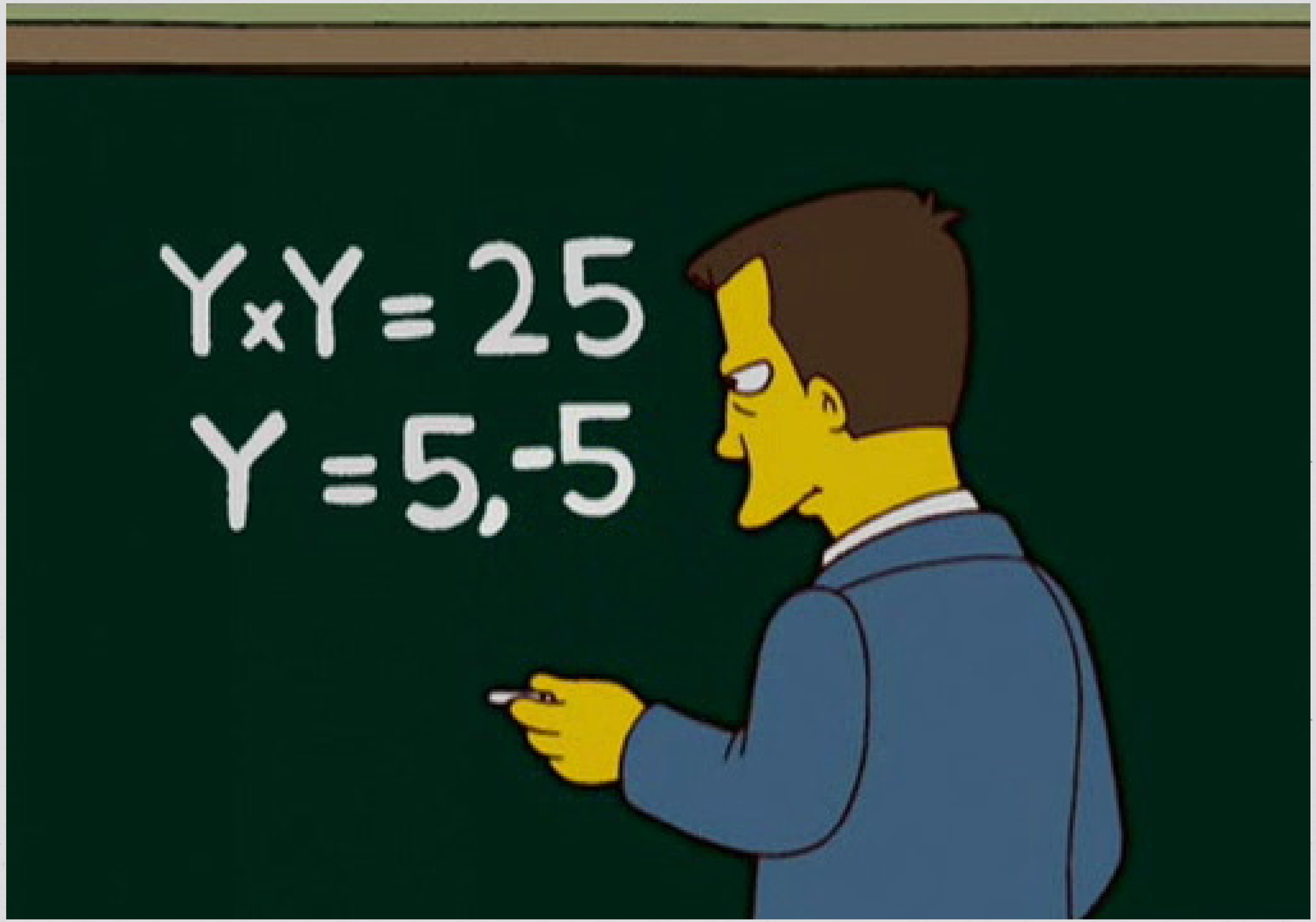
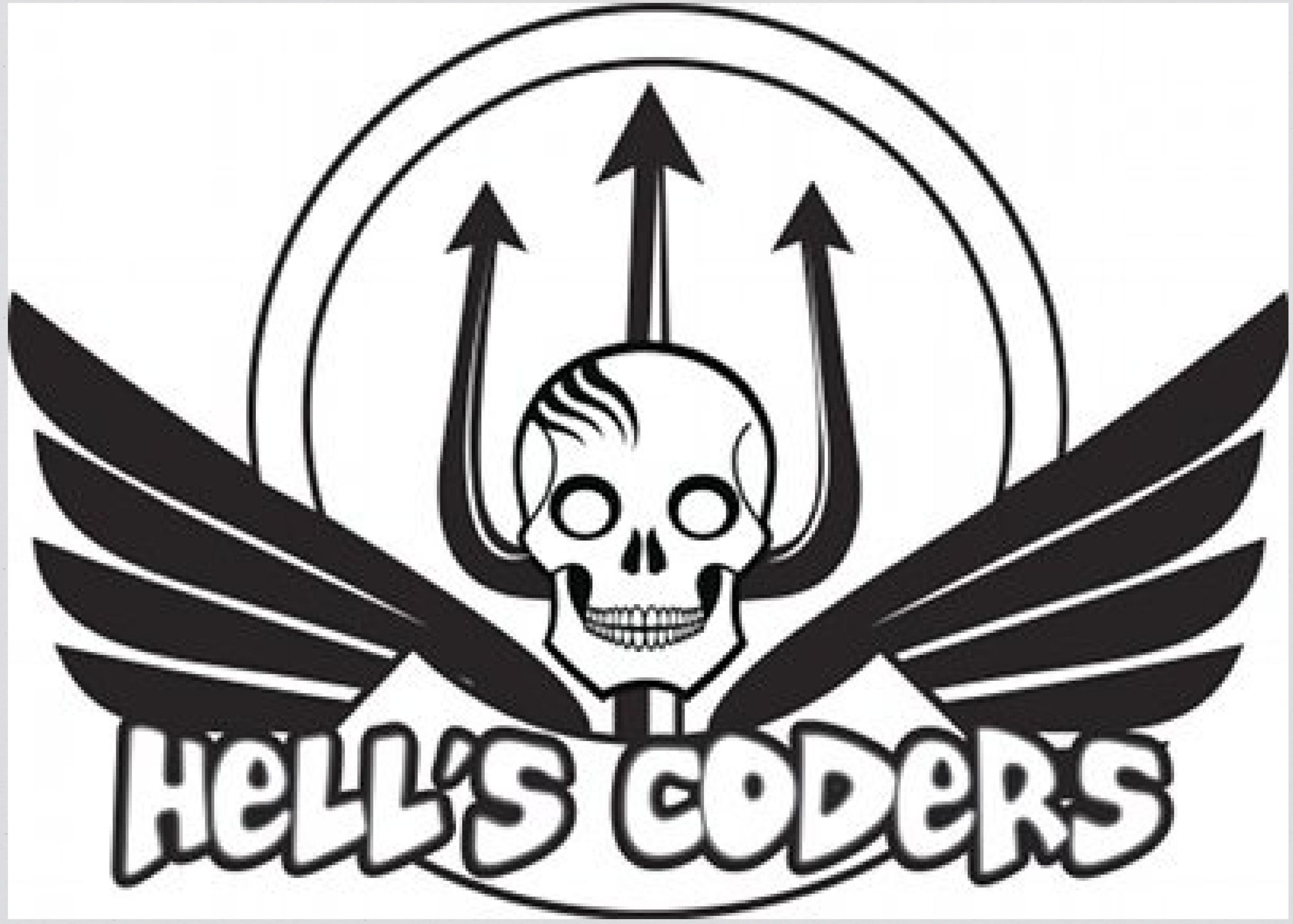
CODE: CLASS 2

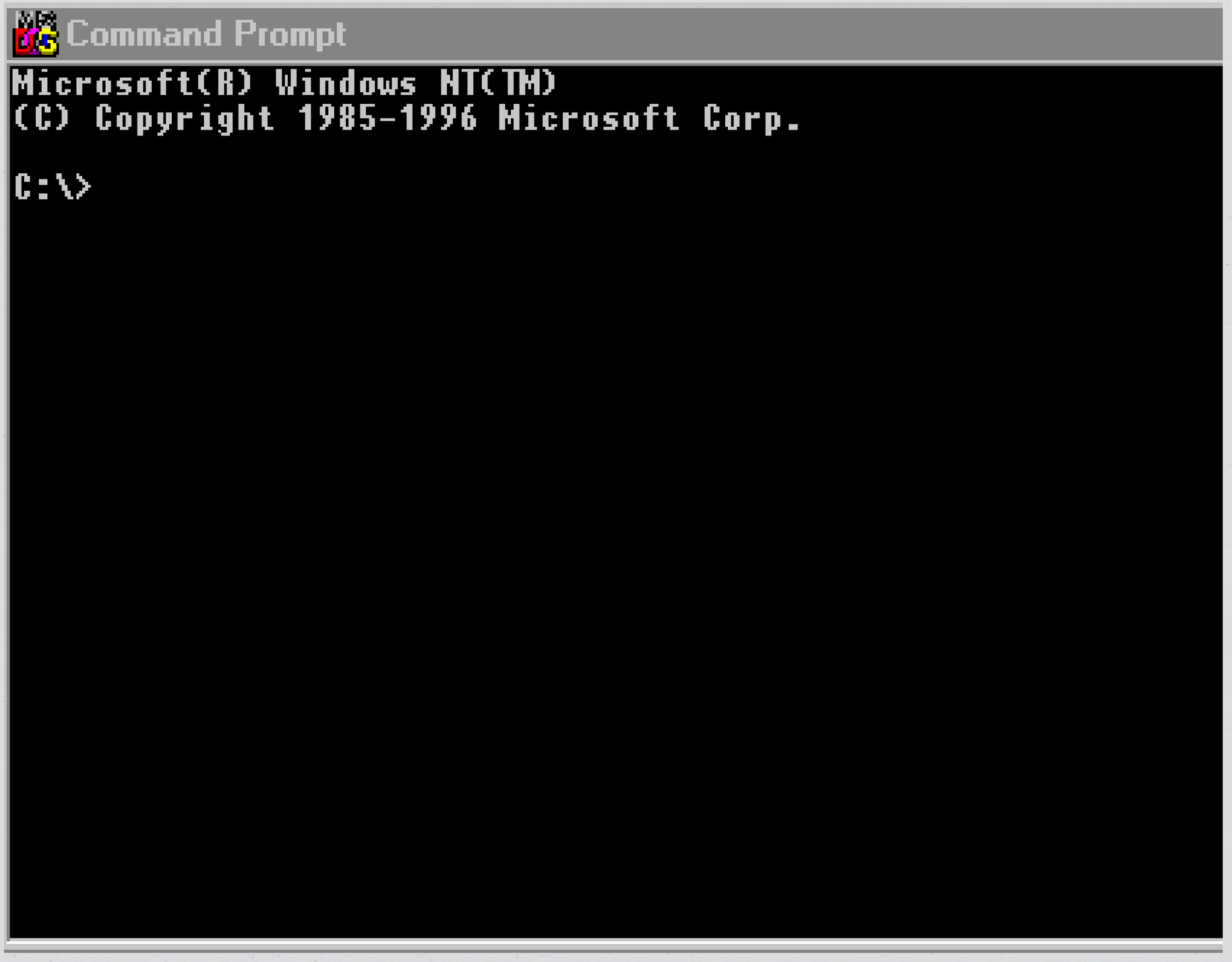
Introduction, Outputs, And Drawin' Stuff!

So You Think You Can't Code?

Monday, July 30, 12


Y×Y=25
Y=5,-5

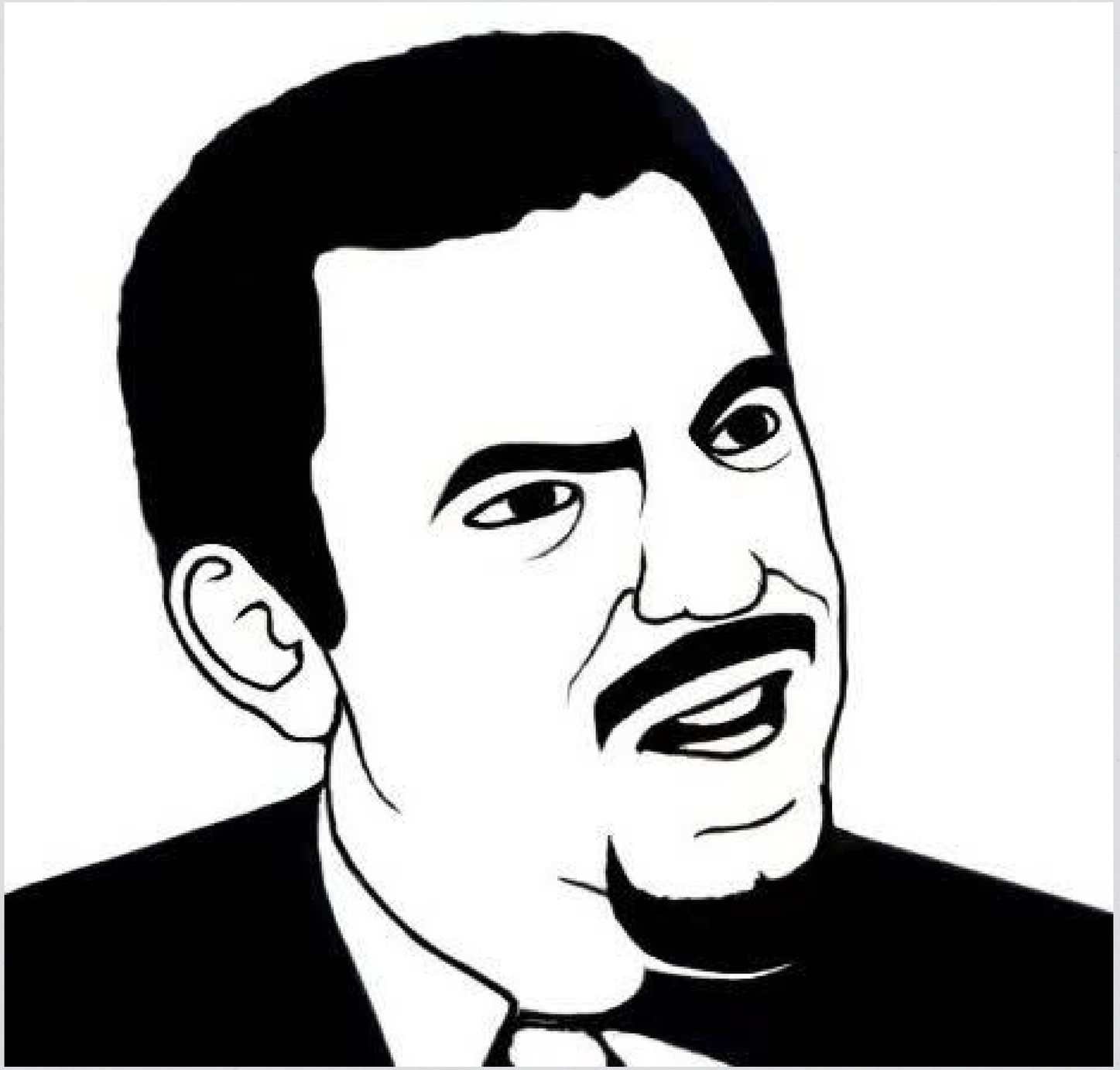

HELL'S CODERS

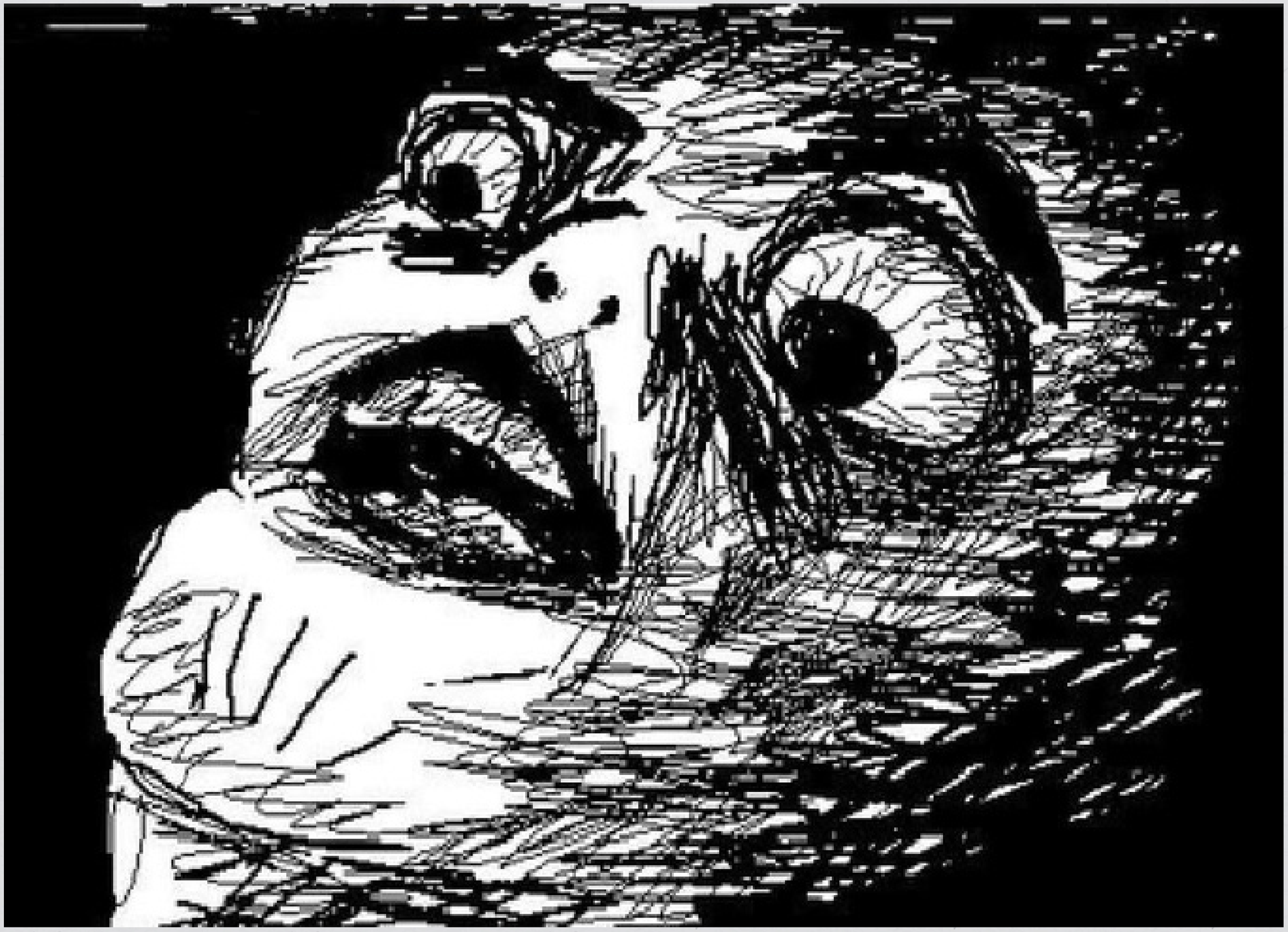

Command Prompt
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
C:\>


HEY
YOU CAN DO IT
Troll.me

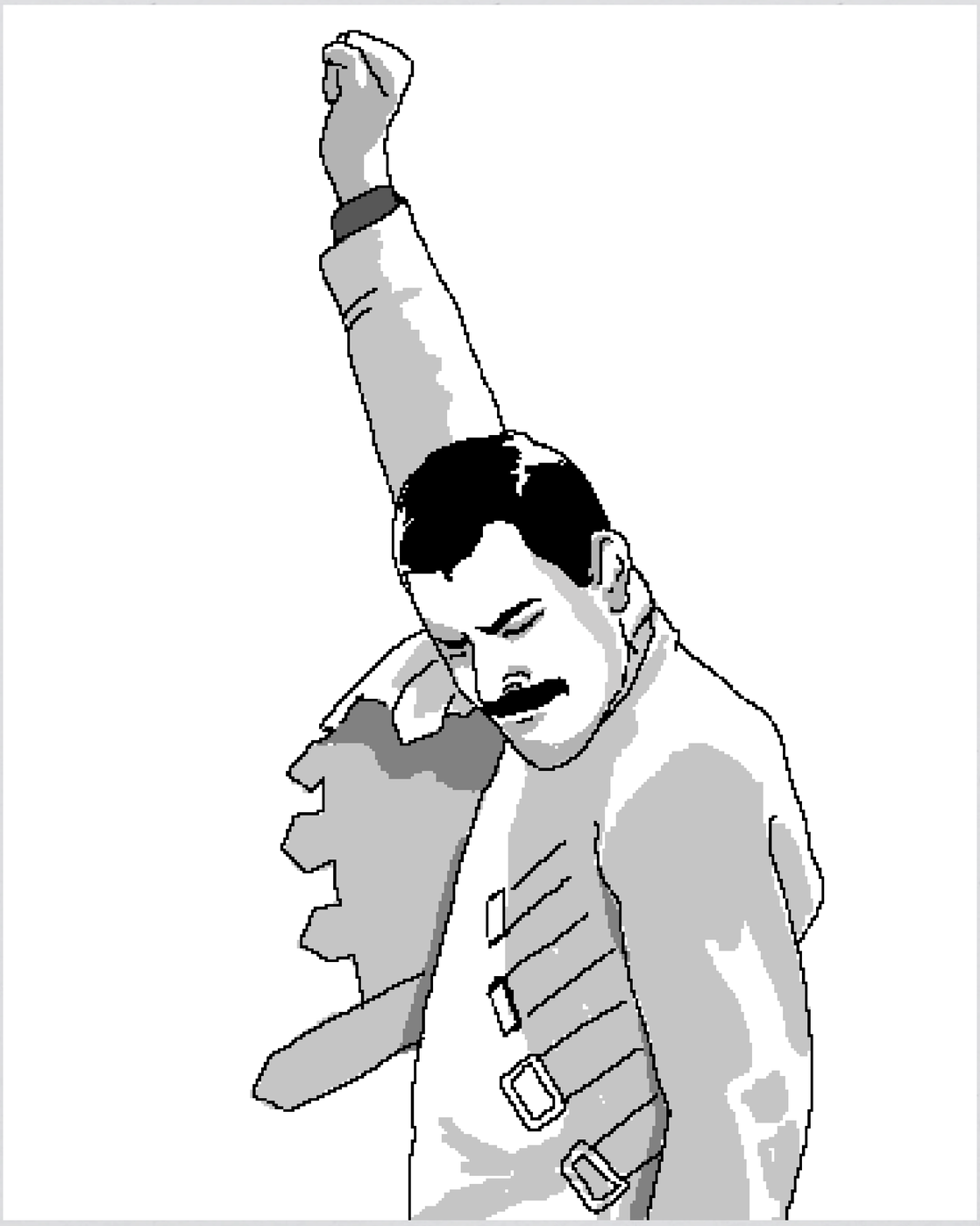
EVER TRIED.
EVER FAILED.
NO MATTER.
TRY AGAIN.
FAIL AGAIN.
FAIL BETTER.

Samuel Beckett (1906-1989)

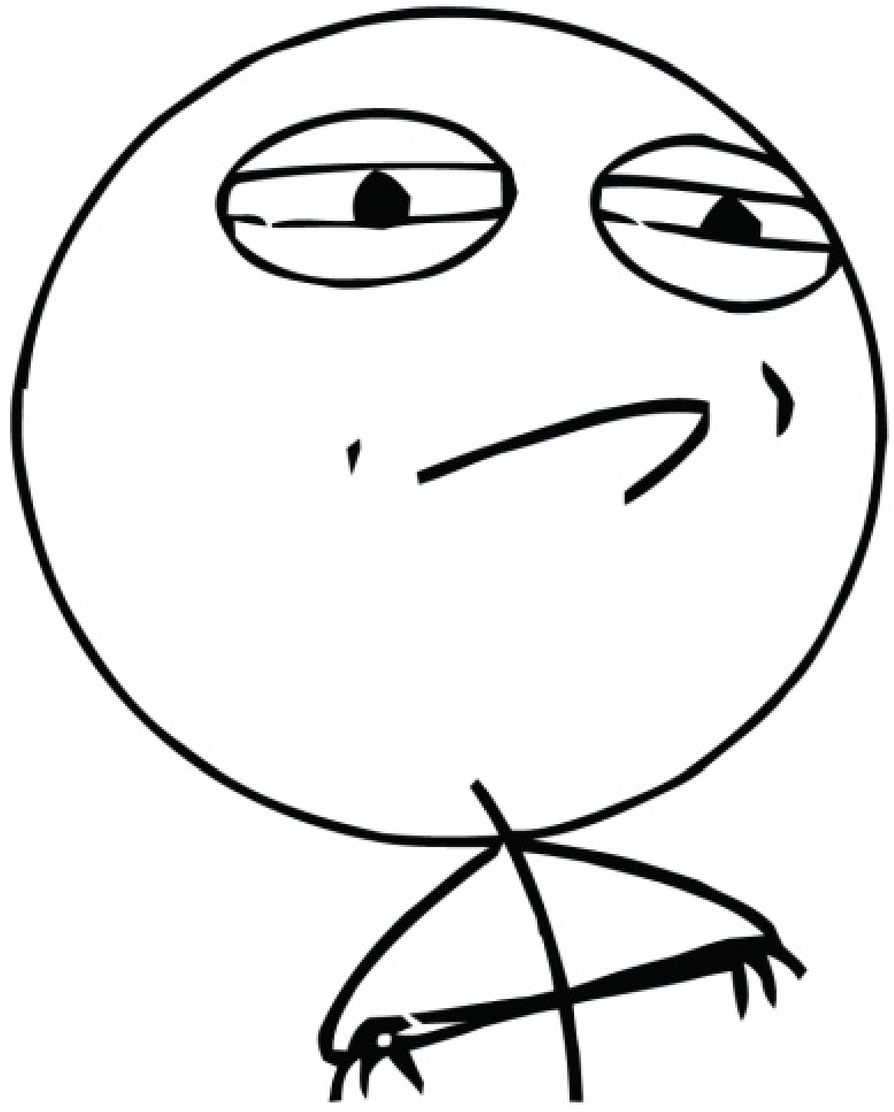





FFFFFFF
FFFFFFF
FFFFFF
FFFUU
UUUU
UUUU
UUUU
UUUU
UUUU-



CHALLENGE ACCEPTED



Jane Friedhoff

game designer + creative coder

- **Porfolio:** janefriedhoff.com
- **Blog:** janefriedhoff.com/blog
- **Email:** jane.friedhoff@gmail.com
- **Twitter:** @jfriedhoff
- **Github:** friej715

interests: mechanic design, interactive installations, physical + vocal creativity, public ridiculousness

code toolbox: Processing, openFrameworks/C++, MaxMSP, Arduino, CSS/HTML. currently learning PHP, javaScript, && jQuery.

What Is Processing?

Monday, July 30, 12

WHAT IS PROCESSING?
- Processing is what we call an IDE.

What Is An IDE?

WHAT IS AN IDE?

- Stands for Integrated Development Environment
- Basically, gives you all the things you need to create an application

What is an IDE?

Cooking

- Recipe/ingredients
- Stove/oven
- Taste-tester

Monday, July 30, 12

WHAT IS AN IDE?
- Think about it like a kitchen. If you're going to bake a cake, you need:
    - Recipe/ingredients/bowl to put them together in
    - An oven to turn those things into a finished product
    - A taste-tester, to prevent you from serving something terrible/poisonous

What is an IDE?

Cooking

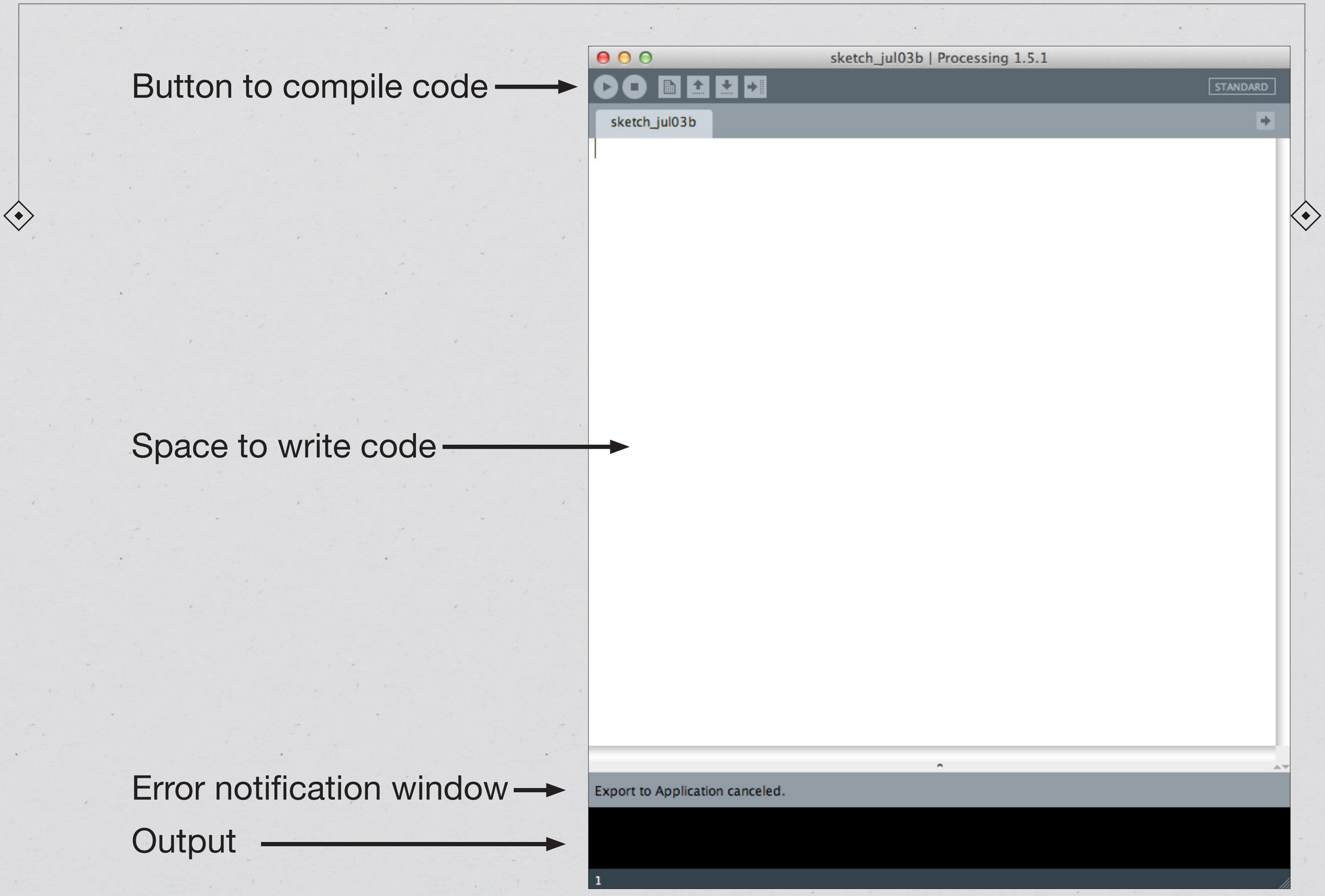
- Recipe/ingredients
- Stove/oven
- Taste-tester

Coding

- Text editor
- Compiler
- Debugger

Monday, July 30, 12

WHAT IS AN IDE?
- The situation with coding is analogous. You need:
    - Space to write your code (which is basically just commands and instructions)
    - A thing to transform that writing into an actual app
    - Something to check the code for errors and notify you of them



Monday, July 30, 12

WHAT IS AN IDE?

- Diagram of where all this is in Processing
- Output: basically, where the computer can feed you information that isn't visually apparent. Like a doctor's stethescope.

Taking Notes In Your Code

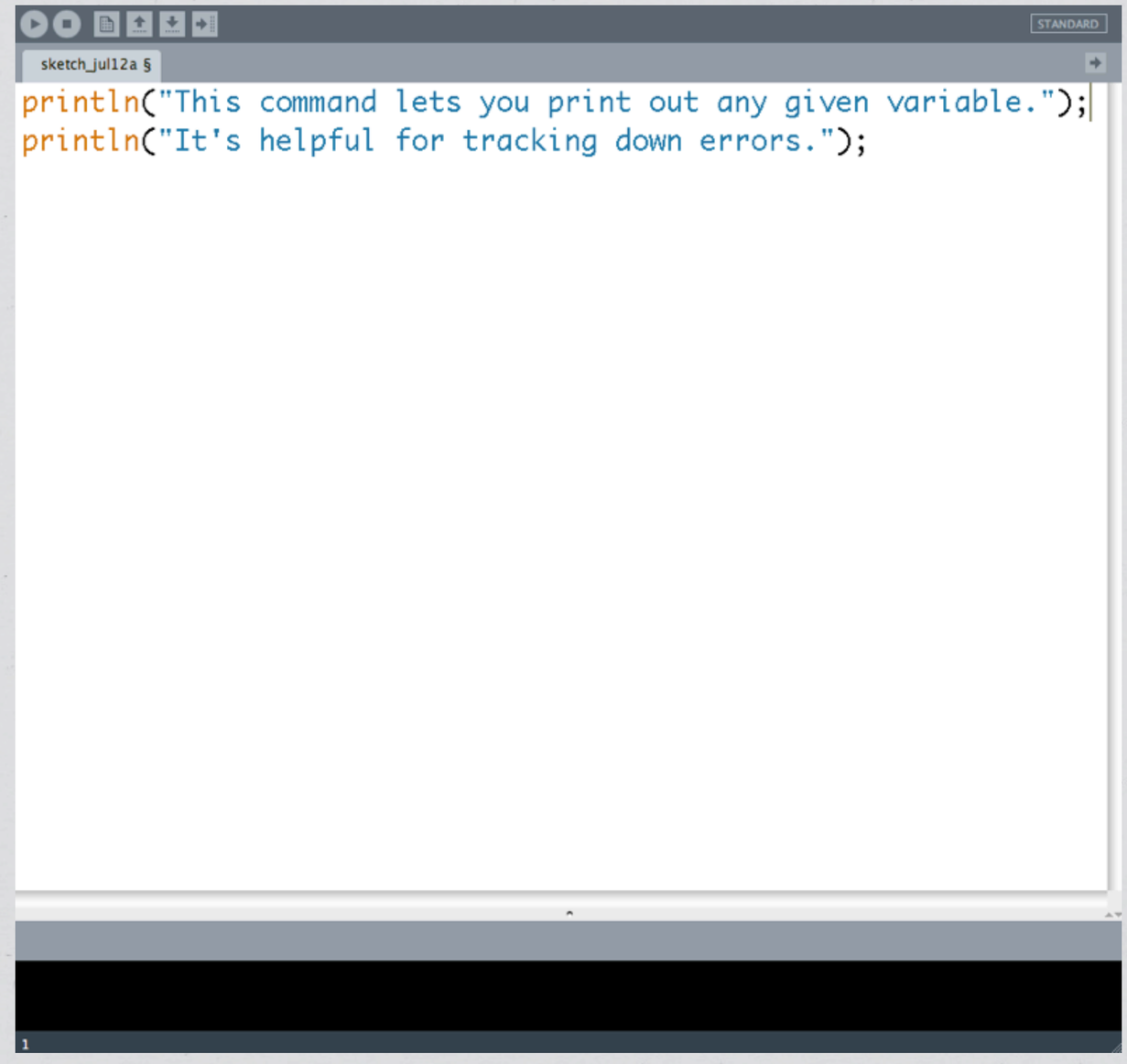
TAKING NOTES

- The same way you take notes to remember a class, you should take notes in your code to help you remember what is going on
- These notes: “comments”


le coding
Imports System
Public Class Employee
Private _name As St
Private _salary As I
Public ReadOnly Pro
Get
Return _nam
End Get
End Property
Public ReadOnly Pro
Get
Return _sal
End Get
End Property
Public Sub Increase
Enough for today, saving time!
Always put enough comments in your code!
Opening file 6 weeks later...
WHAT THE FUCK


thousands of lines and zero comments
merry christmas and a happy bus accident
theprofoundprogrammer.tumblr

Peeking Under The Hood



Monday, July 30, 12

PRINTLN: LOOKING UNDER THE HOOD

- println():
  - Command (indicated by parentheses)
  - Prints out whatever's in its parentheses
- Useful because:
  - Say you try to draw a circle on screen and you can't see it. Can either guess or just print out its location
  - Lets you figure out where your code stops working
  - (Right now) Gives immediate feedback!

~~Syntax Proper~~

Proper Syntax

SYNTAX

- May have noticed () and ; -- computers use punctuation just like we do.
- Syntax (basically code grammar) is super important, because computers are highly literal

Humans

* End sentences with a period (or other punctuation mark)
    * Hi there.
    * Where's the coffee?
* Use opening and closing punctuation
    * So I said, "You look amazing!"

SYNTAX
- Human languages have syntax and accepted grammar. It helps us know what's going on.

Otherwise it would be difficult to know where a sentence begins it would also be difficult to know where it ends all of this is important for our comprehension of language and what it means otherwise we might misconstrue the meaning of a sentence you know

SYNTAX

- Human languages have syntax and accepted grammar. It helps us know what's going on.

Or imagine dialogue in a book--the following things are very different:

- Jane said, “I looked nice.”

- “Jane said I looked nice.”

SYNTAX

- Human languages have syntax and accepted grammar. It helps us know what’s going on.

Computer "Grammar"

Humans

* End sentences with a **period** (or **other punctuation mark**)
  * Hi there.
  * Where's the coffee?
* Use **opening and closing** punctuation
  * So I said, "You look amazing!"

Computers

* Generally end commands with a **semicolon**
  * println("Syntax matters!");
* Use **opening and closing** punctuation
  * println("Syntax matters!");

Monday, July 30, 12

SYNTAX
- Same with computers
- Think about println--if we didn't have closing quotes/parens/semicolon, the thing wouldn't know when to stop printing, and could start printing the rest of our code

Inside The Parentheses

Real Life

getMeSomeLunch(sandwich);

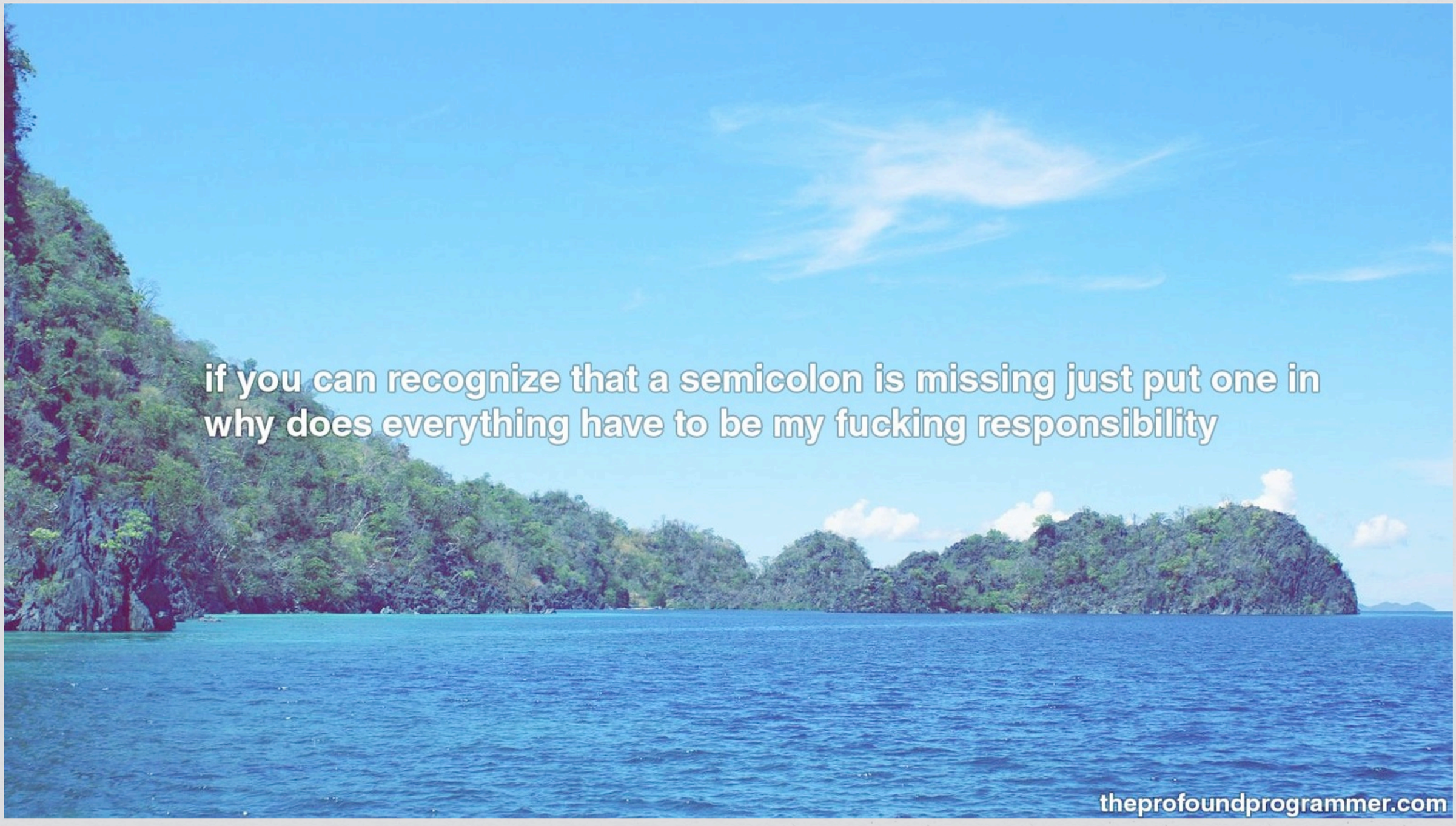
Get me some lunch, specifically, a sandwich.

Coding

println(“Hello!”);

Print something, specifically, hello, to the console.

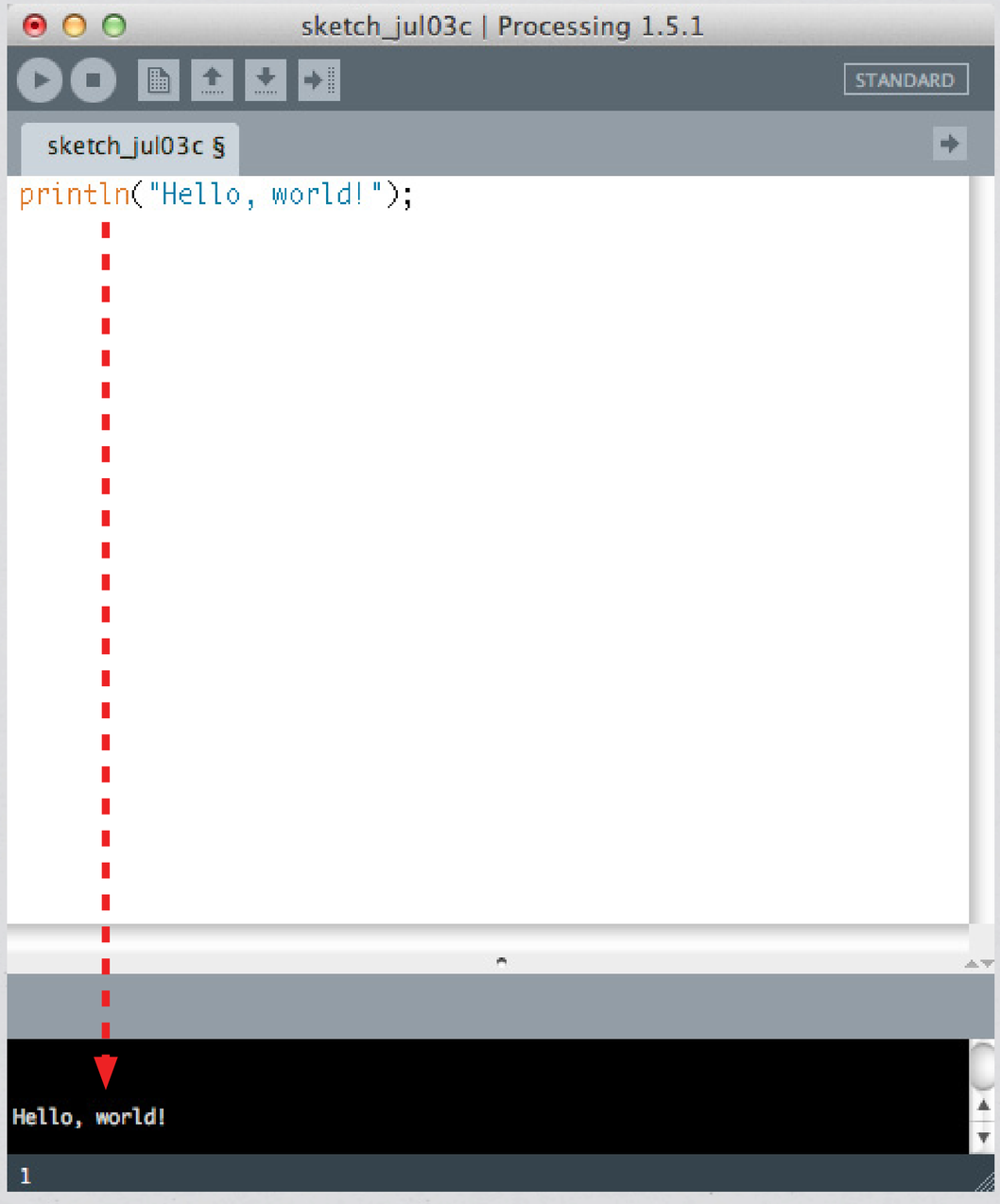
PARAMETERS
- Commands + details of those commands (called **parameters** or **arguments**)
- Specifies exactly what we want the command to do (print what? draw something where?)


if you can recognize that a semicolon is missing just put one in
why does everything have to be my fucking responsibility
theprofoundprogrammer.com

Try using the println(); command to print "Hello, world!"

TRY THIS

- Try out println. Do you see where it pops up?



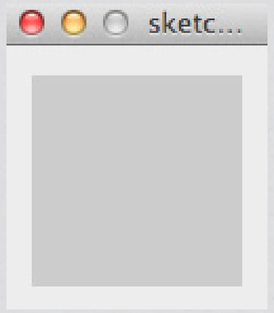
CONGRATS

- You just wrote your first program

Let's Get Visual

Monday, July 30, 12

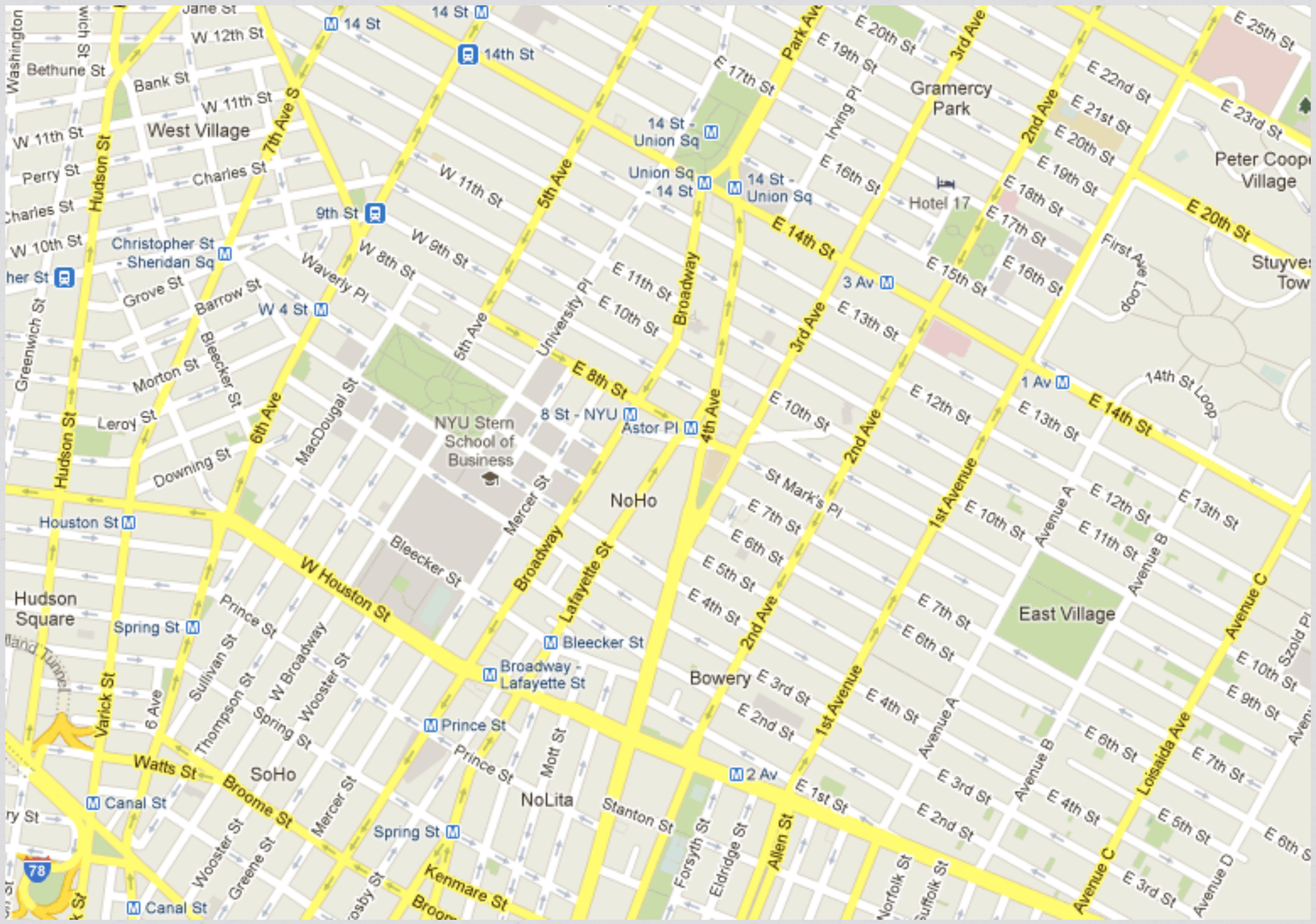
LET'S GET VISUAL
- Moving on to visuals since that's fun



LET'S GET VISUAL

- When you draw things, they show up in this window
- Default size: 100px by 100px (can change, of course)

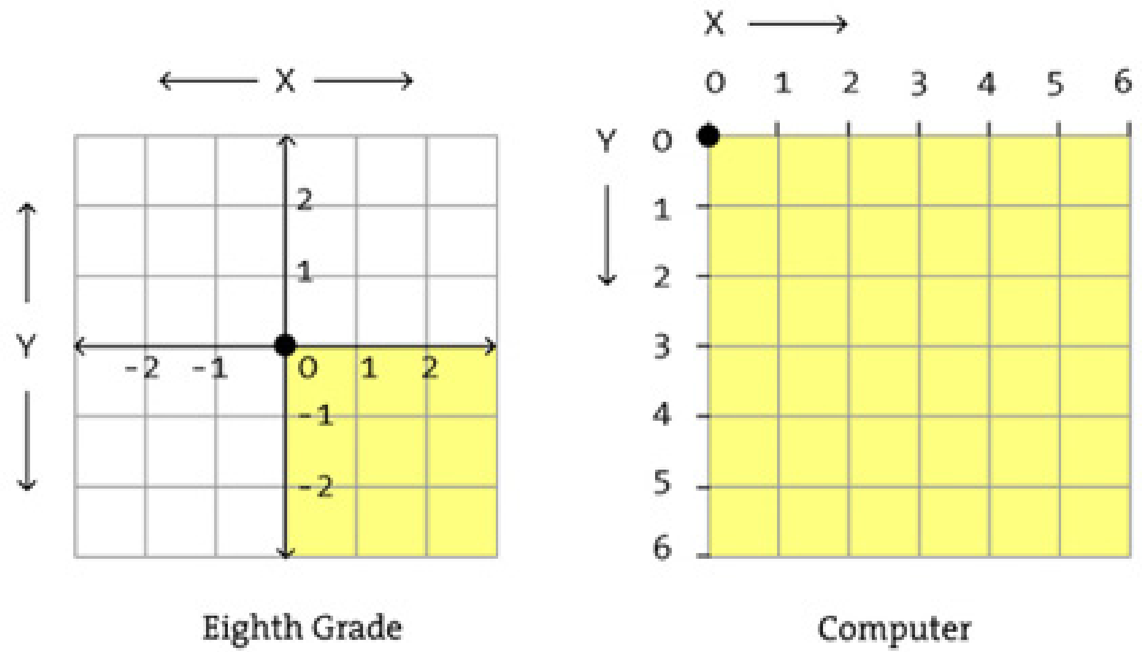
Locations In Processing



Monday, July 30, 12

LOCATIONS IN PROCESSING

- Say I asked you to meet me at a coffeeshop downtown--and that was all I said. Without knowing the intersection, you wouldn't have a clue about where to go, and would either end up at the wrong place or just refuse to go
- Having named, numbered streets and addresses help us orient ourselves and figure out how to get where we're going



As you go right, the x value gets bigger.

As you go down, the y value gets bigger.

Monday, July 30, 12

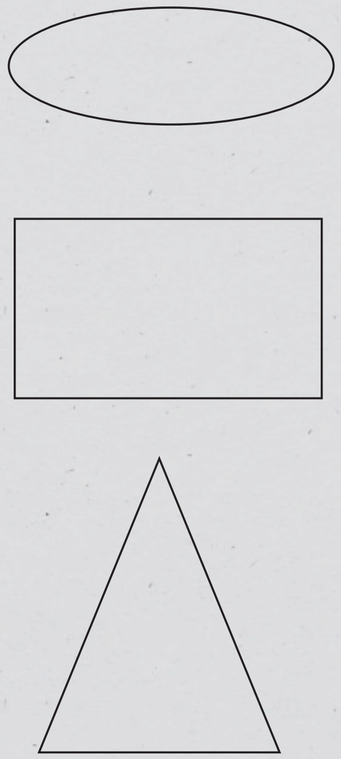
LOCATIONS IN PROCESSING

- Processing also uses a grid system, although not the one we're used to.
- The x-axis refers to our horizontal location (e.g. west or east Manhattan)
    - Right: bigger
    - Left: smaller
- The y-axis refers to our vertical location (e.g. uptown or downtown Manhattan)
    - Up: smaller
    - Down: bigger
- We write coordinates with x first, then y.

Drawin' Things!

DRAWIN' THINGS

- Now that we know how to reference a location on screen, we can start putting stuff there!



ellipse(x position, y position, width, height);

rect(x position, y position, width, height);

triangle(x pos of 1st point, y pos of 1st point, x pos of 2nd point, y pos of 2nd point, x pos of 3rd point, y pos of 3rd point);

Monday, July 30, 12

DRAWIN' STUFF

- Remember the println command? To draw shapes, we use commands as well. Each shape has a different command
- Notice the stuff in parentheses? That's the stuff the computer needs to know before it can draw the shape you want

Real Life

getMeSomeLunch(sandwich, footlong, veggies);

Get me some lunch, specifically, a sandwich that is a footlong with veggies.

WHY PARENTHESES?

- Where we put the **parameters** of the command--basically, the details
- If I asked you to get me some lunch, you'd want to know some stuff about the lunch I wanted before you got it

Real Life

getMeSomeLunch(sandwich, footlong, veggies);

Get me some lunch, specifically, a sandwich that is a footlong with veggies.

Coding

ellipse(10, 20, 50, 60);

Draw me an ellipse at x-position 10 and y-position 20, with a width of 50 pixels and a height of 60 pixels.

Monday, July 30, 12

WHY PARENTHESES?
- In code, the computer needs to know the details of your command before it does it
- Computers, in general, CANNOT GUESS. They don't make do, they just freak out and crash

line(x pos of 1st point, y pos of 1st point, x pos of second point, y pos of 2nd point);

point(x pos, y pos);

MOAR DRAWIN’

- Here are some more things you can draw

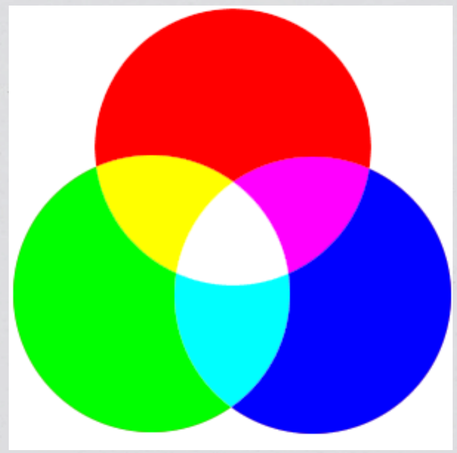
“The Colors, Duke!”

THE COLORS, DUKE!

- Drawin’ isn’t fun without color, so we’ll cover that first

RGB: Red, Green, Blue

Additive Color:
The more you add, the closer you get to white.



Monday, July 30, 12

THE COLORS, DUKE!
- Computers generally think of any color as some mixture of red, green, and blue (RGB)
    - Can do HSB/HSV, but not covering that right this sec, ask

Measured from 0 - 255 (lowest value to highest).

Red: 255
Green: 242
Blue: 0

Red: 37
Green: 64
Blue: 143

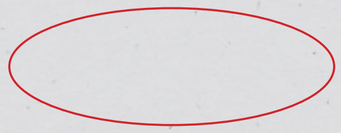
Red: 37
Green: 216
Blue: 48

Red: 255
Green: 255
Blue: 255

- Numerical values go from 0-255

Filling and Outlining

```
fill(210, 35, 42);
ellipse(x position, y position, width, height);
```



```
stroke(210, 35, 42);
ellipse(x position, y position, width, height);
```

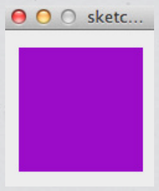
```
noStroke();
fill(210, 35, 42);
ellipse(x position, y position, width, height);
```


- **Fill:** basically using the paint bucket, used for insides of shape/letters
- **Stroke:** coloring the outline of the shape
- Can also use noFill(); or noStroke();

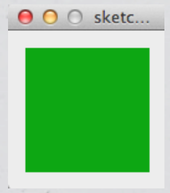
Background Colors

BACKGROUNDS

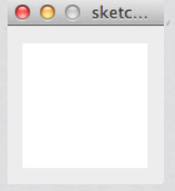
- Can also color the background of the sketch; like the paper color



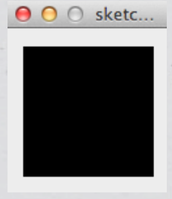
background(155, 12, 200);



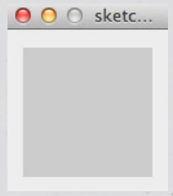
background(13, 167, 19);



background(255);



background(0);



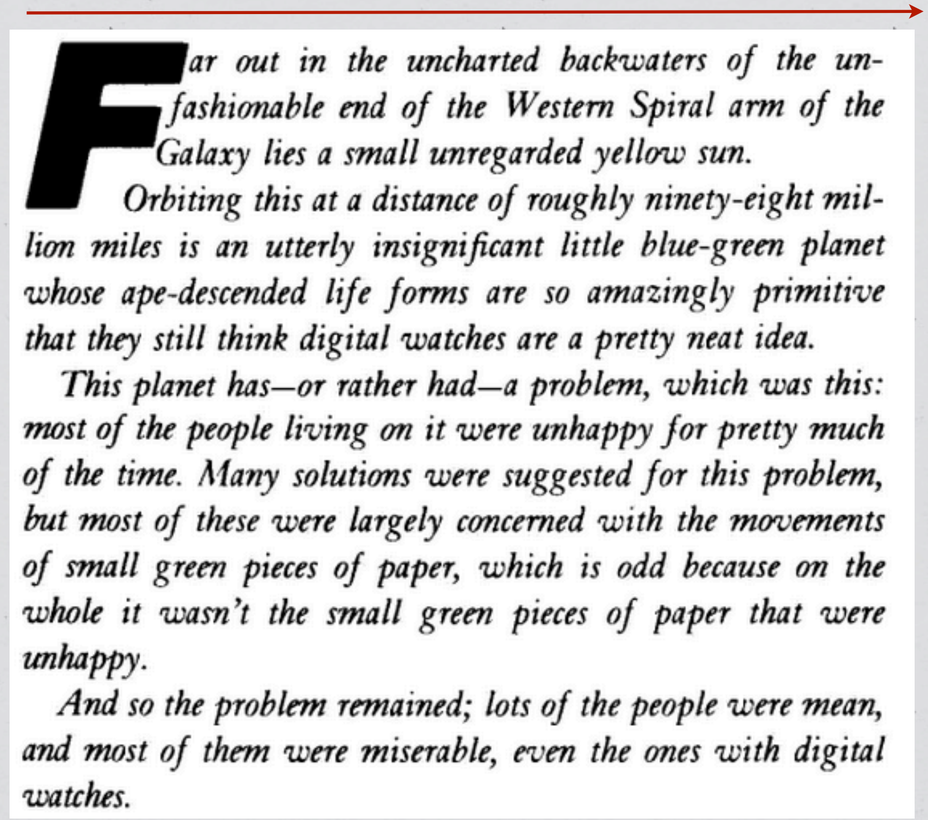
background(100);

- Also RGB
- If you ever want all three values to be the same (for fill, stroke, bg, etc.), you can just type in one number. The R,G,B values will all equal that number--resulting color is between white and black

Inherent Order

INHERENT ORDER:
- Processing interprets code and executes commands in an important way
- Goes from top to bottom

Humans



Far out in the uncharted backwaters of the unfashionable end of the Western Spiral arm of the Galaxy lies a small unregarded yellow sun.

Orbiting this at a distance of roughly ninety-eight million miles is an utterly insignificant little blue-green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea.

This planet has—or rather had—a problem, which was this: most of the people living on it were unhappy for pretty much of the time. Many solutions were suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

And so the problem remained; lots of the people were mean, and most of them were miserable, even the ones with digital watches.

INHERENT ORDER:
- People read in a particular way: from left to right, and top to bottom
- Things that happen later in time happen closer to the bottom, or closer to the end

Humans

Far out in the uncharted backwaters of the unfashionable end of the Western Spiral arm of the Galaxy lies a small unregarded yellow sun.

Orbiting this at a distance of roughly ninety-eight million miles is an utterly insignificant little blue-green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea.

This planet has—or rather had—a problem, which was this: most of the people living on it were unhappy for pretty much of the time. Many solutions were suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

And so the problem remained; lots of the people were mean, and most of them were miserable, even the ones with digital watches.

Processing

We'll talk about automatically repeating these commands (i.e. looping) two classes from now.

Monday, July 30, 12

INHERENT ORDER:
- Same with code: code is read from left to right and top to bottom
- Things that happen later in code happen further down
- How might this affect the stuff you draw?

INHERENT ORDER:
- Things can be drawn on top of other things
- Fills apply to things that happen later on, if nothing stops them

Reliving Preschool

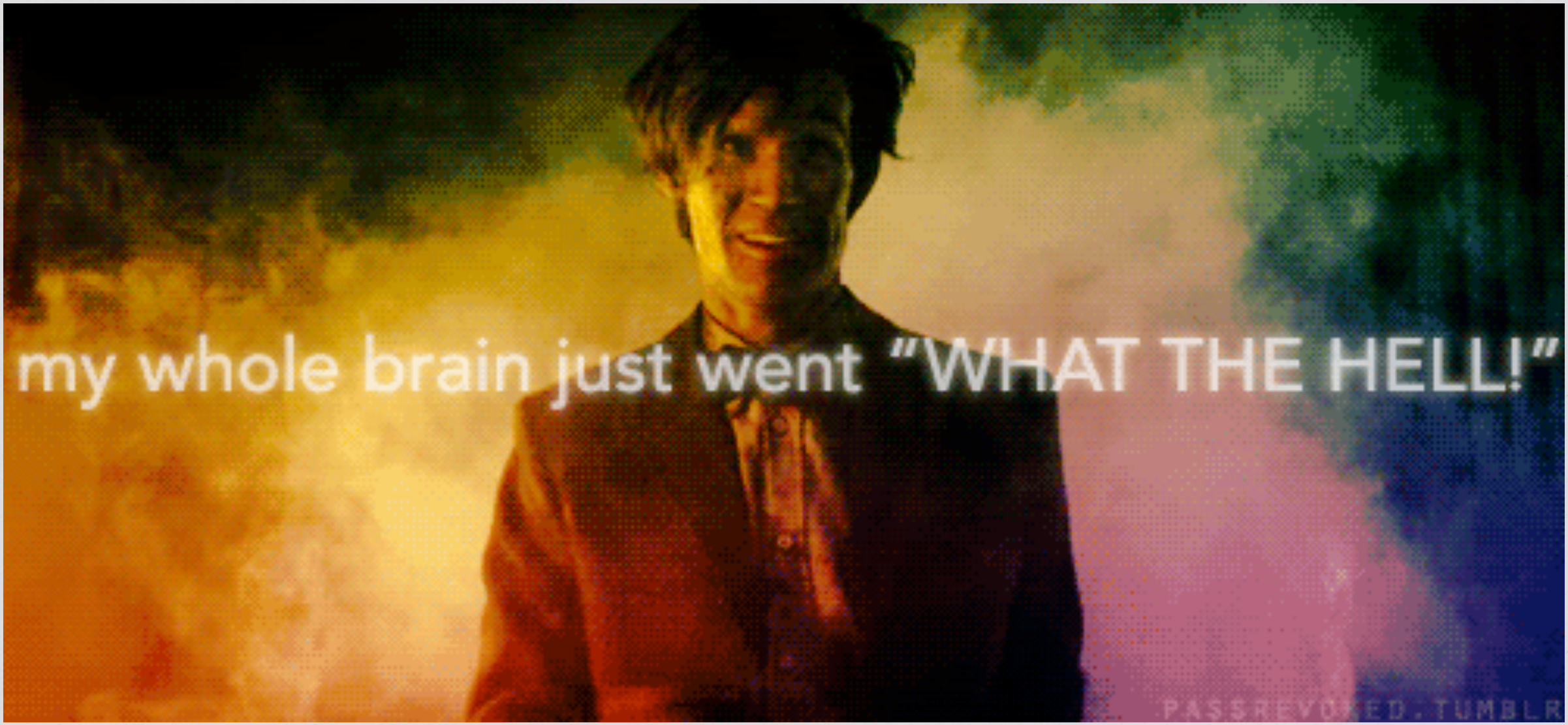
Try drawin' some shapes!
See how order affects the way things are drawn, and how where you put a fill affects what it colors.

Homework

Draw your section of the Mondrian piece.

Print out (or handwrite) your code and bring it to class.

Read Code, Chapter 8.


my whole brain just went "WHAT THE HELL!"
PASSREVOKED.TUMBLR