (256 rows by 256 columns)
- The first five bits can then identify one of the 32 possible planes
- Only 6 planes defined as of today, of which only 4 are actually “filled”
- Plane 0, the Basic Multingual Plane, contains most of the characters used (as of today) by most of the languages present in the Web

The planes of Unicode

The first 5 bits of the Unicode "code point" identify 32 possible "planes".
Each plane can contain 64 K characters

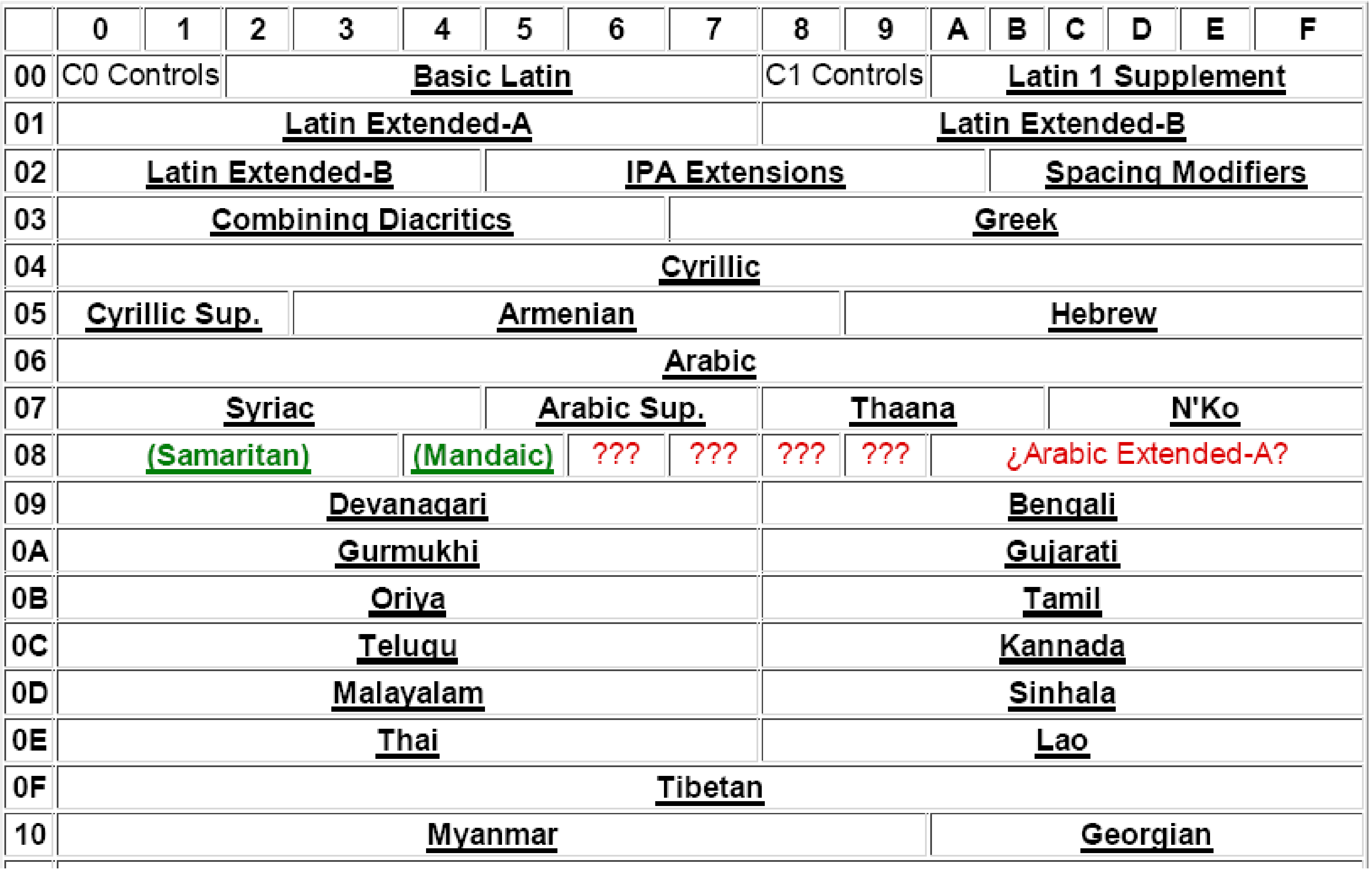


Unicode planes

Plane 0	Basic Multilingual Plane	U+00000 to U+0FFFF	modern languages and special characters. Includes a large number of Chinese, Japanese and Korean (CJK) characters.
Plane 1	Supplementary Multilingual Plane	U+10000 to U+1FFFF	historic scripts and musical and mathematical symbols
Plane 2	Supplementary Ideographic Plane	U+20000 to U+2FFFF	rare Chinese characters
Plane 14	Supplementary Special-purpose Plane	U+E0000 to U+EFFFF	non-recommended language tag and variation selection characters
Plane 15	Supplementary Private Use Area-A	U+F0000 to U+FFFFF	private use (no character is specified)
Plane 16	Supplementary Private Use Area-B	U+100000 to U+10FFFF	private use (no character is specified)

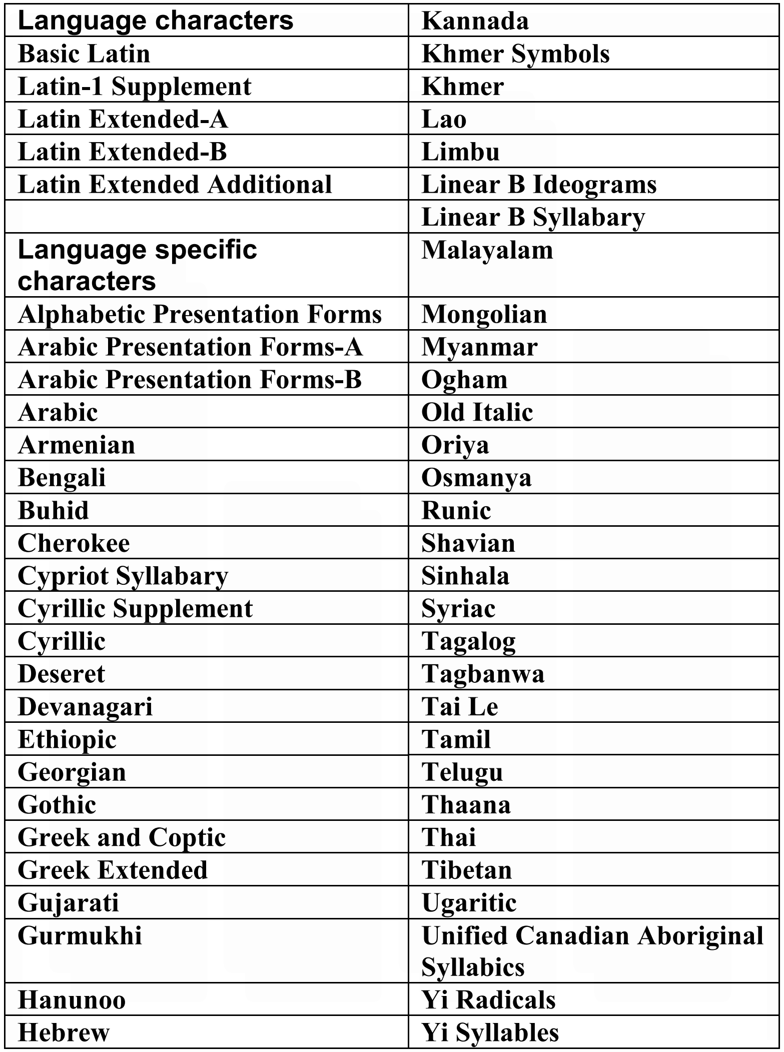
Beginning of BMP

in this table each “column” represents 16 characters

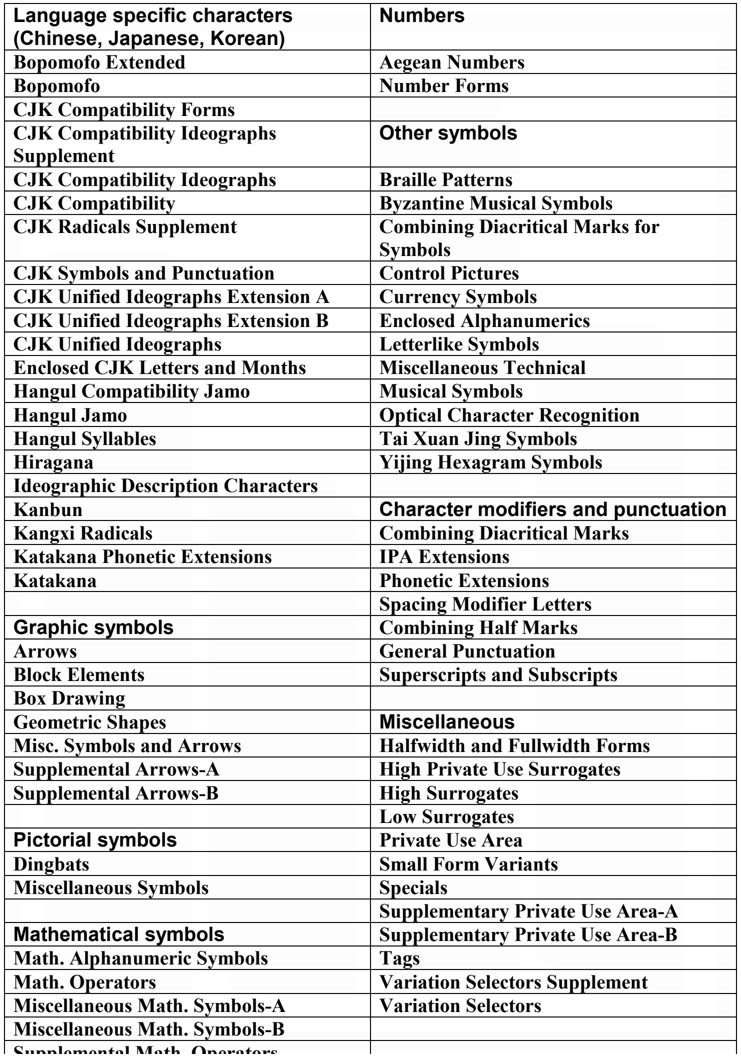


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	C0 Controls		Basic Latin						C1 Controls		Latin 1 Supplement					
01	Latin Extended-A								Latin Extended-B							
02	Latin Extended-B					IPA Extensions						Spacing Modifiers				
03	Combining Diacritics							Greek								
04	Cyrillic															
05	Cyrillic Sup.			Armenian						Hebrew						
06	Arabic															
07	Syriac					Arabic Sup.			Thaana				N'Ko			
08	(Samaritan)				(Mandaic)		???	???	???	???	¿Arabic Extended-A?					
09	Devanagari								Bengali							
0A	Gurmukhi								Gujarati							
0B	Oriya								Tamil							
0C	Telugu								Kannada							
0D	Malayalam								Sinhala							
0E	Thai								Lao							
0F	Tibetan															
10	Myanmar										Georgian					

Unicode charts



Language characters	Kannada
Basic Latin	Khmer Symbols
Latin-1 Supplement	Khmer
Latin Extended-A	Lao
Latin Extended-B	Limbu
Latin Extended Additional	Linear B Ideograms
	Linear B Syllabary
Language specific characters	Malayalam
Alphabetic Presentation Forms	Mongolian
Arabic Presentation Forms-A	Myanmar
Arabic Presentation Forms-B	Ogham
Arabic	Old Italic
Armenian	Oriya
Bengali	Osmanya
Buhid	Runic
Cherokee	Shavian
Cypriot Syllabary	Sinhala
Cyrillic Supplement	Syriac
Cyrillic	Tagalog
Deseret	Tagbanwa
Devanagari	Tai Le
Ethiopic	Tamil
Georgian	Telugu
Gothic	Thaana
Greek and Coptic	Thai
Greek Extended	Tibetan
Gujarati	Ugaritic
Gurmukhi	Unified Canadian Aboriginal Syllabics
Hanunoo	Yi Radicals
Hebrew	Yi Syllables



Language specific characters (Chinese, Japanese, Korean)	Numbers
Bopomofo Extended	Aegean Numbers
Bopomofo	Number Forms
CJK Compatibility Forms	
CJK Compatibility Ideographs Supplement	Other symbols
CJK Compatibility Ideographs	Braille Patterns
CJK Compatibility	Byzantine Musical Symbols
CJK Radicals Supplement	Combining Diacritical Marks for Symbols
CJK Symbols and Punctuation	Control Pictures
CJK Unified Ideographs Extension A	Currency Symbols
CJK Unified Ideographs Extension B	Enclosed Alphanumerics
CJK Unified Ideographs	Letterlike Symbols
Enclosed CJK Letters and Months	Miscellaneous Technical
Hangul Compatibility Jamo	Musical Symbols
Hangul Jamo	Optical Character Recognition
Hangul Syllables	Tai Xuan Jing Symbols
Hiragana	Yijing Hexagram Symbols
Ideographic Description Characters	
Kanbun	Character modifiers and punctuation
Kangxi Radicals	Combining Diacritical Marks
Katakana Phonetic Extensions	IPA Extensions
Katakana	Phonetic Extensions
	Spacing Modifier Letters
Graphic symbols	Combining Half Marks
Arrows	General Punctuation
Block Elements	Superscripts and Subscripts
Box Drawing	
Geometric Shapes	Miscellaneous
Misc. Symbols and Arrows	Halfwidth and Fullwidth Forms
Supplemental Arrows-A	High Private Use Surrogates
Supplemental Arrows-B	High Surrogates
	Low Surrogates
Pictorial symbols	Private Use Area
Dingbats	Small Form Variants
Miscellaneous Symbols	Specials
	Supplementary Private Use Area-A
Mathematical symbols	Supplementary Private Use Area-B
Math. Alphanumeric Symbols	Tags
Math. Operators	Variation Selectors Supplement
Miscellaneous Math. Symbols-A	Variation Selectors
Miscellaneous Math. Symbols-B	
Supplemental Math. Operators	

Unicode encoding

- UTF-32 (fixed length, four bytes)
  - UTF stands for “UCS Transformation Format” (UCS stands for “Unicode Character Set”)
  - UTF-32BE and UTF-32LE have a “byte order mark” to indicate “Big Endian” or ”Little Endian”
- UTF-16 (variable length, two bytes or four bytes)
  - All characters in the BMP represented by two bytes
  - The 21 bits of the characters outside of the BMP are divided in two parts of 11 and 10 bits; to each part is added an offset to bring it in the “surrogate zone” of the BMP (low surrogate at D800 and high surrogate at DC800)
  - in other words, they are represented as two characters in the BMP
  - UTF-16BE and UTF-16LE to indicate “endianness”
- UTF-8 (variable length, one to four bytes)
  - Characters in the 7-bit ASCII represented by one byte
  - Variable length encoding (2, 3 or 4 bytes) for all other characters

UTF-8

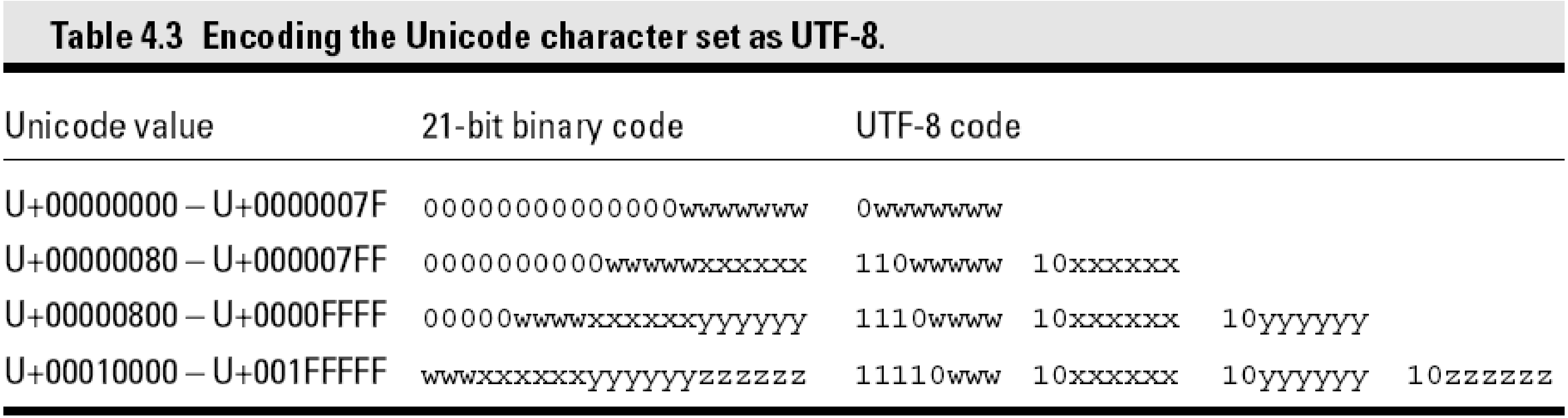
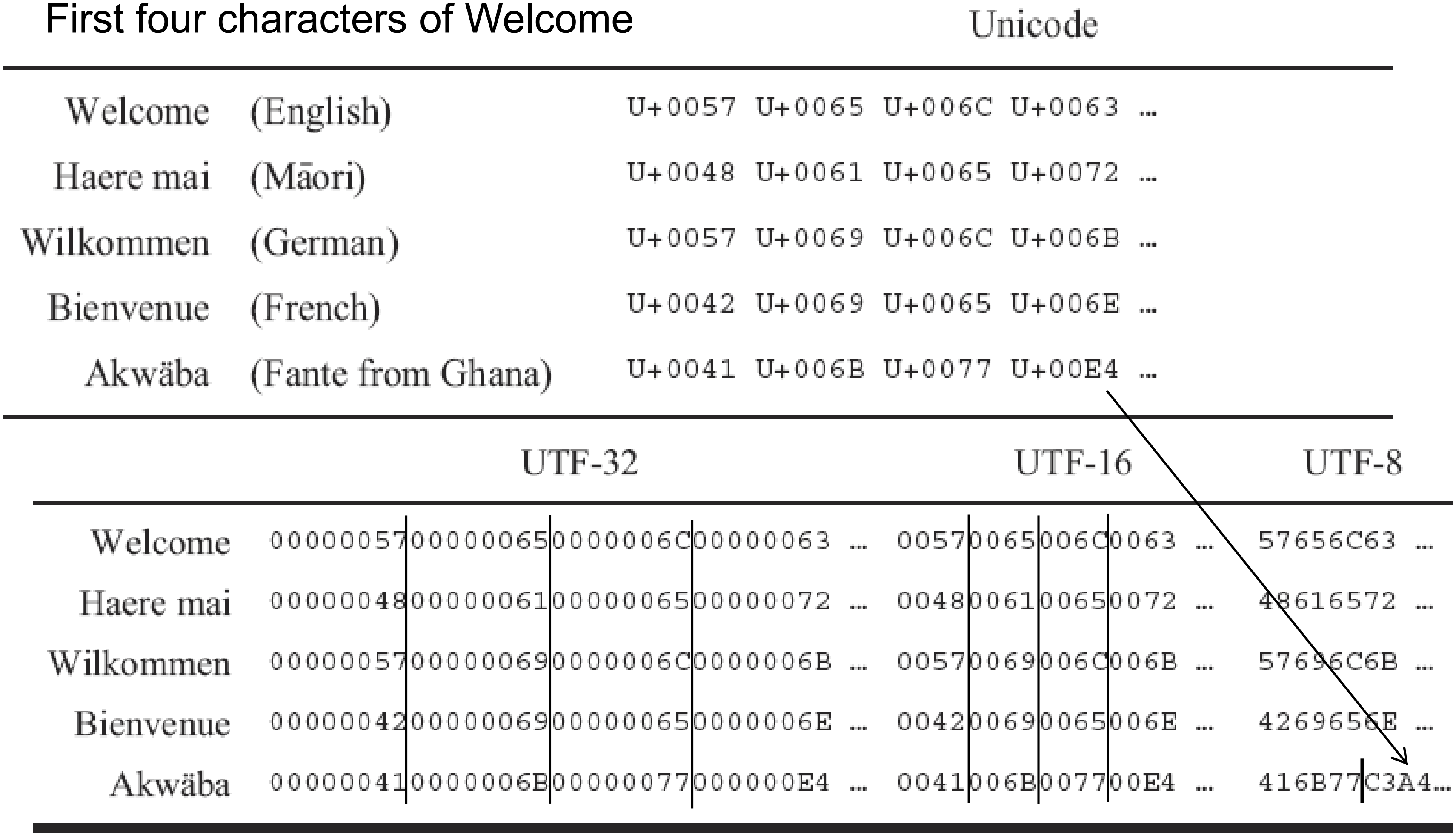


Table 4.3 Encoding the Unicode character set as UTF-8.

Unicode value	21-bit binary code	UTF-8 code			
U+00000000 – U+0000007F	00000000000000wwwwwww	0wwwwwww			
U+00000080 – U+000007FF	0000000000wwwwwxxxxxx	110wwwww	10xxxxxx		
U+00000800 – U+0000FFFF	00000wwwwxxxxxxyyyyyy	1110wwww	10xxxxxx	10yyyyyy	
U+00010000 – U+001FFFFF	wwwxxxxxxyyyyyyzzzzzz	11110www	10xxxxxx	10yyyyyy	10zzzzzz

Unicode example

First four characters of Welcome



		Unicode
Welcome	(English)	U+0057 U+0065 U+006C U+0063 …
Haere mai	(Māori)	U+0048 U+0061 U+0065 U+0072 …
Wilkommen	(German)	U+0057 U+0069 U+006C U+006B …
Bienvenue	(French)	U+0042 U+0069 U+0065 U+006E …
Akwäba	(Fante from Ghana)	U+0041 U+006B U+0077 U+00E4 …

	UTF-32	UTF-16	UTF-8
Welcome	00000057 00000065 0000006C 00000063 …	0057 0065 006C 0063 …	57656C63 …
Haere mai	00000048 00000061 00000065 00000072 …	0048 0061 0065 0072 …	48616572 …
Wilkommen	00000057 00000069 0000006C 0000006B …	0057 0069 006C 006B …	57696C6B …
Bienvenue	00000042 00000069 00000065 0000006E …	0042 0069 0065 006E …	4269656E …
Akwäba	00000041 0000006B 00000077 000000E4 …	0041 006B 0077 00E4 …	416B77 C3A4…

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images
- Video
- Audio

Representing documents

- Plain text
  - No information about structure
  - Different representation for line breaks
    - Windows represent a new line with the sequence “carriage return” followed by “line feed”
    - Unix and Apple/Mac represent a new line with “line feed” only

Knowledge of internal representation needed to extract text

- Page description languages
  - PostScript
  - PDF – Portable Document Format
- Word processors (text editors)
  - ODF – Open Document Format
  - RTF – Rich Text Format
  - Microsoft Word
  - LaTeX

Text editors

Editing of the contents

Editing of the format

- Mark-up languages
- WYSIWYG (What You See Is What You Get)

PostScript

- First commercially available page description language (Adobe 1985)
- It is a real programming language (variables, procedures, etc.) and a PostScript document is actually a “PostScript program”
- A page description comprises a number of graphical drawing instructions, including those that draw letters in a specific font in a specific size
  - Type-1 (Adobe) fonts versus TrueType (Apple)
- The document can be printed (or displayed) by having a “PostScript interpreter” executing the program
- The “abstract” PostScript description is converted to a matrix of dots (“rasterization” or “rendering”)
- PostScript initially designed for printing
  - Photo typesetters resolution up to 12000 dpi (dots per inch)
- PostScripts documents in a Digital Library
  - Extraction of text not always immediate
  - Digital Library must have a PostScript interpreter

PDF
Portable Document Format

- Successor to PostScript, to include good support for displays
- No longer a real programming language
- It defines an overall structure for a pdf document
  - Header, objects, cross-references, trailer
- Support for interactive display
  - Hierarchically structured content
  - Random access to pages
  - Navigation within a document
  - Support of hyperlinks
  - Support of “searchable images”
  - Limited editing capabilities

Representing documents

- Plain text
  - No information about structure
  - Different representation for line breaks
    - Windows represent a new line with the sequence “carriage return” followed by “line feed”
    - Unix and Apple/Mac represent a new line with “line feed” only

Knowledge of internal representation needed to extract text

- Page description languages
  - PostScript
  - PDF – Portable Document Format
- Word processors (text editors)
  - ODF – Open Document Format
  - RTF – Rich Text Format
  - Microsoft Word
  - LaTeX

Text editors

Editing of the contents

Editing of the format

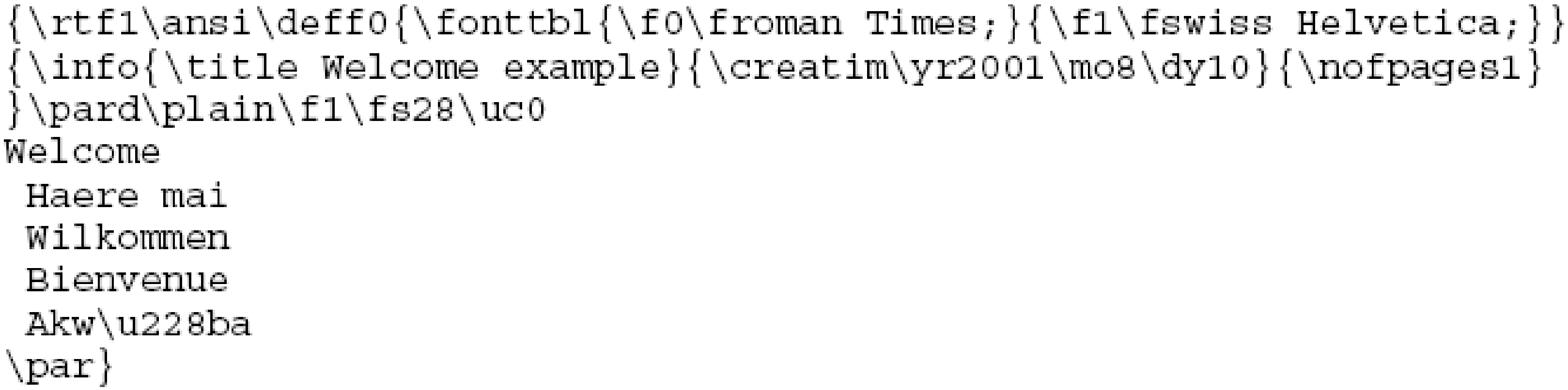
- Mark-up languages
- WYSIWYG
  (What You See Is What You Get)

ODF – Open Document Format

- An ODF document can be a single XML document or a set of XML documents compressed as a ZIP file
  - content.xml - main document content (text, tables, graphical elements)
  - meta.xml - information about the document (author, time of last save, ...)
  - styles.xml - styles that are used in the document
  - settings.xml - document and view settings
- The type of content can be specified with specific file extensions
  - .odt – a text document
  - .ods – a spreadsheet file
  - .odp – a presentation file
  - .odg – an illustration or graphic
- ODF documents can be opened with Microsoft Word, OpenOffice, LibreOffice, etc
- Text extraction is very easy

RTF – Rich Text Format

- Defined by Microsoft in 1987; discontinued in 2008
- Designed primarily to exchange documents among different word processors
- Description must allow a word processor to change “everything” (fonts, typesetting, tables, graphics, etc.)
- It defines an overall structure for a rtf document
  - Header, body



```
{\rtf1\ansi\deff0{\fonttbl{\f0\froman Times;}{\f1\fswiss Helvetica;}}
{\info{\title Welcome example}{\creatim\yr2001\mo8\dy10}{\nofpages1}
}\pard\plain\f1\fs28\uc0
Welcome
 Haere mai
 Wilkommen
 Bienvenue
 Akw\u228ba
\par}
```

Proprietary format, e.g. Word (.doc, .docx)

- Last published specification is that of Word 97
  - Many changes since then
- Internal binary format (more fast and more compact with respect to the other formats)
- Abstract document structure similar to rtf documents
- More rich in functionality, and therefore more complicated
- The “Fast Save” option does not preserve the order of the text
  - Edits are appended at the end of the document
- For text extraction the best alternative is to save in rtf (optionally also in html, but less convenient)

LaTeX

- Widely used in the scientific and mathematical communities
- Based on TeX, defined in the late seventies by Don Knuth, to overcome the limitations of the typesetters available at the time
- LaTeX documents are expressed in plain text, to expose all the details of the internal representation
  - Any text editor on any platform can be used to compose LaTeX document
  - Converted to a page description language (typically PostScript or PDF) to get the formatted document
- Simple document structure
  - Preamble to set the defaults and the global features
  - Structured (sections and subsections) document content
- Highly customizable with “external packages”
- Text extraction not so immediate
  - A single document may occupy several files
  - Possibility of “too much” customization

Representation of information within a computer

- Numbers
- Text (characters and ideograms)
- Documents
- Images ⬅
- Video
- Audio

Welcome

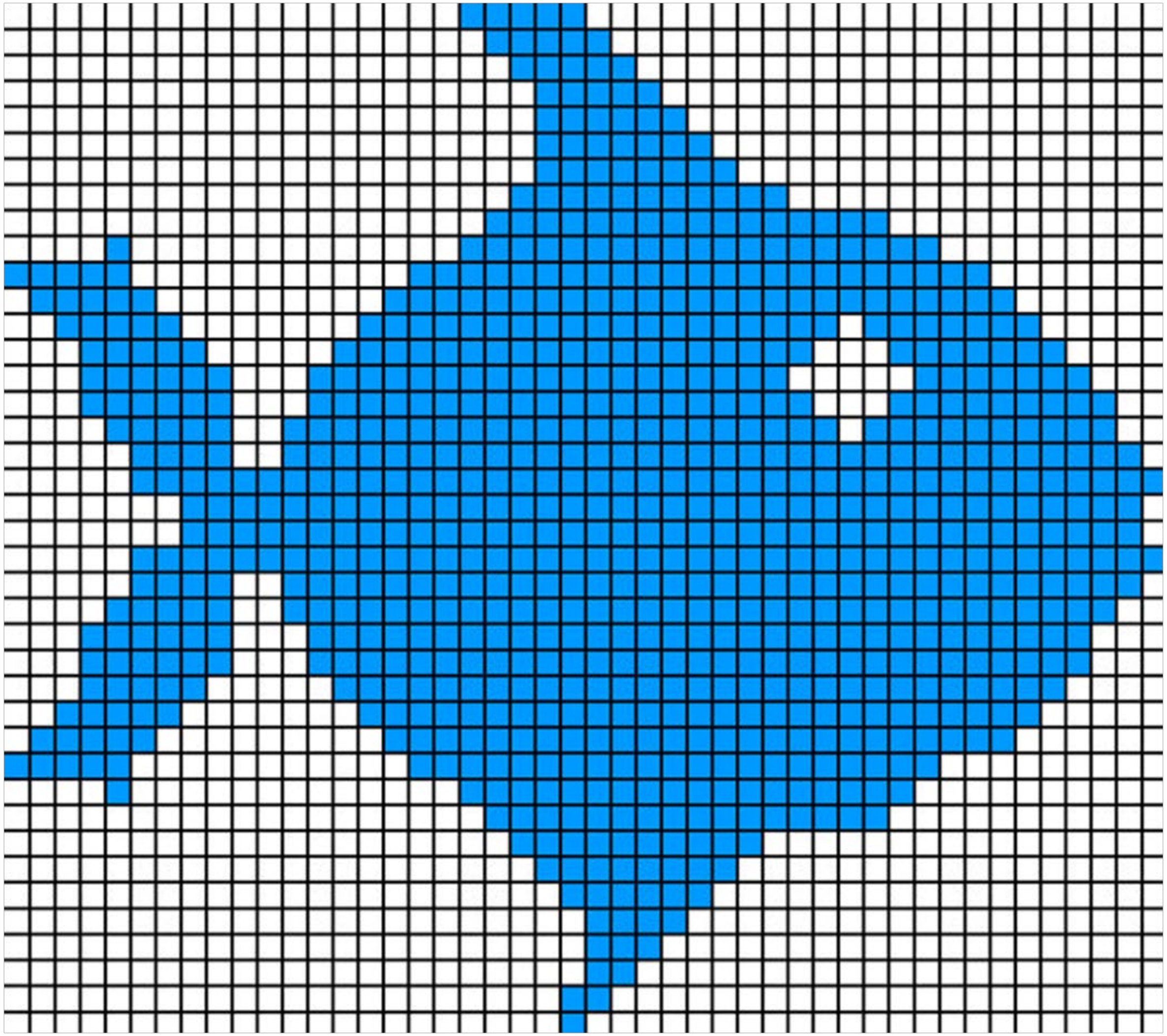
Welcome to image representation and compression

Representation of images

- Vector formats
  (geometric description with points, lines, polygons, etc))
  - EPS (Encapsulated PostScript)
  - PDF (Portable Data Format)
  - SVG (Scalable Vector Graphics)
  - WMF → EMF (Windows MetaFile, Enhanced MetaFile)
  - SWF (ShockWave Flash)

- Raster formats
  (array of “picture elements” called “pixels”)

Picture elements (pixels)



A pixel must be small enough so that its color can be considered uniform for the whole pixel.

Inside the computer, a pixel is represented with a number representing its color.

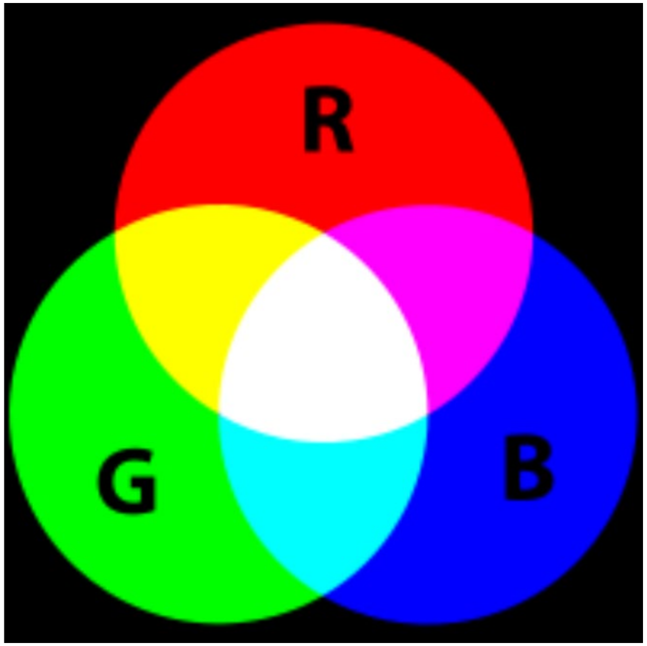
Raster format

- In raster format an image (picture) is represented by a matrix of “pixels”
- A first measure of the quality of a picture is given by the number of pixels, which can be measured in different ways
- Total number of pixels, as in digital cameras and phones
  - from 5-10 MegaPixels to 30-50 and plus MegaPixels
- Number of rows and columns of the matrix, like in TV or PC screens (columns by rows)
  - HDTV 1920x1080, 4K TV 3840 x 2160,
  - PC screen 1024x768, 1280x1024, 1920x1080
- Number of pixels in 1 inch (2,54 cm), called “dpi” (dots per inch)
  - 200-4800 dpi most common ranges

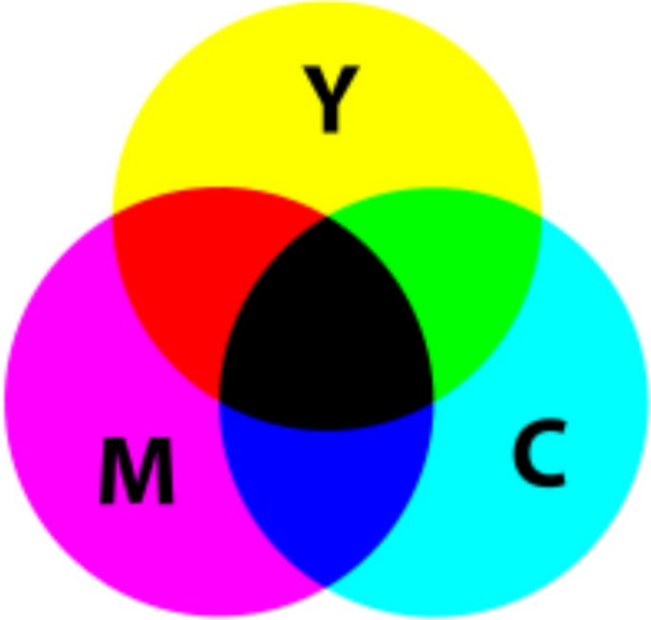
Raster format

- In raster format an image (picture) is represented by a matrix of "pixels"
- The quality of a picture is determined also by the number of bits used to represent one pixel (called depth)
  - 1 bit for black and white
  - 8-16 bits for grey scale (most common ranges)
  - 24-48 bits for color images (most common ranges)
- Usually colors are represented by three numbers, one for each "color component"
- Big file sizes for (uncompressed color) pictures
  - For example, one color page scanned at 600 dpi is about 100 MB

RGB and CMY color components



Additive color mixing



Subtractive color mixing