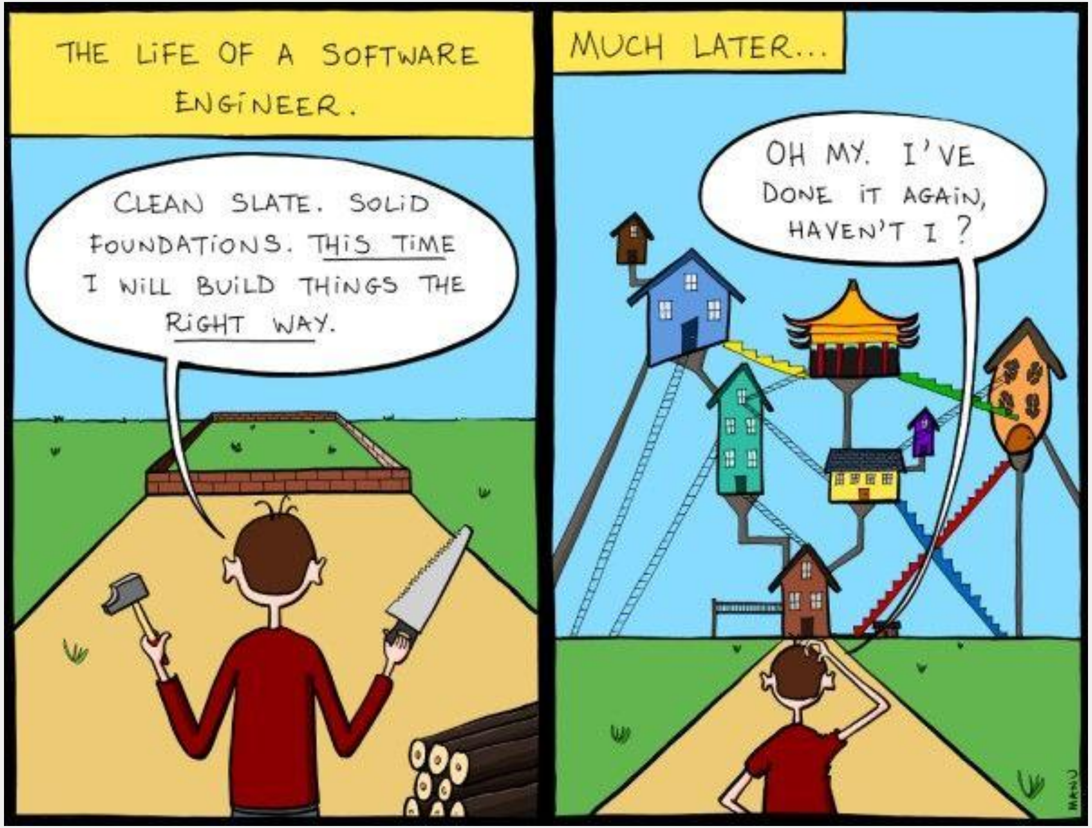


THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?
MANU

CS 110
Local vs Global Variables and Graphics

Adriana Picoral

Announcements

- Office hours (not all are in person, the in person are not all in 837)
- In this class, you are given 3 late days for programming assignments. What this means is that you are allowed to submit up to three programming assignments within 24 hours after the due date throughout the semester, without penalty.

Local and Global

- **Local Variable:** Is a variable with local scope
  - For example: A variable assigned inside of a function can only be used or modified within that function after the initial assignment
- **Global Variable:** Is a variable with global scope
  - For example: a variable declared outside of a function can be accessed or modified across multiple functions

Activity

```python
name = 'NAME'

def process_name():
    name = input('Type your name: ')
    first_letter = name[0]
    after_first = name[1:]
    name = first_letter.upper() + after_first.lower()

process_name()

print('Hi there', name)
```

What will this print?

Input: jACOB

Activity

```
name = 'NAME'

def process_name():
    name = input('Type your name: ')
    first_letter = name[0]
    after_first = name[1:]
    name = first_letter.upper() + after_first.lower()
    print('Hi there', name)

process_name()
```

What will this print?

Input: jACOB

Activity

What will this print?

```
def process_name(name_to_process):
    first_letter = name_to_process[0]
    after_first = name_to_process[1:]
    name = first_letter.upper() + after_first.lower()

name = input('Type your name: ')
process_name(name)

print('Hi there', name)
```

Input: jACOB

Activity

What will this print?

```
def process_name(name_to_process):
    first_letter = name_to_process[0]
    after_first = name_to_process[1:]
    name_to_process = first_letter.upper() + after_first.lower()

name = input('Type your name: ')
process_name(name)

print('Hi there', name)
```

Input: jACOB

Activity

What will this print?

```
def process_name(name_to_process):
    first_letter = name_to_process[0]
    after_first = name_to_process[1:]
    return first_letter.upper() + after_first.lower()

name = input('Type your name: ')
name = process_name(name)

print('Hi there', name)
```

Input: jACOB

Activity

```
name = input('Type your name: ')

def process_name():
    first_letter = name[0]
    after_first = name[1:]
    name = first_letter.upper() + after_first.lower()

process_name()

print('Hi there', name)
```

What will this print?

Input: jACOB

Activity

```
name = input('Type your name: ')

def process_name():
    global name
    first_letter = name[0]
    after_first = name[1:]
    name = first_letter.upper() + after_first.lower()

process_name()

print('Hi there', name)
```

What will this print?

Input: jACOB

Attendance Answer:

local

Graphics in Python

- **GUI:** Graphical User Interface
- TKinter is a module for creating **GUI**s and graphics that is probably already installed on your system
- For this class, you don’t have to use TKinter directly
  - Instead, use **graphics.py** which uses **TKinter**

Homework: graphics.py

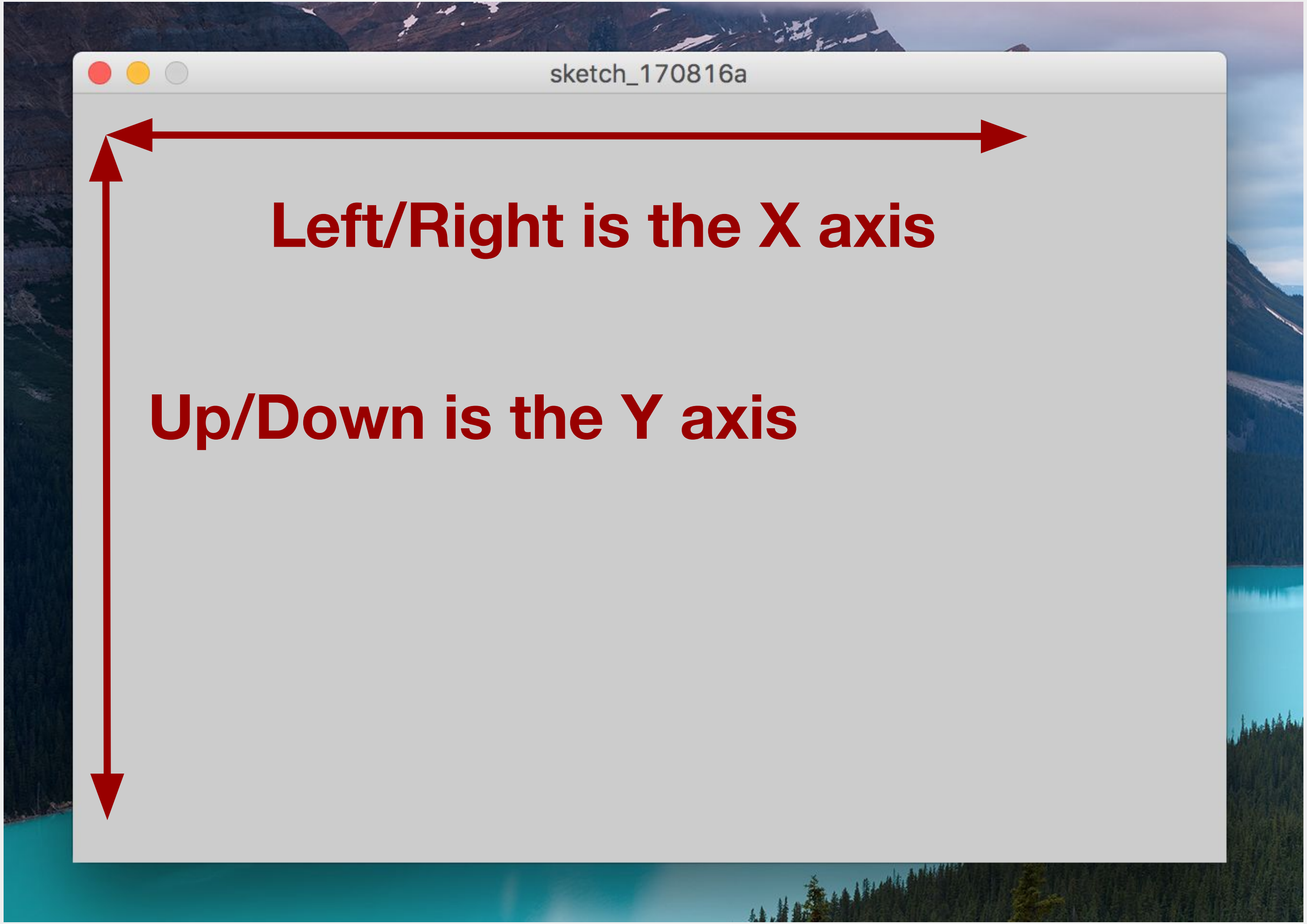
- Download **`graphics.py`**
- Place **`graphics.py`** in the same directory as the program you want to write
  - For some: the **mu_code** directory
  - For some: the **~/PycharmProjects/project_name** directory
- Write an **import** statement to allow you to use the code within:

```
from graphics import graphics
```

```python
from graphics import graphics

def main():
    # What shall we put here?
    pass

main()
```

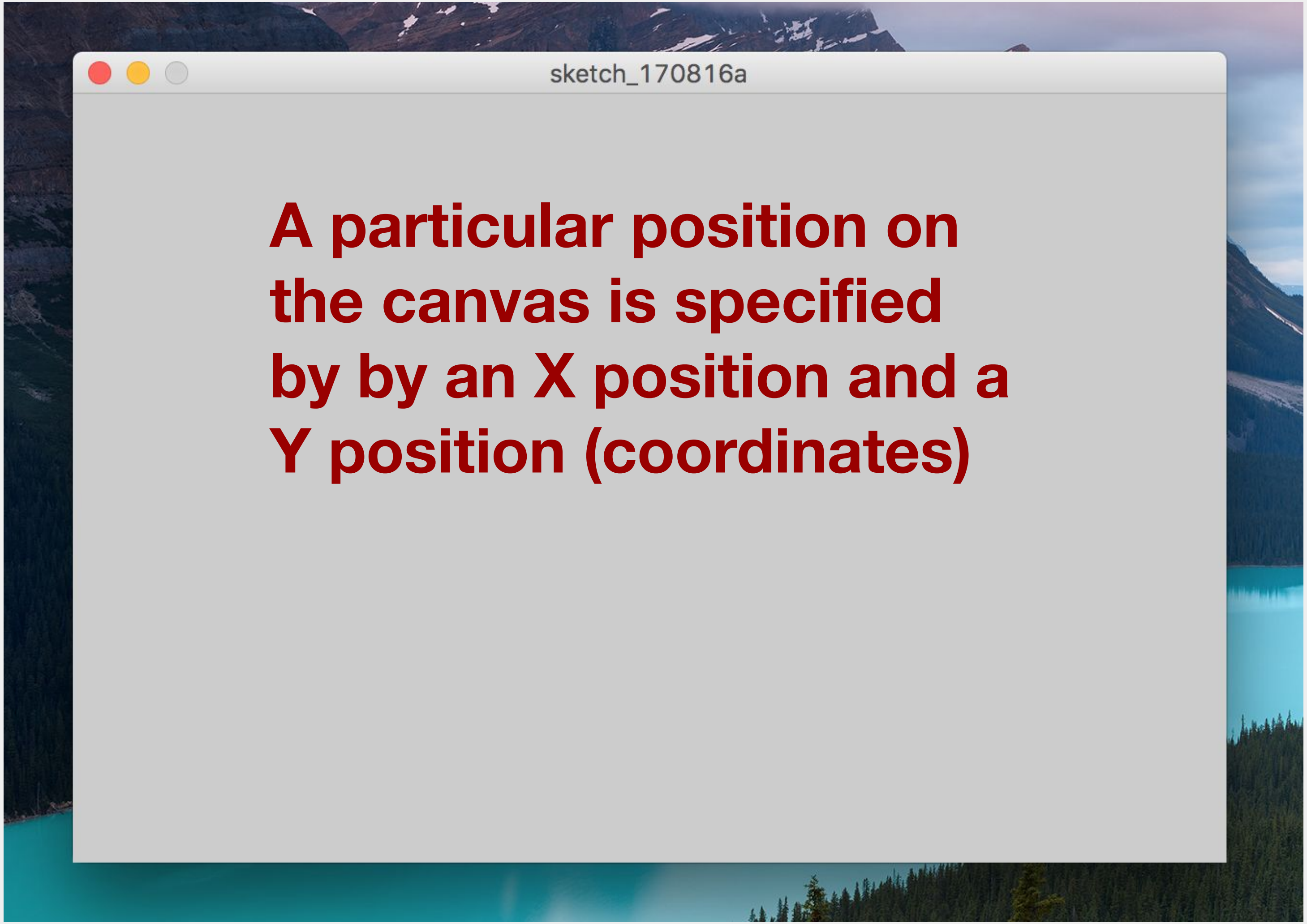
Making graphics

- When using graphics.py, you can create a **canvas**, and then draw shapes and text on the canvas.
- What is a canvas?

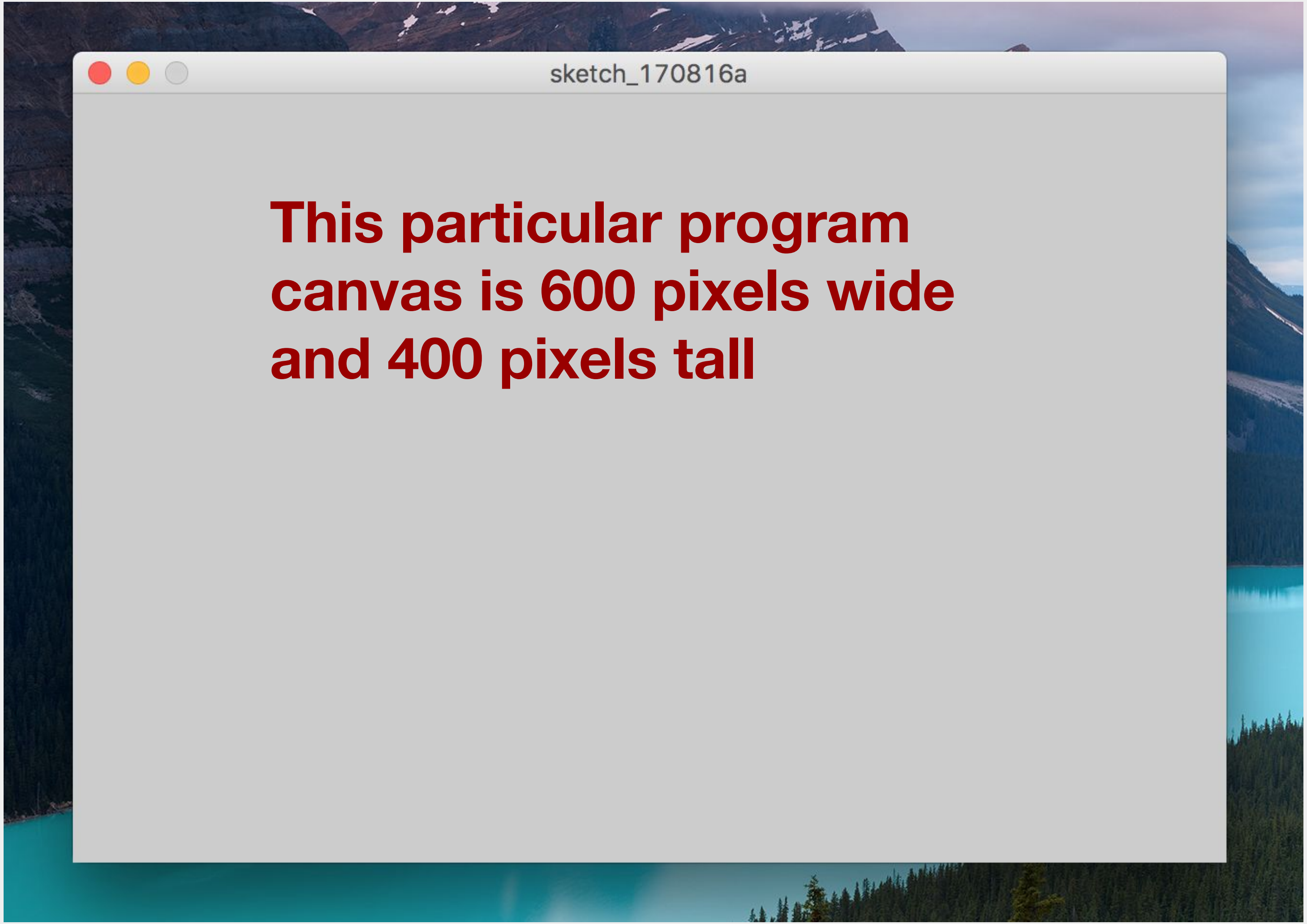


Left/Right is the X axis

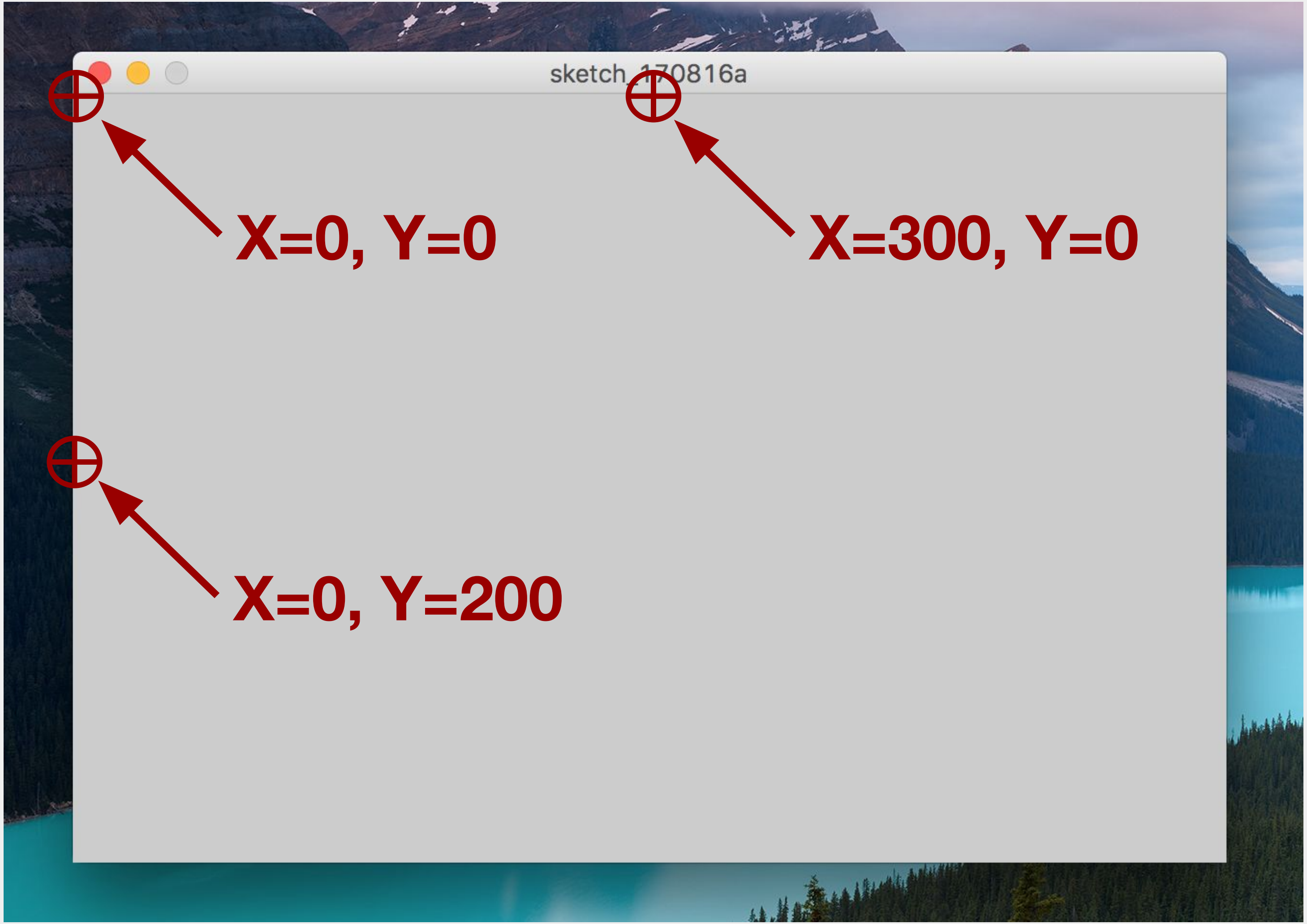
Up/Down is the Y axis



A particular position on the canvas is specified by by an X position and a Y position (coordinates)



This particular program canvas is 600 pixels wide and 400 pixels tall

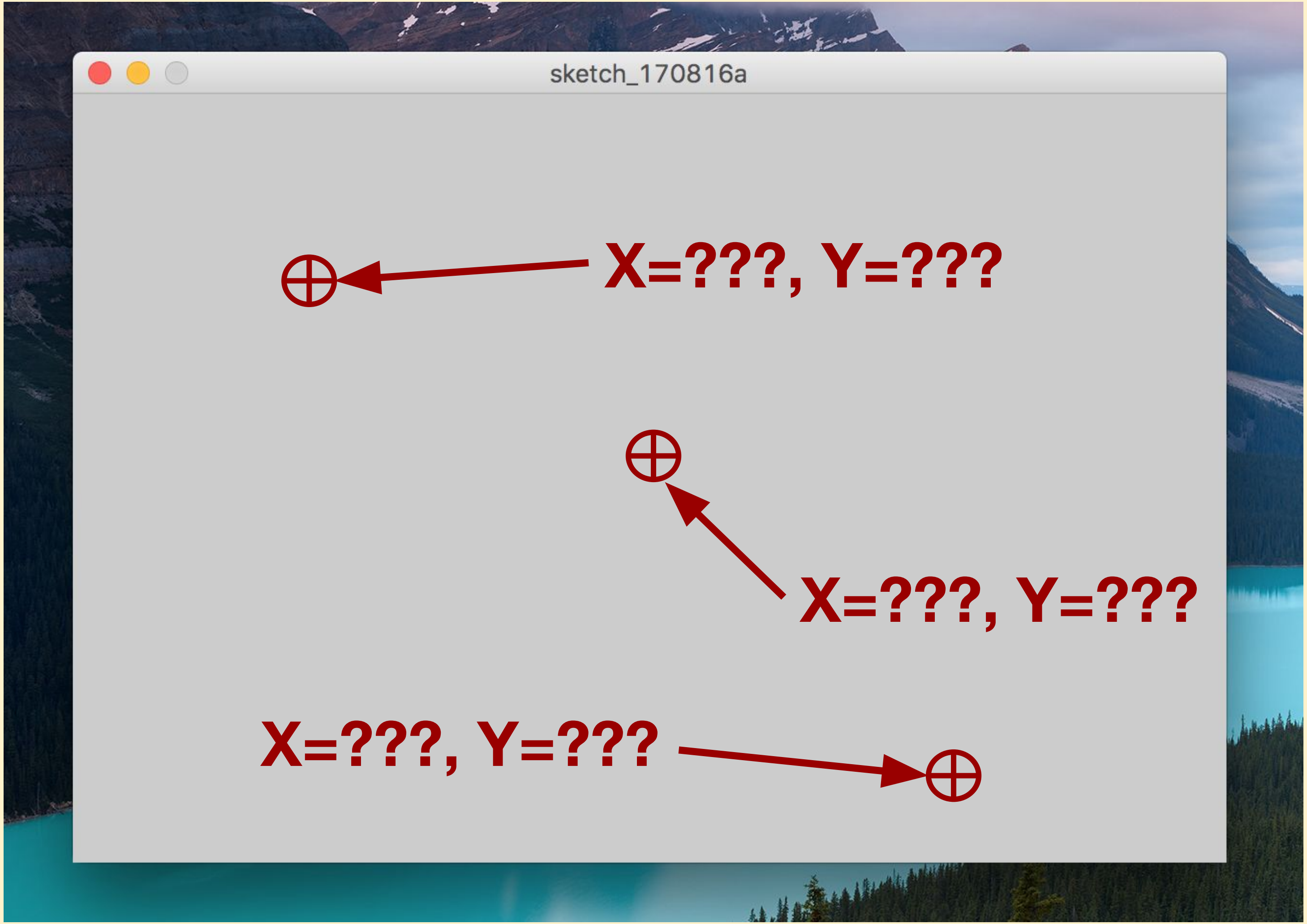


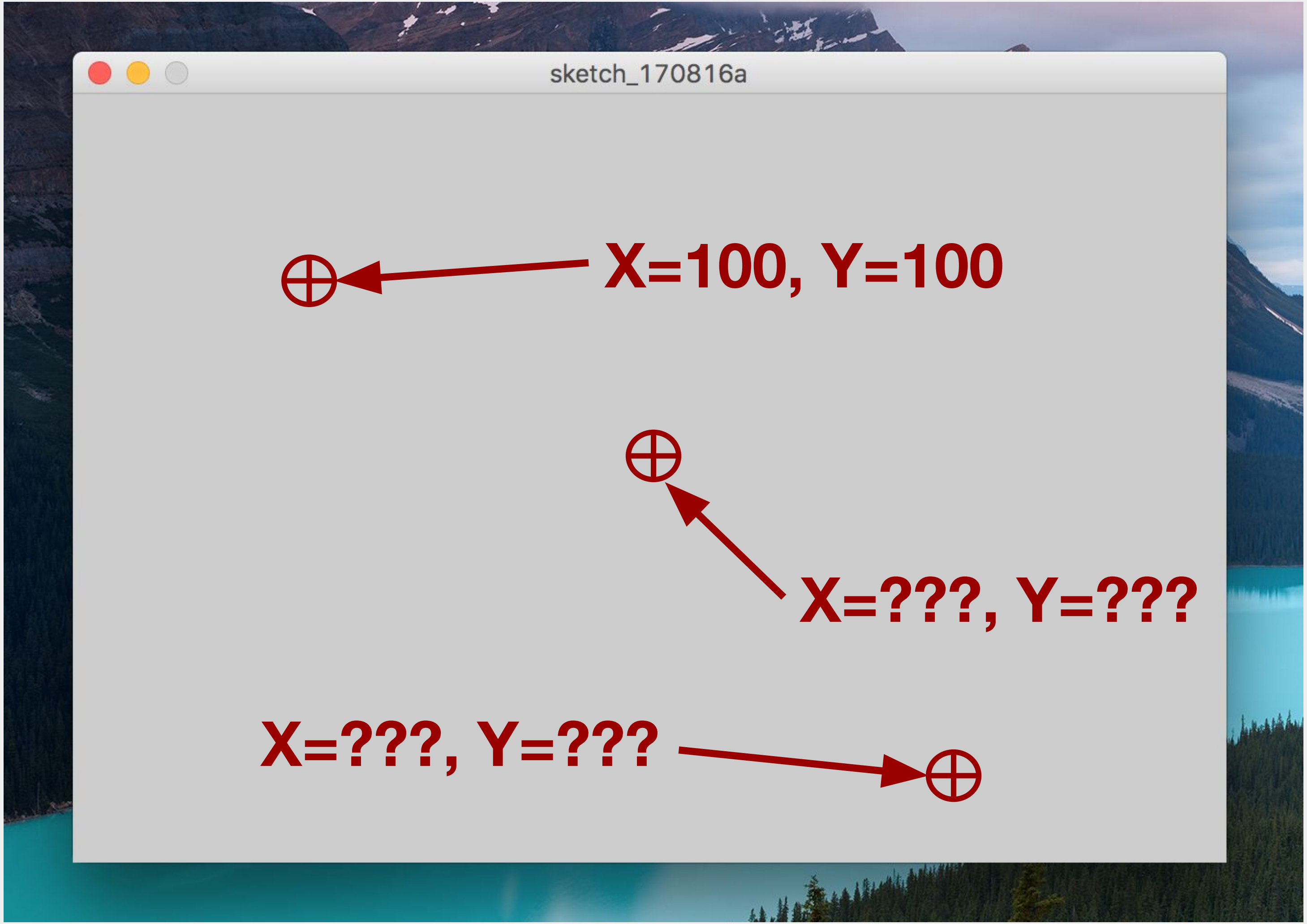
X=0, Y=0

X=300, Y=0

X=0, Y=200

Activity

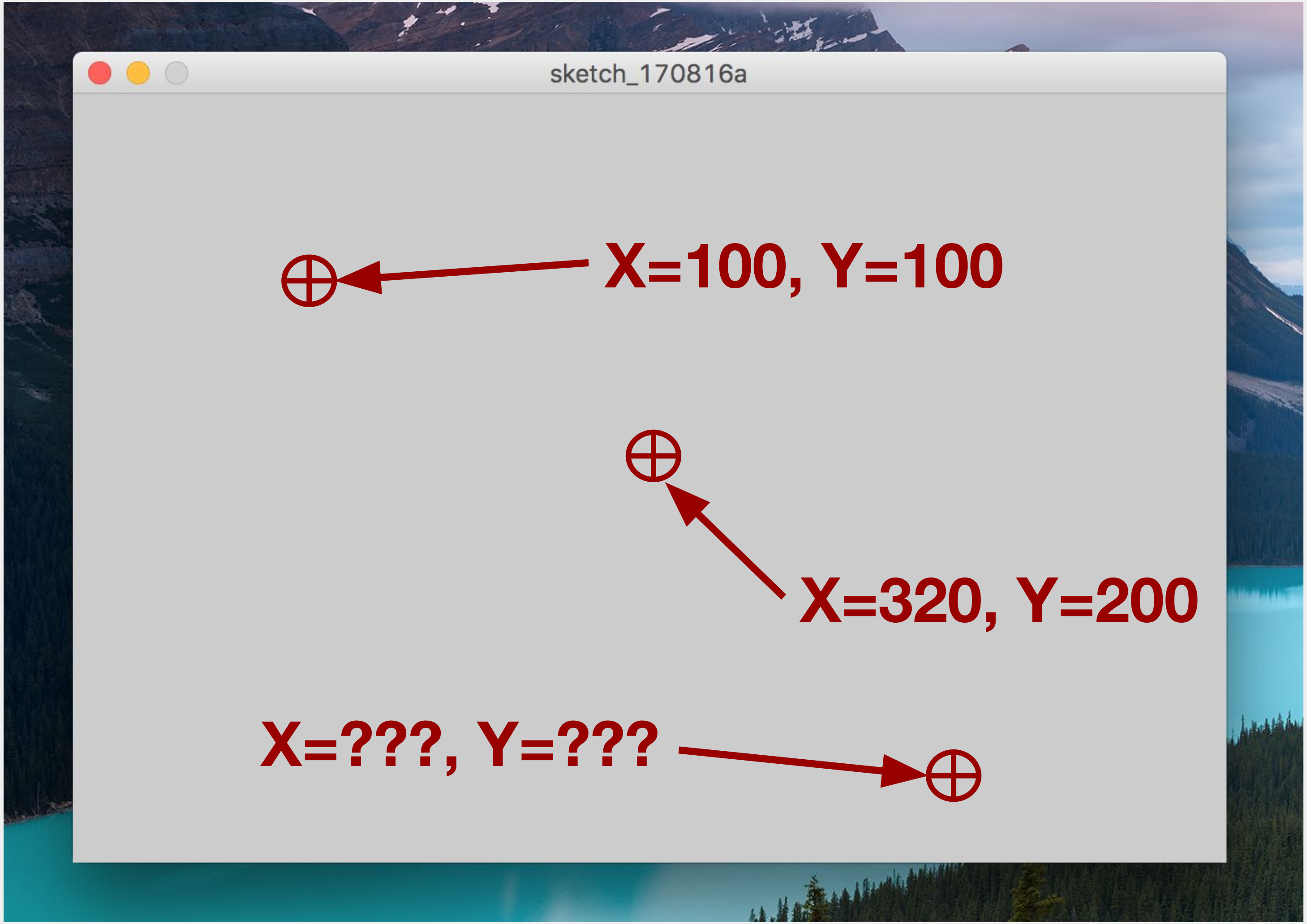




X=100, Y=100

X=???, Y=???

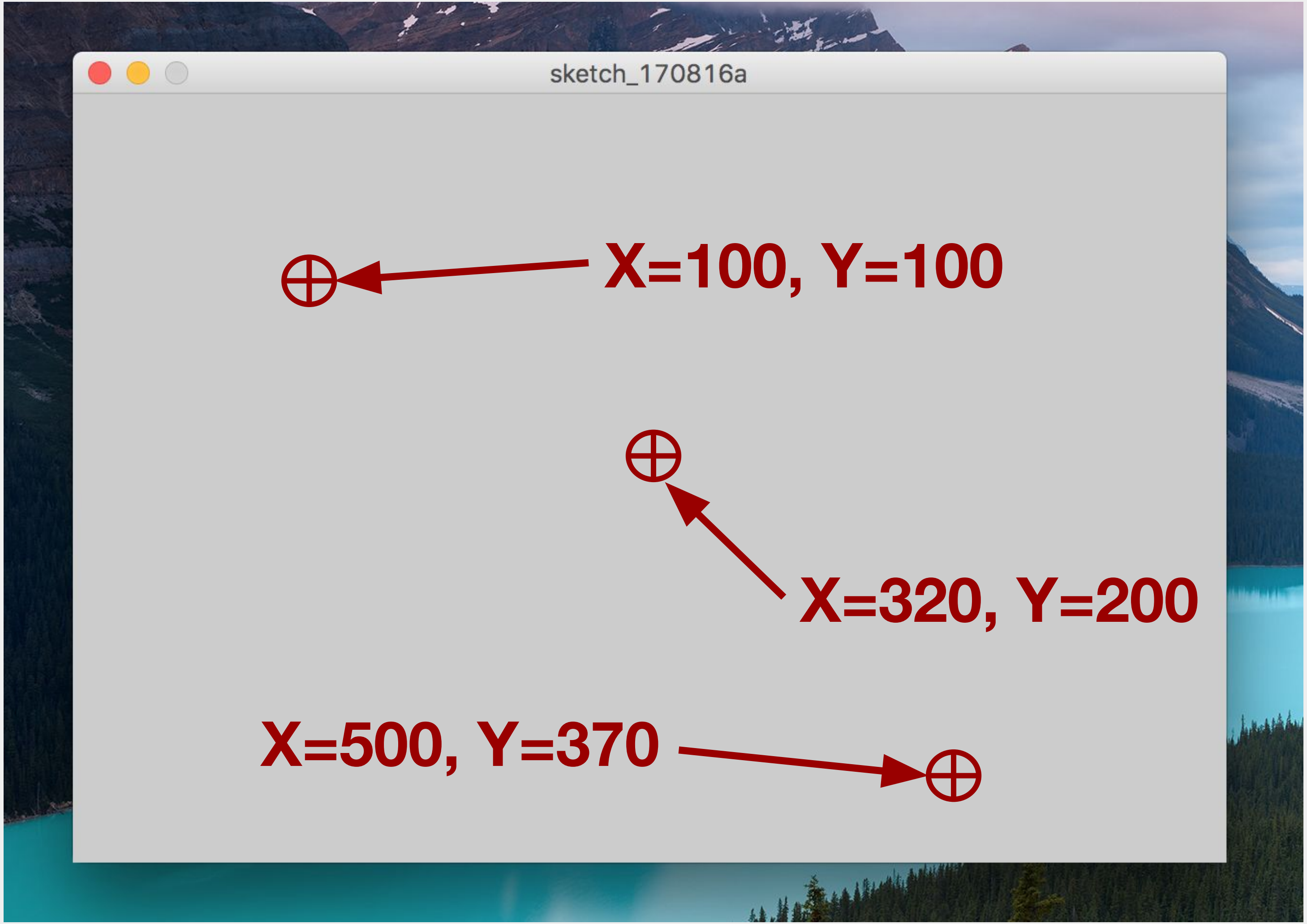
X=???, Y=???



X=100, Y=100

X=320, Y=200

X=???, Y=???


sketch_170816a
X=100, Y=100
X=320, Y=200
X=500, Y=370

Creating a canvas

Use the code below:

```
gui = graphics(width, height, 'title')
```

The **gui** variable is of type **`graphics.graphics`** (we can just refer to it as a **graphics** type). This is a **graphics object**.

We can call methods (functions) using this object.

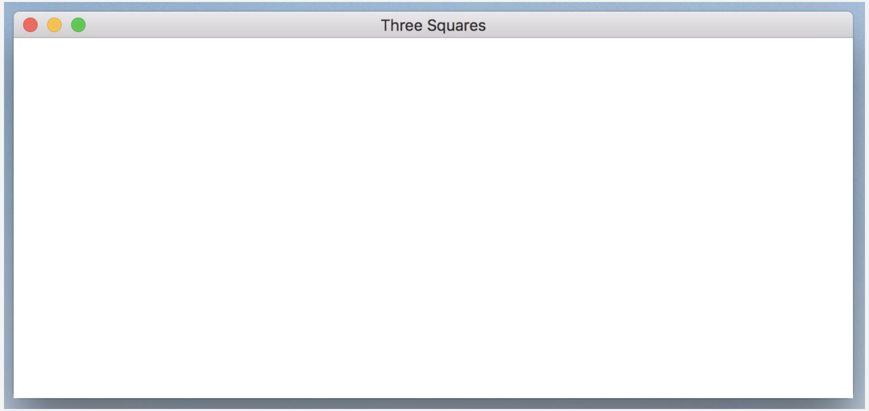
Activity

Draw the canvas

```python
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.draw()

main()
```

The canvas . . .

```
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.draw()

main()
```



Draw a rectangle

```
gui.rectangle(x, y, w, h, fill)
```

Activity

What will this display?

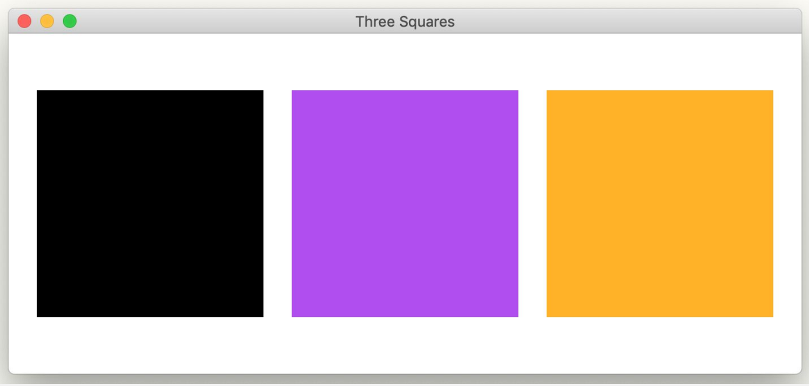
- Label the colors with text

Recall: `gui.rectangle(x, y, w, h, fill)`

```
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.rectangle( 25, 50, 200, 200, 'black')
    gui.rectangle(250, 50, 200, 200, 'purple')
    gui.rectangle(475, 50, 200, 200, 'orange')
    gui.draw()

main()
```

What will this display?

- Label the colors with text



```
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.rectangle( 25, 50, 200, 200, 'black')
    gui.rectangle(250, 50, 200, 200, 'purple')
    gui.rectangle(475, 50, 200, 200, 'orange')
    gui.draw()

main()
```

Activity

What will this display?

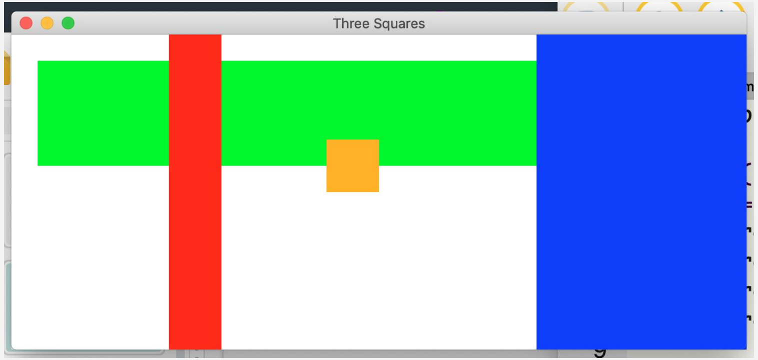
- Label the colors with text

Recall: `gui.rectangle(x, y, w, h, fill)`

```
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.rectangle( 25, 25, 500, 100, 'green')
    gui.rectangle(150, 0, 50, 300, 'red')
    gui.rectangle(500, 0, 200, 300, 'blue')
    gui.rectangle(300, 100, 50, 50, 'orange')
    gui.draw()

main()
```

What will this display?

- Label the colors with text



```
from graphics import graphics

def main():
    gui = graphics(700, 300, 'Three Squares')
    gui.rectangle( 25, 25, 500, 100, 'green')
    gui.rectangle(150, 0, 50, 300, 'red')
    gui.rectangle(500, 0, 200, 300, 'blue')
    gui.rectangle(300, 100, 50, 50, 'orange')
   gui.draw()

main()
```

Activity

What will this display?

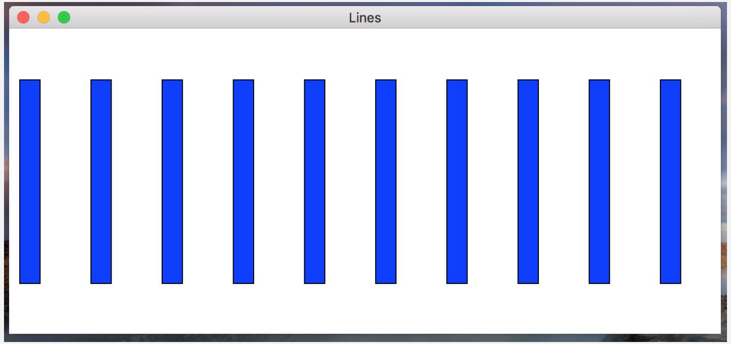
- Label the colors with text

Recall: `gui.rectangle(x, y, w, h, fill)`

```python
def main():
    gui = graphics(700, 300, 'Lines')
    i = 10
    while i < 700:
        gui.rectangle(i, 50, 20, 200, 'blue')
        i += 70
    gui.draw()

main()
```

What will this display?

- Label the colors with text



```
def main():
    gui = graphics(700, 300, 'Lines')
    i = 10
    while i < 700:
        gui.rectangle(i, 50, 20, 200, 'blue')
        i += 70
   gui.draw()

main()
```