

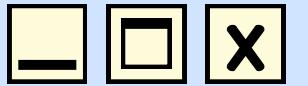
# Nafisa's Section

CSI06A - Code in Place 2025

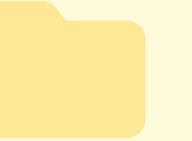
Section 5



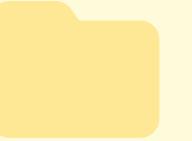
## Week Five Overview



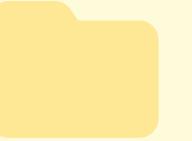
Ice Breaker & Check-in



Quick Concept Review



Understanding the  
problem



Write code  
together



Testing and Refinement  
(Optional) Fun  
Extensions!

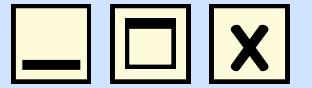


Wrap-up and Q&A

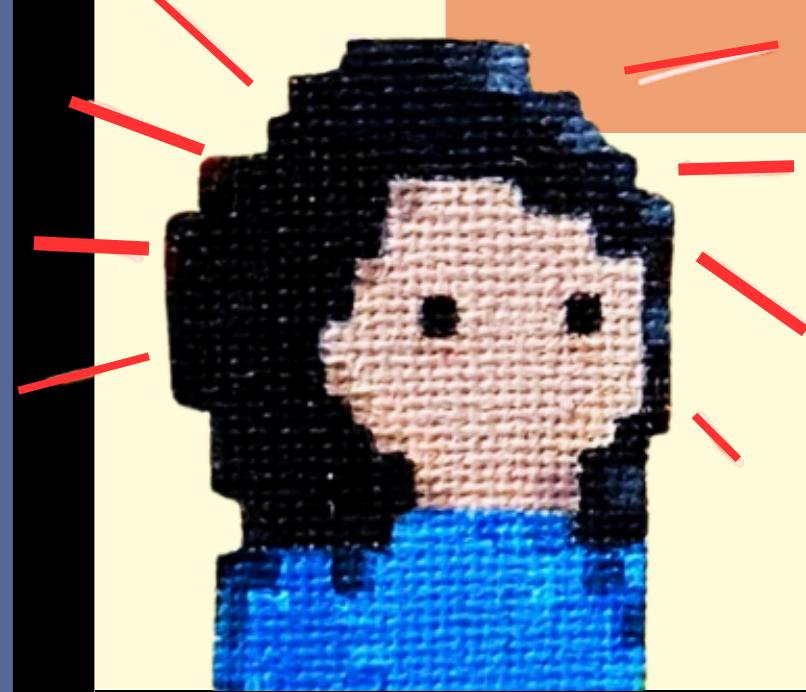




## Let's Connect!

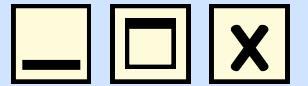


- React with a thumbs-up emoji if you had a good week!



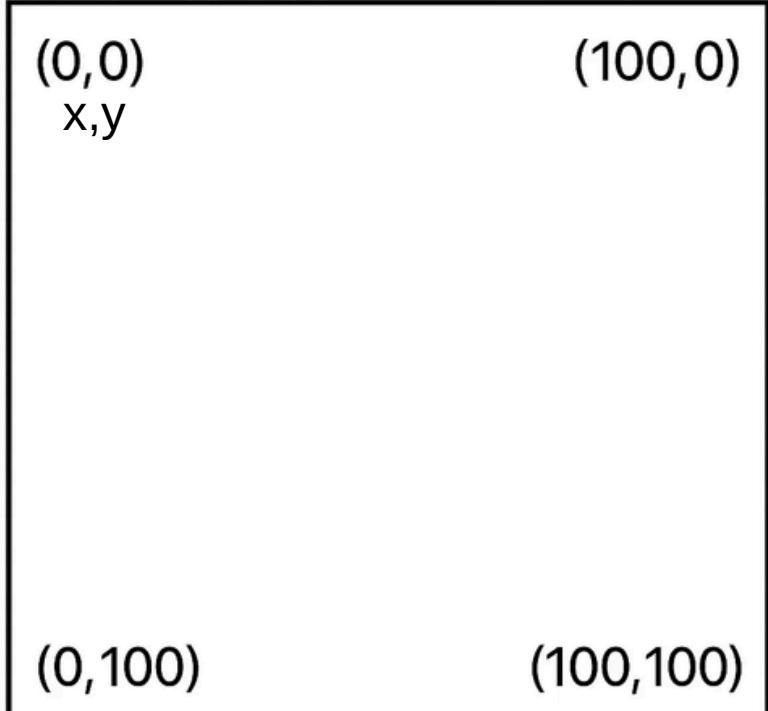


## Quick Review: What's a Canvas?



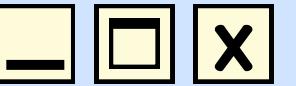
- Think of it as a digital drawing board.
- It has a width and height.
- Coordinates:
  - (0, 0) is the TOP-LEFT corner.
  - X increases to the right.
  - Y increases downwards.

Canvas 100×100





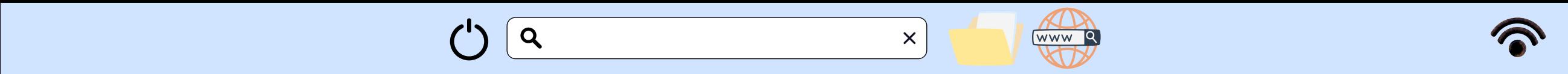
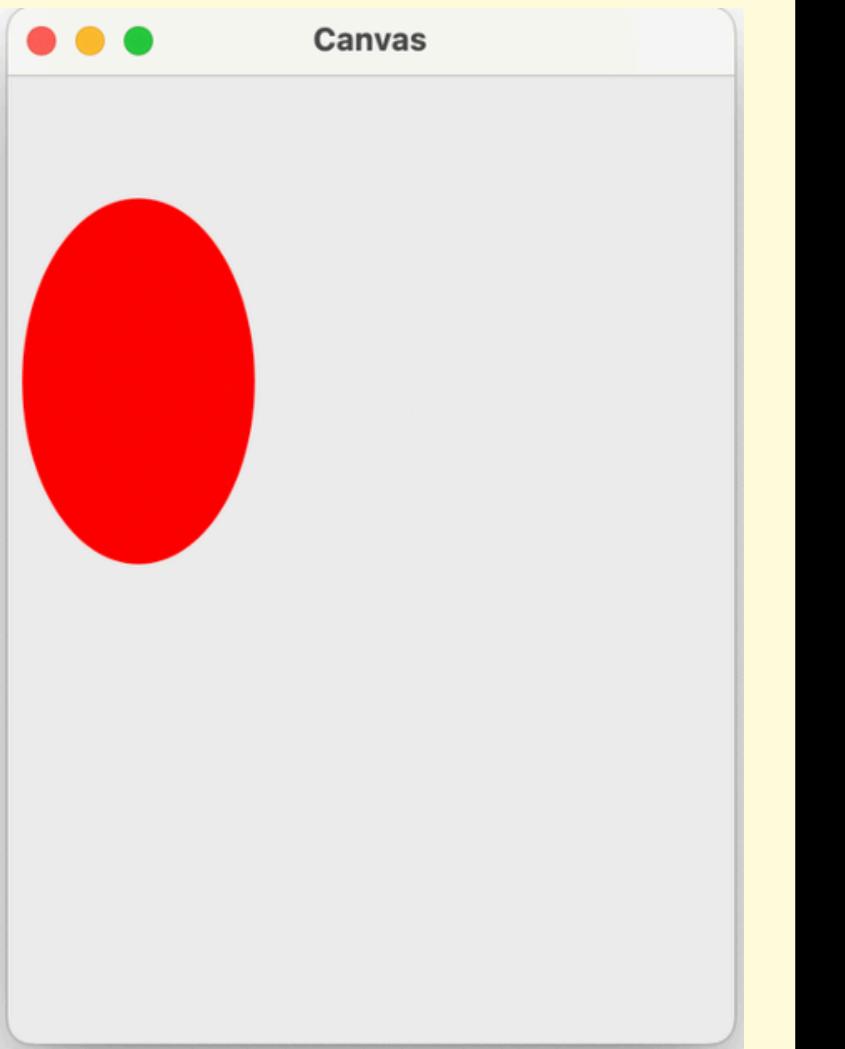
## Quick Review: Drawing Ovals (and Circles!)



- `canvas.create_oval(xl, yl, x2, y2, color)`
- `xl, yl`: Coordinates of the TOP-LEFT corner of the imaginary box.
- `x2, y2`: Coordinates of the BOTTOM-RIGHT corner of the imaginary box.
- The oval is drawn inside this imaginary box.
- A circle is just an oval where width equals height!

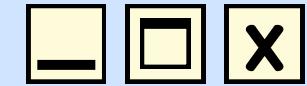
`canvas.create_oval(5, 50, 100, 200, red)`

Here,  $x1 = 5$ ,  $y1 = 50$ ,  $x2 = 100$ ,  $y2 = 200$





## Quick Review: Why use Functions?



- Organize Code: Break down big problems into smaller, manageable pieces.
- Reusable: Write code once, use it many times.
- Abstract Away Details: Hide complexity.

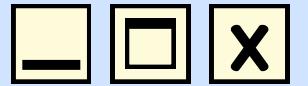
Syntax Reminder:

```
def function_name(parameter1, parameter2):  
    # code to do something  
    return value # optional
```





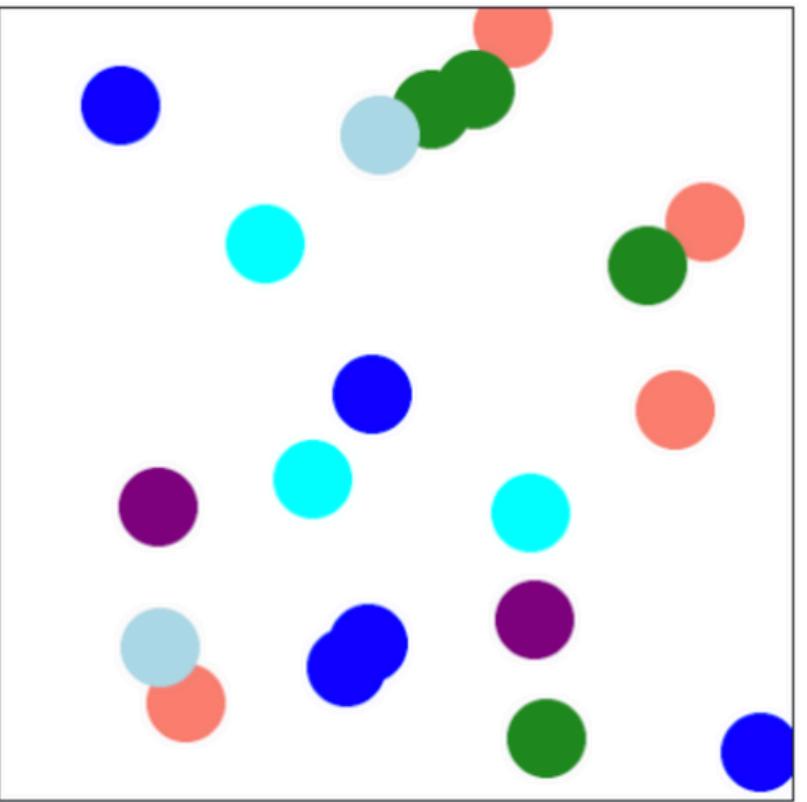
## Today's Goal



Write a program that draws N\_CIRCLES (e.g., 20) circles.

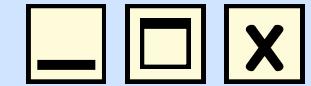
- Each circle:
  - Has a fixed CIRCLE\_SIZE.
  - Is at a random position.
  - Has a random color from a given list.
  - (Ideally) Stays fully within the canvas.

Canvas





## Extensions!



Random Number of Circles: Draw a random number of circles (e.g., between 1 and N\_CIRCLES).

- How? `num_to_draw = random.randint(1, N_CIRCLES)`
- Change loop: `for i in range(num_to_draw):`

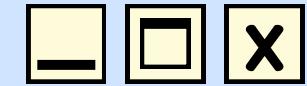
Random Circle Size:

- How? `size = random.randint(10, 50)` (choose min/max)
- Where would size replace CIRCLE\_SIZE in `draw_random_circle?`
- Careful: xl and yl max values now depend on this random size!





## What We Practiced Today



- Problem Decomposition: Breaking a big problem ( $N$  random circles) into smaller ones (1 random circle).
- Functions: `draw_random_circle` made our main function cleaner and the logic reusable.
- Graphics Coordinates: `create_oval` uses a bounding box. Top-left is (0,0).
- Randomness: `random.randint()` for numbers, `random.choice()` for lists.
- Constants: Make code easier to read and modify.





**THANK YOU**

See you in the next session!

