Objective

- For loop

Lab Review

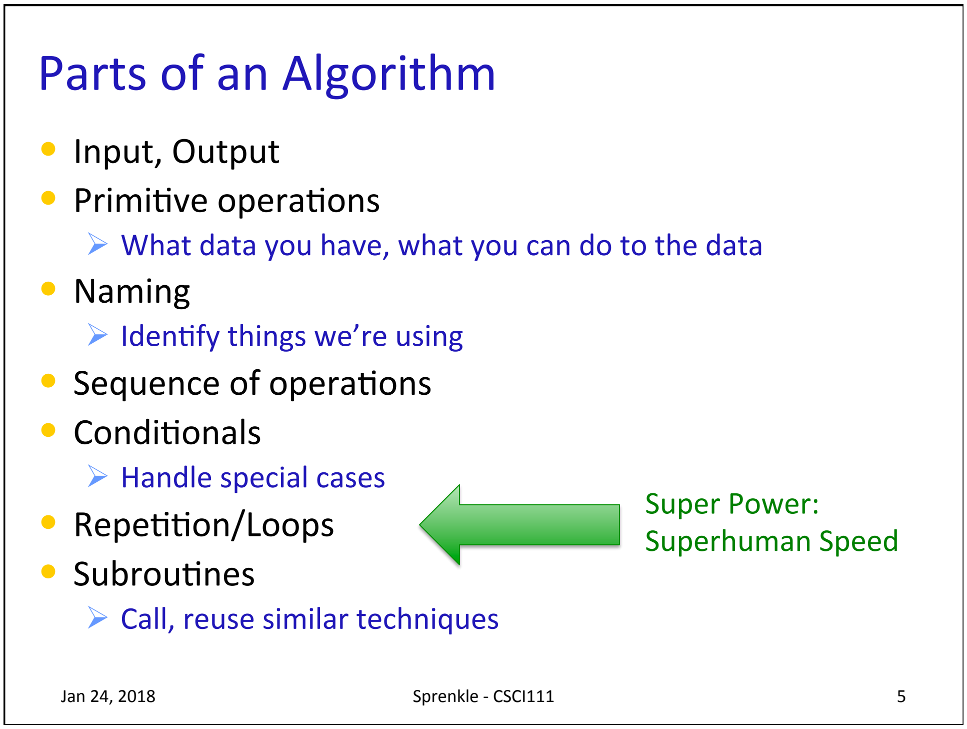
- Follow examples
  - Find solutions to similar problems
  - Understand the solution
  - Adapt the solution to your problem

Task	Objective
Creating a Text object	Confirming that you know how to use the API, using a class that you hadn't used previously.
Making a picture	Allow you to show creativity

Review

- How do we create objects?
- How do we call operations on objects?
- How do we get access to the code in `graphics.py` in our code?
- How can we make a duplicate of a drawable object using the Graphics API?

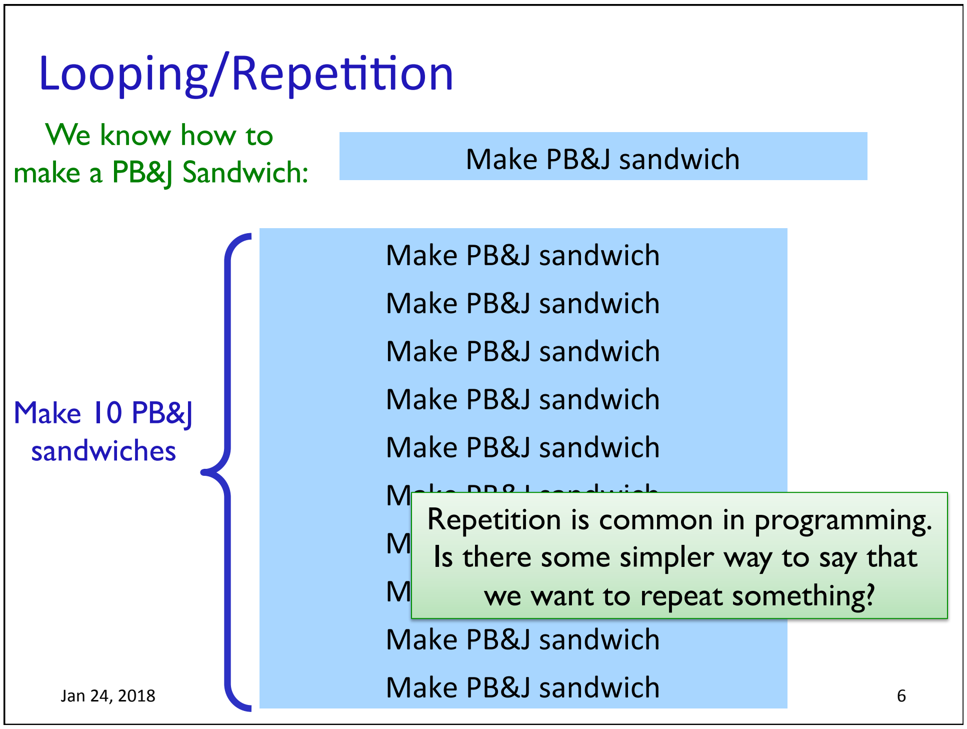
FOR LOOPS



Parts of an Algorithm

- Input, Output
- Primitive operations
  - What data you have, what you can do to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

Super Power:
Superhuman Speed



Looping/Repetition

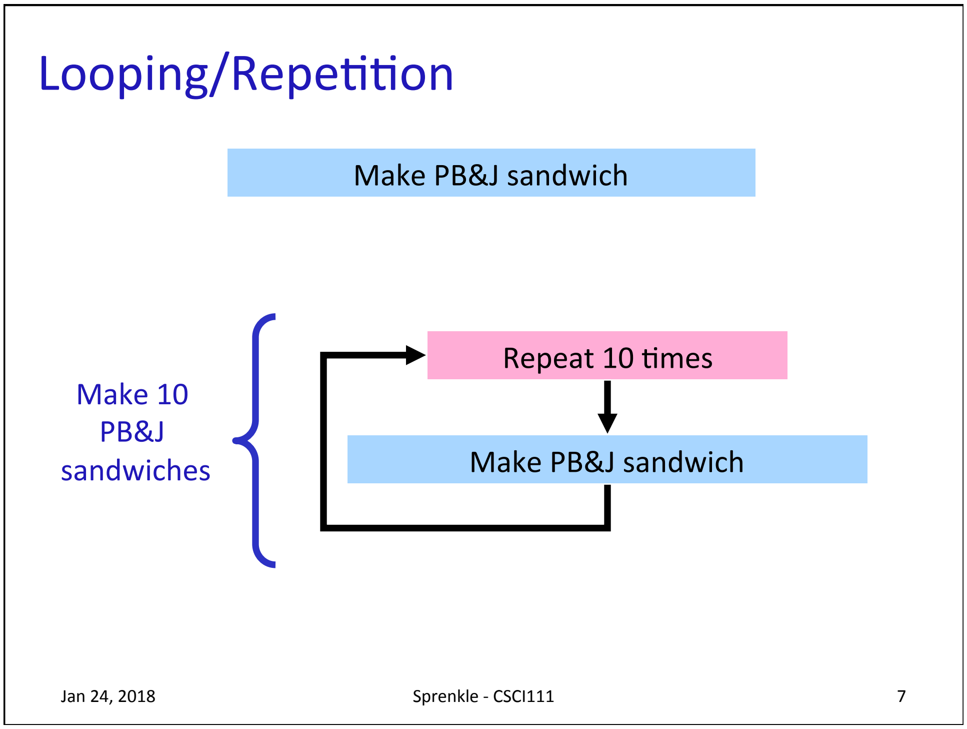
We know how to make a PB&J Sandwich:

Make PB&J sandwich

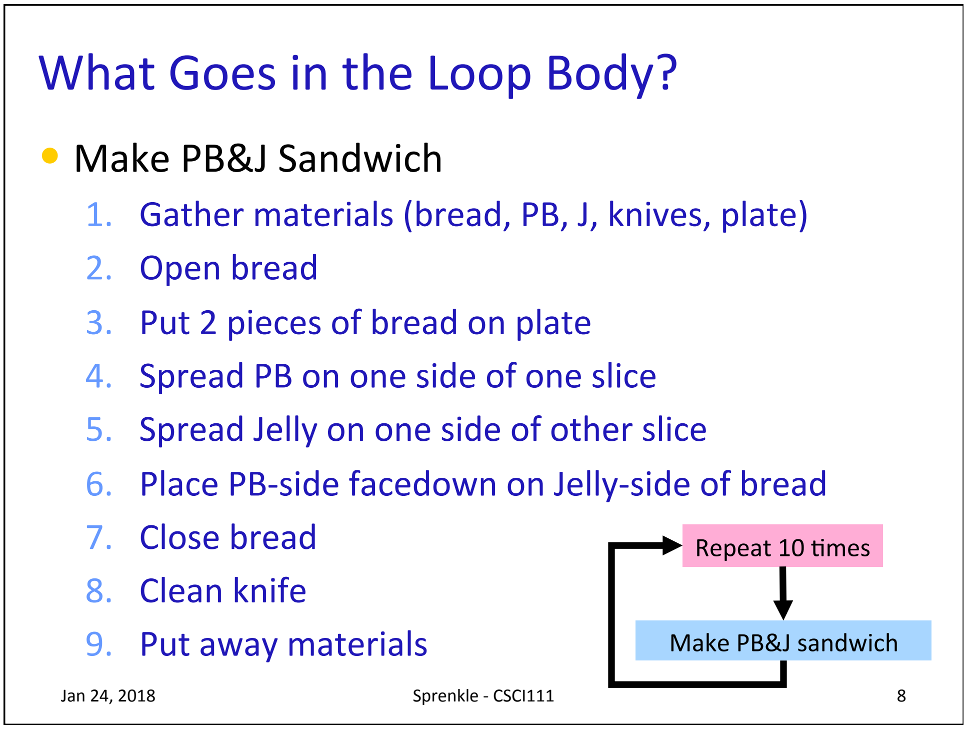
Make 10 PB&J sandwiches

Make PB&J sandwich
Make PB&J sandwich
Make PB&J sandwich
Make PB&J sandwich
Make PB&J sandwich
Make PB&J sandwich
M
M
Make PB&J sandwich
Make PB&J sandwich

Repetition is common in programming. Is there some simpler way to say that we want to repeat something?



Looping/Repetition



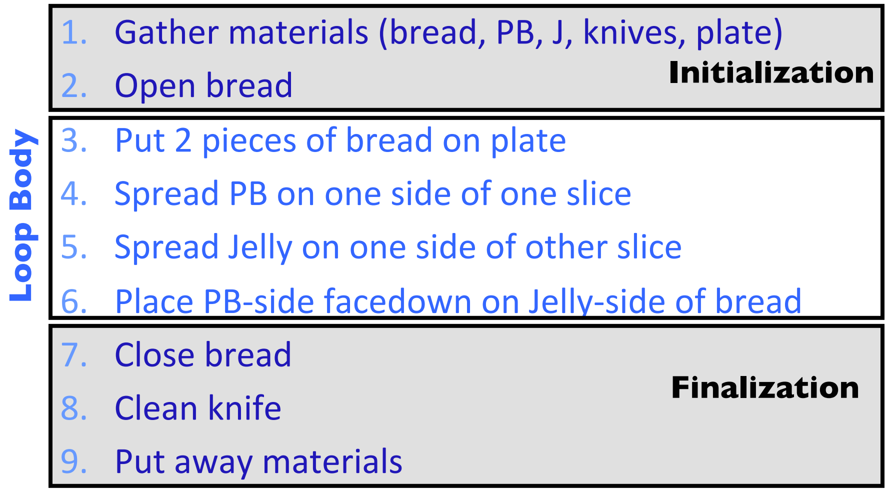
What Goes in the Loop Body?

- Make PB&J Sandwich
  1. Gather materials (bread, PB, J, knives, plate)
  2. Open bread
  3. Put 2 pieces of bread on plate
  4. Spread PB on one side of one slice
  5. Spread Jelly on one side of other slice
  6. Place PB-side facedown on Jelly-side of bread
  7. Close bread
  8. Clean knife
  9. Put away materials

Repeat 10 times
Make PB&J sandwich

What Goes in the Loop Body?

- Make PB&J Sandwich



1. Gather materials (bread, PB, J, knives, plate)
2. Open bread

Initialization

Loop Body

3. Put 2 pieces of bread on plate
4. Spread PB on one side of one slice
5. Spread Jelly on one side of other slice
6. Place PB-side facedown on Jelly-side of bread

7. Close bread
8. Clean knife
9. Put away materials

Finalization

Repetition in Action

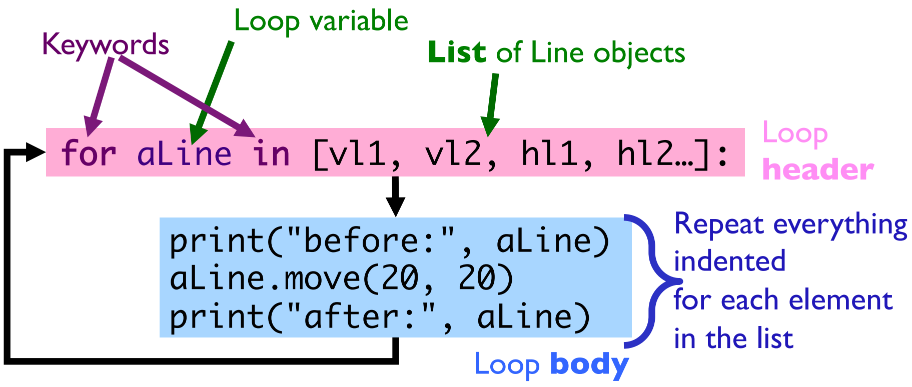
```
# recall: created two vertical and two horizontal lines

…


for aLine in [vertLine1, vertLine2, horizLine1, horizLine2]:
    print("before:", aLine)
    aLine.move(20, 20)
    print("after:", aLine)
```

Run the program several times. What happened?
Change it a bit – what happens now?
Can we explain this code?

tictactoe_withfor.py

The for Loop

Do <something> for each element in the list

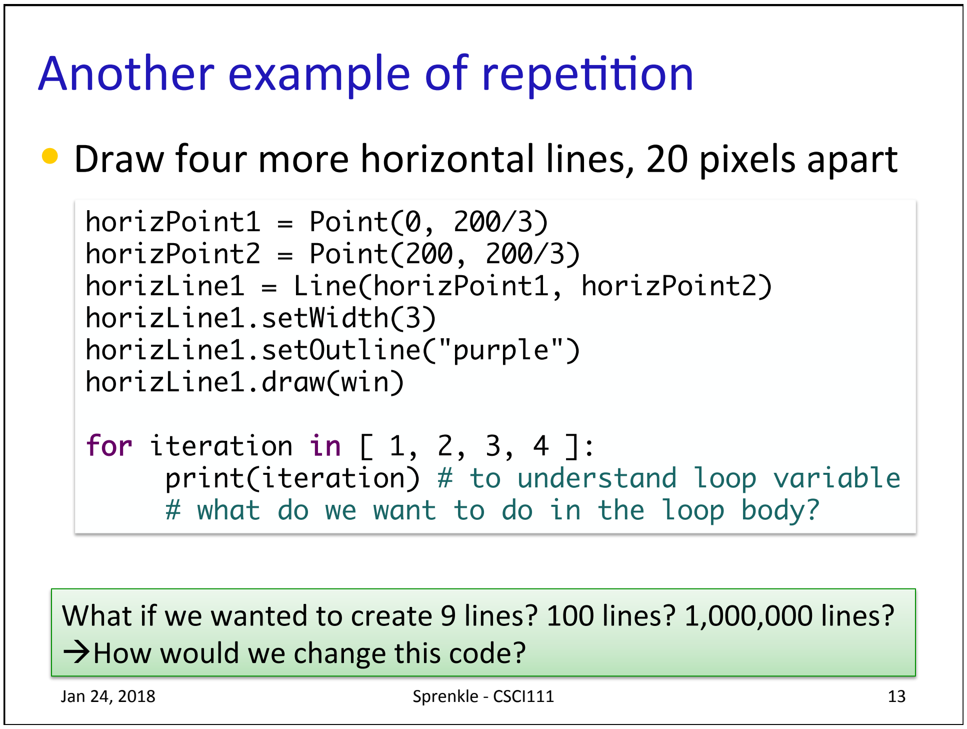


```
for aLine in [vl1, vl2, hl1, hl2…]:
    print("before:", aLine)
    aLine.move(20, 20)
    print("after:", aLine)
```

Another example of repetition

- Draw four more horizontal lines, 20 pixels apart

```
horizPoint1 = Point(0, 200/3)
horizPoint2 = Point(200, 200/3)
horizLine1 = Line(horizPoint1, horizPoint2)
horizLine1.setWidth(3)
horizLine1.setOutline("purple")
horizLine1.draw(win)

for iteration in [ 1, 2, 3, 4 ]:
     print(iteration) # to understand loop variable
     # what do we want to do in the loop body?
```

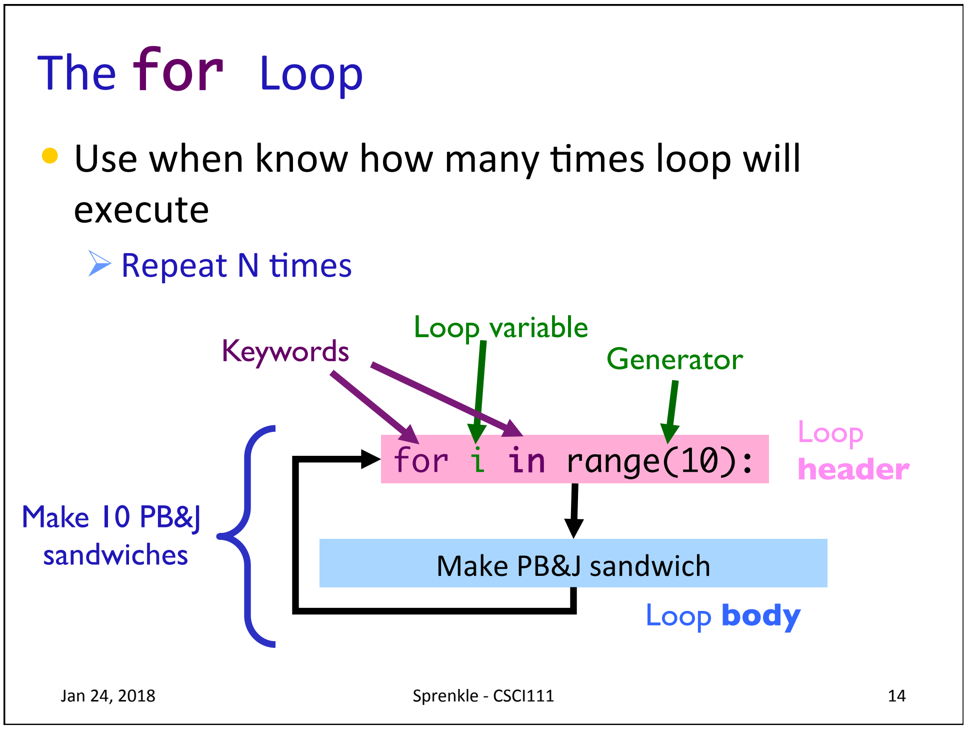


Another example of repetition

- Draw four more horizontal lines, 20 pixels apart

```
horizPoint1 = Point(0, 200/3)
horizPoint2 = Point(200, 200/3)
horizLine1 = Line(horizPoint1, horizPoint2)
horizLine1.setWidth(3)
horizLine1.setOutline("purple")
horizLine1.draw(win)

for iteration in [ 1, 2, 3, 4 ]:
     print(iteration) # to understand loop variable
     # what do we want to do in the loop body?
```

What if we wanted to create 9 lines? 100 lines? 1,000,000 lines?
→How would we change this code?



The for Loop

- Use when know how many times loop will execute
  - Repeat N times

Keywords

Loop variable

Generator

```
for i in range(10):
```

Loop **header**

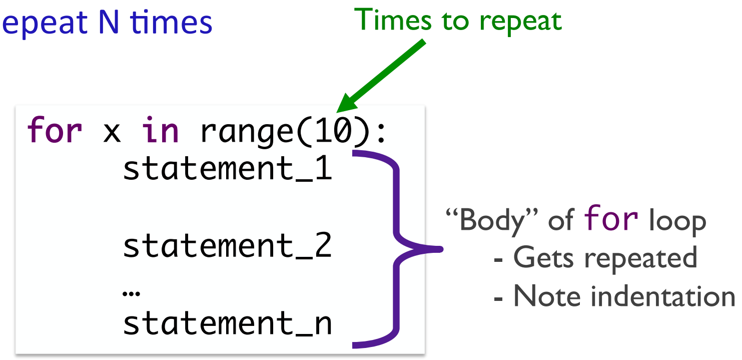
Make 10 PB&J sandwiches

Make PB&J sandwich

Loop **body**

for Loop Syntax and Semantics

- Use when know how many times loop will execute
  - Repeat N times



Times to repeat

```
for x in range(10):
      statement_1

      statement_2
      …
      statement_n
```

"Body" of for loop
- Gets repeated
- Note indentation

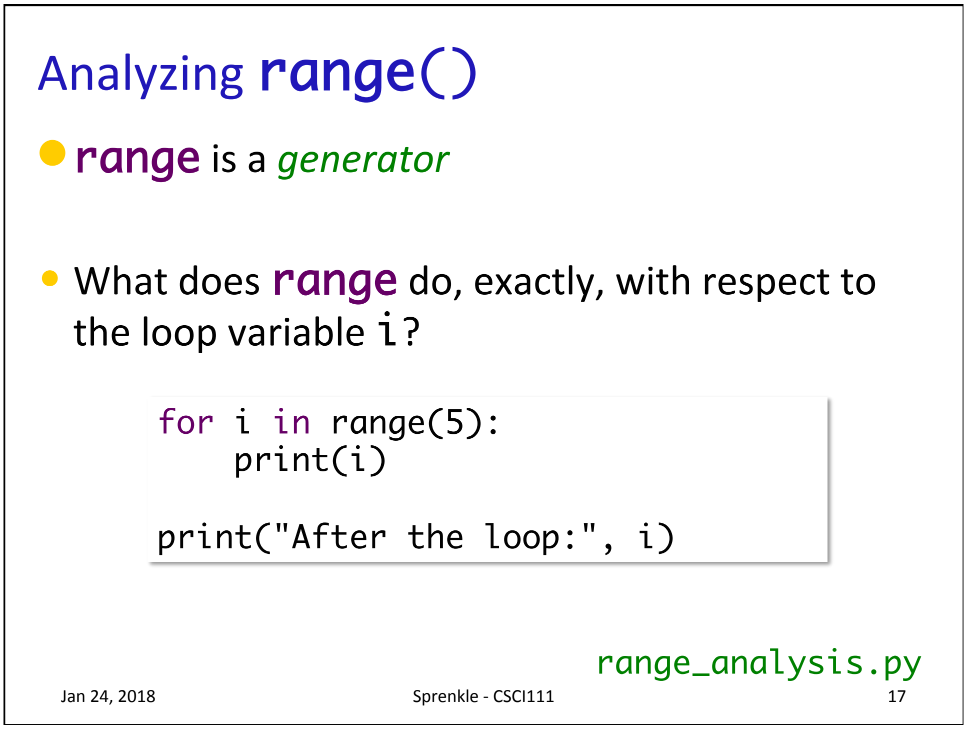
Using the for Loop

- If only ***one*** statement to repeat,
  - Body can be on same line as header

```
for variable in range(5): print("Hello!")
```

In general, I don't recommend writing this way.
Not that difficult to have on a separate line.
Often need to put more in body anyway.

simple_for.py

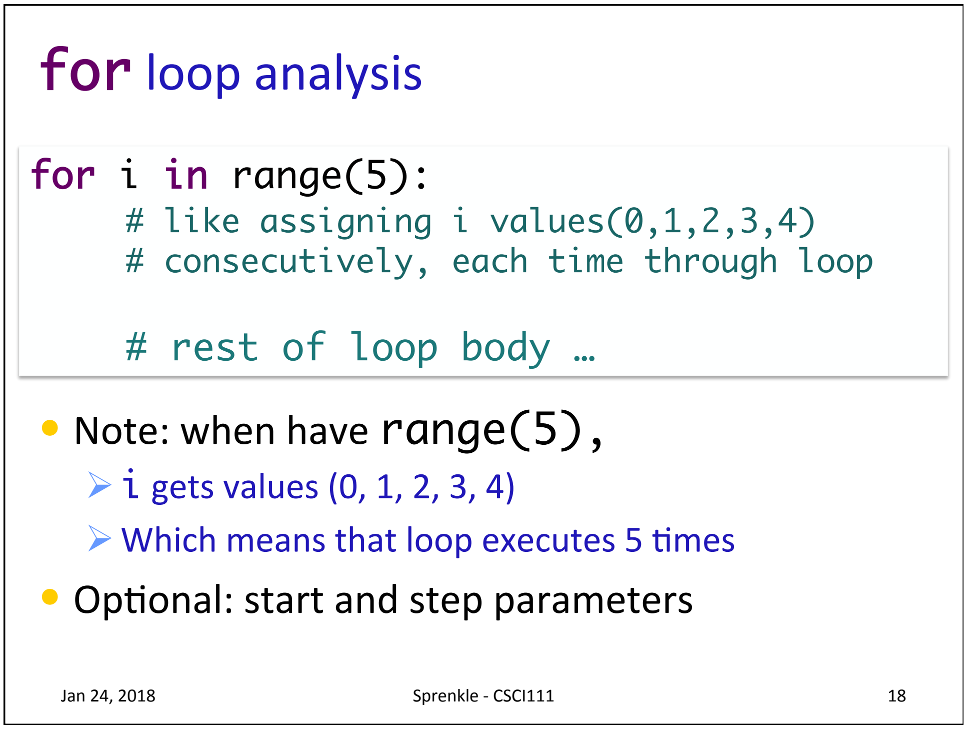


Analyzing **range()**

- **range** is a *generator*
- What does **range** do, exactly, with respect to the loop variable `i`?

```
for i in range(5):
    print(i)

print("After the loop:", i)
```

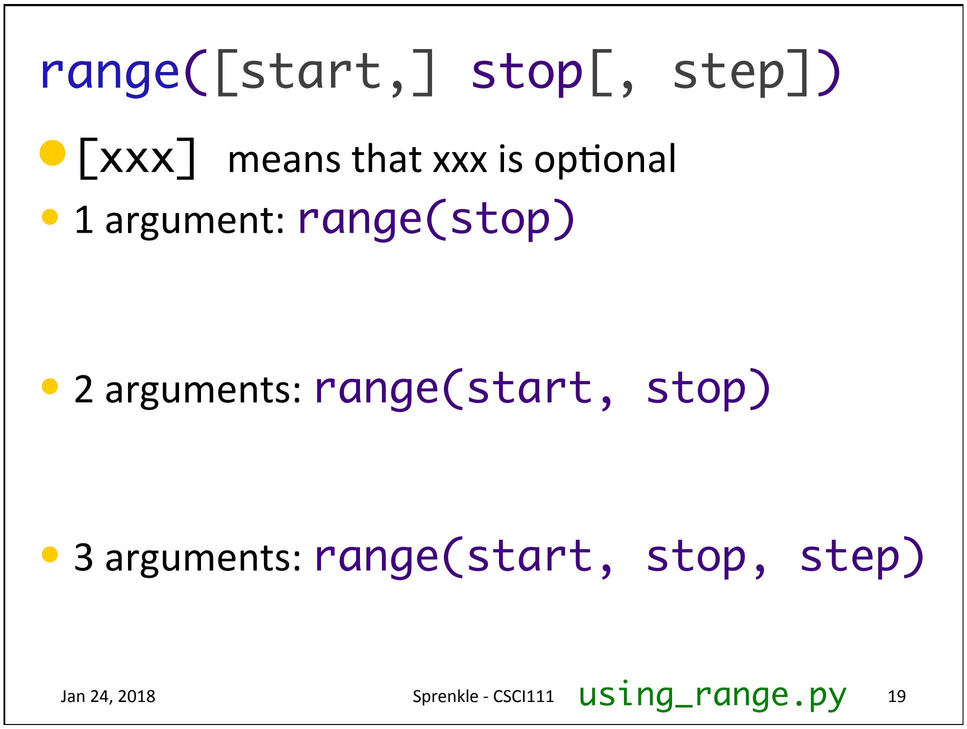
range_analysis.py



for loop analysis

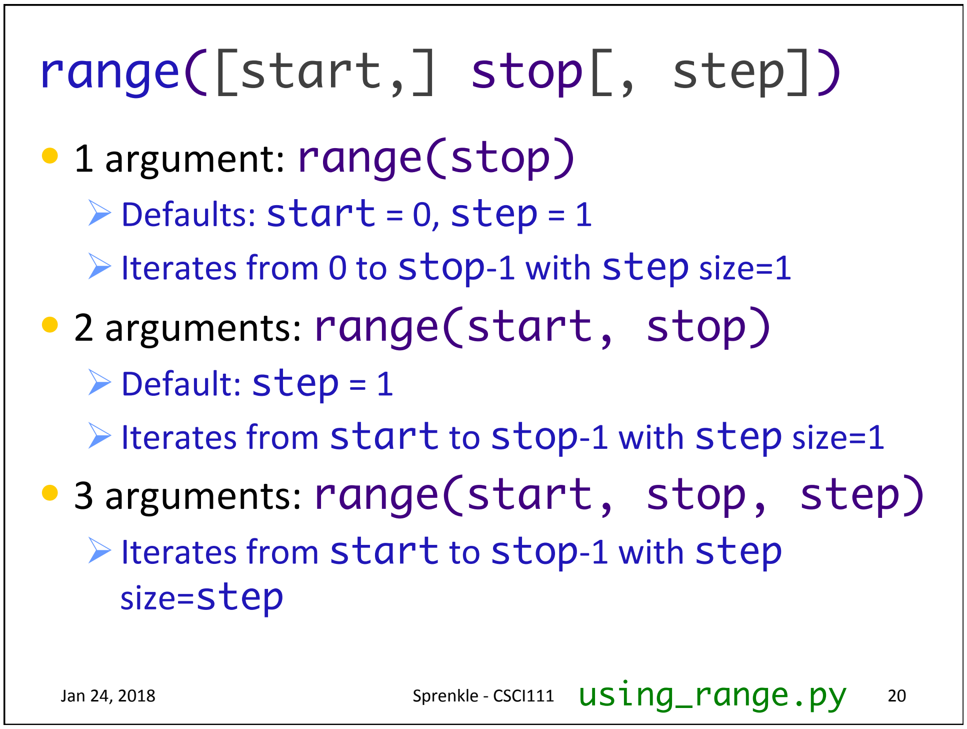
```
for i in range(5):
     # like assigning i values(0,1,2,3,4)
     # consecutively, each time through loop

     # rest of loop body …
```

- Note: when have `range(5)`,
  - `i` gets values (0, 1, 2, 3, 4)
  - Which means that loop executes 5 times
- Optional: start and step parameters



range([start,] stop[, step])

- [xxx] means that xxx is optional
- 1 argument: range(stop)
- 2 arguments: range(start, stop)
- 3 arguments: range(start, stop, step)

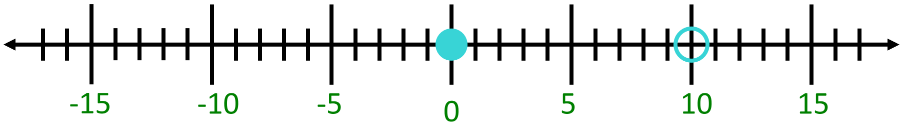


range([start,] stop[, step])

- 1 argument: range(stop)
  - Defaults: start = 0, step = 1
  - Iterates from 0 to stop-1 with step size=1
- 2 arguments: range(start, stop)
  - Default: step = 1
  - Iterates from start to stop-1 with step size=1
- 3 arguments: range(start, stop, step)
  - Iterates from start to stop-1 with step size=step

range

- range is a number generator
  - 1 argument: `range(stop)`
  - 2 arguments: `range(start, stop)`
  - 3 arguments: `range(start, stop, step)`

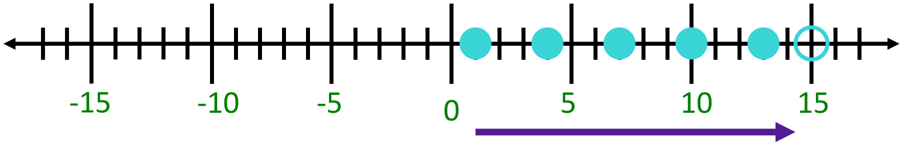


```
range(10)
range(0,10)
range(0,10,1)
```

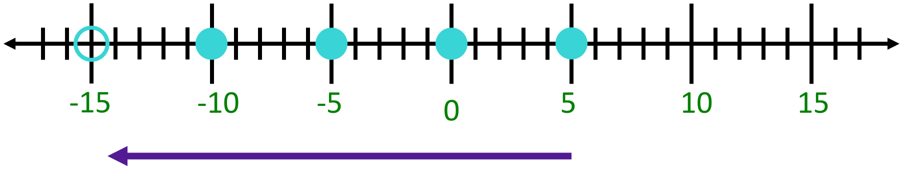
[start, stop)

Sequence generated by range

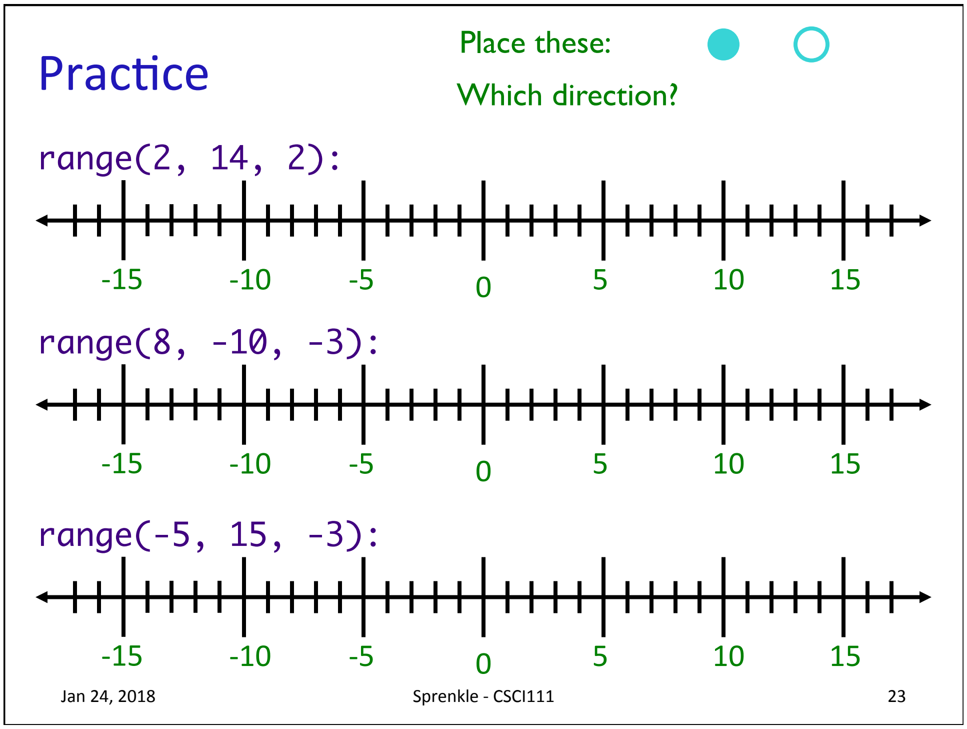
`range(1, 15, 3):`



`range(5, -15, -5):`



more_range_examples.py



Practice

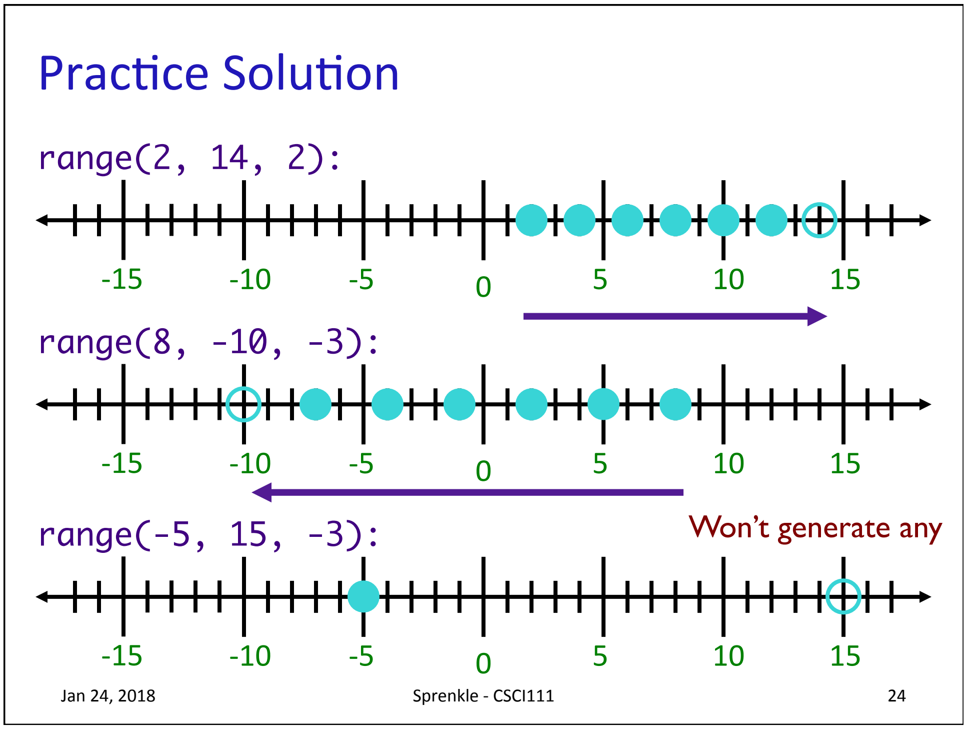
Place these: ● ○

Which direction?

range(2, 14, 2):

range(8, -10, -3):

range(-5, 15, -3):



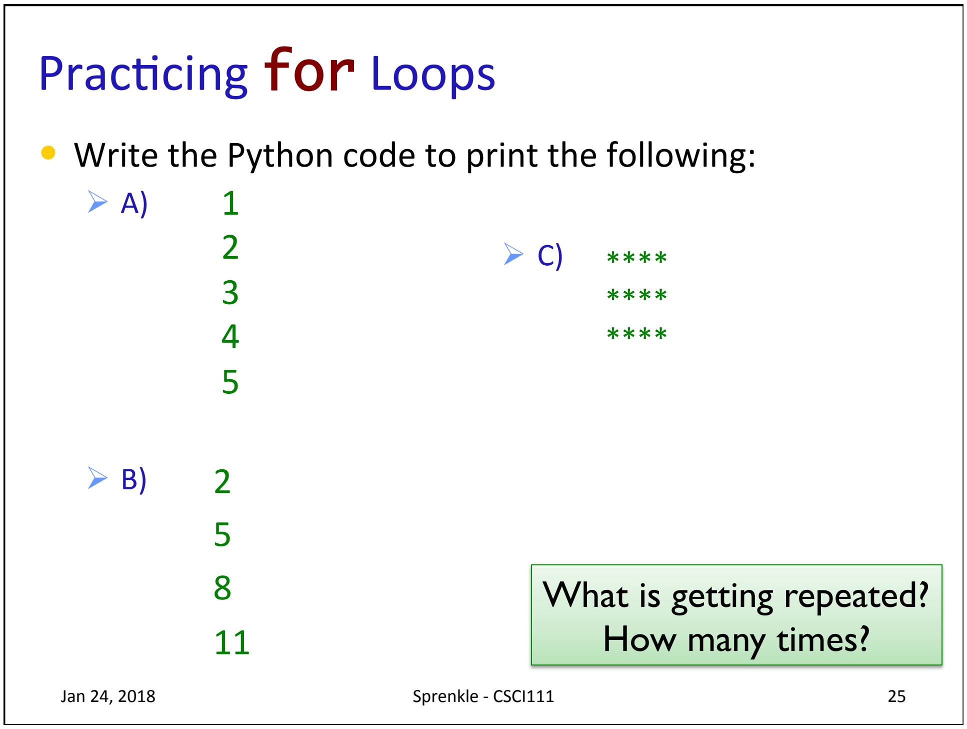
Practice Solution

range(2, 14, 2):

range(8, -10, -3):

range(-5, 15, -3):

Won't generate any



Practicing for Loops

- Write the Python code to print the following:
  - A)

```
1
2
3
4
5
```

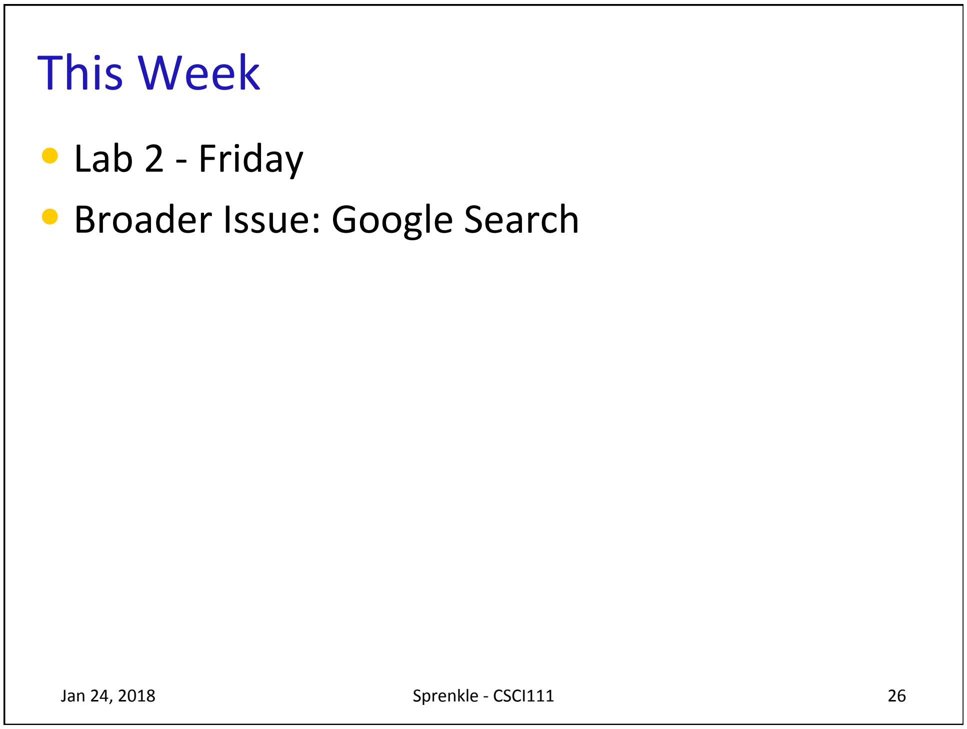
  - B)

```
2

5

8

11
```

  - C)

```
****
****
****
```

What is getting repeated?
How many times?



This Week

- Lab 2 - Friday
- Broader Issue: Google Search