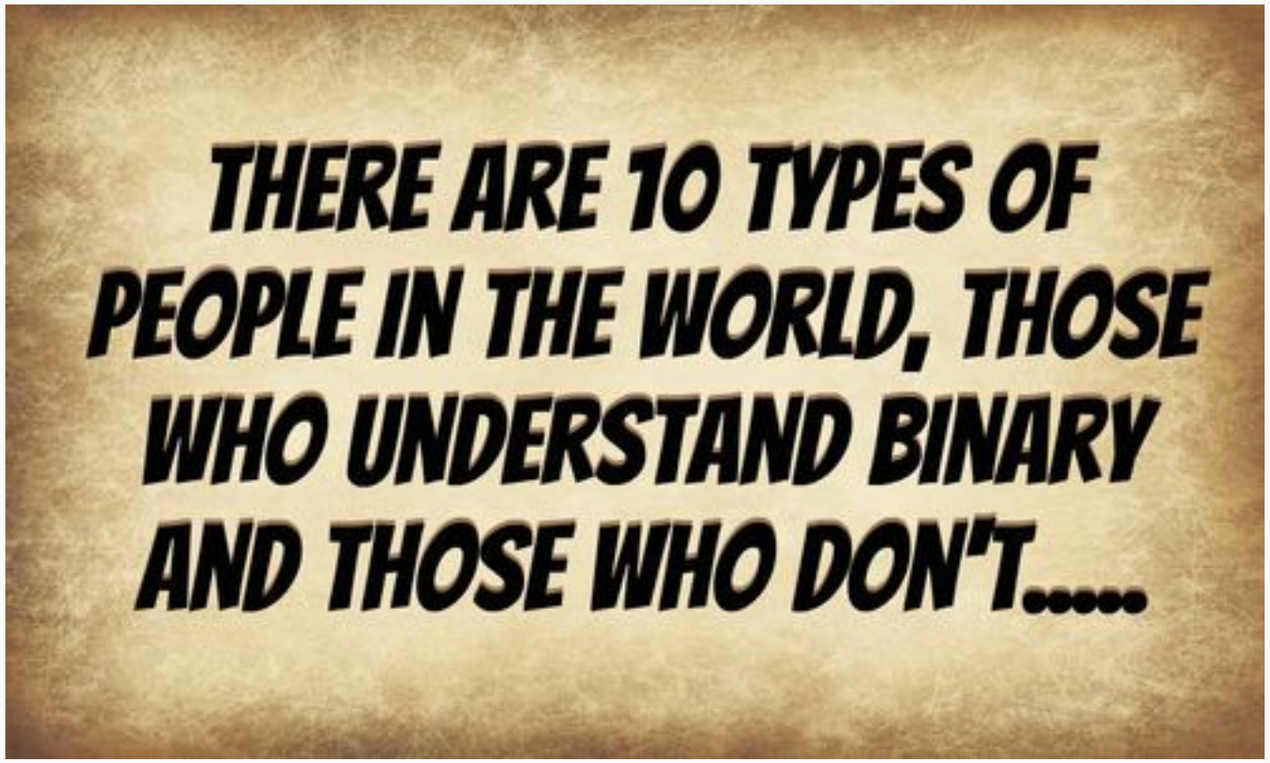
Unit 0: Binary



Dave Abel

January 29, 2016

Outline of Class

- Binary!
- Representing Stuff with Binary
- Binary Addition
- Binary Subtraction
- Binary Multiplication

Your First Bit

- Chemistry has its *molecules*
- Physics has its *strings*
- Math has its *sets*.
- Computer science has its *bits*.
  - They are the smallest unit of information
    - **True:** (1, on)
    - **False:** (0, off)

Your First Bit

- Chemistry has its *molecules*
- Physics has its *strings*
- Math has its *sets*.

BInary
digi**T**

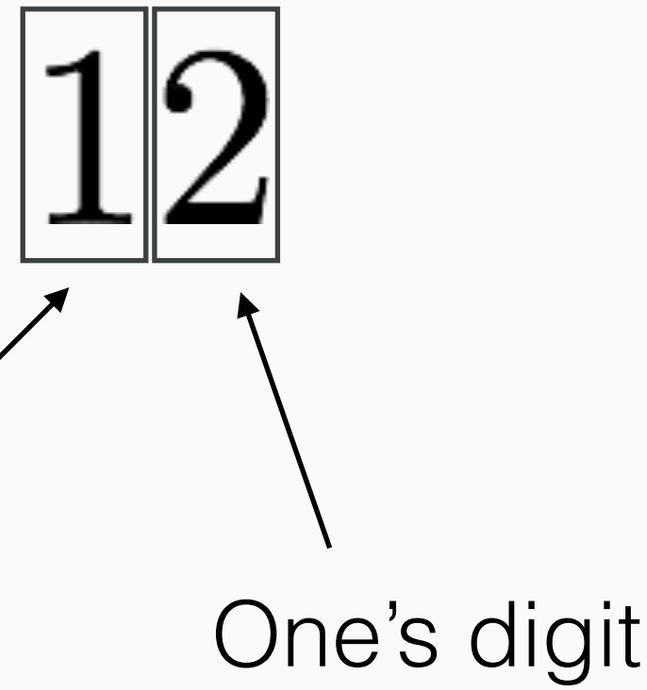
- Computer science has its *bits*.
  - They are the smallest unit of information
    - **True:** (1, on)
    - **False:** (0, off)

Why a Bit?

- Bits are:
  - **simple:** there's no smaller discrete distinction to be made!
  - **powerful:** bit sequences can still represent seemingly anything!
- Fundamental! (1 vs. 0, on vs. off, true vs. false)

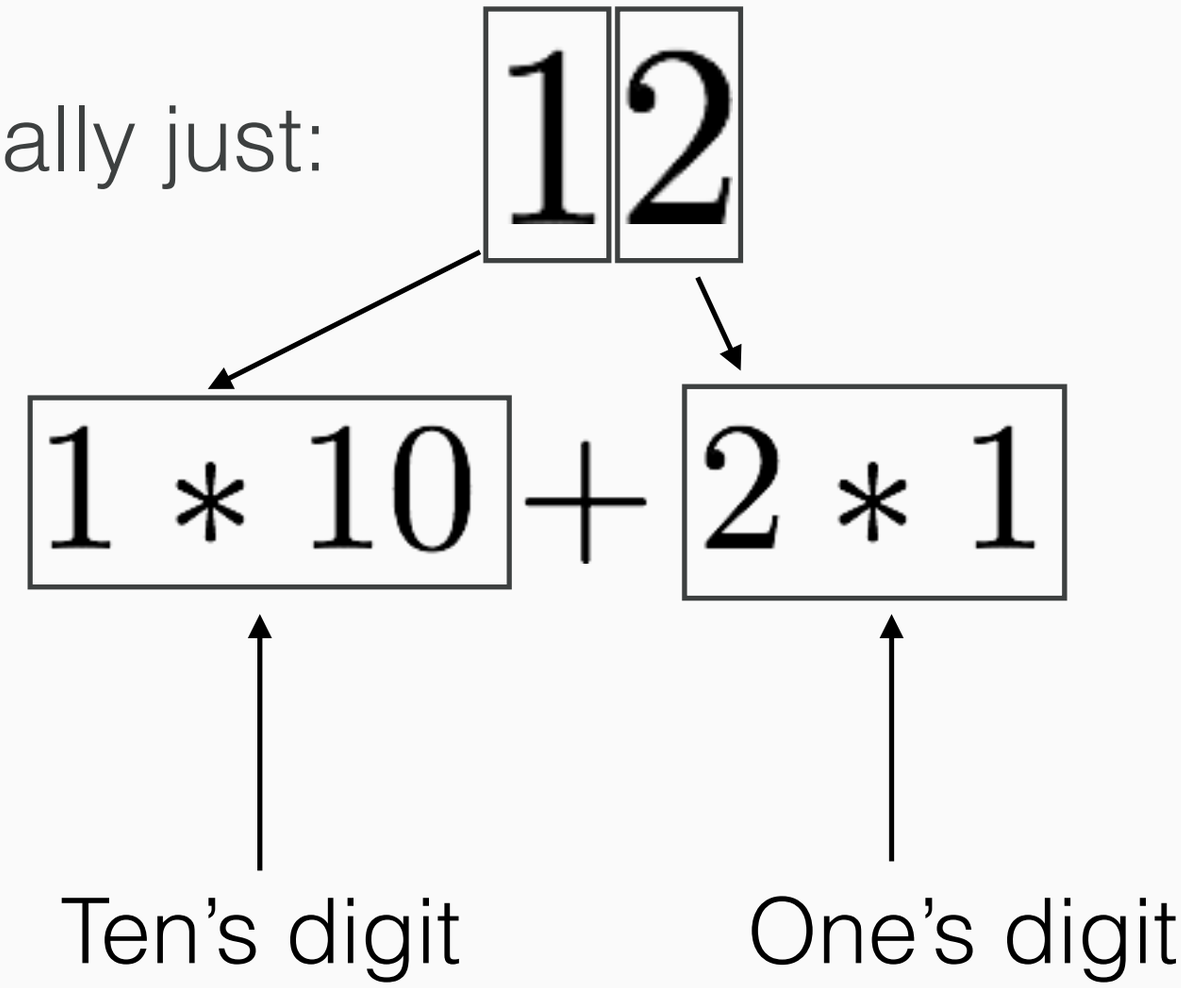
Base Ten

- We use base *ten* for our representation
- So twelve is really just:



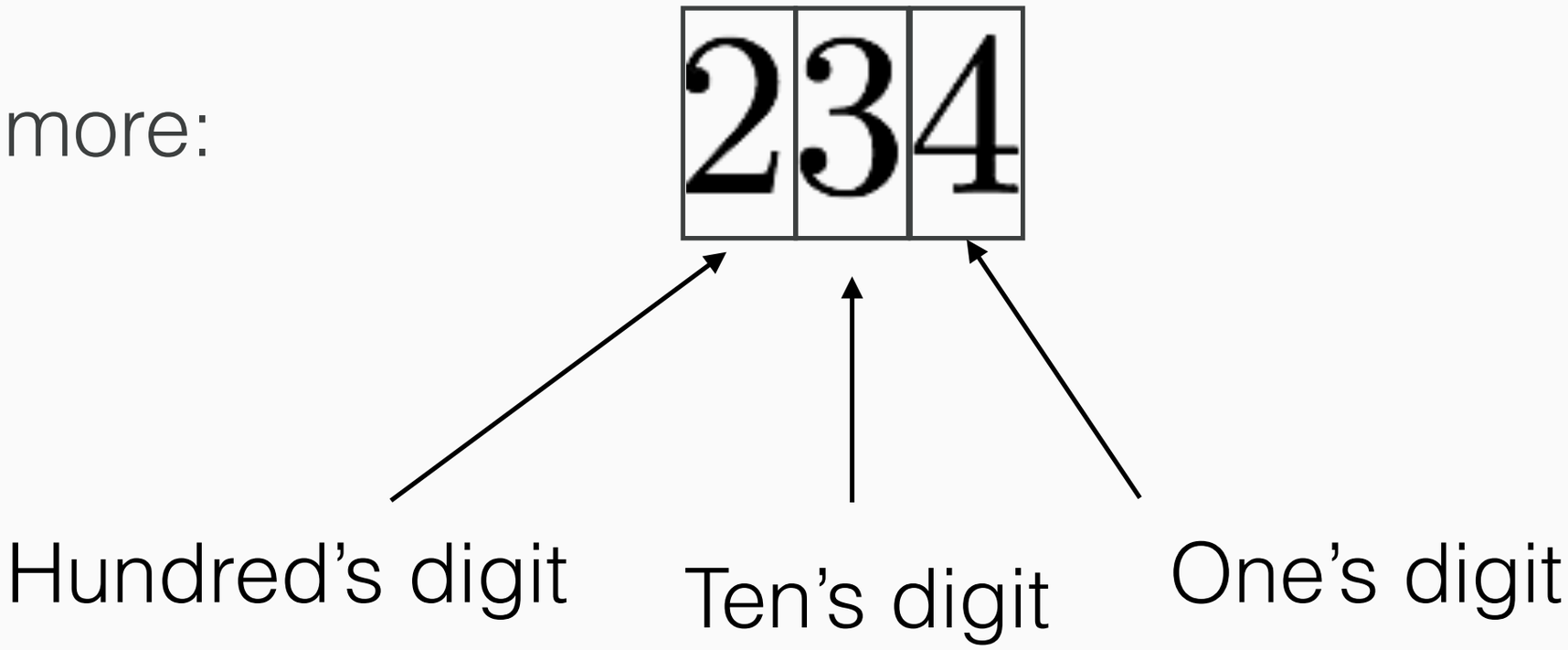
Base Ten

- We use base *ten* for our representation
- So twelve is really just:



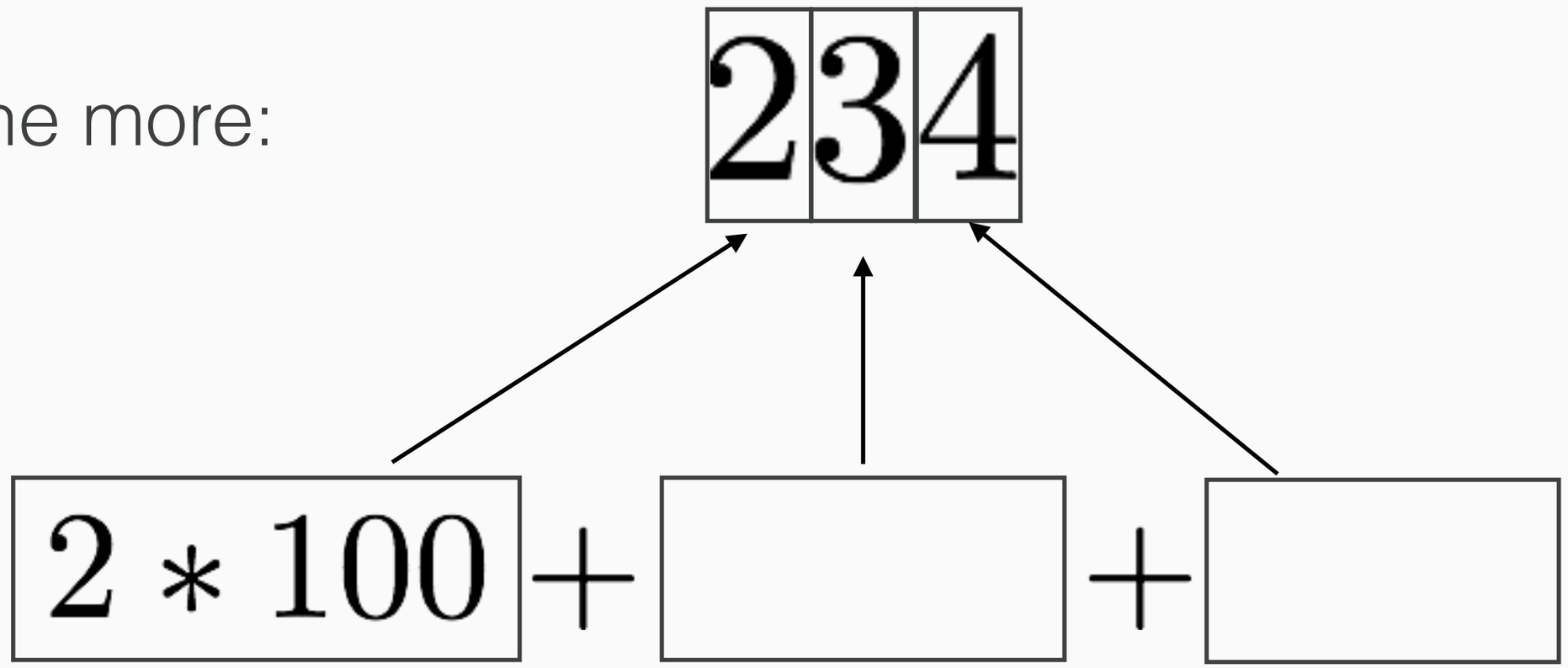
Base Ten

- We use base *ten* for our representation
- One more:



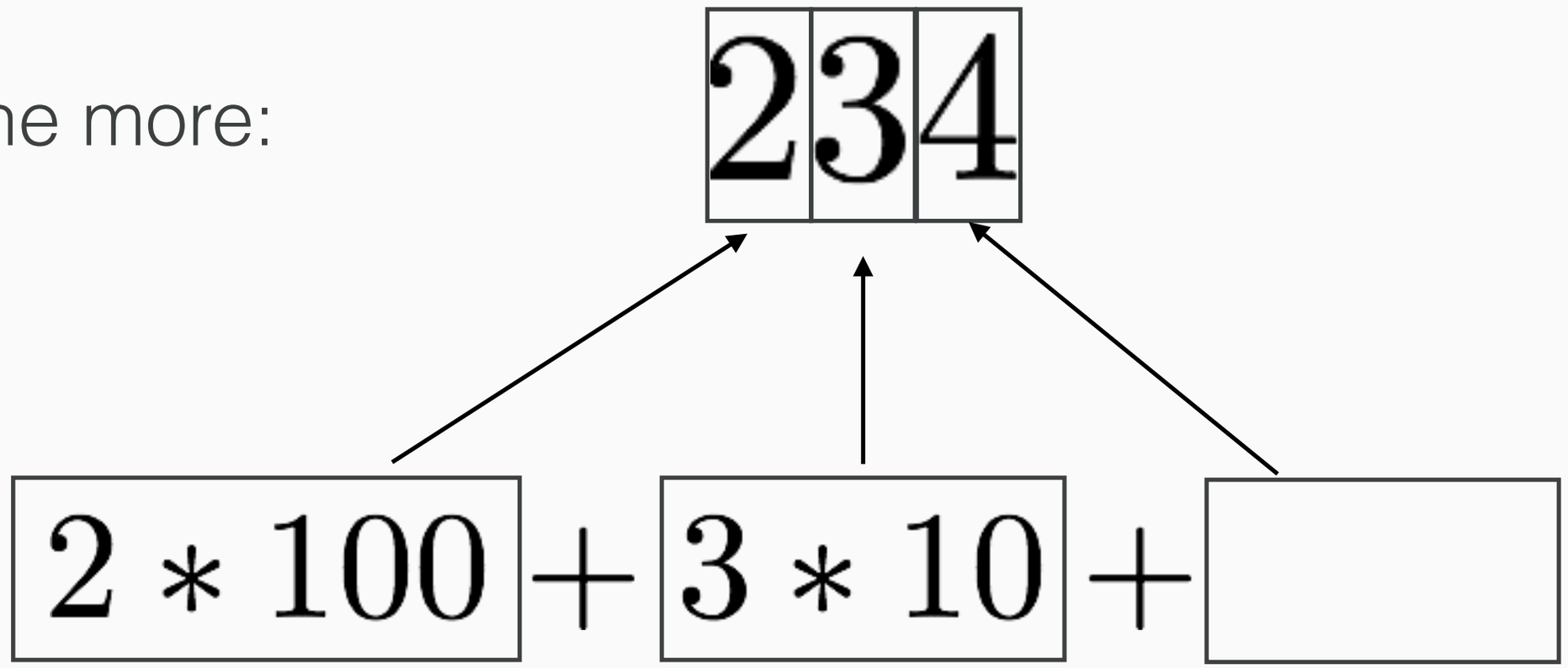
Base Ten

- We use base *ten* for our representation
- One more:



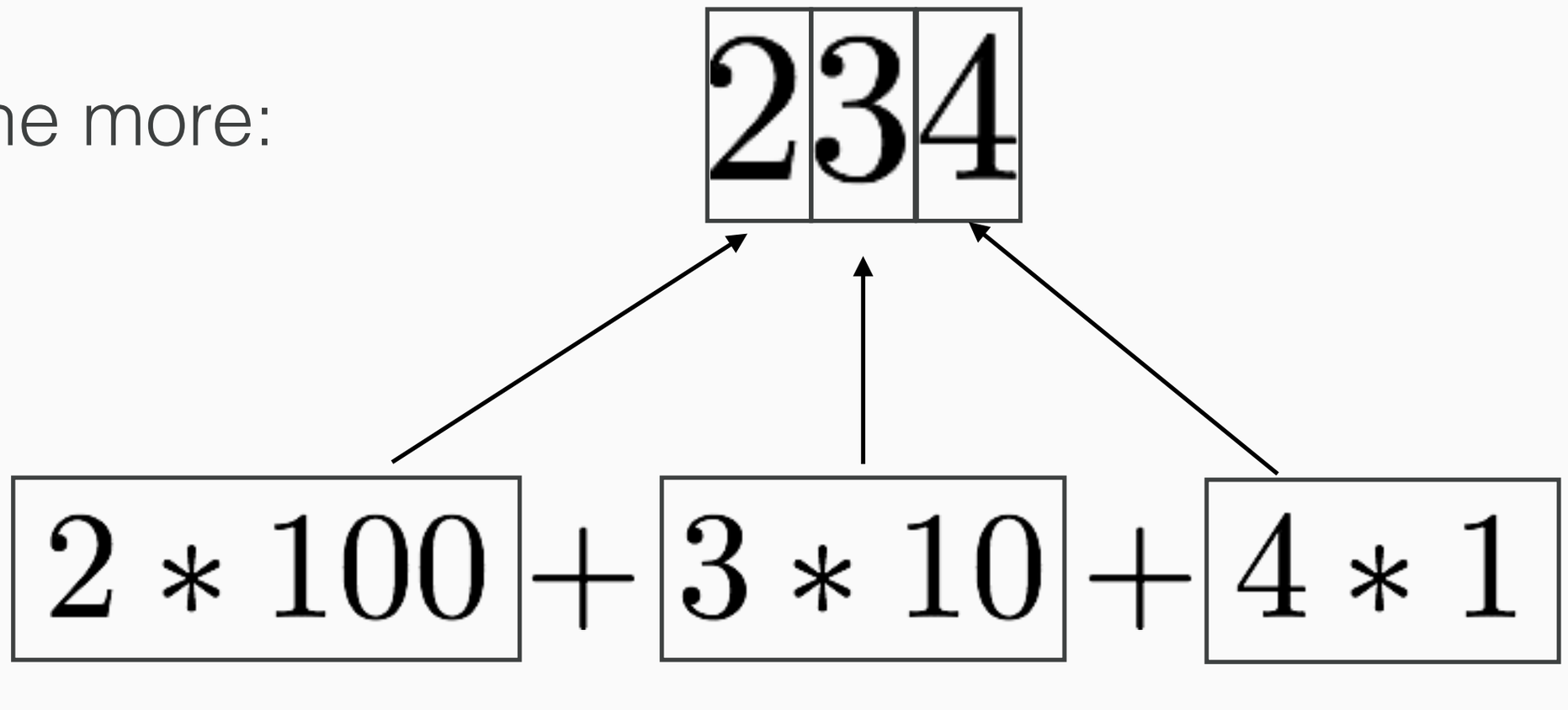
Base Ten

- We use base *ten* for our representation
- One more:



Base Ten

- We use base *ten* for our representation
- One more:



Base Ten

- We use base *ten* for our representation

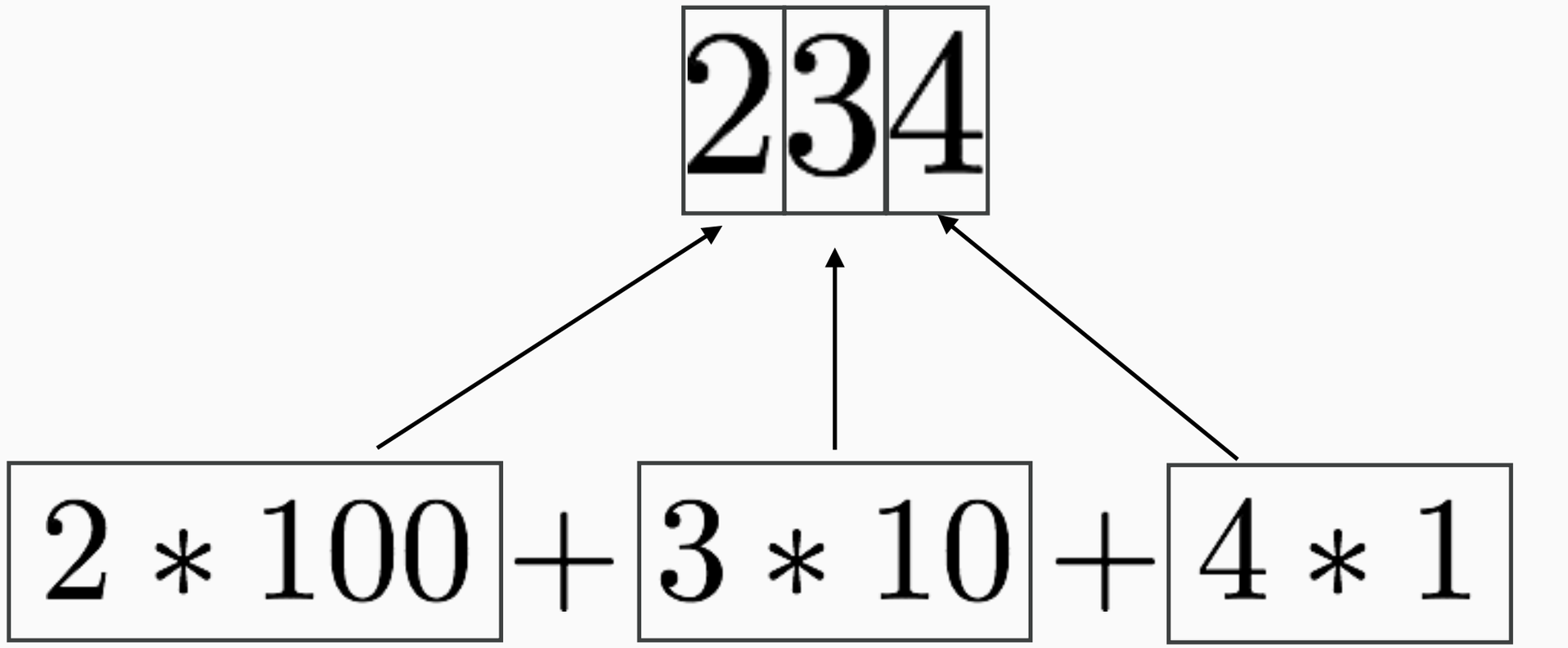
Q: Why do you think humans use base ten?

Base Ten

- We use base *ten* for our representation

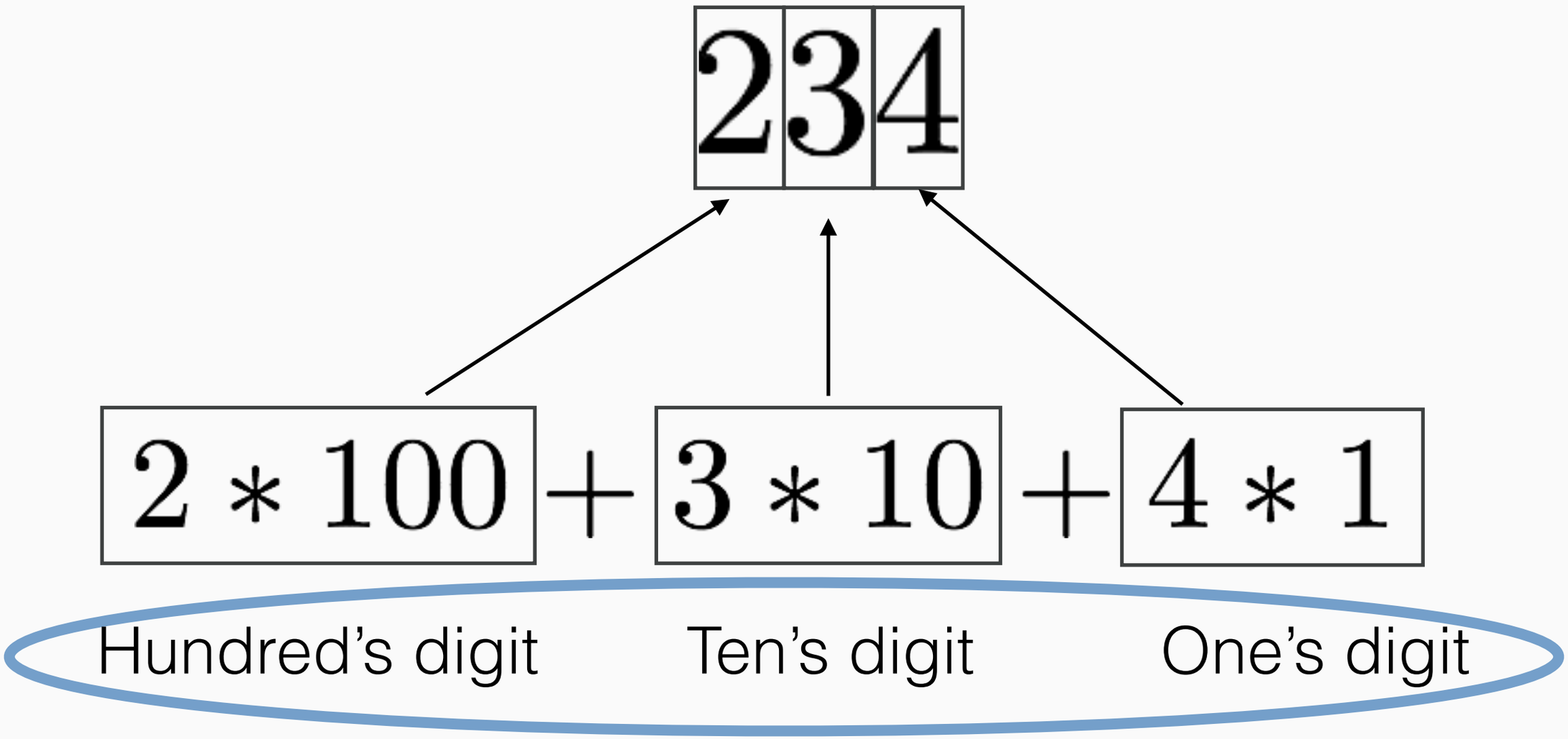
Q: Why do you think humans use base ten?

Revisiting Two Hundred Thiry Four



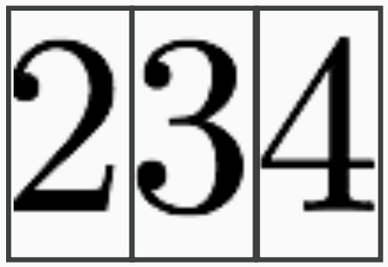
Hundred's digit Ten's digit One's digit

Revisiting Two Hundred Thiry Four



Q: What's the pattern here?

Revisiting Two Hundred Thiry Four

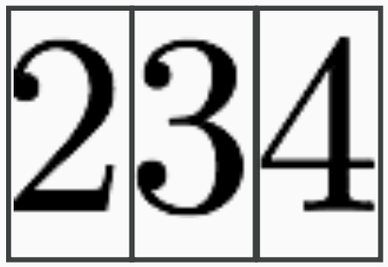


A: Powers of ten!

Hundred's digit Ten's digit One's digit

Q: What's the pattern here?

Revisiting Two Hundred Thiry Four



A: Powers of ten!

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

Hundred's digit Ten's digit One's digit

Q: What's the pattern here?

In Binary…

Main Idea: Each digit must be either a 0 or a 1, instead of 0,1,2,3,4,…,9

How About Binary?

Powers of TWO!

……. Four’s digit Two’s digit One’s digit

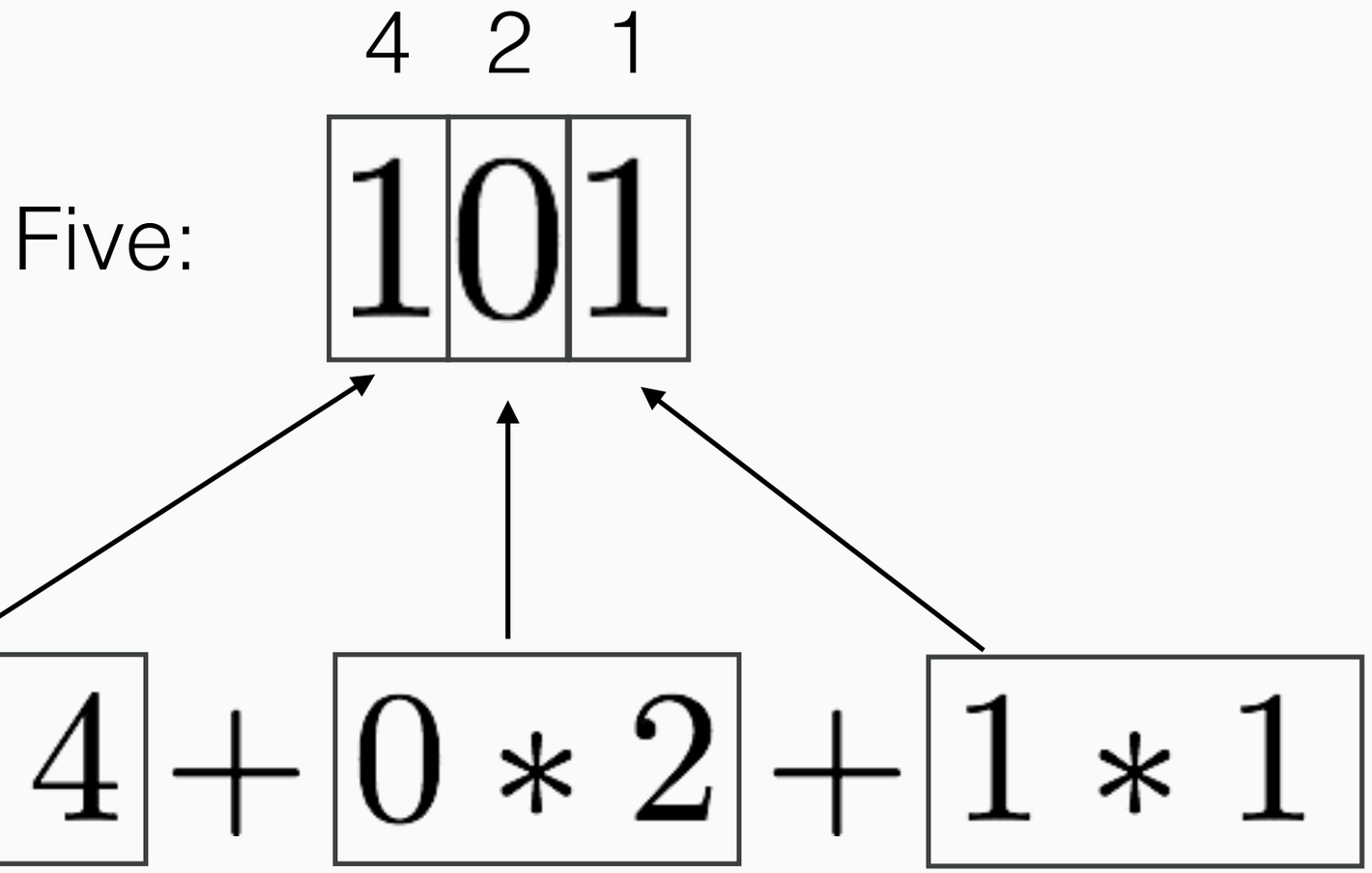
How About Binary?

Powers of TWO!

$$2^2 = 4 \quad 2^1 = 2 \quad 2^0 = 1$$

…….. Four’s digit Two’s digit One’s digit

Our First Binary Number

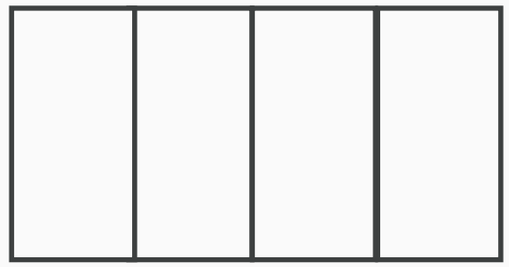


Convert: Twelve

With a partner next to you, walk through converting twelve into binary.

Convert Twelve:

- **Start with the biggest power of 2 no bigger than your number.**
- Write down a 1 in that power of two's column.
- Subtract that power of 2 from your number.
- Go to the next smallest power of 2:
- If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.
- If not, write down 0.
- Repeat until you get to 1.



8 4 2 1

Our Number: Twelve

Convert Twelve:

- Start with the biggest power of 2 no bigger than your number.
- **Write down a 1 in that power of two's column.**

8	4	2	1
1			

- Subtract that power of 2 from your number.
- Go to the next smallest power of 2:
- If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.
- If not, write down 0.
- Repeat until you get to 1.

8	4	2	1
1	1	0	0

Our Number: Twelve

Convert Twelve:

- Start with the biggest power of 2 no bigger than your number.
- Write down a 1 in that power of two's column.
- **Subtract that power of 2 from your number.**
- Go to the next smallest power of 2:
- If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.
- If not, write down 0.
- Repeat until you get to 1.

8	4	2	1
1			

8	4	2	1

Our Number: **Twelve - Eight = Four**

Convert Twelve:

- Start with the biggest power of 2 no bigger than your number.
- Write down a 1 in that power of two's column.
- Subtract that power of 2 from your number.

8	4	2	1
1			

- **Go to the next smallest power of 2:**
- If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.
- If not, write down 0.
- Repeat until you get to 1.

8	4	2	1

Our Number: **Four**

Convert Twelve:

- Start with the biggest power of 2 no bigger than your number.
- Write down a 1 in that power of two's column.
- Subtract that power of 2 from your number.

8	4	2	1
1	1		

- Go to the next smallest power of 2:
  - **If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.**
- If not, write down 0.
- Repeat until you get to 1.

Our Number: **Four - Four = Zero**

Convert Twelve:

- Start with the biggest power of 2 no bigger than your number.
- Write down a 1 in that power of two's column.
- Subtract that power of 2 from your number.

1	1	0	0

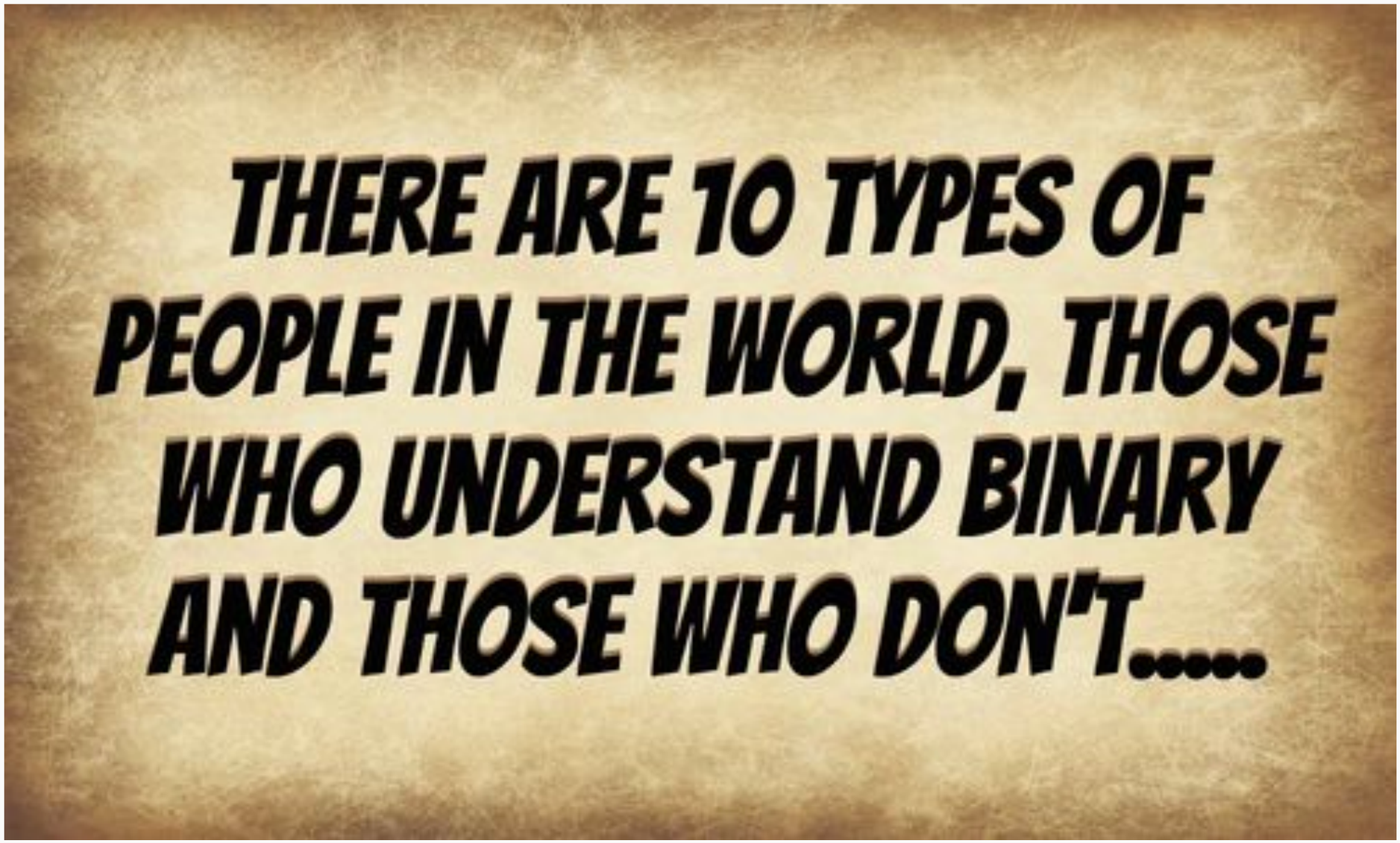
- Go to the next smallest power of 2:

8 4 2 1

- If your remaining number, (four in our case) is greater than or equal to the power of 2, write down a 1 and subtract the power of 2.
- If not, write down 0.
- **Repeat until you get to 1.**

Our Number: **Zero**

Welcome to the Club



Welcome to the Club

- Fun fact! 8 bits…. e.g.: $0\ 1\ 0\ 0\ 1\ 0\ 0\ 1$

Your First Official Byte

- Fun fact! 8 bits…. e.g.: 0 1 0 0 1 0 0 1
- Is called a Byte

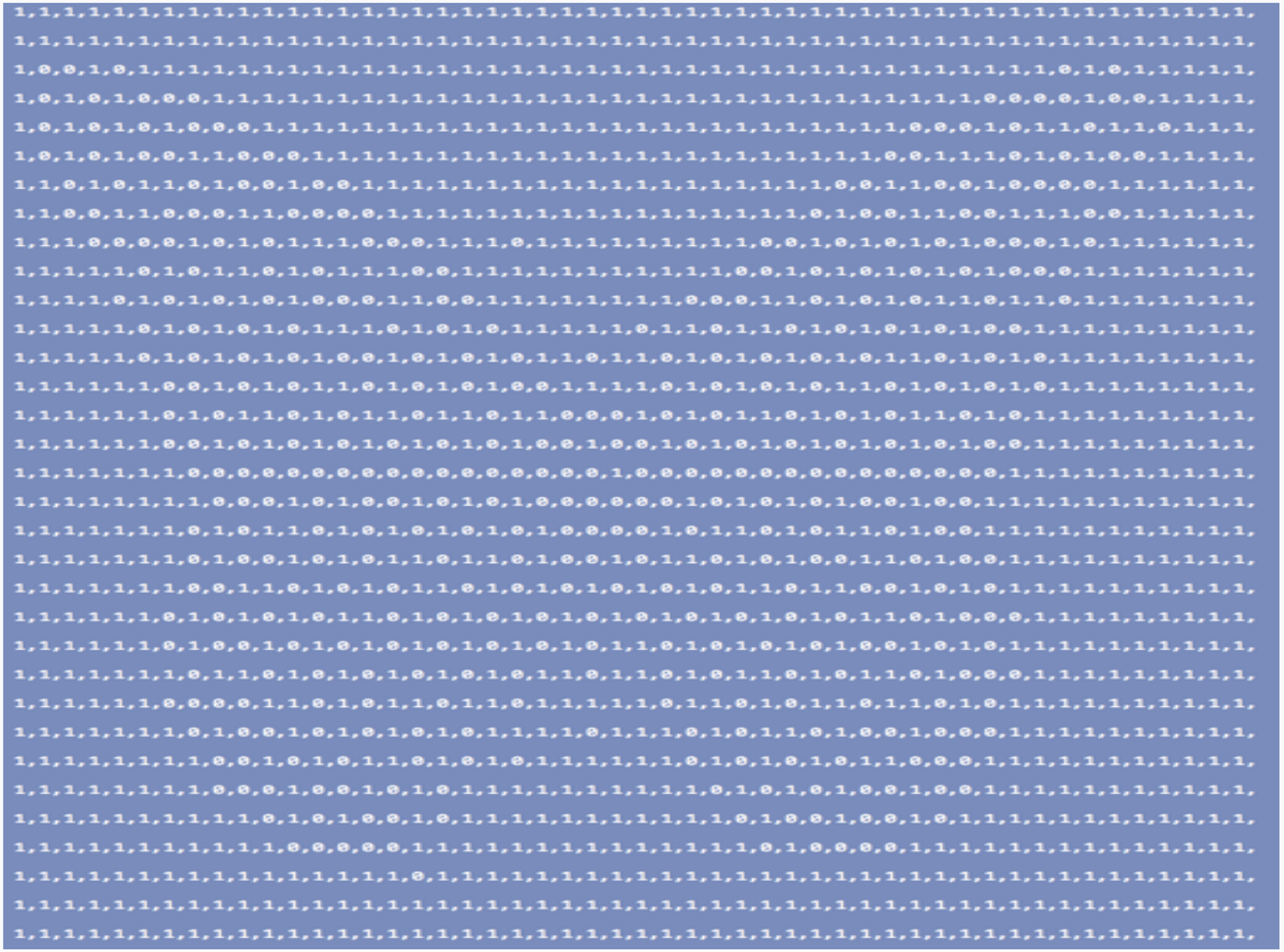
Your First Official Byte

- Fun fact! 8 bits…. e.g.: 0 1 0 0 1 0 0 1
- Is called a Byte
- Even more fun fact! 4 bits…

Your First Official Nibble

- Fun fact! 8 bits…. e.g.: $0\ 1\ 0\ 0\ 1\ 0\ 0\ 1$
- Is called a Byte
- Even more fun fact! 4 bits…
- Is called a nibble

Representing Images!



Representing Images!

Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0

Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about grey scale?

Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about grey scale?
- A: Pick a range, say one nibble: 0000-1111
- 0000 = white
- 0001-1110 = light grey through dark grey
- 1111 = black

Representing Images!

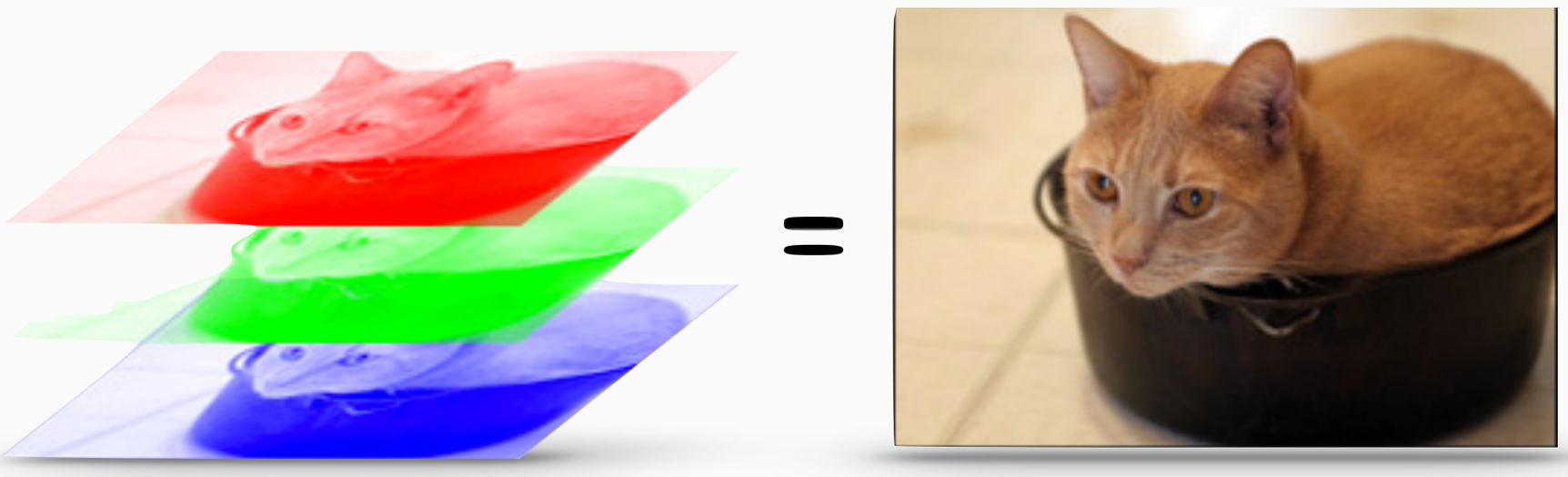
- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about color?

Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about color?
- A: Multiple Channels!

Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about color?
- A: Multiple Channels!



Representing Images!

- Idea: Black Pixel = 1, White Pixel = 0
- Q: How about color?
- A: Multiple Channels!

Representing Images!

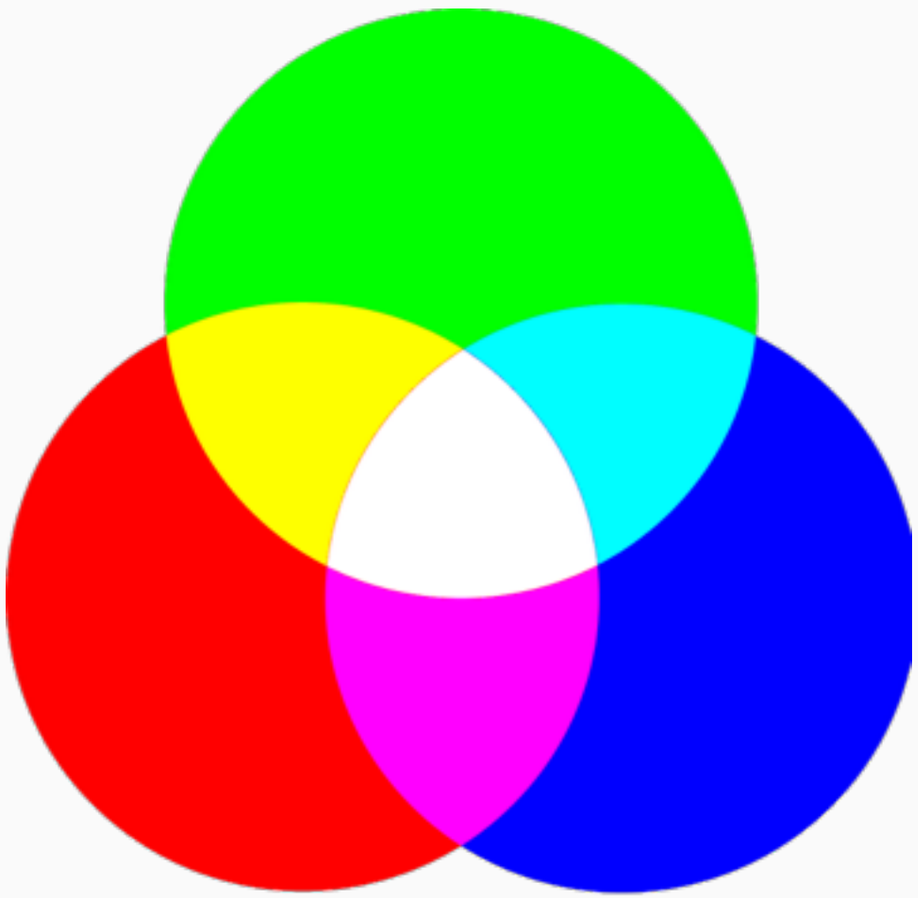
- Idea: Black Pixel = 1, White Pixel = 0
- ~~Q: How about color?~~
- A: Multiple “channels”: Red, Green, Blue.
- **Q: Why is this weird?**

Representing Images!

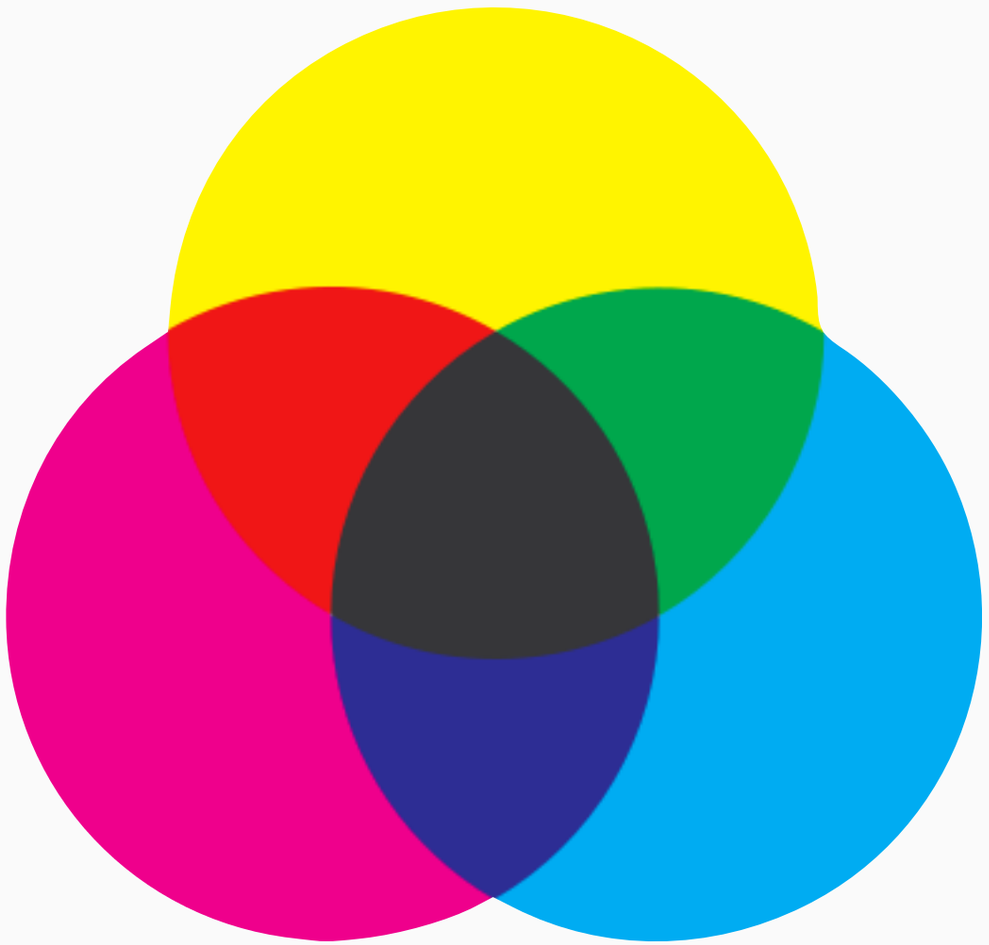
- Idea: Black Pixel = 1, White Pixel = 0
- ~~Q: How about color?~~
- A: Multiple “channels”: Red, Green, Blue.
- **Q: Why is this weird?**
- **A: But… But… Painting!**

Representing Images!

Additive Color (computers)



Subtractive Color (paint)



Representing Video!

Representing Text!

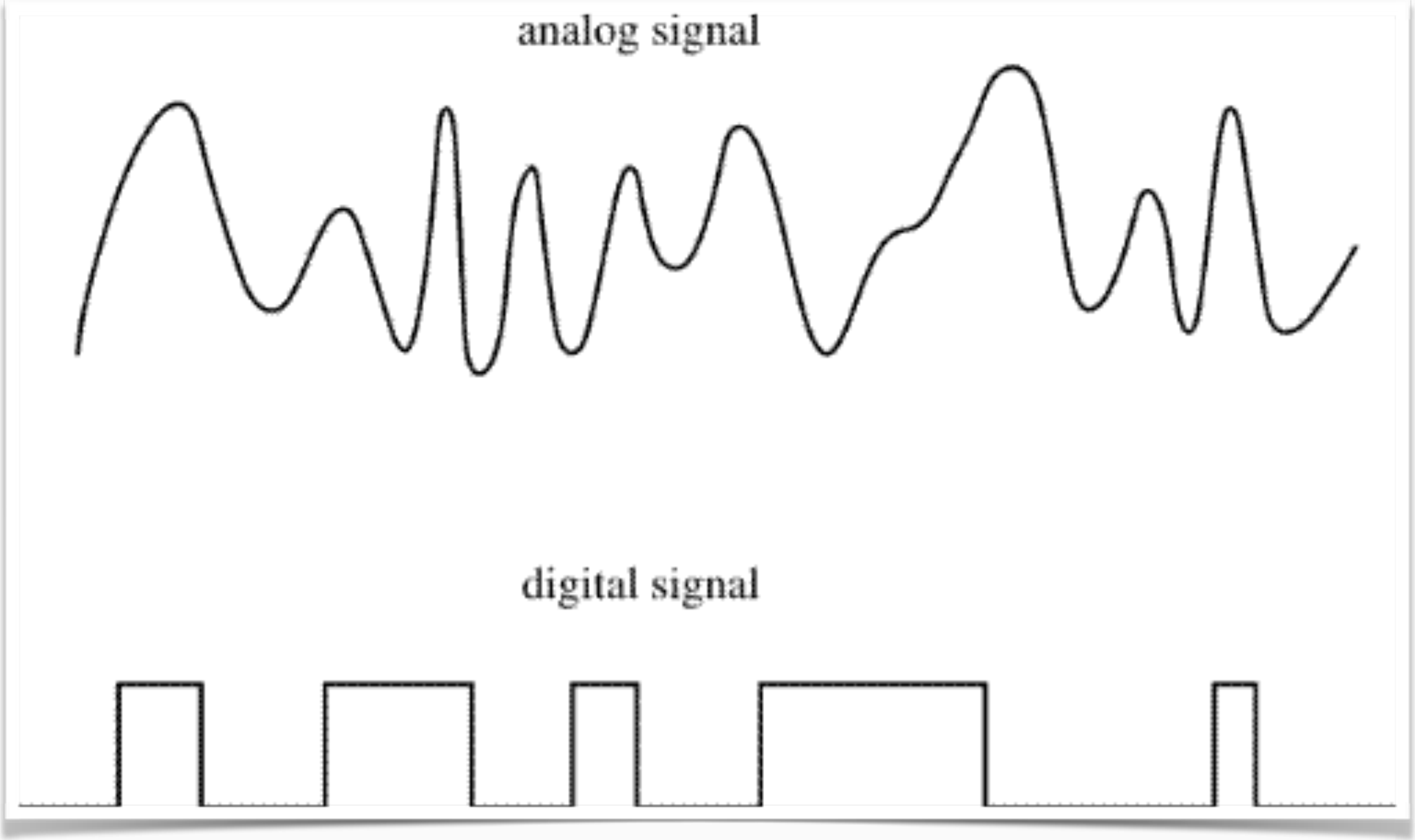
0	1	10	11	…	11100	11101	11110
a	b	c	d	…	<space>	!	?

Representing Text!

0	1	10	11	…	11100	11101	11110
a	b	c	d	…	<space>	!	?

hi ⟶ 1000 1001

Representing Sounds!



Representing…?

Q: Does anything else come to mind?

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$\begin{array}{r} 1001010 \\ +\ 0101000 \\ \hline \\ \end{array}$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$
\begin{array}{r}
1001010 \\
+\ 0101000 \\
\hline
0
\end{array}
$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$
\begin{array}{r}
1001010 \\
+\ 0101000 \\
\hline
10
\end{array}
$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$\begin{array}{r} 1001010 \\ +\ 0101000 \\ \hline \\ 010 \end{array}$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$\begin{array}{r} {\scriptstyle 1} \\ 1001010 \\ +\ 0101000 \\ \hline 0010 \end{array}$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$\begin{array}{r} {\scriptstyle 1} \\ 1001010 \\ +\ 0101000 \\ \hline 10010 \end{array}$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:

$$\begin{array}{r} {\scriptstyle 1} \\ 1001010 \\ +\ 0101000 \\ \hline 1110010 \end{array}$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:
- Can check by converting:

$$1001010 = 74$$

$$0101000 = 40$$

$$\text{---}$$

$$114$$

$$\begin{array}{r} 1001010 \\ +\ 0101000 \\ \hline 1110010 \end{array}$$

$$(1 * 64 + 1 * 32 + 1 * 16 + 1 * 2)$$

Binary Addition

- Just like base ten addition.
  - Line up the numbers, add:
- Can check by converting:

$$1001010 = 74$$

$$0101000 = 40$$

$$\overline{}$$

$$114$$ ✓

$$\begin{array}{r} 1001010 \\ +\ 0101000 \\ \hline 1110010 \end{array}$$

$$(1 * 64 + 1 * 32 + 1 * 16 + 1 * 2)$$

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- Three simple cases:

A:

$$\begin{array}{rr} & 1 \\ - & 0 \\ \hline & 1 \end{array}$$

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- Three simple cases:

A:

$$\begin{array}{r} 1 \\ -\ 0 \\ \hline 1 \end{array}$$

B:

$$\begin{array}{r} 1 \\ -\ 1 \\ \hline 0 \end{array}$$

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- Three simple cases:

A:

$$\begin{array}{r} 1 \\ -\ 0 \\ \hline 1 \end{array}$$

B:

$$\begin{array}{r} 1 \\ -\ 1 \\ \hline 0 \end{array}$$

C:

$$\begin{array}{r} 0 \\ -\ 0 \\ \hline 0 \end{array}$$

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rrr} & 1 & 0 \\ - & 0 & 1 \\ \hline \end{array}$$

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rrr} & 1 & 0 \\ - & 0 & 1 \\ \hline \end{array}$$

In this class: we always assume the top number is bigger (result always positive)

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rrr} & 1 & 0 \\ - & 0 & 1 \\ \hline \end{array}$$

In this class: we always assume the top number is bigger (result always positive)

Then, there must be a 1 somewhere to the left of the top number

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rrr} & 1 & 0 \\ - & 0 & 1 \\ \hline \end{array}$$

Here, we *borrow* the next power of two

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

1

D:

$$\begin{array}{rcc} & & 1 \\ & \not{1} & 0 \\ - & 0 & 1 \\ \hline \end{array}$$

Here, we *borrow* the next power of two

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rcc} & & 1 \\ & \not{1} & 0 \\ - & 0 & 1 \\ \hline & & 1 \end{array}$$

Here, we *borrow* the next power of two

Binary Subtraction

- Just like base ten subtraction, with one weird case.
- One weird case:

D:

$$\begin{array}{rrr} & 1 & 0 \\ - & 0 & 1 \\ \hline & & 1 \end{array}$$

Can always convert to sanity check:

2 - 1 = 1

Binary Multiplication

- Actually *way easier than base then!*
- Q: Why?
- A: All 0's and 1's! Multiplying by those is easy!

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add a the top number, otherwise, shift over and repeat:

$$\begin{array}{cccccc} & & 1 & 0 & 0 & 1 \\ * & & 1 & 0 & 1 & \boxed{0} \\ \hline \end{array}$$

← First, 0 so do nothing, shift over.

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$\begin{array}{rcccc} & 1 & 0 & 0 & 1 \\ * & 1 & 0 & \boxed{1} & 0 \\ \hline \end{array}$$

← Now, 1, so write the top number

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$\begin{array}{cccccc} & & 1 & 0 & 0 & 1 \\ & * & 1 & 0 & 1 & 0 \\ \hline & 1 & 0 & 0 & 1 & \end{array}$$

← Now, 1, so write the top number

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$
\begin{array}{cccccc}
 & & 1 & 0 & 0 & 1 \\
 & * & 1 & \boxed{0} & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 &
\end{array}
$$

⟵ Another 0…

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$
\begin{array}{cccccc}
 & & 1 & 0 & 0 & 1 \\
 & * & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 &
\end{array}
$$

← A one! Fun. Write the top number again.

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$
\begin{array}{cccccc}
 & & 1 & 0 & 0 & 1 \\
 & * & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 & \\
+ & 1 & 0 & 0 & 1 & \\
\hline
\end{array}
$$

← Now add these

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$
\begin{array}{cccccccc}
 & & & 1 & 0 & 0 & 1 & \\
 & & * & 1 & 0 & 1 & 0 & \\
\hline
 & & 1 & 0 & 0 & 1 & & \\
+ & 1 & 0 & 0 & 1 & & & \\
\hline
1 & 0 & 1 & 1 & 0 & 1 & 0 &
\end{array}
$$

← Now add these

Binary Multiplication

- Idea: move over bottom number, if there's a 1, add the top number, otherwise, shift over and repeat:

$$
\begin{array}{cccccc}
 & & 1 & 0 & 0 & 1 \\
 & * & 1 & 0 & 1 & 0 \\
\hline
 & & 1 & 0 & 0 & 1 \\
+ & 1 & 0 & 0 & 1 & \\
\hline
1 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

Convert to sanity check

nine * ten = ninety

(64 + 16 + 8 + 2 = 90)

Reflection

- Binary!
- Representing Stuff with Binary
- Binary Addition
- Binary Subtraction
- Binary Multiplication
- **Next up: Logic!**

A Quick Game…

You will have *fifteen* seconds.

Using math symbols, English words, or both, write down the biggest number you can (not an infinity). Be precise enough so that your classmates can determine what number you've written by consulting only your card (and possibly Wikipedia).

GO!

Game: Discussion