



CS106A Code in Place

Section 4: Functions & Graphics

Week 4: May 19th, 2023



Today's plan

- Check-in
 - Concept review
 - Advanced Functions
 - Graphics
 - Section Problem: Random Circles
 - Section Problem Bonus
-

Catching-up

- Glad to see the section forum being used, thanks for being active!
- Let's do a quick check-in question per person!
(Feel free to answer in the chat!)
- Choose 1 of the following questions, then name someone else:
 - a) Where would you rather live: the city, or the countryside?
 - b) What is one thing you're excited for? (It can be anything!)
 - c) Favorite movie/tv show?

Questions

Any questions before jumping into the concept review?

Concept Review

- Functions:
 - Parameters
 - Returns
- Graphics:
 - Canvas
 - Drawing

Concept Review: Function - Parameters VS Return

- How do functions share information?
- From parent to child:
 - Parameters
- From child to parent:
 - Return statement

Concept Review: Function - Parameters

Parameters

- 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.

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

Incorrect

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

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

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

Correct

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

We use a parameter to pass the data instead

Concept Review: Function - Returns

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!)

```
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!

- Whenever you have a function that returns something, you usually want a variable to store that data!

Concept Review: Function - Example

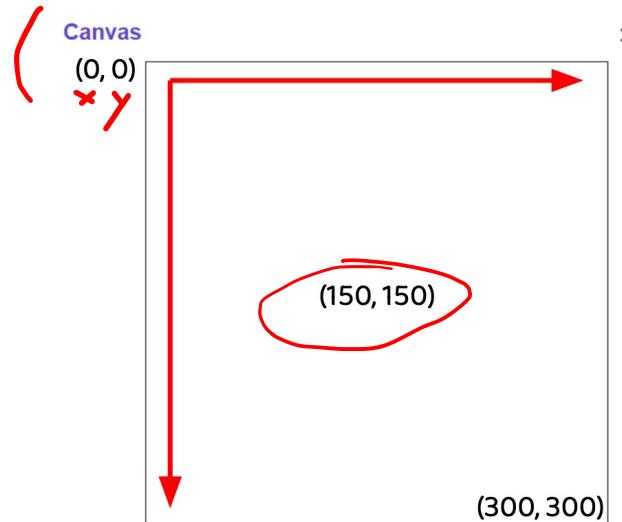
```
1 def main():
2     human_age = float(input('Enter human age: '))
3     dog_age = convert_to_dog(human_age) = 21
4     print('Dog age is: ', dog_age)
5
6
7 def convert_to_dog(human_age):
8     return human_age * 7
```

The diagram illustrates the execution flow of the code. It starts with an orange arrow labeled "Enter 3" pointing to the input statement in line 2. This value is then passed to the convert_to_dog function in line 3. Inside the convert_to_dog function, another orange arrow labeled "Still = 3" points to the parameter in the function call. The function returns the value 21, which is then assigned to the variable dog_age in line 3. Finally, an orange arrow labeled "Return 21" points from the convert_to_dog function back to the assignment statement in line 3.

Concept Review: Graphics

Canvas

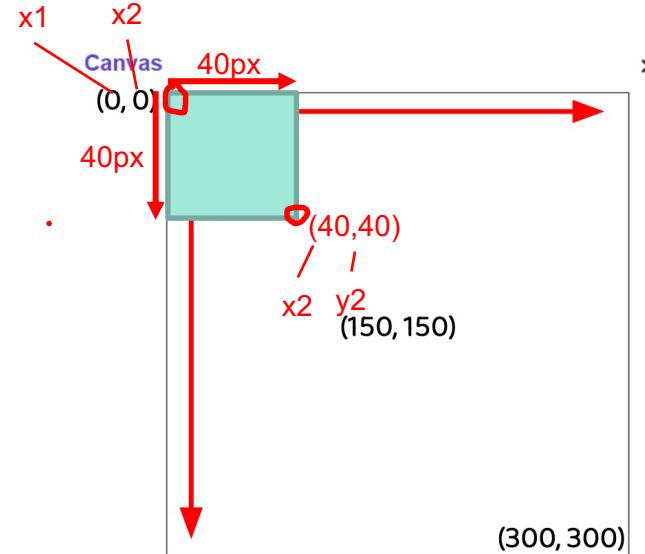
- The canvas is a fun way for us to visualize and interact with our code in a somewhat similar way to Karel!
- Remember how pixels are laid out: starting at $(0, 0)$ for the top left and increasing in x/y-values as you go right/down respectively



Concept Review: Graphics

Drawing

- When we want to draw something, we need to use four parameters which represent two points 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, "cyan")`



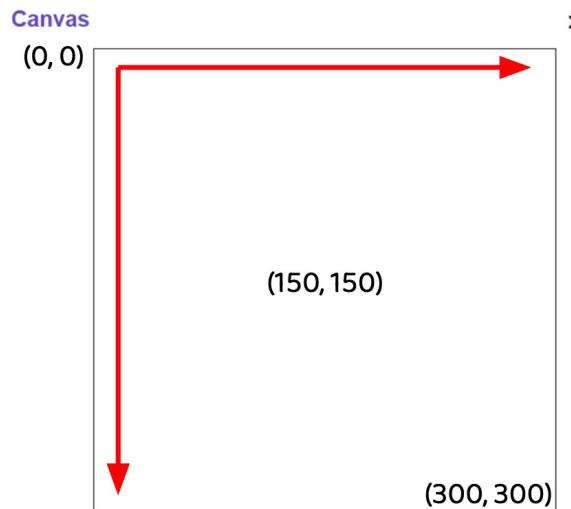
Section Problem: Drawing Circles

Task: write a program that draws 20 circles at random positions with random colors on the canvas.

Drawing Circles

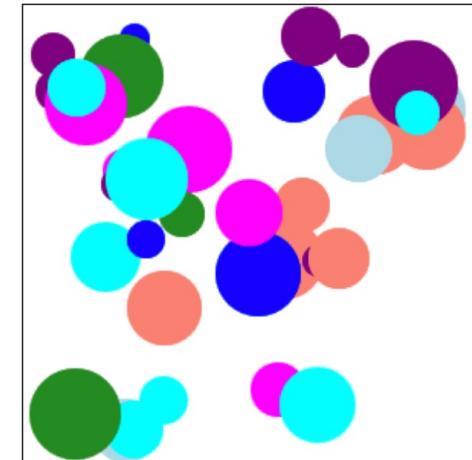
Initial state

- Our program takes in a blank canvas



Goal

- Draws 20 circles at random positions with random colors on the canvas.



Drawing Circles

- Details:
 - Canvas: The canvas is created for you already with some functions provided
 - Colors: We provide a function that returns a random color
 - Canvas size: Canvas width/height are provided as global variables
 - Circle details: you are given the diameter size of the circles and the number of circles to make
- Breakout rooms to discuss approaches for 3 minutes:
 - How do we draw 1 circle
 - How do we extend to 20?

Drawing Circles: Extension

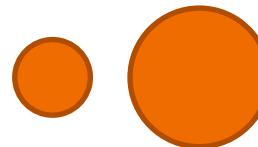
- Number of circles:

- Random amount

#

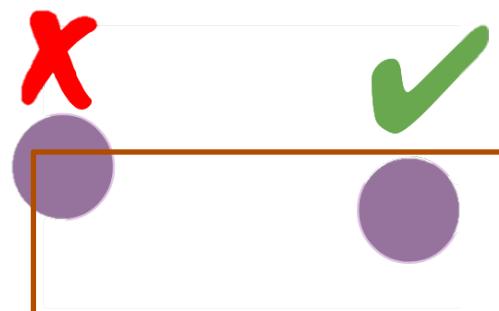
- Circle size:

- Random instead of fixed for every circle



- Circle bound:

- All fit in canvas



See you next week!

— Section 4: Graphics —
Next week: Animations
