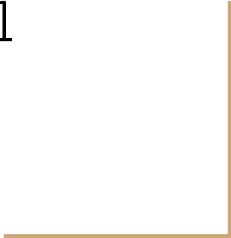


Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1



Announcements

Lab this week: git (source code control) and PIL (Python Imaging Library)

“Problem of the Day” continues today!

Today:

Designing and Implementing object functionality via member functions

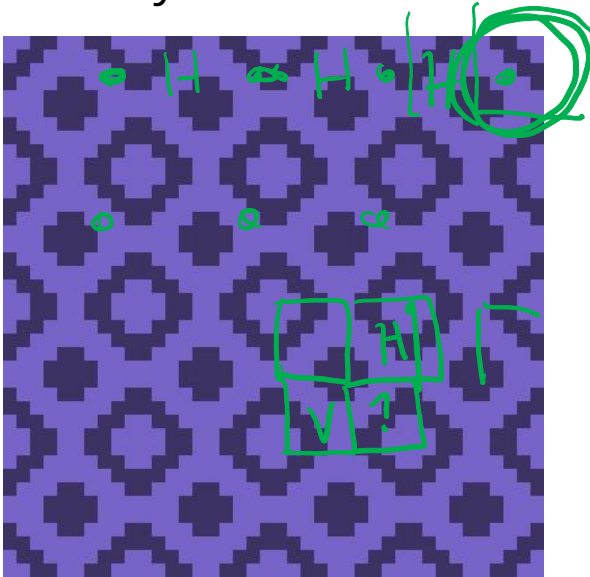
POTD

Design Strategies

1. Decompose a problem into classes
2. List the data associated with each class
3. Write the “driver” code that illustrates the functionality you expect from each class.
4. Implement the functions you expect.
5. Run the driver code to test your functionality.

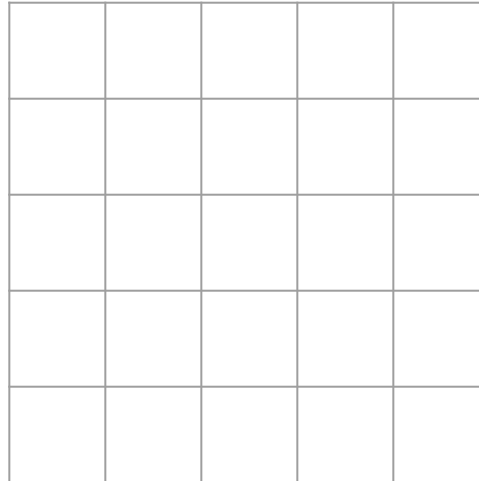
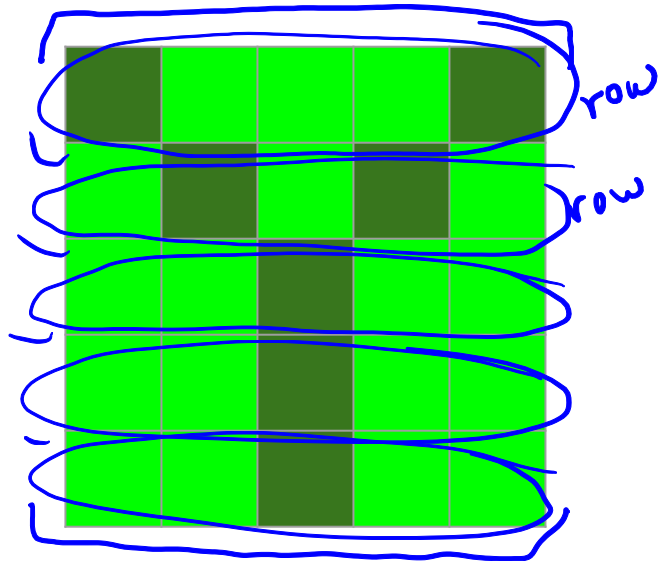
Adding Functionality

Creating blocks is an arduous task. We'd like a way to make new blocks out of old ones! How many different kinds of blocks are found in this image? How are they related to one another?



Flip Vertical

Suppose we want to create a new block which is just the vertical reflection of a given block. Sketch the new block. Describe how you would accomplish the flipped block, in terms of the block representation in our code (list of rows).



1. *New list, copy of old.*
2. *Reverse the new list.*
3. *Build block from new list.*
4. *Return new block.*

Flip Vertical

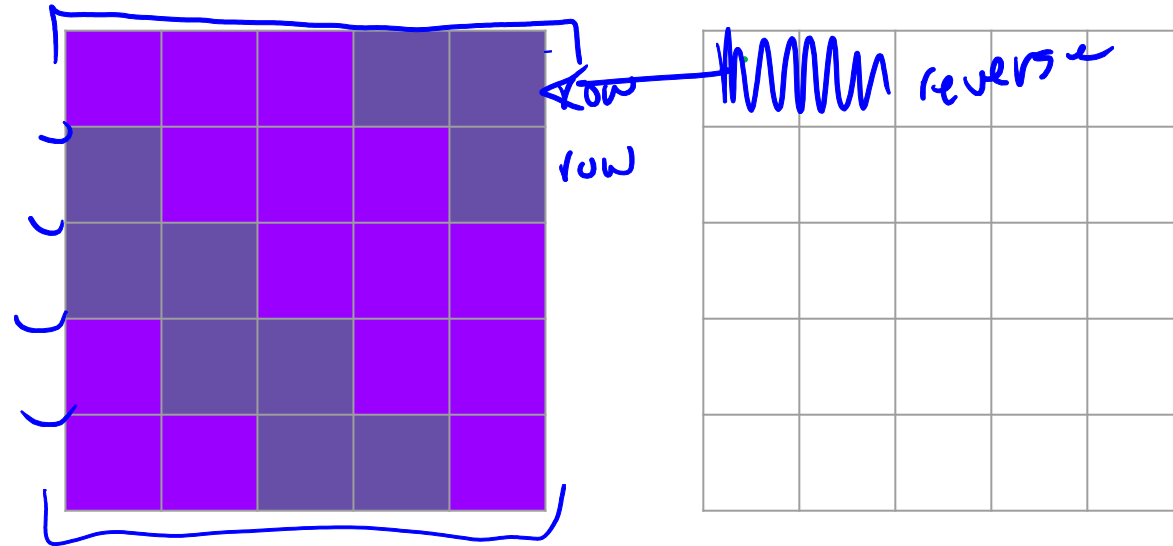
<https://repl.it/@ckh205/knittingSkeleton>

Review the code we've written and make 3 observations:

1. Line __: _____
2. Line __: _____
3. Line __: _____

Flip Horizontal

Suppose we want to create a new block which is just the horizontal reflection of a given block. Sketch the new block. Describe how you would accomplish the flipped block, in terms of the block representation in our code.



1. make new list of rows
2. where each row has a reversed list of stitches
3. use new list to make a block & return H

Need 2 new things...

List Comprehensions in Python

A new way to create lists from old!

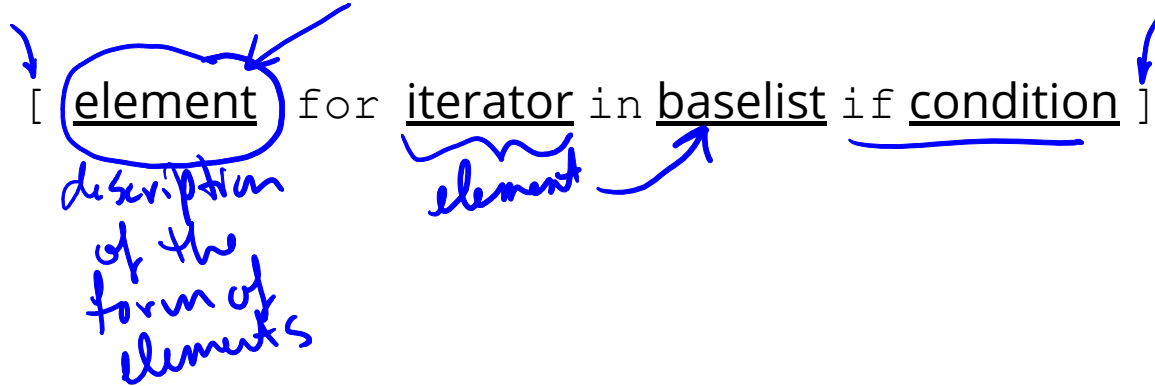
<https://repl.it/@ckh205/ListComprehensionsExamples>

General Form:

[element for iterator in baselist if condition]

description of the form of elements

element

The diagram shows the general form of a list comprehension: [element for iterator in baselist if condition]. The word 'element' is circled in blue, with a handwritten note 'description of the form of elements' pointing to it. The words 'iterator', 'baselist', and 'condition' are underlined in blue. A wavy blue line under 'iterator' has a handwritten note 'element' below it. Blue arrows point from the top left and top right towards the opening and closing brackets of the list comprehension.

Flip Horizontal

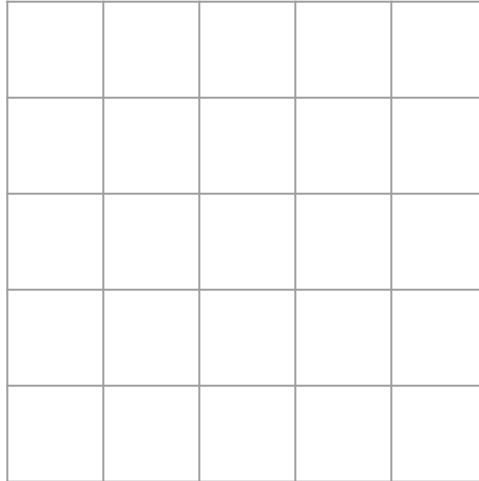
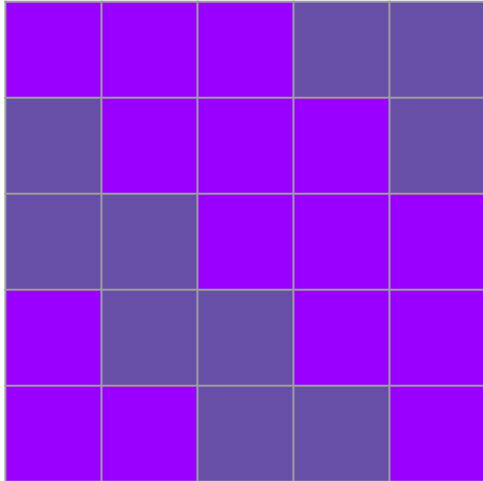
<https://repl.it/@ckh205/knittingSkeleton>

Review the code we've written and make 3 observations:

1. Line __: _____
2. Line __: _____
3. Line __: _____

Rotate 180

Suppose we want to create a new block which is a 180 degree rotation of a given block. Sketch the new block. Describe how you would accomplish the flipped block.



1.

2.

3.

4.

Rotate 180

<https://repl.it/@ckh205/knittingSkeleton>

Review the code we've written and make 3 observations:

1. Line __: _____
2. Line __: _____
3. Line __: _____

POTD #3 Tue

<https://cheeren.github.io/cpsc203test/>

Describe any snags you run into:

1. Line ____: _____
2. Line ____: _____
3. Line ____: _____
4. Line ____: _____
5. Line ____: _____

ToDo for next class...

POTD: Continue Fri and Mon. Submit to repo.

Reading: TLACS Ch 4 (intro to turtle graphics and iteration)

Play: <https://www.google.com/search?q=color+picker>

Adapt: Use today's demo to create a handcraft of your own! Upload your creation to <https://bit.ly/2ly0z8Q> (please make unique file names)

References:

TLACS Ch 17

<https://docs.python.org/3/library/dataclasses.html>