

# Simple and compound shapes For loops

Oliver W. Layton

CS151: Computational Thinking: Visual Media

Lecture 05, Spring 2021

Friday February 19

# Quizzes

- There will be a 10-15 min quiz that is available for you to take after class today.
- Given most weeks on online over [Moodle](#) after class on Fridays.
- You have from after Friday's class to Sunday 11:59pm to take it.
- Weekly homework is a great study tool.
- **Focus:** read short code samples, answer questions about them.
- Usual question types: T/F, multiple choice, type in the number.

# Project grading

- How projects are graded.
- What counts as extensions changes as you advance through CS151.

# Project 2

**Goal:** Draw two different space-themed scenes!

- Convert `lab2.py` into your "shape library" ( `shapelib.py`): Holds all functions to draw simple and compound shapes.
- In shape library, make two more simple shapes (other than triangle — rectangles, pentagon, parallelogram, circle, etc.)
- In shape library, make two space-themed compound shape by calling your simple shape functions (e.g. rocket ship, planets, stars, etc.).
- Design a space themed scene using all your shapes.

# Project 2 guidelines

- Use best practices for code organization that we discussed on Wednesday (e.g. headers, docstrings, code comments, main code at bottom etc.).
- Use single empty lines in between blocks of code that accomplish subtasks to improve readability (analogous to new paragraphs when writing an essay).
- Use function names that make it clear what they are doing (e.g. `triangle()`).

# Practice: make a house out of a triangle on top of a rectangle

- Adding position and scale parameters to a compound shape function
- Let's also show how we can access functions in a different file.
- Changing color, fill, and pen width in a function

# For loops

# Simplifying code for repetitive tasks

Until now, we've had to write one line of code per command.

- Functions have helped reduce code, but for highly repetitive tasks, it is still annoying!
- Example: Draw a square, we need 4 sets of `forward` and `right` commands.
- With a **for loop**, we can *substantially* reduce the amount of code we need to write for repetitive tasks.

# For loops in Swift Playgrounds

# For loops in Python

In general, for loops have the following structure:

```
for <loop index> in range(<number of loop iterations>):  
    <loop body>
```

Let's try a simple example: `lecture_05_for_loop.py`

Example: Using a for loop to simplify  
drawing hexagon

Example: Using a for loop to draw  
any polygon