# Data Representation

15-110 – Friday 09/03

#### **Announcements**

Eberly Study Information

- No class or office hours next Monday
  - Enjoy Labor Day!

- Check1 due next Wednesday at noon
  - Don't forget about Piazza and Office Hours!
  - **Demo:** how to submit files on Gradescope

#### **Muddiest Points**

- Hard to keep up with new terminology
  - Try having the slides open during lecture! Download from <a href="https://www.cs.cmu.edu/~110/schedule.html">https://www.cs.cmu.edu/~110/schedule.html</a> before lecture starts.
  - If we don't define a term in the slides, let us know so we can add it

- How to tell apart different error types
  - We'll learn more about errors in Week 3
  - For now, just try using the debugging approach we introduced

# Learning Objectives

Understand how different number systems can represent the same information

Translate binary numbers to decimal, and vice versa

 Interpret binary numbers as abstracted types, including colors and text

# Number Systems

#### Computers Run on 0s and 1s

Computers represent everything by using 0s and 1s. You've likely seen references to this before.

How can we represent text, or images, or sound with 0s and 1s? This brings us back to **abstraction**.

#### Abstraction is About Representation

Recall our definition of abstraction from the first lecture:

**Abstraction** is a technique used to make complex systems manageable by changing the amount of detail used to **represent** or interact with the system.

We'll use abstraction to translate 0s and 1s to decimal numbers, then translate those numbers to other types.

# Number Systems – Coins

A **number system** is a way of representing numbers using symbols.

One example of a number system is currency. In the US currency system, how much is each of the following symbols worth?









Penny 1 cent

Nickel 5 cents

Dime 10 cents

Quarter 25 cents

#### Number Systems – Dollars

Alternatively, we can represent money using **dollars and cents**, in decimal form.

For example, a medium coffee at Tazza is \$2.65.



#### Converting Coins to Dollars

We can **convert between number systems** by translating a value from one system to the other.

For example, the coins on the left represent the same value as \$0.87

Using pictures is clunky. Let's make a new representation system for coins.



# Coin Number Representation

To represent coins, we'll make a number with four digits.

The first represents quarters, the second dimes, the third nickels, and the fourth pennies.

$$c.3.1.0.2 =$$

$$3*$0.25 + 1*$0.10 + 0*$0.05 + 2*$0.01 =$$

\$0.87













	Q	D	N	P
С	3	1	0	2

#### Converting Dollars to Coins

In recitation, you created an algorithm to convert money from dollars to coins, minimizing the number of coins used.

How did your algorithm work?

# **Conversion Example**

What is \$0.59 in coin representation?

$$$0.59 = 2*$0.25 + 0*$0.10 + 1*$0.05 + 4*$0.01 = c.2.0.1.4$$

# **Activity: Coin Conversion**

You do: Now try the following calculations:

What is c.1.1.1.2 in dollars?

What is \$0.61 in coin representation?

# Number Systems - Decimal

When we work with ordinary numbers outside of any specific context, we usually use the **decimal** number system.

Moving from the right, the first digit is the number of 1s, the second is 10s, the third is 100s, etc. Each digit represents a **power of 10**. For example, 1980 in decimal is 1 \* 1000 + 9 \* 100 + 8 \* 10 + 0 \* 1

But this isn't the only abstract number system we can use!

1000 <b>10</b> 3	10 <sup>2</sup>	10 <sup>1</sup>	10° <b>1</b>
1	9	8	0

# Number Systems – Binary

We can represent numbers using only 0s and 1s with the **binary number** system.

Instead of counting the number of 1s, 5s, 10s, and 25s in coins, or 1s, 10s, 100s, and 1000s in abstract amounts, count the number of 1s, 2s, 4s, and 8s. For example, 1101 in binary is 1\*8+1\*4+0\*2+1\*1.

Why these numbers? They're **powers of 2**. This is a number in **base 2**, which only needs the digits 0 and 1.

<sup>23</sup> 8	<sup>22</sup> <b>4</b>	<sup>21</sup> <b>2</b>	<sup>20</sup> <b>1</b>
1	1	0	1

#### Bits and Bytes

When working with binary and computers, we often refer to a set of binary values used together to represent a number.

A single binary value is called a **bit**.

A set of 8 bits is called a **byte**.

We commonly use some number of **bytes** to represent data values.

#### **Counting in Binary**

$$O = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$2 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{4} = \begin{bmatrix} \mathbf{128} & \mathbf{64} & \mathbf{32} & \mathbf{16} & \mathbf{8} & \mathbf{4} & \mathbf{2} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \end{bmatrix}$$

$$6 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$1 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$3 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$5 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$7 = \begin{bmatrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

# **Converting Binary to Decimal**

To convert a binary number to decimal, just add each power of 2 that is represented by a 1.

For example, 00011000 = 16 + 8 = 24

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

Another example:

$$10010001 = 128 + 16 + 1 = 145$$

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

# Converting Decimal to Binary

Converting decimal to binary uses the **same process** as converting dollars to coins.

Look for the largest power of 2 that can fit in the number and subtract it from the number. Repeat with the next power of 2, etc., until you reach 0.

For example, 36 = 32 + 4 = 00100100

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

Another example:

$$103 = 64 + 32 + 4 + 2 + 1$$

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

# **Activity: Converting Binary**

You do: Now try converting numbers on your own.

First: what is **01011011** in decimal?

Second: what is **75** in binary?

# **Abstracted Types**

# Binary and Abstraction

Now that we can represent numbers using binary, we can represent everything computers store using binary.

We just need to use **abstraction** to interpret bits or numbers in particular ways.

Let's consider numbers, images, and text.

#### Discussion: Representing Negative Numbers

It can be helpful to think logically about how to represent a value before learning how it's done in practice. Let's do that now.

**Discuss:** We can convert binary directly into positive numbers, but how do we represent negative numbers?

# **Answer: Representing Negative Numbers**

**Simple Approach**: reserve one bit to represent whether the number is positive or negative. Convert the rest normally.

**Actual Approach:** mathematically, X + (-X) = 0. Set up the negative binary so that when it is added to the binary of the positive version of the number, the result is 0. Do this by **restricting the number of bits** that will be considered to a preset amount (perhaps 8).

The value 11111111 is 255, but it is also -1 because 11111111 + 1 = 100000000, which becomes 00000000 if we only have 8 bits. 11111110 is -2 (or 254), and so on.

How do we decide whether 11111111 is 255 or -1? It all depends on our **interpretation**. Is the integer signed (possibly negative) or unsigned (definitely positive)?

#### Size of Integers

Your machine is either classified as 32-bit or 64-bit. This refers to the size of integers used by your computer's operating system.

The largest signed integer that can be represented with N bits is  $2^{N}-1$  (why?). This means that...

Largest int for 32 bits: 4,294,967,295 (or 2,147,483,647 with negative numbers)

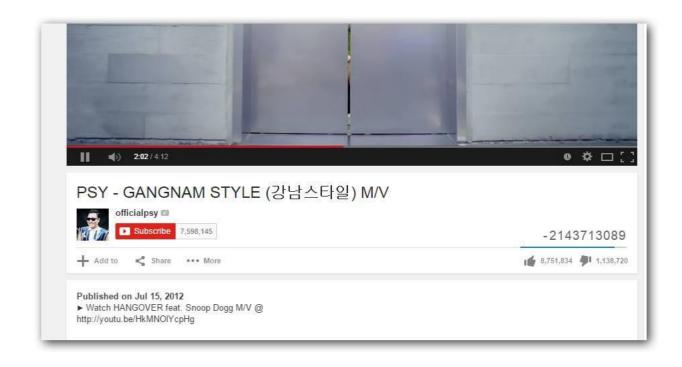
Largest int for 64 bits: 18,446,744,073,709,551,615 (18.4 quintillion)

# Integer Overflow

Why does this matter?

By late 2014, the music video Gangnam Style received more than **2 billion** views. When it passed the largest positive number that could be represented with 32 bits, YouTube showed the number of views as **negative** instead!

Now YouTube uses a 64-bit counter instead.



#### Represent Images as Grids of Colors

What if we want to represent an image? How can we convert that to numbers?

First, break the image down into a grid of colors, where each square of color has a distinct hue. A square of color in this context is called a **pixel**.

If we can represent a pixel in binary, we can interpret a series of pixels as an image.



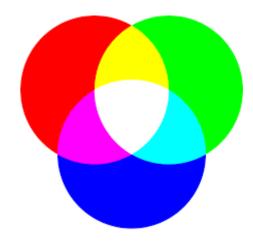
# Representing Colors in Binary

We need to represent a single color (a pixel) as a number.

There are a few ways to do this, but we'll focus on **RGB**. Any color can be represented as a combination of Red, Green, and Blue.

Red, green, and blue intensity can be represented using one **byte** each, where 00000000 (0) is none and 1111111 (255) is very intense. Each pixel will therefore require 3 bytes to encode.

Try it out here: w3schools.com/colors/colors\_rgb.asp



#### Represent Text as Individual Characters

Next, how do we represent text?

First, we break it down into smaller parts, like with images. In this case, we can break text down into individual **characters**.

For example, the text "Hello World" becomes H, e, l, l, o, space, W, o, r, l, d

# Use a Lookup Table to Convert Characters

Unlike colors, characters don't have a natural connection to numbers.

Instead, we can use a **lookup table** that maps each possible character to an integer.

As long as every computer uses the same lookup table, computers can always translate a set of numbers into the same set of characters.

# ASCII is a Simple Lookup Table

For basic characters, we can use the encoding system called ASCII. This maps the numbers 0 to 255 to characters. Therefore, one character is represented by one byte.

Check it out here: www.asciitable.com

Dec Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0 0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	
1 1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
<b>2</b> 2	002	Start of Text	34	22	042	"	n	66	42	102	B	В	98	62	142	b	b
<b>3</b> 3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	C
4 4	004	<b>End of Transmission</b>	36	24	044	\$	\$	68	44	104	D	D	100	64	144	d	d
<b>5</b> 5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
<b>6</b> 6		Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
<b>7</b> 7	007	Bell	39	27	047	'	1	71	47	107	G	G	103	67	147	g	g
88	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
<b>9</b> 9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
<b>10</b> A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
<b>11</b> B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12 C	014	Form feed	44	2C	054	,	1	76	4C	114	L	L	108	6C	154	l	1
<b>13</b> D	015	Carriage return	45	2D	055	-	_	77	4D	115	M	M	109	6D	155	m	m
<b>14</b> E		Shift Out	46	2E	056	.		78	4E	116	N	N	110	6E		n	n
<b>15</b> F	017	Shift In	47	2F	057	/	/	79	4F	117	O	0	111	6F	157	o	0
<b>16</b> 10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	р
<b>17</b> 11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
<b>18</b> 12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
<b>19</b> 13	023	Device Control 3	51	33	063	3	3	83		123	S	S	115	73	163	s	5
20 14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21 15		Negative Ack.	53			5	5	85	55		U	U	117			u	u
22 16		Synchronous idle	54		066	6	6		56	126	V	V	118		166	v:	V
23 17		End of Trans. Block	55	37		7	7	87	57	127	W	W	119	77	167	w	w
24 18	030	Cancel	56		070	8	8	88		130	X	X	120		170	x	X
<b>25</b> 19	031	End of Medium	57	39	071	9:	9	89	59	131	Y:	Y	121	79		y	V
26 1A	032	Substitute	58		072	:		90	5A	132	Z	Z	122			z	Z
27 1B		Escape	59			;:	;		5B		[	1	123			{	{
28 1C		File Separator	60			<	<		5C		\	\	124				Ì
29 1D		Group Separator	61			=	=		5D		]	1	125	7D		}	}
30 1E		Record Separator	62			>	>		5E		^	^	126			~	~
31 1F		Unit Separator	63			?	?		5F		_		127				Del
Samuel Ar (1976)						,						_				asciichars	com

# **Translating Text to Numbers**

"Yay" =
01011001 -> 89
01100001 -> 97
01111001 -> 121

0 0 000 NULL 32 0 040   Space 64 40 100 @ @ 96 60 140 ` \\ 1 1 001 Start of Header 33 21 041 ! ! 65 41 101 A A 97 61 141 a a \\ 2 2 002 Start of Text 34 22 042 " " 66 42 1012 B B 98 62 142 b b \\ 3 3 03 End of Text 35 23 043 # # 67 43 103 C C 99 63 143 c c \\ 4 4 004 End of Transmission 36 24 044 \$ S 68 44 104 D D 100 64 144 d e \\ 6 6 006 Acknowledgment 38 26 045 & & 70 46 106 F F 102 66 145 e e \\ 6 6 006 Acknowledgment 38 26 046 & & 70 46 106 F F 102 66 146 f f \\ 7 7 007 Bell 39 27 047 ' " 71 47 107 G G 103 67 147 g g \\ 8 8 010 Backspace 40 28 050 ( ( 72 48 110 H H 104 68 150 h h \\ 9 9 011 Horizontal Tab 41 29 051 ) ) 73 49 111 I I 105 69 151 i i \\ 10 A 012 Line feed 42 2A 052 * * 74 4A 111 J J 106 6A 152 j j \\ 11 B 013 Vertical Tab 43 28 053 + + 75 48 113 K K 107 6B 153 k k \\ 12 C 014 Form feed 44 2C 054 , \\ 13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m \\ 14 E 016 Shift Out 46 2E 056 Ɣ \\ 15 F 017 Shift In 47 2F 057 Ɨ \\ 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 111 67 155 o o \\ 18 12 02 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 160 o o \\ 19 13 023 Device Control 3 51 33 063 3 3 88 55 125 V U 117 75 165 o u \\ 21 15 025 Negative Ack, 53 35 065 7 7 87 57 127 W W 119 77 16 o u \\ 22 16 026 Synchronous idle 54 36 066 6 6 88 55 125 V U 117 75 165 o u \\ 23 16 035 Group Separator 60 3C 074 < > 94 55 133 ] 1 125 70 175  \\ 24 18 031 End of Medium 57 39 071 < > 94 55 133 ^ 1 127 77 177  \\ 25 10 035 Group Separator 61 30 075 < > 94 55 133 ^ 1 127 77 177  \\ 25 16 037 Unit Separator 63 3F 077 B > 94 55 133 ^ 1 127 77 177  \\ 25 16 037 Unit Separator 63 3F 077 B > 94 55 133 ^ 1 127 77 177 \\ 25 16 037 Unit S	Dec Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
2 2 002 Start of Text	0 0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	,
3 3 003 End of Text	<b>1</b> 1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
4 4 004 End of Transmission  36 24 044 \$ \$ 68 44 104 D D 100 64 144 d d 5 5 005 Enquiry  37 25 045 % % 69 45 105 E E 101 65 145 e e 6 6 006 Acknowledgment  38 26 046 & & 70 46 106 F F 102 66 146 f f 7 7 007 Bell  39 27 047 ' ' 71 47 107 G G 103 67 147 g g 8 8 010 Backspace  40 28 050 ( ( 72 48 110 H H 104 68 150 j i 10 A 012 Line feed  42 2A 052 * * 74 4A 112 J J 106 6A 152 j i 11 B 013 Vertical Tab  43 2B 053 + + 75 4B 113 K K 107 6B 153 k k 12 C 014 Form feed  44 2C 054 , , 76 4C 114 L L 108 6C 154 l I 13 D 015 Carriage return  45 2D 055 - - 77 4D 115 M M 109 6D 155 m I 14 E 016 Shift Out  46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape  48 30 060 0 0 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 7 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 30 68 5 5 85 55 125 X U 117 75 165 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 X U 117 75 165 s u 21 15 025 Negative Ack. 53 35 065 7 7 87 57 127 X U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 7 7 87 57 127 X U 117 75 165 n u 24 18 030 Cancel 55 38 073 ; 9 91 58 133 [ [ 123 78 173 z X 27 18 033 Escape 59 38 073 ; 9 91 58 133 [ [ 123 78 173 z { 28 1C 034 File Separator 60 3C 074 B > 92 5C 134 \ \ 124 7C 174     29 10 035 Group Separator 61 28 076 B > 94 5E 136 ^ \ 1 126 7E 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 176 ~ \ 177 ~ \ 177 ~ \ 177 ~ \ 177 ~ \ 177 ~ \ 177  \ 177 ~ \ 177 ~ \ 177 ~ \ 177 ~ \ 177 	<b>2</b> 2	002	Start of Text	34	22	042	"	n	66	42	102	B	В	98	62	142	b	b
5 5         005         Enquiry         37 25         045         %         %         69 45         105         E         E         101         65         145         e         e           6 6         006         Acknowledgment         38         26         046         &         &         70         46         06         F         F         102         66         146         f         f           7 7 007         Bell         39         27         047         '         '         71         47         107         G         G         103         67         147         g         g           8 8         010         Backspace         40         28         050         (         (         72         48         110         H         H         104         68         150         h         h           10 A         011         Horizontal Tab         41         29         051         )         )         73         49         111         H         K         105         69         151         h         h           11 B         013         Vertical Tab	<b>3</b> 3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	C
6 6       006 Acknowledgment       38 26       046 &       & 70 46       106 F       F       102 66       146 f       f         7 7       007 Bell       39 27 047 '       71 47       107 G       G       103 67       147 g       g         8 8 010 Backspace       40 28 050 (       72 48 110 H       H       104 68 150 h       h         9 9 011 Horizontal Tab       41 29 051 )       73 49 111 I       I 105 69 151 i       i         10 A 012 Line feed       42 2A 052 *       74 4A 112 J       J 106 6A 152 j       j         11 B 013 Vertical Tab       43 2B 053 +       + 75 4B 113 K       K 107 6B 153 k       k         12 C 014 Form feed       44 2C 054 ,       - 76 4C 114 L       L 108 6C 154 l       l         13 D 015 Carriage return       45 2D 055 -       - 77 4D 115 M       M 109 6D 155 m       m         14 E 016 Shift Out       46 2E 056 .       - 78 4E 116 M       M 100 6E 156 n       n         15 F 017 Shift In       47 2F 057 /       / 79 4F 117 O       O 111 6F 157 o       o         16 10 020 Data Link Escape       48 30 060 0       0 80 50 120 P       P 112 70 160 p       p	4 4	004	<b>End of Transmission</b>	36	24	044	\$	\$	68	44	104	D	D	100	64	144	d	d
7 7 007 Bell 39 27 047 ' ' 71 47 107 G G 103 67 147 g g 8 8 010 Backspace 40 28 050 ( ( 72 48 110 H H 104 68 150 h h 9 9 011 Horizontal Tab 41 29 051 ) ) 73 49 111 I I 105 69 151 i i 10 A 012 Line feed 42 2A 052 * * 74 4A 112 J J 106 6A 152 j j 11 B 013 Vertical Tab 43 2B 053 + + 75 4B 113 K K 107 6B 153 k k 12 C 014 Form feed 44 2C 054 , , 76 4C 114 L L 108 6C 154 l I 13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m 14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 112 70 160 p p 17 1021 Device Control 1 49 31 061 1 1 81 51 121 ̬ Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 4 52 34 064 4 4 84 50 42 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 073 7 9 89 59 131 Q Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 7 9 90 5A 132 c J 122 7D 175 z Z 27 1B 033 Escape 59 38 073 8 9 90 5A 132 c J 125 7D 175 } } 30 1E 036 Record Separator 60 3C 67 < > 94 5E 136 ^ ^ 126 7E 176 ~ ~	<b>5</b> 5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
8 8 010 Backspace	<b>6</b> 6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
9 9 011 Horizontal Tab 41 29 051 ) ) 73 49 111 I I 105 69 151 i i 10 A 012 Line feed 42 2A 052 * * 74 4A 112 J J 106 6A 152 j j 11 B 013 Vertical Tab 43 2B 053 + + 75 4B 113 K K 107 6B 153 k k 12 C 014 Form feed 44 2C 054 , , 76 4C 114 L L 108 6C 154 l I 13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m 14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 27 18 033 Escape 59 3B 073 7 5 91 58 133 ] I 123 78 173 x x 27 18 035 Escape 59 3B 073 8 8 8 55 01 135 c I 124 7C 174 z z 27 18 036 Record Separator 60 3C 074 < < 92 5C 134 c \ 124 7C 174   \ 124 7C 174   \ 129 10 036 Record Separator 61 3D 075 B > 94 5E 136 ^ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \		007	Bell	39	27	047	'	1			107	G	G	103	67	147	g	g
10 A 012 Line feed 42 2A 052 * * 74 4A 112 J J 106 6A 152 j j 118 013 Vertical Tab 43 2B 053 + + 75 4B 113 K K 107 6B 153 k k 12 C 014 Form feed 44 2C 054 , , 76 4C 114 L L 108 6C 154 l I 13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m 14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 26 1A 032 Substitute 58 3A 072 7 7 87 57 127 W W 119 77 167 w w 26 1A 032 Substitute 58 3A 072 7 9 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 7 9 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 7 9 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 7 9 15 B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 125 7D 175 } \} 30 1E 036 Record Separator 61 3D 075 B > 94 5E 136 ^ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	<b>8</b> 8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
11 B	<b>9</b> 9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
12 C 014 Form feed 44 2C 054 , , 76 4C 114 L L 108 6C 154 l I 13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m 14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 112 70 160 ј p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 51 125 U U 117 75 165 u u 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 7 7 88 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 17 18 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < > 94 5E 136 ^ ^ 126 7E 176 ~ ~	10 A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
13 D 015 Carriage return 45 2D 055 - - 77 4D 115 M M 109 6D 155 m m 14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 0 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 12 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 18 030 Cancel 56 38 070 7 7 87 57 127 W W 119 77 167 p y 26 1A 032 Substitute 58 3A 072 : 5 90 5A 132 Z Z 122 7A 172 z z 27 1B 033 Escape 59 3B 073 ; 9 15 135 ] J 125 7D 175 } } 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~		013	Vertical Tab					+					K					k
14 E 016 Shift Out 46 2E 056 . . 78 4E 116 N N 110 6E 156 n n 15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 O 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 7 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 7 ; 91 58 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 126 7E 176     29 1D 035 Group Separator 61 3D 075 = = 93 5D 135 ] ] 125 7D 175 } } 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~	<b>12</b> C	014	Form feed			054	,	1	76	4C	114	L	L	108	6C	154	l	1
15 F 017 Shift In 47 2F 057 / / 79 4F 117 O O 111 6F 157 o o 16 10 020 Data Link Escape 48 30 060 0 O 80 50 120 P P 112 70 160 p p 17 11 021 Device Control 1 49 31 061 1 1 81 51 121 Q Q 113 71 161 q q 18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 126 7E 176 } \} 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ \ \frac{107}{2} 126 7E 176 ~ \ \frac{117}{2} 160 o o 160 p p 170 p o 160 o o 160 o o 160 o o 160 p o		015	Carriage return			055	-	-	77	4D	115	M	M	109	6D	155	m	m
16 10 020 Data Link Escape  48 30 060 0 0  80 50 120 P P  112 70 160 p p  17 11 021 Device Control 1  49 31 061 1 1  81 51 121 Q Q  113 71 161 q q  18 12 022 Device Control 2  50 32 062 2 2  82 52 122 R R  114 72 162 r r  19 13 023 Device Control 3  51 33 063 3 3  83 53 123 S S  115 73 163 s s  20 14 024 Device Control 4  52 34 064 4 4  84 54 124 T T  116 74 164 t t  21 15 025 Negative Ack.  53 35 065 5 5  85 55 125 U U  117 75 165 u u  22 16 026 Synchronous idle  54 36 066 6 6  86 56 126 V V  118 76 166 v v  23 17 027 End of Trans. Block  55 37 067 7 7  87 57 127 W W  119 77 167 w w  24 18 030 Cancel  56 38 070 8 8  88 58 130 X X  120 78 170 x x  25 19 031 End of Medium  57 39 071 9 9  89 59 131 Y Y  121 79 171 y y  26 1A 032 Substitute  58 3A 072 : :  90 5A 132 Z Z  122 7A 172 z z  27 1B 033 Escape  59 3B 073 A *  91 5B 133 [ [  123 7B 173 { {  28 1C 034 File Separator  60 3C 074 < <  92 5C 134 \ \ 126 7E 176 ~ ~	<b>14</b> E	016	Shift Out	46	2E	056	.		78	4E	116	N	N	110	6E	156	n	n
17 11 021 Device Control 1		017	Shift In			057	/	/	79	4F	117	O	0	111	6F	157	o	0
18 12 022 Device Control 2 50 32 062 2 2 82 52 122 R R 114 72 162 r r 19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 126 7E 176 } \} 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ \(^{\text{A}}\)	<b>16</b> 10					060	0	0	80	50	120	P	P			160	p	р
19 13 023 Device Control 3 51 33 063 3 3 83 53 123 S S 115 73 163 s s 20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 126 7E 176 } \} 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ \(^{\text{\chi}}\) 126 7E 176 ~ \(^{\text{\chi}}\)	<b>17</b> 11	021	Device Control 1			061	1	1			121	Q	Q			161	q	q
20 14 024 Device Control 4 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 Negative Ack. 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 Synchronous idle 54 36 066 6 6 86 56 126 V V 118 76 166 v v 23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 124 7C 174     29 1D 035 Group Separator 61 3D 075 = = 93 5D 135 ] ] 125 7D 175 } } 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^	<b>18</b> 12	022	Device Control 2			062	2	2			122	R	R	114	72	162	r	r
21 15 025 Negative Ack. 22 16 026 Synchronous idle 23 17 027 End of Trans. Block 24 18 030 Cancel 25 19 031 End of Medium 26 1A 032 Substitute 27 1B 033 Escape 28 1C 034 File Separator 29 1D 035 Group Separator 30 1E 036 Record Separator 31 5 065 5 5 85 55 125 U U 117 75 165 u u 117 75 165 u u 118 76 166 v v 118 76 167 w w 119 77 167	<b>19</b> 13	023	Device Control 3			063	3	3	83	53	123	S	S	115	73	163	s	S
22 16 026 Synchronous idle 23 17 027 End of Trans. Block 24 18 030 Cancel 25 19 031 End of Medium 26 1A 032 Substitute 27 1B 033 Escape 28 1C 034 File Separator 29 1D 035 Group Separator 30 1E 036 Record Separator 30 16 08 #054; 6 86 56 126 V V 118 76 166 v V 119 77 167 w W 110 78 170 x X 120 78 170 y Y 121 79 171 y Y 122 7A 172 z Z 123 7B 133 Z Z 124 7C 174     125 7D 175 } } 126 7E 176 ~ ~	20 14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
23 17 027 End of Trans. Block 55 37 067 7 7 87 57 127 W W 119 77 167 w w 24 18 030 Cancel 56 38 070 8 8 88 58 130 X X 120 78 170 x x 25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z z 27 1B 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 124 7C 174     29 1D 035 Group Separator 61 3D 075 = = 93 5D 135 ] ] 125 7D 175 } } 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~	<b>21</b> 15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
24 18 030 Cancel       56 38 070 8 8       88 58 130 X X       120 78 170 x x         25 19 031 End of Medium       57 39 071 9 9       89 59 131 Y Y       121 79 171 y y         26 1A 032 Substitute       58 3A 072 : : 90 5A 132 Z Z       122 7A 172 z Z         27 1B 033 Escape       59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { {         28 1C 034 File Separator       60 3C 074 < < 92 5C 134 \ \ 124 7C 174	<b>22</b> 16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	V
25 19 031 End of Medium 57 39 071 9 9 89 59 131 Y Y 121 79 171 y y 26 1A 032 Substitute 58 3A 072 : : 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 Escape 59 3B 073 ; ; 91 5B 133 [ [ 123 7B 173 { { 28 1C 034 File Separator 60 3C 074 < < 92 5C 134 \ \ 124 7C 174     29 1D 035 Group Separator 61 3D 075 = = 93 5D 135 ] ] 125 7D 175 } } 30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~	23 17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	W
26 1A 032 Substitute       58 3A 072 : :       90 5A 132 Z Z       122 7A 172 z Z         27 1B 033 Escape       59 3B 073 ; ;       91 5B 133 [ [       123 7B 173 { {         28 1C 034 File Separator       60 3C 074 < <	<b>24</b> 18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	X
27 1B 033 Escape       59 3B 073 ; ;       91 5B 133 [ [       123 7B 173 { {         28 1C 034 File Separator       60 3C 074 < <	<b>25</b> 19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	У
28 1C 034 File Separator       60 3C 074 < <	<b>26</b> 1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	Z
29 1D 035 Group Separator       61 3D 075 = =       93 5D 135 ] ]       125 7D 175 } }         30 1E 036 Record Separator       62 3E 076 > >       94 5E 136 ^ ^       126 7E 176 ~ ~	<b>27</b> 1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
30 1E 036 Record Separator 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~	<b>28</b> 1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	1	124	7C	174		
	<b>29</b> 1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
31 1F 037 Unit Separator 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177  Del	<b>30</b> 1E			62	3E	076	>	>			136	^	٨			176	~	~
	<b>31</b> 1F			63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

#### For More Characters, Use Unicode

There are plenty of characters that aren't available in ASCII (characters from non-English languages, advanced symbols, emoji...) due to the limited size.

The Unicode system represents every character that can be typed into a computer. It uses up to 5 bytes, which can represent up to 1 trillion characters! Find all the Unicode characters here: <a href="www.unicode-table.com">www.unicode-table.com</a>

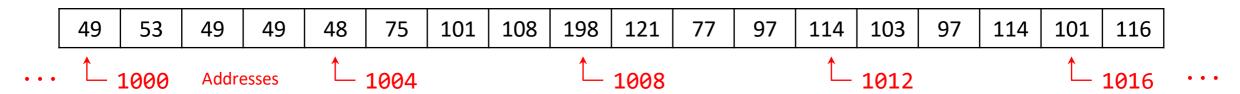
The Unicode system is also **actively under development**. The Unicode Consortium regularly updates the standard to add new types of characters and emoji.

**Discuss:** what are the potential repercussions of using a single standard for all text on computers?

# Computer Memory is Stored as Binary

Your computer keeps track of saved data and all the information it needs to run in its **memory**, which is represented as binary. You can think about your computer's memory as a really long list of bits, where each bit can be set to 0 or 1. But usually we think in terms of bytes, groups of 8 bits.

Every byte in your computer has an **address**, which the computer uses to look up its value.



#### Binary Values Depend on Interpretation

When you open a file on your computer, the application goes to the appropriate address, reads the associated binary, and **interprets** the binary values based on the file encoding it expects. That interpretation depends on the **application** you use when opening the file, and the **filetype**.

You can attempt to open **any file** using **any program**, if you convince your computer to let you try. Some programs may crash, and others will show nonsense because the binary isn't being interpreted correctly.

**Example:** try changing a .docx filetype to .txt, then open it in a plain text editor. .docx files have extra encoding, whereas .txt files use plain ASCII.

# We Use Lots of Bytes!

In modern computing, we use a **lot** of bytes to represent information.

**Smartphone Memory:** 64 gigabytes = 64 **billion** bytes

**Google databases:** Over 100 million gigabytes = 100 quadrillion bytes!

CMU Wifi: 15 million bytes per second

# Learning Objectives

 Understand how different number systems can represent the same information

• Translate binary numbers to decimal, and vice versa

Interpret binary numbers as abstracted types, including colors and text

Feedback form: <a href="https://bit.ly/110-f21-feedback">https://bit.ly/110-f21-feedback</a>