Representing Information

The Digital World

Computer Science studies the

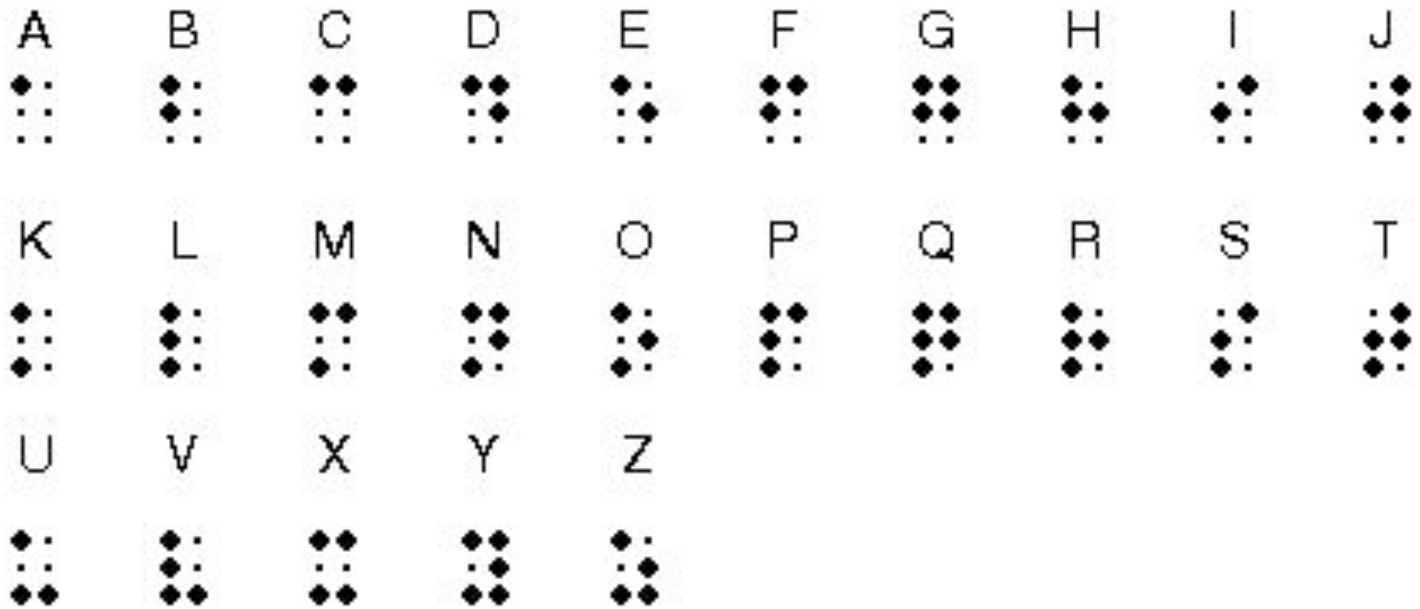
representation and manipulation of information

Writing :The First Software?

- First writing appeared in Sumer (present-day Iraq) about 5000 years ago.
- Spoken language encoded by a fixed repertory of symbols.
- Essence of the invention is the *coding scheme itself*, the physical medium (e.g., stone engraving, clay tablets, paper and ink) is irrelevant.

- It doesn't even matter what the symbols look like—the coding scheme is still the same.

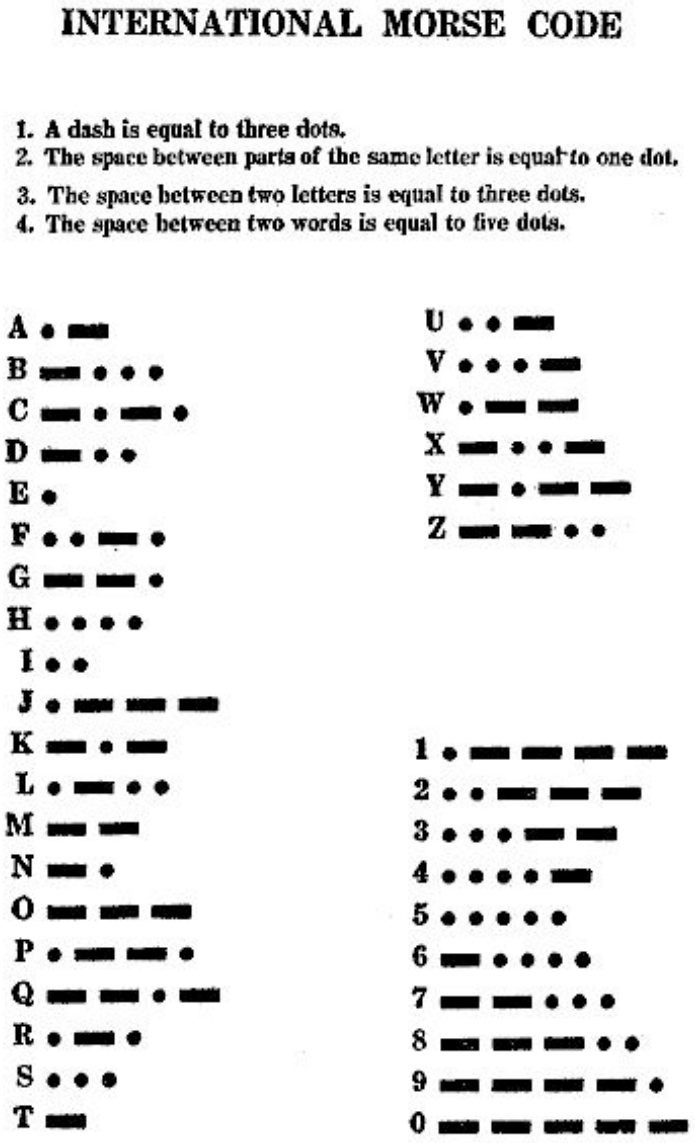
- The symbols don’t have to *look* like anything.



The Braille tactile alphabet for the blind

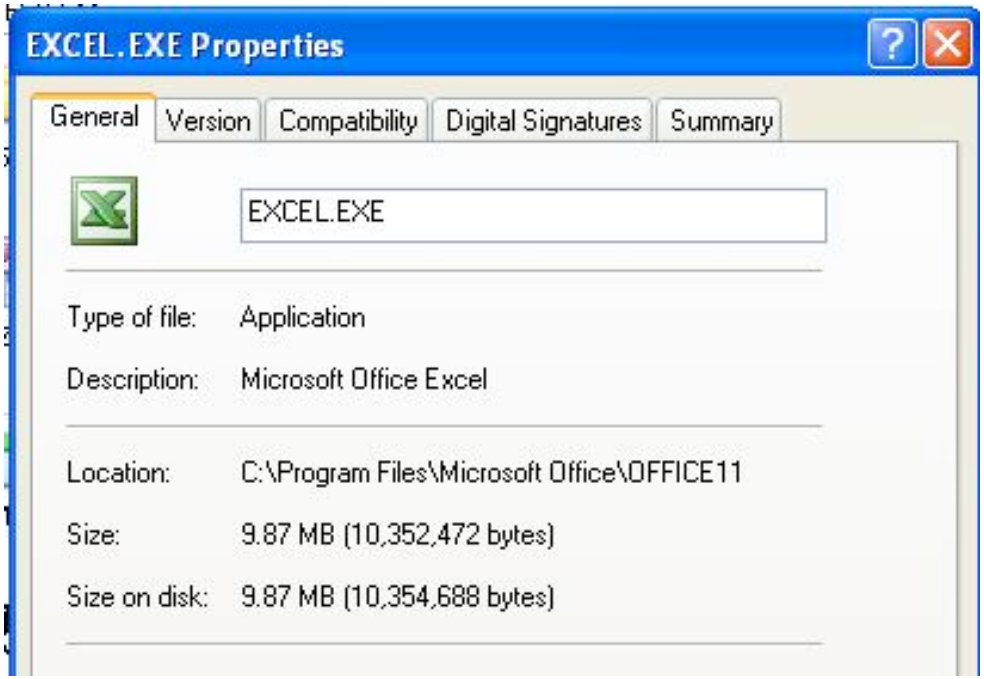
- The symbols don't have to *look* like anything.

- (Morse code, formerly used for telegraph communication, is audible, not visible.)



How do we *measure* information?

- In a computer file system, you measure it in units called ***bytes*** (what's that?)



Whatever they are, this one (the program code for Microsoft Excel) has a lot of them.

How do we *measure* information?

- In a computer file system, you measure it in units called ***bytes*** (what's that?)

My iPod has even more bytes(not as many as yours, though, I'll bet)

What's the *Smallest* Amount of Information Possible?

Attorney: Let's get you ready for your testimony. Do you know what color my dress is?

Client: It's blue.

Attorney: Wrong! My dress *is* blue, but the right answer is "Yes".

Bits

- Smallest unit of information distinguishes between two alternatives: on-off, yes-no, left-right.
- We write these alternatives 1-0.
- One bit (= Binary digIT)
- Computers encode **all** information (numbers, text, pictures, music,…) as sequences of bits .

We can't stress this point strongly enough!

- Computers encode **all** information (numbers, text, pictures, music,…) as sequences of bits .

We can't stress this point strongly enough!

- Your computer successfully creates the illusion that it contains photographs, letters, songs and movies. All it really contains is bits, lots of them, patterned in ways you can't see. Your computer was designed to store just bits---all the files and folders and different kinds of data are illusions created by computer programmers….We couldn't live without those more intuitive concepts, but they are artifices. Underneath, it's all just bits.

*---from **Blown to Bits** (2008), by Hal Abelson, Ken Ledeen and Harry Lewis*

We can't stress this point strongly enough!

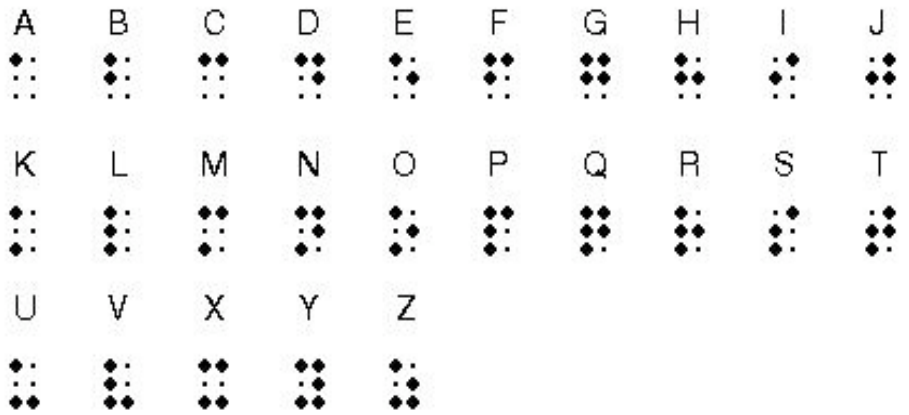
- 'Multimedia' is the wrong word. We have created a **unimedia**, really. Bits are bits.

---Nicholas Negroponte, founder of the MIT Media Lab, quoted in Newsweek Magazine (1993).

Bits and Bytes

- There are four (=2^2) different sequences of two bits (00,01,10,11), eight (2^3) different sequences of three bits, … In general, 2^n different sequences of n bits.
- Computers represent most information in blocks of eight bits. **One byte = eight bits**.
- There are 2^8=256 different possible values for a byte.

Braille Alphabet (early example of information encoded by bits)



- Braille tactile alphabet for the blind encodes each letter by six bits (raised dot =1, no raised dot = 0).
- $26<32=2^5$, so 5 bits would have sufficed, but the additional bit is put to use to represent other symbols.

Bits and bytes for humans

- It's hard to read something like 00101101 and tell at a glance that it is different from 00101011.
- A more readable way to write bits and bytes, called hexadecimal notation, uses a single letter or digit to represent each block of four bits.
- In hexadecimal, the two bytes above would be written 2D and 2B.

Hexadecimal Notation

- Human-readable binary: each block of four bits encoded by a single symbol (0,..,9,A,B,..,F)

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

Bytes and more bytes

- 1 *kilobyte* = 1KB = 2^{10} bytes = 1024 bytes (not 1000 bytes!).
- Likewise 1 *megabyte* = 1MB = 2^{20} bytes = 1,048,576 bytes.
- And 1 *gigabyte* = 1GB = 2^{30} bytes = 1,073,741,824 bytes.
- But watch out…1 Kb means 1 kilobit, and that's just 1000 bits (I think).

Bytes and more bytes

- The text of an average novel encoded in standard manner (what's that?) will fit into < 1 MB of storage.
- A compact disk holds about 700MB, enough for an hour of stereo audio.
- A DVD holds about 5GB, my laptop's hard drive more than 50 GB.
- But just *how* are text, sounds, movies, etc., represented by bytes?

Encoding Text with Bits

- ASCII (American Standard Code for Information Interchange)—each character encoded by a single byte.
- Ninety-five printable characters

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

encoded by byte values 20-8F (hex). [20 is a blank space.] Nonprinting characters (like tab) encoded by values less than 20.

Table of ASCII Values

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	□

- For example, “Boston” is 426F73746F6E in hexadecimal. If we write it out in individual bits, it’s 0100 0010 0110 1111 0111 0011 0111 0100 0110 1111 0110 1110

Encoding Text with Bits-Unicode

- **Unicode** uses a *two*-byte code to include a large number of international character sets.
- For normal Latin letters, the Unicode encoding is gotten by placing the byte 00 (hex) before the ASCII encoding.
- For instance, the Unicode encoding of ‘m’ is 006D (hex).

Encoding Text with Bits-Unicode

Some Greek, Russian and other letters, displaying Unicode encoding (from the Insert Symbol menu in Microsoft Word).