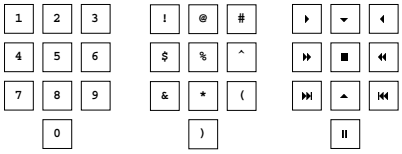
Digital Representation of Information

FIT 100

Digital encoding of information means the data is stored in discrete units -- effectively numbers. Text is represented using one byte for each of the keyboard characters

FIT 100 Phone Numbers, SSNs, ISBNs, ...

- Dictionary says “digitize” means represent something using the ten decimal digits … its more general
- Why use digits for phone numbers?
  - Numerical properties unimportant
  - Must only specify the sequence for pressing buttons
  - Digits are familiar, have short names
- Adopt any symbols and relabel the buttons

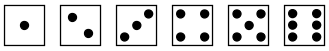
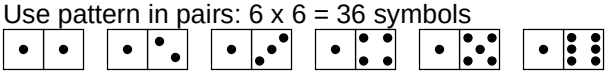
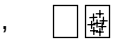


Standard Shift Player

800 325 3535

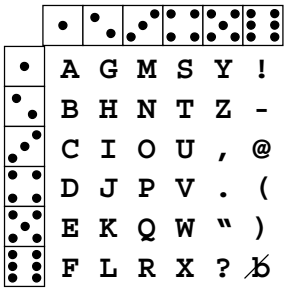
*)) #@% #%#%

FIT 100 Patterns to Symbols

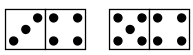
- A die's patterns can make symbols
  - Use pattern once: 6 symbols 
  - Use pattern in pairs: 6 x 6 = 36 symbols  ...
  - Use pattern in triples: 6 x 6 x 6 = 216
  - In general, using m patterns in sequences of n: m^n
- Fundamental pattern (*PandA*): The presence or absence of a phenomenon at a specific place/time
  - It's there or not -- Light: on/off, Water: flow/still,  Magnetism: charged/neutral, Checkbox marked/empty, Door: closed/open, …
  - The states must be discrete -- distinguishable, unambiguous
  - The two states represent one *bit* of information

FIT 100 Representing Information

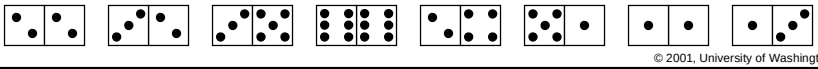
- Keyboard characters can be represented exactly
- Imagine you and your friend are prohibited from talking -- its too noisy? -- and so you use dice to encode the letters and punctuation to communicate



With two dice encode the 26 Latin letters and punctuation (spell numbers). So,

 is UW

Order matters: row then col!



FIT 100 Bits and Bytes

- It's customary to name the two possible patterns of a bit 1 and 0, though any names would be OK
- Sequences of 8 bits create a *byte*
- Two patterns in sequences of 8 …
  m = 2, n = 8, 2^8 = 256 possibilities
  from 0000 0000 through 1111 1111
- The two pattern alternatives motivate the term *binary* for this representation

Names for Patterns

Present	Absent
On	Off
Yes	No
1	0
True	False
+	-
Black	White
For	Against
Yin	Yang
Lisa	Bart
…	...

FIT 100 Character Representations

- Keyboard characters are encoded into a byte or two
- ASCII is one of many byte encodings of characters

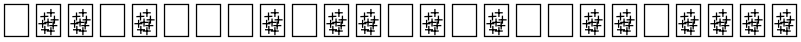
ASCII	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0010		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
1000																
1001																
1010		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

ASCII, pronounced AS·key, stands for American Standard Code for Information Interchange

A is represented 0100 0001
B is represented 0100 0010
C is represented 0100 0011
...

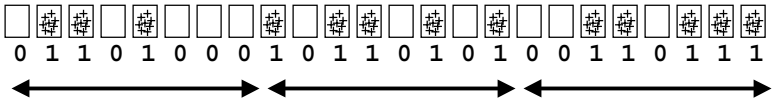
FIT 100 Storing Text

- Information is often stored by charge or magnetic field



Schematic diagram of magnetic spots, say on a disk

- Its presence or absence can be detected, leading to a natural association with 1 and 0 to charged/neutral states (alternatively, plus-charged / not plus charged)



- Text is stored as a sequence of keyboard characters

FIT 100 Embellishing Text

- Often, text is to have specific properties, e.g be printed in a specific font, be italic, etc.
- To distinguish the text from the modifiers that describe its properties, tag the modifiers
  - A tag is a text string, <tag> or </tag>, that modifies text
  - Pairs of tags surround the tagged text, e.g. <title>Gone with the Wind</title>
  - The “opening” and “closing” tags differ in that the close is indicated by a slash
  - Not all tags have a “match”
- Software *interprets* the tags when the text is being processed, e.g. printed or displayed as a web page

FIT 100 Numbers

- In addition to text computers must store numbers
- Numbers are sometimes stored as text characters:

0100 0110 0100 1001 0101 0100 0011 0001 0011 0000 0011 0000

F I T 1 0 0

- Mostly numbers are stored directly using *binary notation*, since it has only two digits, 0 and 1
- Binary numbers and arithmetic are very much like decimal, except restricted to the two digits 0 and 1 … what number is 1100100?

0
1
10
11
100
101
110
111
1000

Binary is counting on your fists instead of your fingers

FIT 100 Decimal and Binary

	Decimal	Binary
Symbols:	0, 1, …, 9	0, 1
Base	10	2
Number xyz	$x\cdot10^2+y\cdot10^1+z\cdot10^0$	$x\cdot2^2+y\cdot2^1+z\cdot2^0$
Ex: 101	$1\cdot10^2+0\cdot10^1+1\cdot10^0$	$1\cdot2^2+0\cdot2^1+1\cdot2^0$
Place Value	101	5
Powers	1, 10, 100, 1000, …	1, 2, 4, 8, 16, 32, 64, ...

Binary works just like decimal, except that the base is 2

- What binary numbers are: 1000_2, 1010_2 and 1111_2

Use a subscript to indicate the number base, e.g $5_{10} = 101_2$

2^3
2^2
2^1
2^0

FIT 100 Adding Is Familiar

- To add in binary use the same technique (algorithm), as decimal but restrict yourself to 0 and 1 … everything else works the same way

```
1 1       1              1      <-- Carries
  1 1 0 0 1 1 (2)          51
  1 1 1 0 1 0 (2)          58
1 1 0 1 1 0 1 (2)         109
```

- Binary is pretty tedious for humans because there are so many digits … circuitry benefits however because it uses the two states (on/off) efficiently

FIT 100 Summary

- Patterns are used to create symbols, symbols are used to represent information
- The binary digits (bits) 0 and 1 are a natural way to interpret the presence or absence of a phenomenon
- Bytes are composed of 8 bits, ASCII represents text as one character per byte
- Binary numbers and arithmetic are like decimal except they are limited to the two numerals 0 and 1
- Tags are used to insert modifiers into text and keep it separated from the text