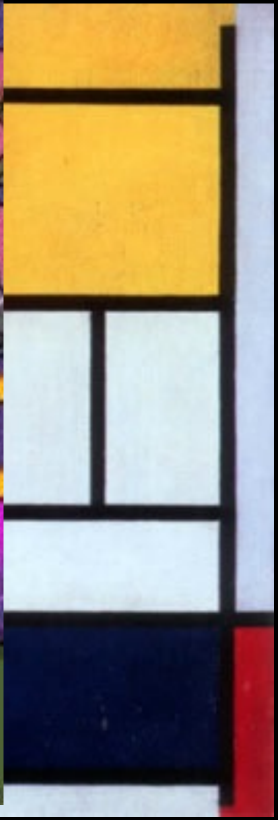
GRAY AREA
FOUNDATION
FOR THE ARTS

ONE ART, PLEASE!
quickmeme.com

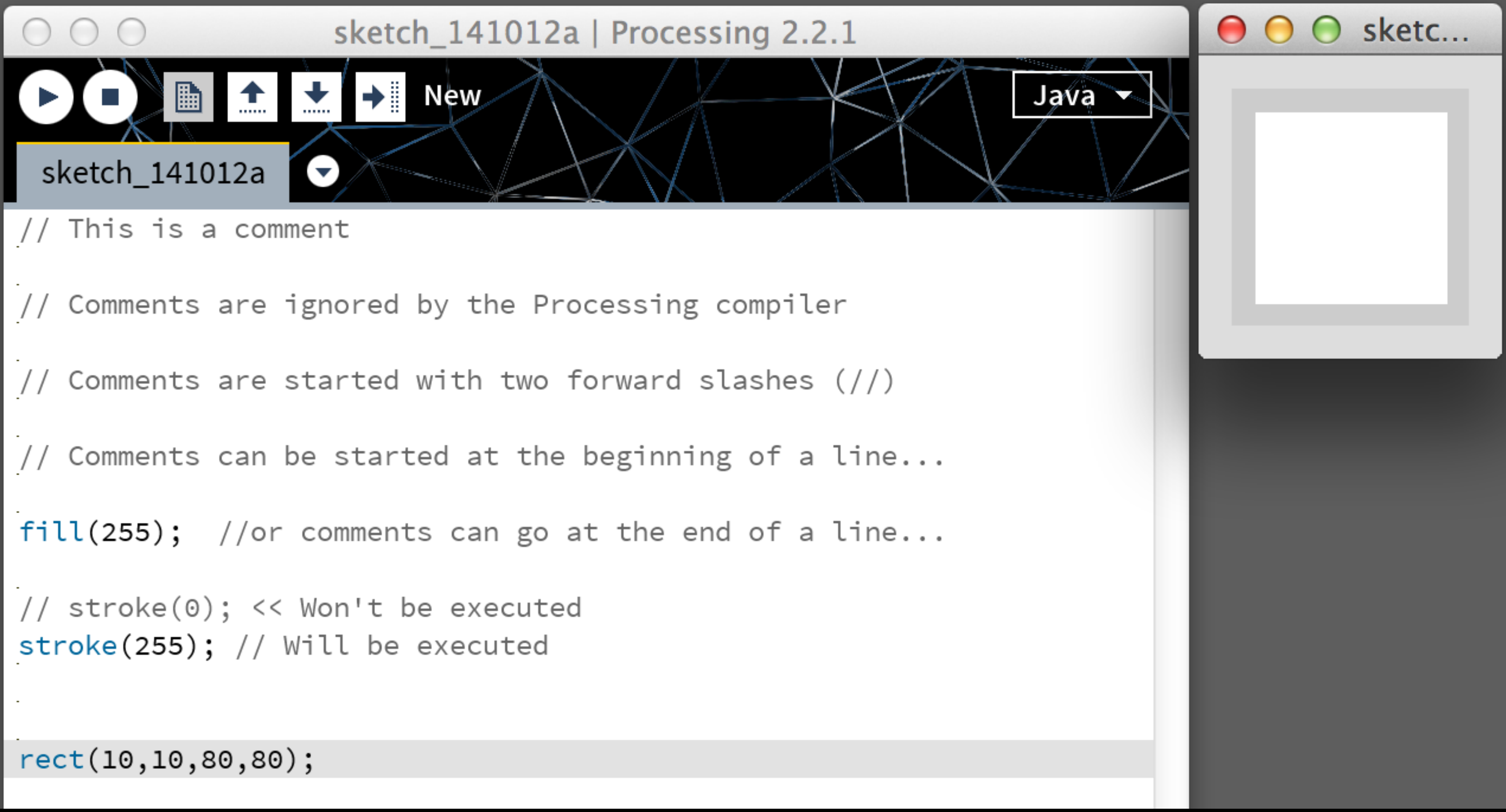


Draw me a house!

WAR
IS A
RACKET

A Few Rules

Comment your code!



```
// This is a comment

// Comments are ignored by the Processing compiler

// Comments are started with two forward slashes (//)

// Comments can be started at the beginning of a line...

fill(255);  //or comments can go at the end of a line...

// stroke(0); << Won't be executed
stroke(255); // Will be executed


rect(10,10,80,80);
```

A Few Rules

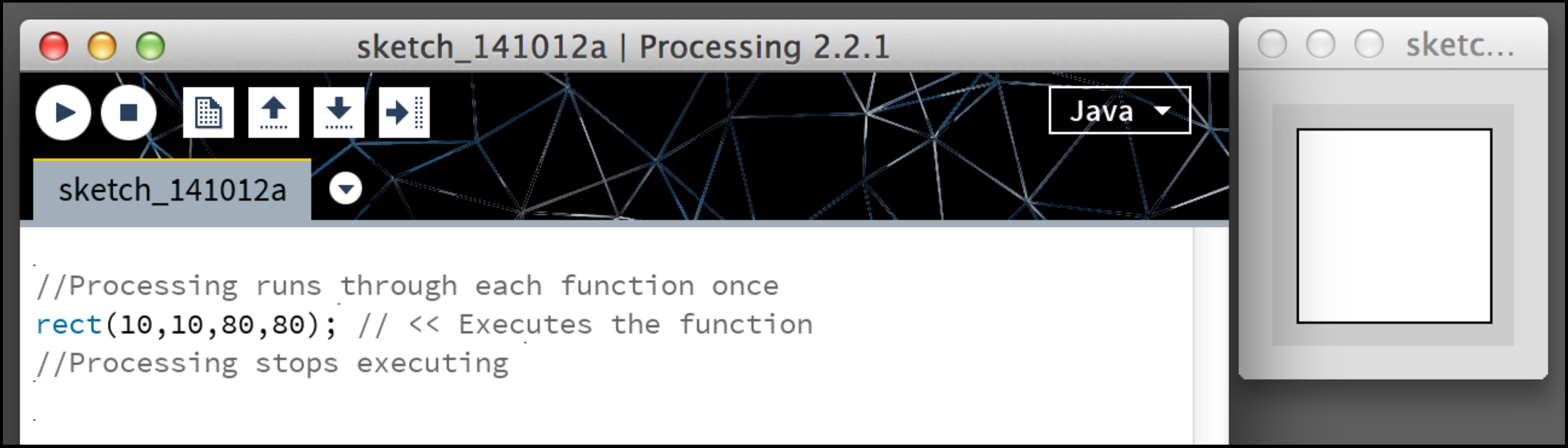
println() is your friend!

A Few Rules

Setup, Draw, ???, **Profit!**

What happens when we Run?

- Processing performs two basic operations:
  - Take input from the user (key/mouse/etc.)
  - Change pixel colors in the window



```
//Processing runs through each function once
rect(10,10,80,80); // << Executes the function
//Processing stops executing
```

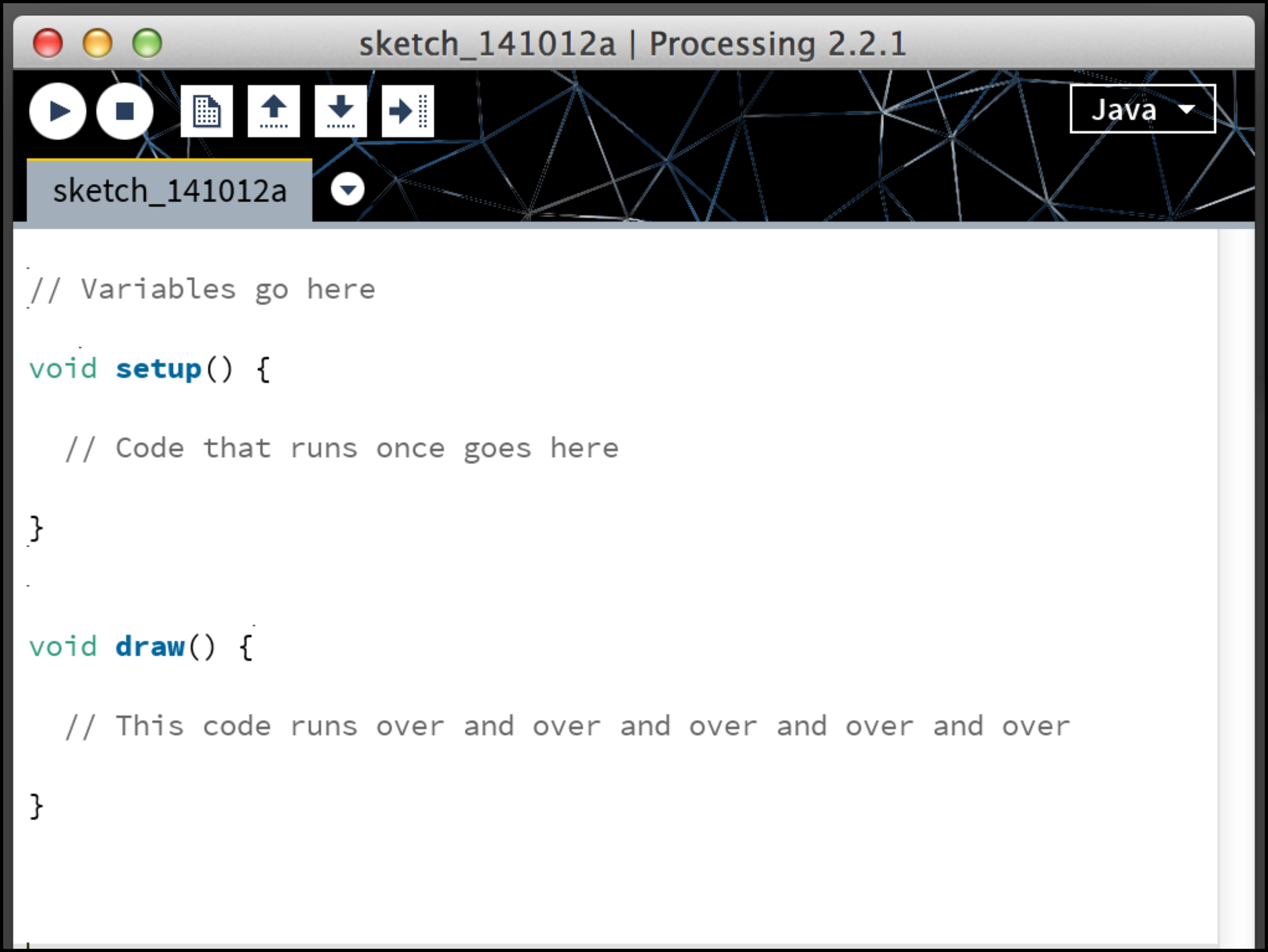
Setup, Draw, ???, Profit!

- Code only executes once
    - Because of this, no animation or interaction is possible
- Processing has two very important built-in functions, setup() and draw()

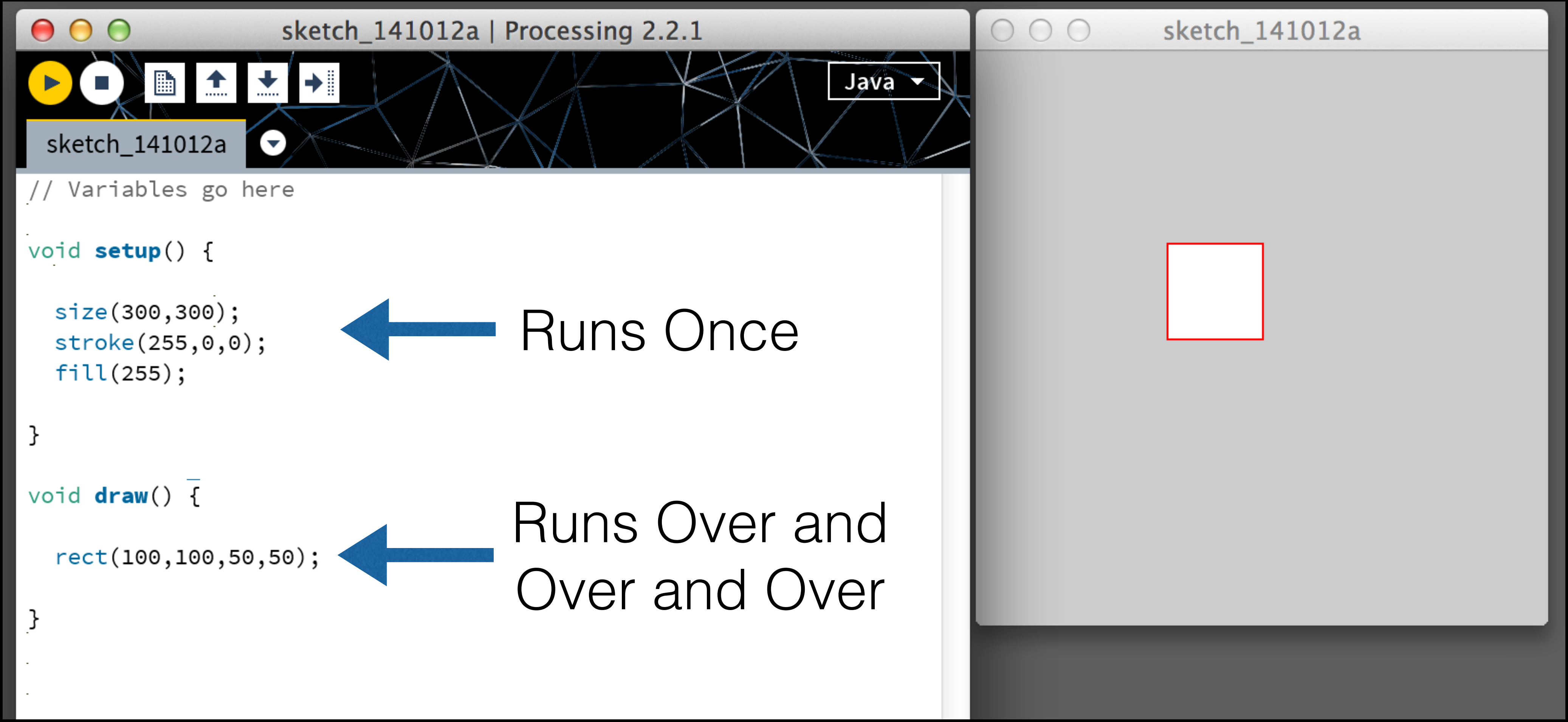
Setup, Draw, ???, Profit!

- setup() is used for commands that should be run only once
- draw() runs over and over, so drawing code goes in here

Setup, Draw, ???, Profit!



```
// Variables go here

void setup() {

  // Code that runs once goes here

}


void draw() {

  // This code runs over and over and over and over and over

}
```



```
// Variables go here

void setup() {

  size(300,300);
  stroke(255,0,0);
  fill(255);

}

void draw() {

  rect(100,100,50,50);

}
```

Runs Once

Runs Over and Over and Over

Setup, Draw, ???, Profit!

- The speed at which the draw command runs is controlled by frameRate() function
    - frameRate(# of runs per second)

Variables

- This is a core concept in all of programming
- Variables are tiny lockers to store data
- You first create a locker, then store data in it

```
data_type  variable_name;

variable_name = value;
```

Variables

- Processing gives us a few variables that we can use in our programs
    - width
    - height
    - mouseX
    - mouseY

Variables

- Width/Height
  - correspond to the values passed to the size(width,height) function
- Useful for creating sketches that do not depend on a specific sized window

Variables

- mouseX/mouseY
    - corresponds to the current position of the mouse cursor

Variables

- There are different types of variables in Processing:
    - int - whole number
    - float - decimal number
    - char - a single letter
    - string - a group of letters

Variables

- Processing has block level scope
  - Variables declared inside curly braces, can only be used inside those curly braces
  - Variables declared outside of all curly braces (i.e. at the top of the editor) can be used everywhere. They are called **global**

If Then What

- Another core concept in programming is conditionals
- Using if statements, we can ask questions mid-stream in our code
- We can change how our code acts depending on the answer

Code Challenge

- Check where the mouse is, and change the background color accordingly

Code Challenge

- Create two circles, one that is always under the mouse, and another that is always 20 pixels to the right of the mouse

Loops and Loops

- Remember the pain of writing line() functions over and over and over? No more!!
- Another core programming concept: **the for loop**

For Loops

```
for (int i = 0; i < something; i++) {

    //Code Goes Here



}
```

Code Challenge

- Create 4 squares that are drawn around the current mouse position, using a for loop

Functions

- Functions are blocks of code which can be called
- We’ve been using functions since starting with Processing
- Writing your own function can be easy and powerful!

Functions

```
void myfunction() {

    rect(0,0,100,100);

}
```

Functions

- Return type
- Function name
- Code block

Functions

- Functions can be declared anywhere
- It is typically a good idea not to declare them in your draw loop or setup loop
- Once declared, you can use your function anywhere!

The Mouse and the Keyboard

- We know mouseX and mouseY
- There is also keyPressed
    - if (keyPressed == SOME KEY) …

The Mouse and the Keyboard

- Processing also provides a few functions for the Mouse
    - mouseClicked()
    - mouseDragged()

Code Challenge!

- Draw lines with mouseClicked and mouseDragged

save()

- Writes the current state of the Processing screen to a file
  - save(‘filename.tif’);

Code Challenge - MS PAINT!