Welcome to Section!

Week 5

Slides by Cameron Mohne and Maggie Lee

Agenda

1.

Check-in

2.

Recap

3.

Section Problem:
Random Circles

Before we start...

Let's do a quick check-in activity!

In breakout rooms:
Grab the closest thing to you and try to sell it to your partner.

As a group, one-by-one:
What did you try to sell?

Questions?

Before we jump into the recap, are there any questions?

Lecture Recap

Recap - Functions 2.0

Functions 2.0

Parameters

Parameters are a way to pass data to your helper functions. Variables made in a function are invisible to other functions unless it is passed through a parameter.

Incorrect

```
def function_one():
    variable_one = 0
    function_two()

def function_two():
    variable_two = variable_one
```

Not possible! *function_two* doesn't know that *variable_one* exists!

Correct

```
def function_one():
    variable_one = 0
    function_two(variable_one)

def function_two(example_param):
    variable_two = example_param
```

We use a parameter to pass the data instead

Functions 2.0

Returns

Similar to parameters, but in the reverse direction! We use return statements to pass data from a function to wherever it was called.

One example is the `input()` function that you've been using! Whenever you have a function that returns something, you usually want a variable to store that data!

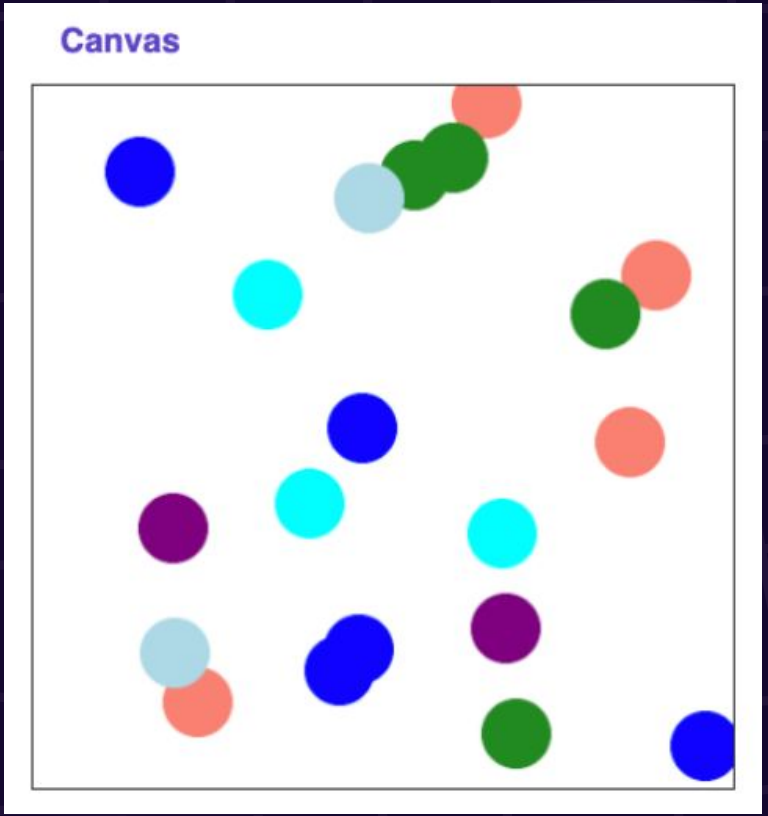
```
def function_one():
    name = input("Enter your name: ")

def input(prompt):
    ...  # Some code
    return user_input
```

`input()` returns data for you to use, wherever you called it from!

Recap - Graphics

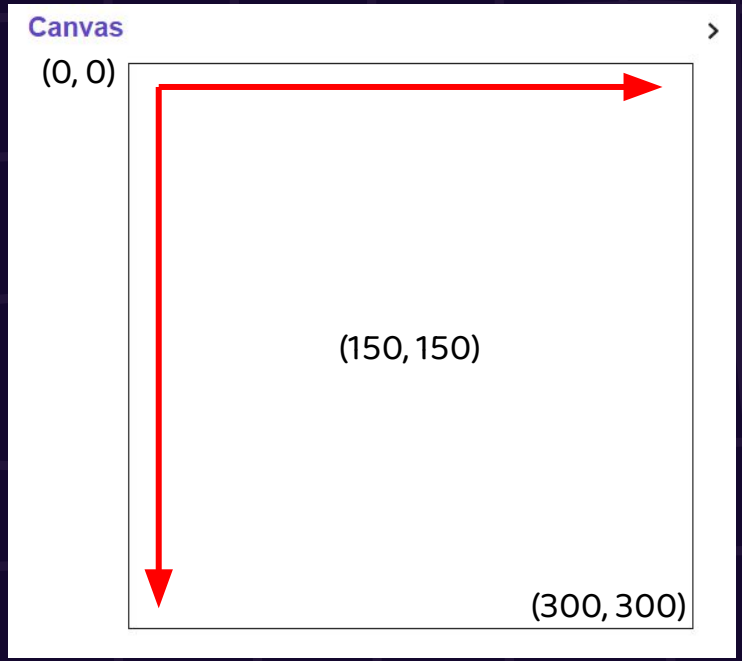
Canvas

The canvas is a fun way for us to visualize and interact with our code in a somewhat similar way to Karel!



Canvas

Remember how pixels are laid out: starting at (0, 0) for the top left and increasing in x/y-values as you go right/down respectively

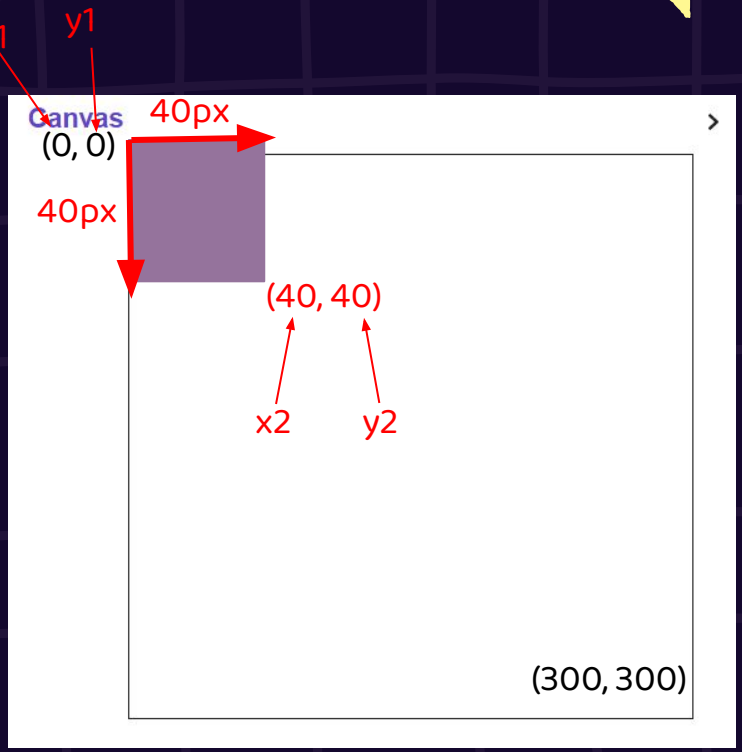


Canvas

To draw something, we need to use four parameters. These numbers represent the coordinates of the two points used to draw the image. Since the function is a part of our canvas, we use our canvas followed by a period and then the function name. We can also add an extra parameter for color if we want.

Example:

```
canvas.create_rectangle(0, 0, 40, 40, “Purple”)
                        x1 y1 x2  y2     color
```



Recap Questions

Wow! That was a lot! You've also been learning a lot.
Are there any questions over lecture content or anything else?

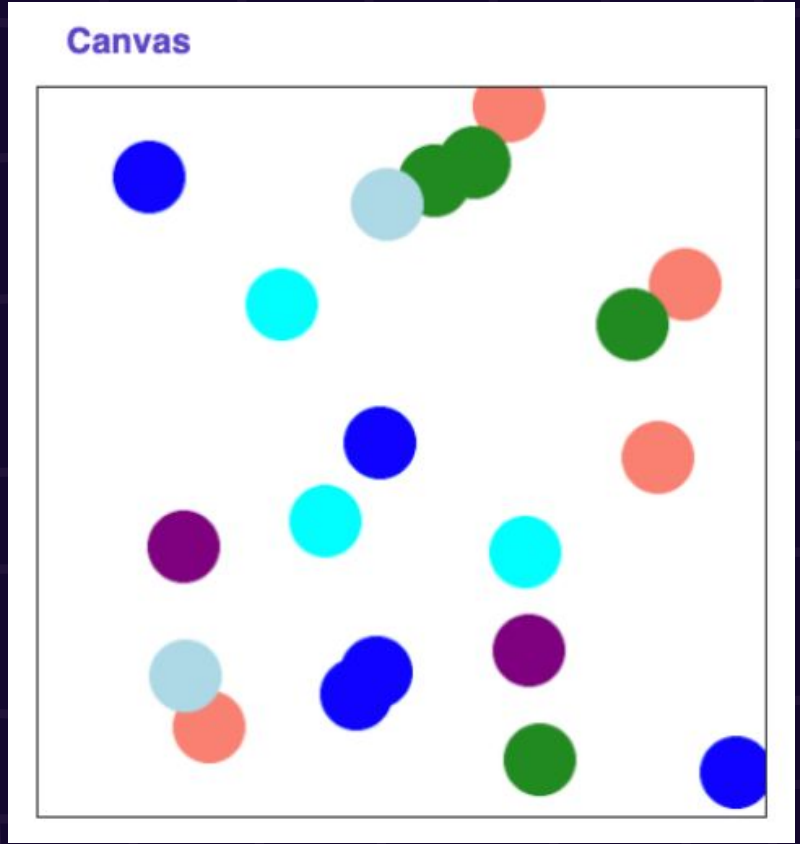
Section Problem: Random Circles

Random Circles

As an intro to functions & graphics, we want to get comfortable with:

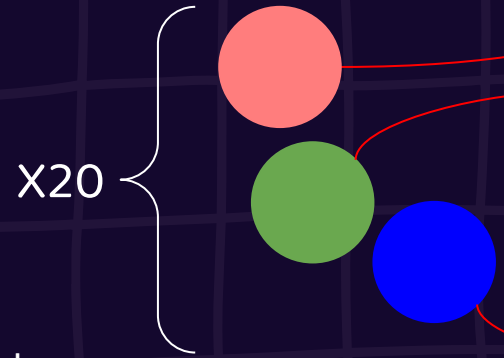
1) The conceptuals of drawing and indexing with pixels
2) Calling functions with parameters/returns.

To practice, let's write a program that **draws 20 circles at random positions with random colors** on the canvas.

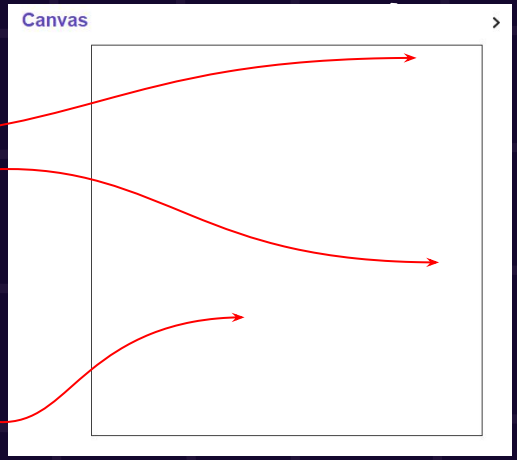


End Goal

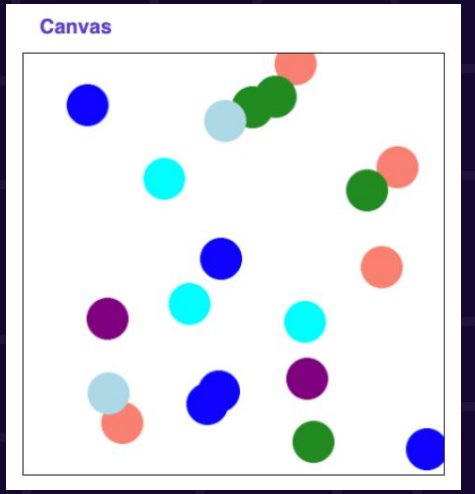
Our program takes in a blank canvas and then **draws 20 circles at random positions with random colors** on the canvas.



Example Starting Canvas



Example Output Canvas



Problem Details

Canvas

The canvas is created for you already with some functions provided

Colors

We provide a function that returns a random color

Canvas Details

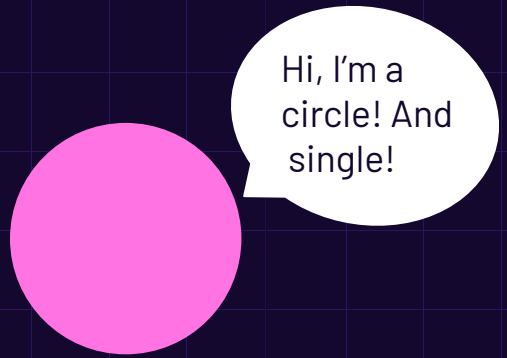
Canvas width/height are provided as global variables

Circle Details

We give you the diameter size of the circles and the number of circles to make

Pre-code Discussion

Say we want to make a single circle, how would we do that?



Are there any questions regarding the canvas, graphics, drawing, or anything else?

Coding Time!

Let's split into breakout rooms
to code up the solution!

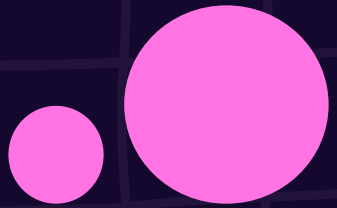
Post-code Discussion

Are there any questions at all?

Extensions

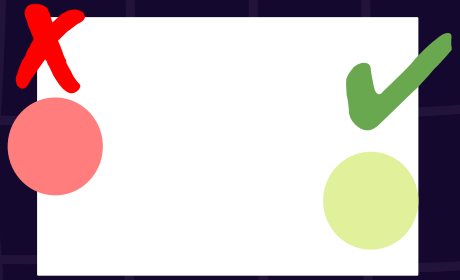
Number of Circles

Create a random amount of circles instead of a set amount!



Size of Circles

Have the circles be of random size instead of a set size!



Bounds of Circles

Make sure all circles are in the bounds of the canvas!