Code in Place 2025

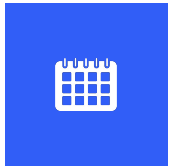
Stanford CS106A

Section - Week 5

Functions & Graphics

David Tsai, Code in Place

Today's Agenda



1. Check-In

How are we all doing?

2. Concepts Review

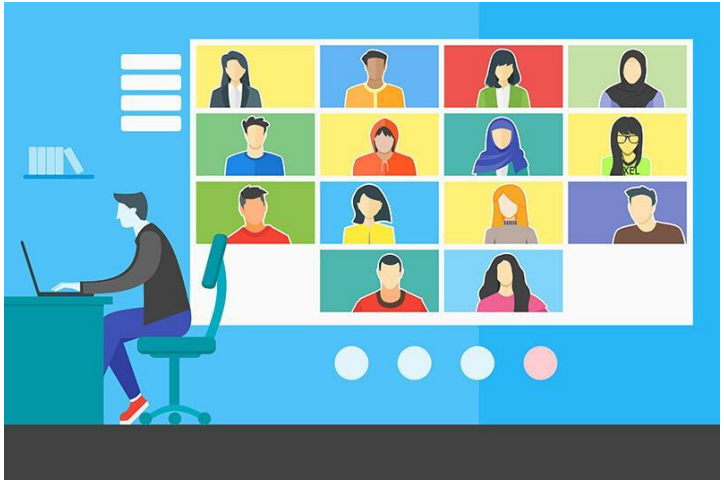
Functions, Graphics

3. Practice Problem

"Draw Random Circles"

Please Turn On Your Camera

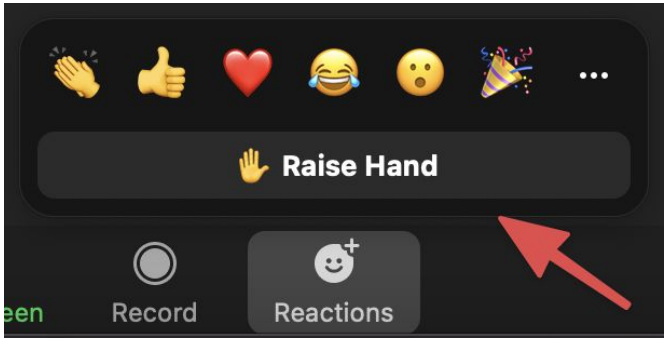
If you're able, please **turn on your camera**!
. . . . It can really make the section come to life!

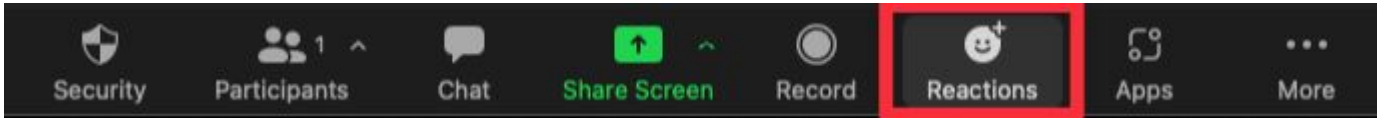


(Image source: https://as.virginia.edu/eight-ways-get-more-out-zoom)

Zoom Reactions

- 👍 **Thumbs Up:** If you understand.
- ✋ **Raise Hand:** If you have a question (or just speak in the mic).





Introductions

Hi, I'm **David**!

- Head TA, here at my 4th Code in Place.
  - Started as a CIP student! 2x volunteer Section Leader.
- CS @ Massachusetts Institute of Technology (MIT)
- Produced Manager and former Software Engineer
- Love photography, video games, movies
- Guilty pleasure: Reality competition shows like Survivor

Check-In

How are you all doing?
Hopefully your fourth week of CIP went well!

- What is something you've found most fun and/or challenging in CIP so far?

Concepts Review

Functions: Passing Data

Arguments & Parameters

- Pass data from a calling function to a helper function.

`return` statement

- Return data from helper function to calling function.

```
def main():
      mid = average(7.2, 31.5)      # Function call.  Parameters: 7.2, 31.5
      print(mid)

def average(a, b):                  # Function definition.  Arguments: a, b
      sum = a + b
      return sum / 2                # Ends the function and gives back a value.
```

Functions: Variable Scope

Variable scope

- When you define a variable within a function, it will have **local scope** (its scope lies ONLY within the function).
  - These types of variables are called **local variables**.
- Local variables exist only for as long as the function is running.
- Local variables cannot be changed or accessed from outside the function.

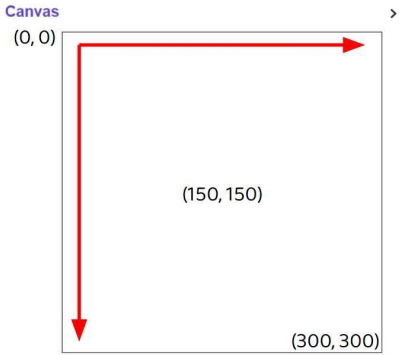
```
def greet():
      message = “Hello”       # Local variable within the greet() function.
      print(message)          # Print the local variable.

def main():
      print(message)          # Error!  The variable message does not exist!
```

Graphics: Canvas

- We'll use Python's "Tkinter" package to draw on a **Canvas**.
  - Pixels start at position (0, 0) on the top-left corner.
  - x increases going right, y increases going down.

```
CANVAS_WIDTH = 300
CANVAS_HEIGHT = 300
def main():
      canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
```

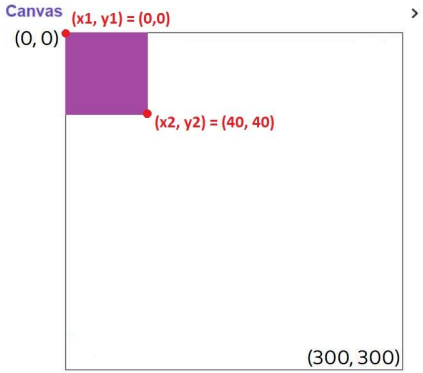


(Image source: Cameron Mohne)

Graphics: Draw Rectangle

- `canvas.create_rectangle(x1, y1, x2, y2, color)`
  - `(x1, y1)` = top-left corner
  - `(x2, y2)` = bottom-right corner

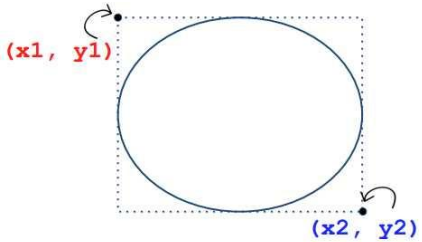
```
def main():
      canvas = Canvas(300, 300)
      canvas.create_rectangle(0, 0, 40, 40, “purple”)
```



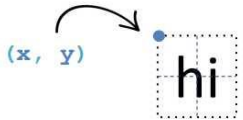
Graphics: More Shapes

- `canvas.create_line(x1, y1, x2, y2, color)`

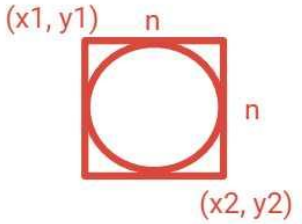
- `canvas.create_oval(x1, y1, x2, y2, color)`



- `canvas.create_text(x, y, text = “hi”)`



How to Draw a Square or Circle?

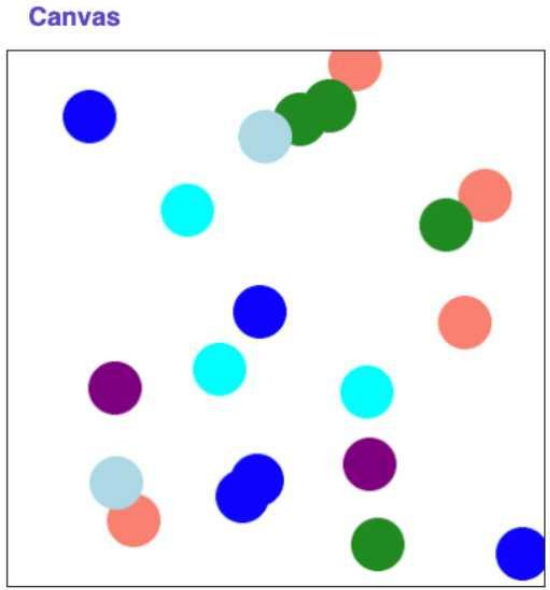


- Let's say you know the following:
  - Top-left corner has coordinates `(x1, y1)`
  - The side of the square has length `n`
    - `n` is also the diameter of the inscribed circle.
- How do you derive the bottom-right coordinate?
  - `x2 = x1 + n`
    `y2 = y1 + n`

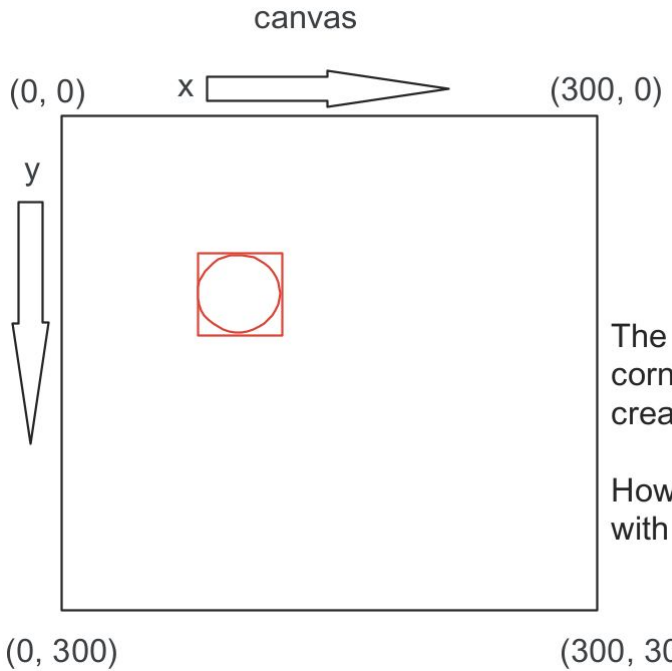
- `canvas.create_rectangle(x1, y1, x2, y2, color)`
- `canvas.create_oval(x1, y1, x2, y2, color)`

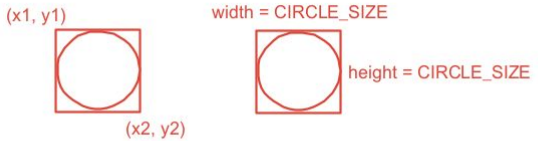
Section Exercise: “Draw Random Circles”

Draw 20 circles at random positions with random colors.



Section Exercise: “Draw Random Circles”






```
x1 = random.randint(0, CANVAS_WIDTH)
y1 = random.randint(0, CANVAS_HEIGHT)

canvas.create_oval(x1, y1, x2, y2, "red")
canvas.create_oval(x1, y1, x1 + CIRCLE_SIZE, y1 + CIRCLE_SIZE, "red")
```

The code above will randomly generate the top-left corner coordinates of a circle (x1, y1) and use them to create a circle on the canvas.

How do we ensure that the circle is always created with its entire body inside the canvas borders?

```
x1 = random.randint(0, CANVAS_WIDTH - CIRCLE_SIZE)
y1 = random.randint(0, CANVAS_HEIGHT - CIRCLE_SIZE)
```