Bits, Bytes, and Representation of Information

Interpretation of bits depends on context

- **meaning of a group of bits depends on how they are interpreted**
- **1 byte could be**
  - 1 bit in use, 7 wasted bits (e.g., M/F in a database)
  - 8 bits storing a number between 0 and 255
  - an alphabetic character like W or + or 7
  - part of a character in another alphabet or writing system (2 bytes)
  - part of a larger number (2 or 4 or 8 bytes, usually)
  - part of a picture or sound
  - part of the location or address of something in memory
  - part of an instruction for a computer to execute
  - …

"part of an instruction for a computer to execute"

- instructions are just bits, stored in the same memory as data
- different kinds of computers use different bit patterns for their instructions
  - laptop, cellphone, game machine, etc., all potentially different
  - old powerPC CPU different from Intel CPU
- **one program's instructions are another program's data**
  - when you download a new program from the net, it's data
  - when you run it, it's instructions

Getting a binary representation of information

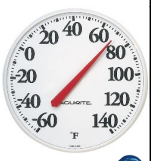
- **the usual sequence:**
  - something (sound, pictures, text, instructions, …) is converted into numbers by some mechanism
  - the numbers can be stored, retrieved, processed, transmitted
  - the numbers might be reconstituted into a version of the original
- **for sound, pictures, other real-world values**
  - make accurate measurements
  - convert them to numeric values

Encoding sound

- need to **measure intensity/loudness often enough and accurately enough** that we **can reconstruct it well enough**
- higher frequency = higher pitch
- **human ear can hear ~ 20 Hz to 20 KHz**
  - taking samples at twice the highest frequency is good enough (Nyquist)
- **CD audio usually uses**
  - 44,100 samples / second
  - accuracy of 1 in 65,536 (= 2^16) distinct levels
  - two samples at each time for stereo
  - data rate is 44,100 x 2 x 16 bits/sample
    = 1,411,200 bits/sec = 176,400 bytes/sec ~ 10.6 MB/minute
- **MP3 audio compresses by clever encoding and removal of sounds that won't really be heard**
  - data rate is ~ 1 MB/minute

Analog versus Digital

- **analog: "analogous" or "the analog of"**
  - smoothly or continuously varying values
  - volume control, dimmer, faucet, steering wheel
  - value varies smoothly with something else
    - no discrete steps or changes in values
    - small change in one implies small change in another
    - infinite number of possible values
  - the world we perceive is largely analog



- **digital: discrete values**
  - only a finite number of different values
  - a change in something results in sudden change from one discrete value to another
    digital speedometer, digital watch, push-button radio tuner, …
  - values are represented as numbers

Transducers

- **devices that convert from one representation to another**
  - microphone
  - loudspeaker / earphones
  - camera / scanner
  - printer / screen
  - mouse
  - touch screen
  - etc.
- **something is usually lost by conversion (in each direction)**
  - the ultimate copy is not as good as the original

Inherently Discrete values

- **another kind of conversion**
  - letters are converted into numbers when you type on a keyboard
  - the letters are stored (a Word document), retrieved (File/Open…), processed (paper is revised), transmitted (submitted by email)
  - printed on paper
- **letters and other symbols are <u>inherently discrete</u>**
- **encoding them as numbers is just assigning a numeric value to each one, without any intrinsic meaning**

Representing letters as numbers

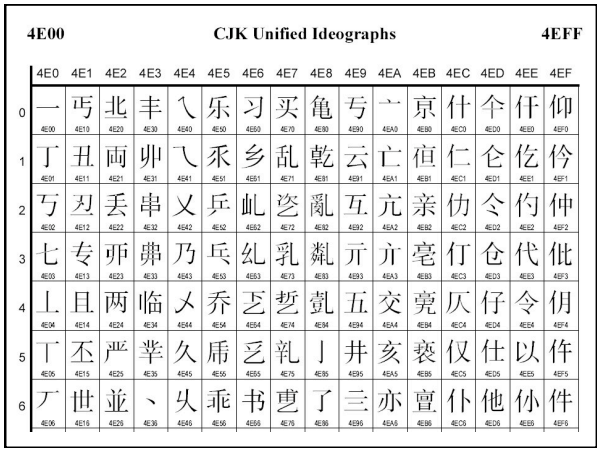
- **what letters and other symbols are included?**
- **how many digits/letter?**
  - determined by how many symbols there are
  - how do we disambiguate if symbols have different lengths?
- **how do we decide whose encoding to use?**
- **the representation is arbitrary**
- **but everyone has to agree on it**
  - if they want to work together

ASCII (hex encoding)

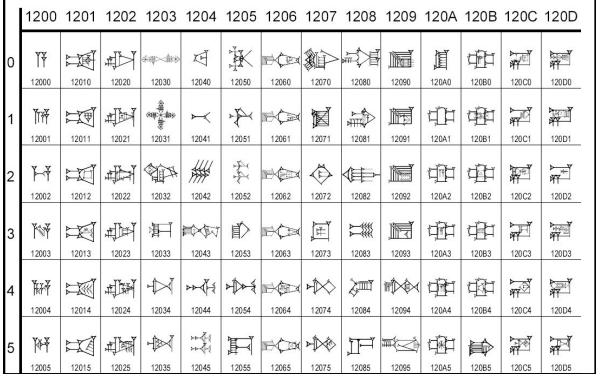
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d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

Other alphabets

- **what if we wanted Cyrillic or Hebrew instead of English?**
- **how do we handle a document with mixed Cyrillic and English?**
- **how do we interpret a string of digits?**
  - is this group of digits an English letter or a Cyrillic letter?
- **what if we wanted to include Chinese?**
  - how many digits might it now take?



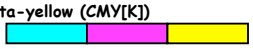
Cuneiform (from unicode.org)



Color

- **TV & computer screens use Red-Green-Blue (RGB) model**

- **each color is a combination of red, green, blue components**
  - R+G = yellow, R+B = magenta, B+G = cyan, R+G+B = white
- **for computers, color of a pixel is usually specified by three numbers giving amount of each color, on a scale of 0 to 255**
- **this is often expressed in hexadecimal so the three components can be specified separately (in effect, as bit patterns)**
  - 000000 is black, FFFFFF is white
- **printers, etc., use cyan-magenta-yellow (CMY[K])**



Things to remember

- **digital devices represent everything as numbers**
  - discrete values, not continuous or infinitely precise
- **all modern digital devices use binary numbers (base 2)**
  - instead of decimal (base 10)
- **it's all bits at the bottom**
  - a bit is a "binary digit", that is, a number that is either 0 or 1
  - computers ultimately represent and process everything as bits
- **groups of bits represent larger things**
  - numbers, letters, words, names, pictures, sounds, instructions, ...
  - the interpretation of a group of bits depends on their context
  - the representation is arbitrary; standards (often) define it
- **the number of digits used in the representation determines how many different things can be represented**
  - number of values = base $^{\text{number of digits}}$
  - e.g., 10^2 , 2^{10}