Data Representation

Kelly Rivers and Stephanie Rosenthal

15-110 Fall 2019

Announcements

- Add Deadline
- Homework 1 Check-in was due on Monday

  How was it?
- Homework 1 is due Monday at 12-noon!

  Start Early! You can already do more than half of the questions.

Recap – Unit 1

Monday – Algorithms

Wednesday – Programming Basics

Friday – Computer Organization

Today – Data Representation

Friday – Programming Functions

Abstraction

What steps do we do to manipulate that data?
How do we program our computers?
How do we represent our data?


ON
OFF
ON
OFF
ON
OFF

How do we use 0/1s to make everything we see and do on computers?

1

0

Making Change

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

8 cents in the fewest coins

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

Q	D	N	P

8 cents in the fewest coins

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

Q	D	N	P
0	0	1	3

1 nickel + 3 pennies =
1 x (5 cents) + 3 x (1 cent) =
5 cents + 3 cents =
8 cents

36 cents in the fewest coins

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

Q	D	N	P

36 cents in the fewest coins

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

Q	D	N	P
1	0	1	1

1 quarter + 1 dime + 1 penny =
1 x (25 cents) + 1 x (10 cents) + 1 x (1 cent) =
25 cents + 10 cents + 1 cents =
36 cents

Piazza Poll

Penny
1 cent

Nickel
5 cents

Dime
10 cents

Quarter
25 cents

Q	D	N	P
1	1	0	1

What is your algorithm for making change?

1. Use as many quarters as possible that's less than the total amount
2. Use as many dimes as possible that's less than the remaining amount
3. Use as many nickels as possible that's less than the remaining amount
4. Use the number of pennies equal to the remaining amount

New Money System

Penny
1 cent

Dime
10 cents

Dollar coin
100 cents

$	D	P

198 cents in the fewest coins

Penny
1 cent

Dime
10 cents

Dollar coin
100 cents

$	D	P

198 cents in the fewest coins

Penny
1 cent

Dime
10 cents

Dollar coin
100 cents

$	D	P
1	9	8

1 dollar + 9 dimes + 8 pennies =
1 x (100 cents) + 9 x (10 cents) + 8 x (1 cent) =
100 cents + 90 cents + 8 cents =
198 cents

Decimal Number System

$$100 = 10^2$$
$$10 = 10^1$$
$$1 = 10^0$$

"Base 10"

100	10	1
$	D	P
1	9	8

Decimal Number System

$$100 = 10^2$$
$$10 = 10^1$$
$$1 = 10^0$$

"Base 10"

100	10	1
$	D	P
1	9	8

What is the highest number that any column can be?

Newer Money System

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

G	B	C	I

How many cents?

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

G	B	C	I
1	0	1	1

How many cents?

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

8	4	2	1
1	0	1	1

How many cents?

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

8	4	2	1
1	0	1	1

1 gold + 1 copper + 1 iron =
1 x (8 cents) + 1 x (2 cents) + 1 x (1 cent) =
8 cents + 2 cents + 1 cent =
11 cents

How many cents?

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

8	4	2	1
0	1	1	0

How many cents?

Iron
1 cent

Copper
2 cents

Bronze
4 cents

Gold
8 cents

8	4	2	1
0	1	1	0

1 bronze + 1 copper =
1 x (4 cents) + 1 x (2 cents) =
4 cents + 2 cents =
6 cents

Binary Number System

$8 = 2^3$
$4 = 2^2$
$2 = 2^1$
$1 = 2^0$

8	4	2	1
0	1	1	0

"Base 2"

What is the highest number that any column can be?

Computers are Binary

1

0

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	1

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	0

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	1	1	0

Counting in Binary

128	64	32	16	8	4	2	1
0	0	0	0	0	1	1	1

Converting to Decimal

128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	1

Converting to Decimal

128	64	32	16	8	4	2	1
1	0	0	1	0	0	0	1

Converting to Binary

3 6

Converting to Binary

36

128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	0

Converting to Binary

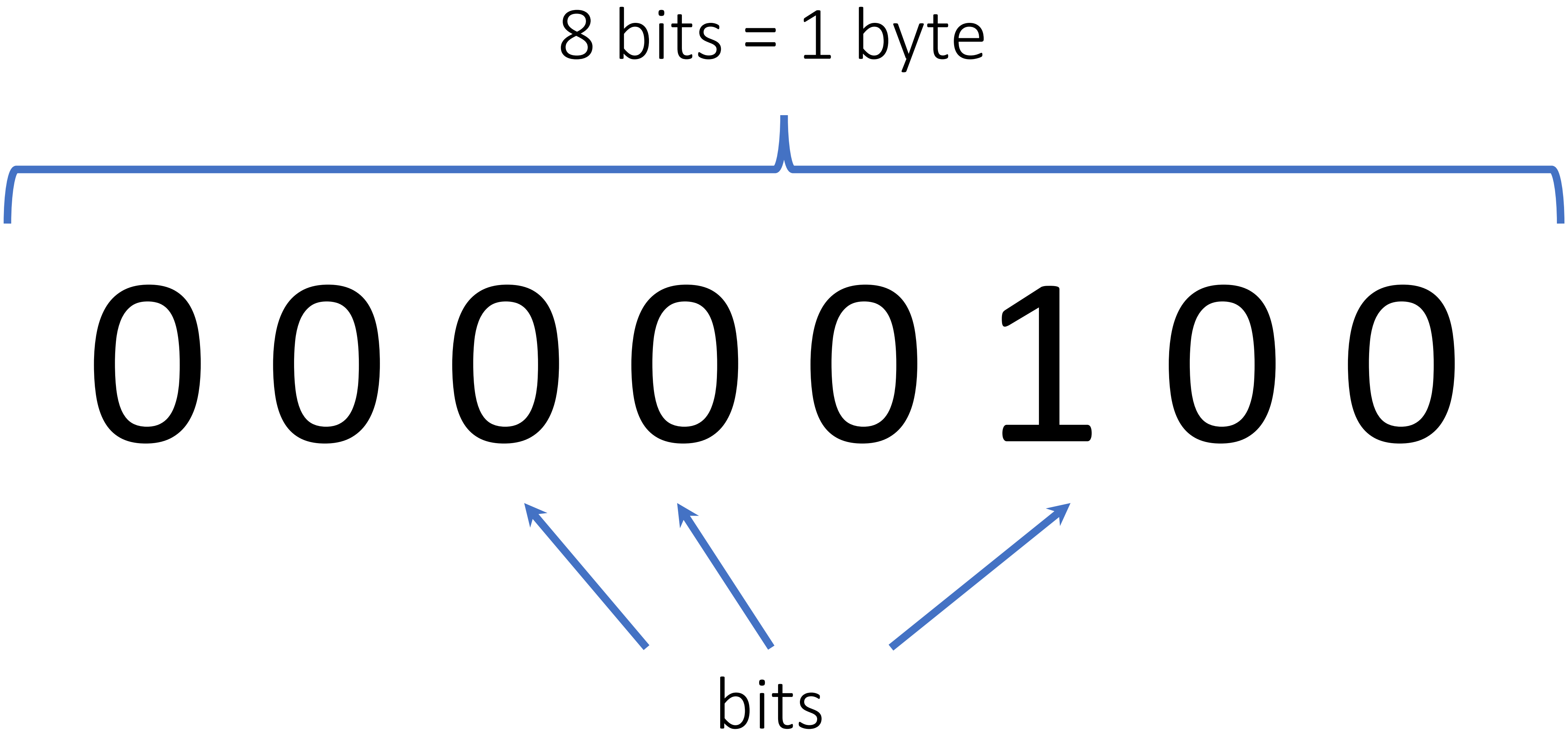
1 0 4

Converting to Binary

104

128	64	32	16	8	4	2	1
0	1	1	0	1	0	0	0

Bits and Bytes



Maximum Value of a Byte

8 bits = 1 byte

1 1 1 1 1 1 1 1

bits

Hexadecimal (Base 16)

4 bits = ½ byte | 4 bits = ½ byte

1 1 0 1 0 0 1 1

16 values: 0,1,2,...,9,A,B,C,D,E,F

Hexadecimal

4 bits = ½ byte   4 bits = ½ byte

1 1 0 1 0 0 1 1

D   3

Scale

Wifi 600 Mbit/s = 600 million bits every second
NES Game 8kB = 8000 bytes = 64000 bits
iPhone X 256 GB ~= 256 billion bytes
Google 15 exabytes = 15 billion GB

Abstraction - Everything is Bits

Integers
Letters and Symbols
Pixels / Colors
Computer Instructions
Locations in Computer Memory
Computer Addresses
Real Numbers

Integers

Idea: make 1 bit a negative sign

-	64	32	16	8	4	2	1
1	1	0	1	0	0	1	1

Integers

Idea: make 1 bit a negative sign

-	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

Integers

Idea: make 1 bit a negative sign

-	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

vs

1	0	0	0	0	0	0	0

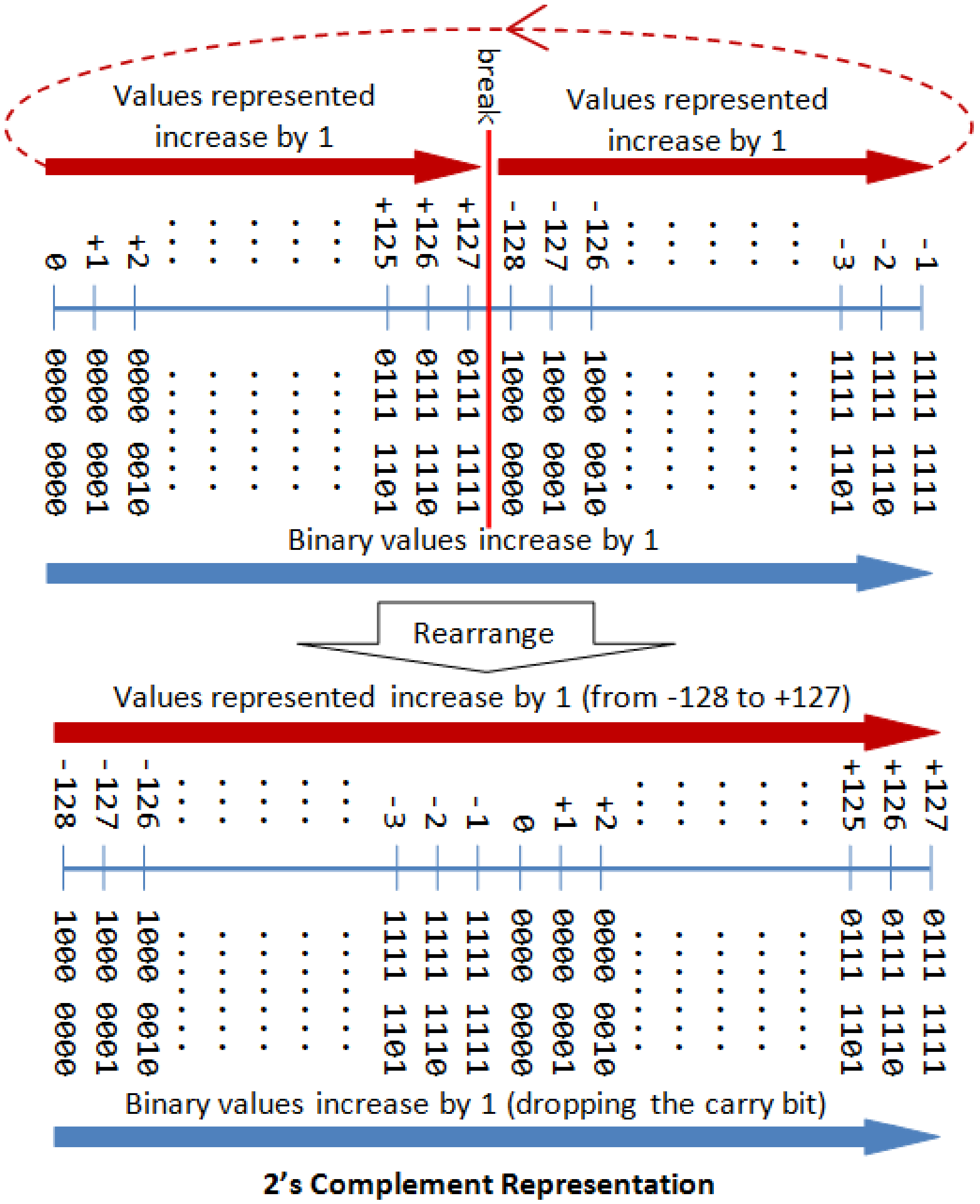
Integers – 2's Complement

Goal of 2's Complement:

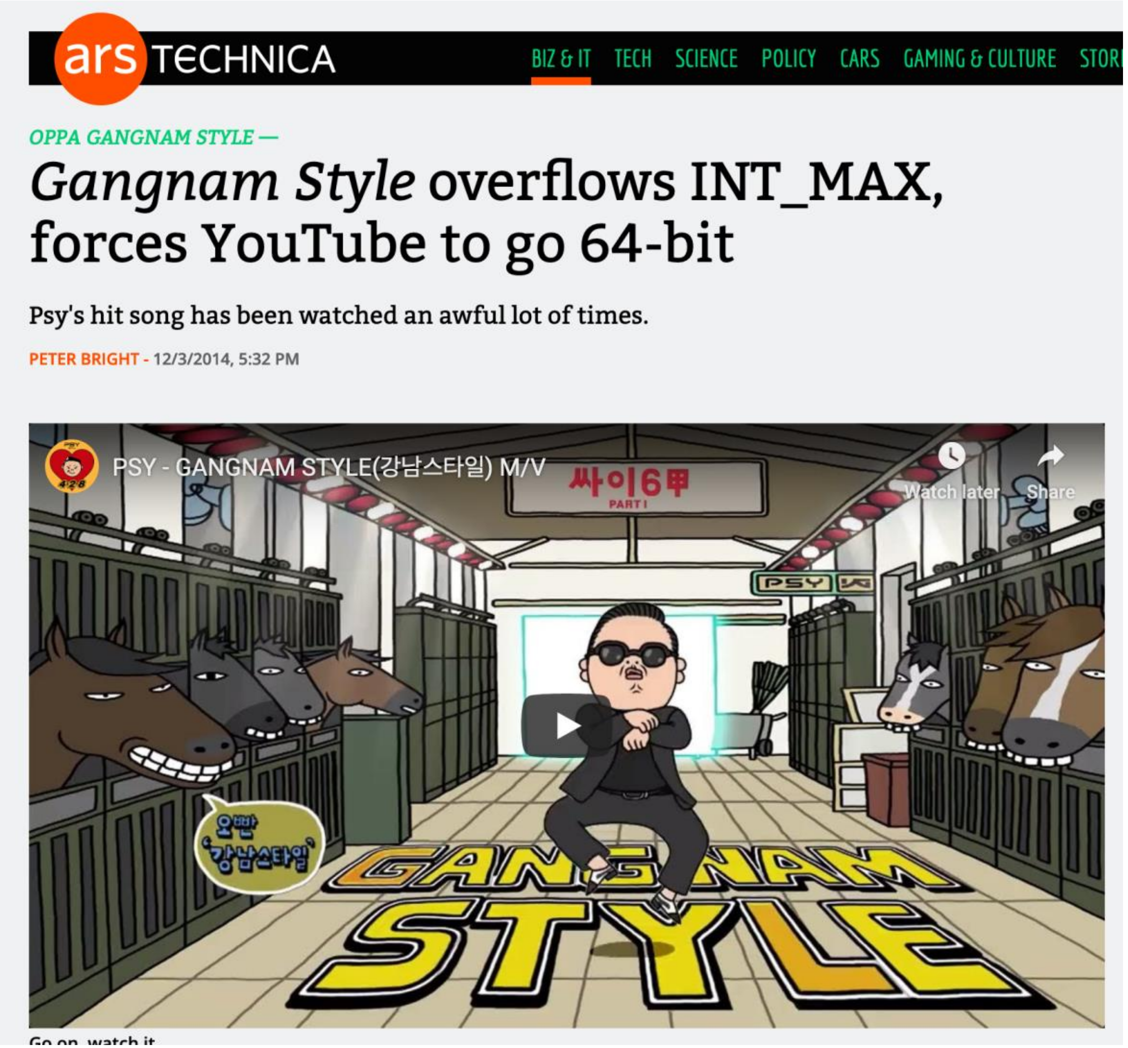
Avoid having +0 and -0

Algorithm for computing negative #'s from binary representation:

1) If positive, whole number in binary
2) If negative:
    a) flip all the bits
    b) compute the whole number
    c) multiply by -1 and subtract 1



How many bytes in an Integer?



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STOR

OPPA GANGNAM STYLE —

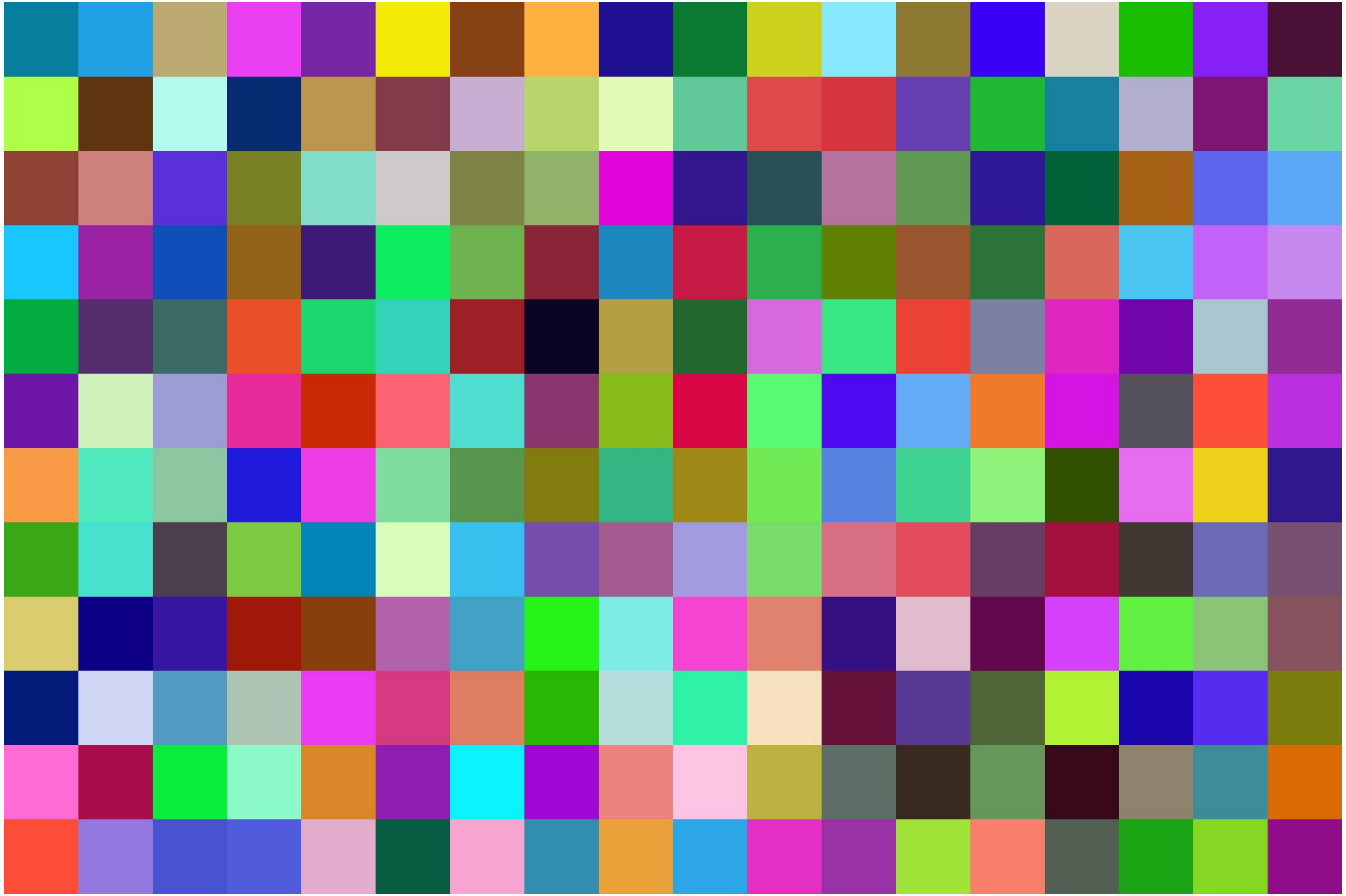
***Gangnam Style* overflows INT_MAX, forces YouTube to go 64-bit**

Psy's hit song has been watched an awful lot of times.

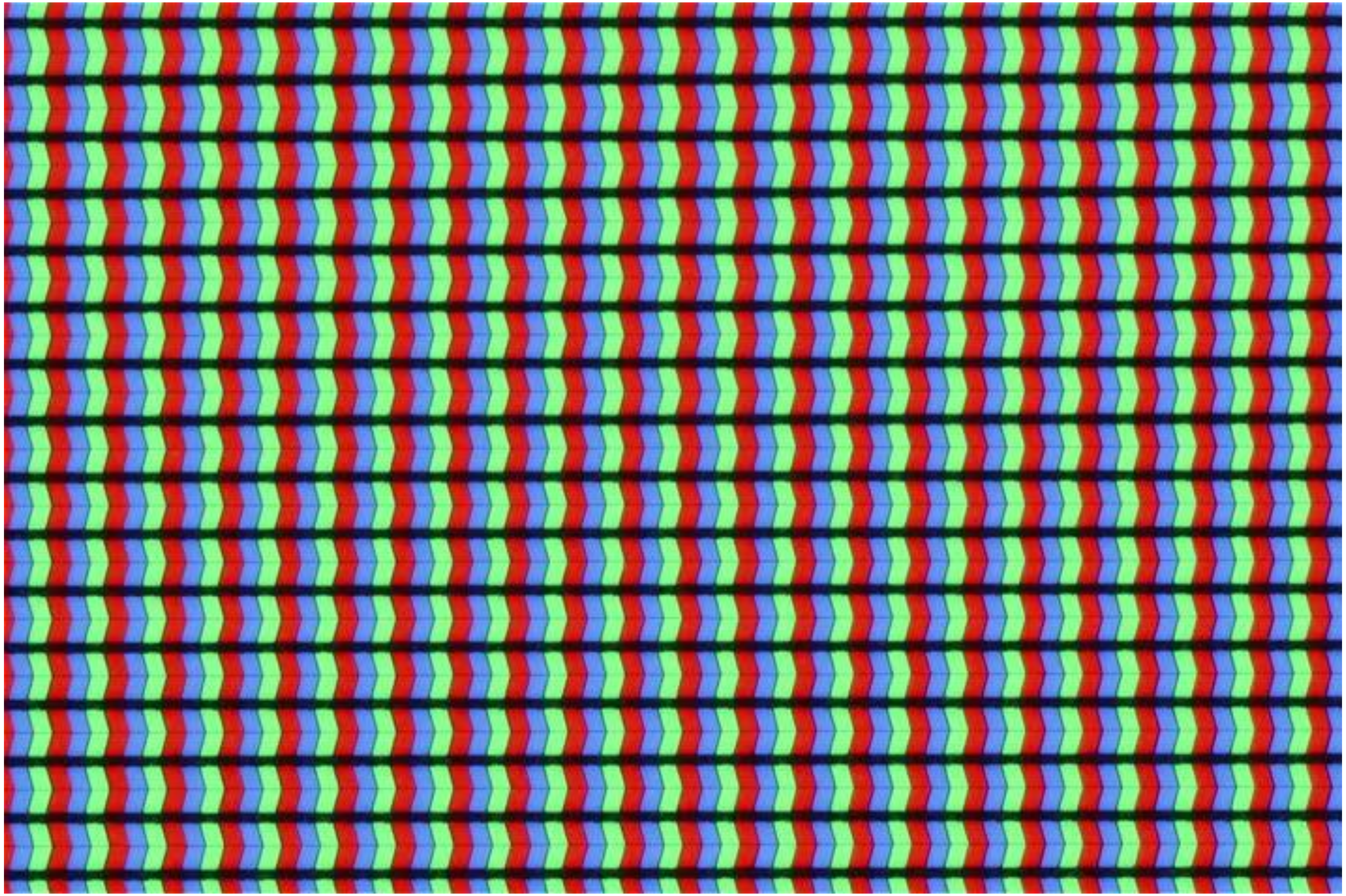
PETER BRIGHT - 12/3/2014, 5:32 PM

Go on, watch it

Pixels



Pixels - RGB



Pixels


GRAY = 1 SET OF DIGITS
11111111
11100110
11001101
10110100
10011011
01110011
01010000
00101000
00000000
'RGB' = 3 SETS OF DIGITS
11111111 00000000 00000000
01100110 01100110 11111111
00110011 11001100 10011001
11111111 11111111 01100110
11111111 00000000 11001100
00110011 11001100 11111111
00110011 00110011 00000000
00110011 00110011 10011001
11111111 10011001 10011001
'CMYK' = 4 SETS OF DIGITS
00000000 11000101 10111000 00000000
01000000 00111001 00000000 00000000
01010010 00000000 00110110 00000000
00000000 00000000 00111100 00000000
00000000 01001010 00000000 00000000
01010011 00000000 00000100 00000000
01001100 00111110 01011100 00110110
01100000 01011010 00000000 00000000
00000000 00110010 00011010 00000000

Pixels – How many bytes in an RGB pixel?

Abstraction

Need to know how many bytes and what to do with them:

- Integers
- Pixels
- Real Numbers
- Locations in Computer Memory
- IP Addresses (internet computer addresses)
- …

Need a look-up table (dictionary) to decode:

- Letters
- Computer Code
- Image Files
- Music Files
- …

Letters - ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters - ASCII

"15-110" =

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters - ASCII

"15-110" =

0011 0001
0011 0101
0010 1101
0011 0001
0011 0001
0011 0000

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters - ASCII

"Happy" =

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters - ASCII

"Happy" =

0100 1000
0110 0001
0111 0000
0111 0000
0111 1001

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters – How many bytes for 1 letter?

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
32	20	[SPACE]	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	[DEL]

Letters – Unicode (2-6 bytes)

0000000001000001 = 65 = "A" - 16 bits

Basic Latin - 32-126
More Latin - 16-669 ĀĎĠķŊè
Greek and Coptic - 880-1023 ΓΔΘΣπε

0010000000111101 - 8253 - ‽
1111111011111000 - 65272 - ﻸ

11110000100111111001100010000100 - 😄
11110000100111111001001010101001 - 💩

Computer Instructions, Bytecode, etc

Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	And memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and Save retum address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMS	7200		Complement AC
CME m	7100	e	Comp
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E is 0
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrup
IOF	F040		Inter

How does a computer know to use ASCII, Unicode, Integers, Code?

Everything is Bits

Integers
Letters and Symbols
Pixels / Colors
Computer Instructions
Locations in Computer Memory
Computer Addresses
Real Numbers