# On your way in…

Pick-up:

1. Your graded Homework 1

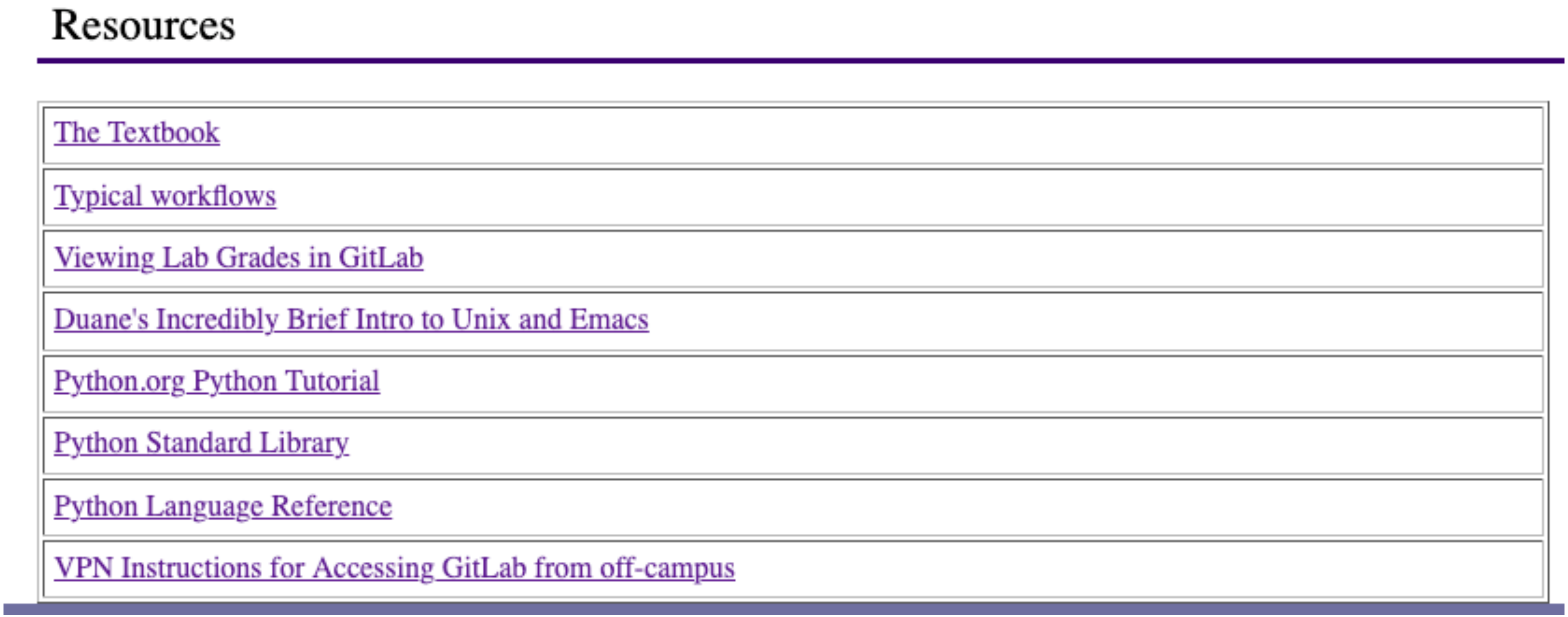
2. POGIL 16: Lists

# CS THESIS PROPOSALS THIS FRIDAY 2:35P IN TCL 202

- Phoebe Huang
- Yo Akiyama
- Emmie Hine

Come see what CS seniors are up to!
If you're interested in majoring in Computer Science, you'll need 20 colloquium to graduate, and any you sign-in for now will count!

# “Resources” on Course Website are Helpful



## Resources

- The Textbook
- Typical workflows
- Viewing Lab Grades in GitLab
- Duane's Incredibly Brief Intro to Unix and Emacs
- Python.org Python Tutorial
- Python Standard Library
- Python Language Reference
- VPN Instructions for Accessing GitLab from off-campus

Forgot how to git clone, or move through directories in Terminal? WORKFLOWS
Need to check your lab grades? VIEWING LAB GRADES
Not sure of commands in Terminal? DUANE’S INCREDIBLY BRIEF INTRO
New Python concept and the book/POGIL/lecture isn’t helping? PYTHON TUTORIAL

# Lab 3 will be released today

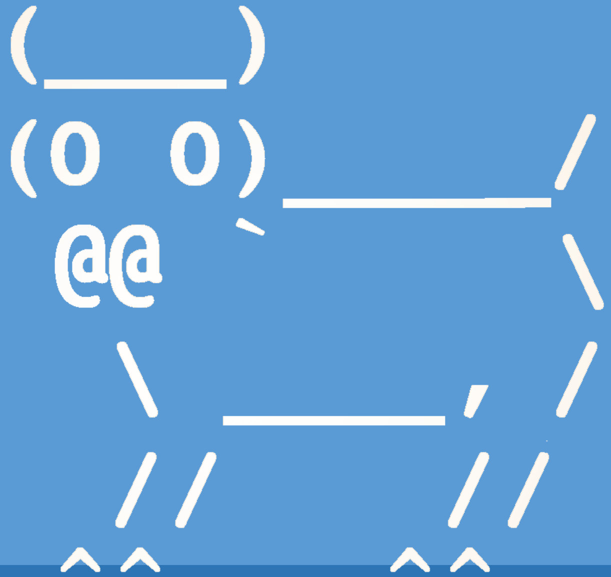
- It’s focused on strings, but requires some iteration, lists, and file reading
  - We’ll go over all but strings today
- Might want to focus on `words(filename)` and `sized(n, l)` methods this weekend
  - Others require strings!

# Welcome to CS 134!

Introduction to Computer Science

Iris Howley

-Lists-

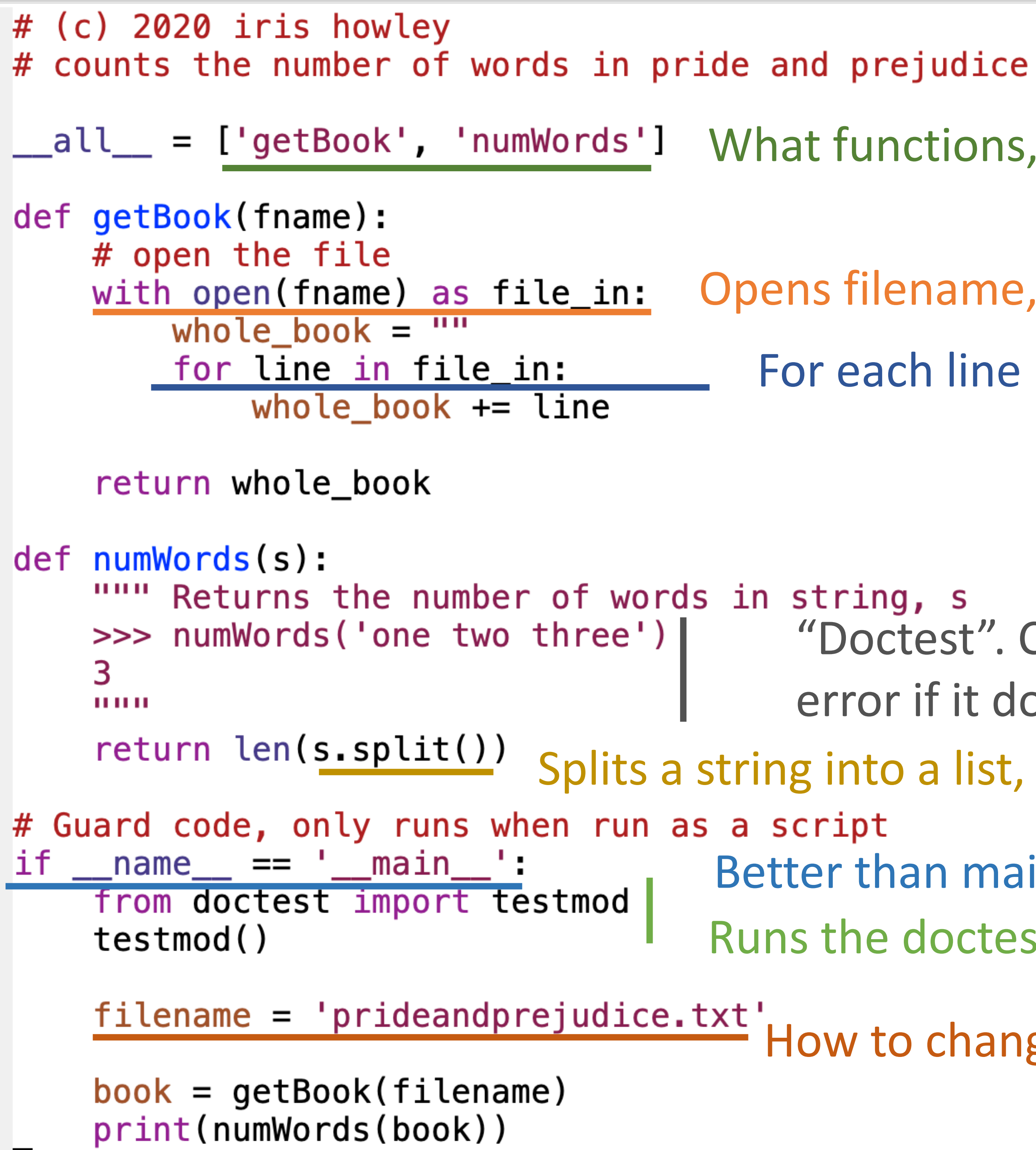


Spring 2020

# WELL-BUILT MODULES

## Making our code reusable

(Handy for this week's lab)



```
# (c) 2020 iris howley
# counts the number of words in pride and prejudice

__all__ = ['getBook', 'numWords']

def getBook(fname):
    # open the file
    with open(fname) as file_in:
        whole_book = ""
        for line in file_in:
            whole_book += line

    return whole_book

def numWords(s):
    """ Returns the number of words in string, s
    >>> numWords('one two three')
    3
    """
    return len(s.split())

# Guard code, only runs when run as a script
if __name__ == '__main__':
    from doctest import testmod
    testmod()

    filename = 'prideandprejudice.txt'

    book = getBook(filename)
    print(numWords(book))
```

What functions, variables, etc. are imported with import *

Opens filename, fname, calls it file_in

For each line in our file, file_in

“Doctest”. Calls the code on the >>> line. Throws an error if it doesn’t return what’s shown on next line

Splits a string into a list, default is by ” “

Better than main() function. Only called when run as a script.

Runs the doctests when code is run as a script

How to change to asking user for filename?

# Currently…

Inside example.py:

- `def main():`
  - `print(“Inside main function”)`

- `# Function call to main`
- `main()`

**main() is called every time example.py is interpreted  by python**

**(1) Imported as a module [in interactive python]**

**(2) Run as a script**

**Not ideal!**

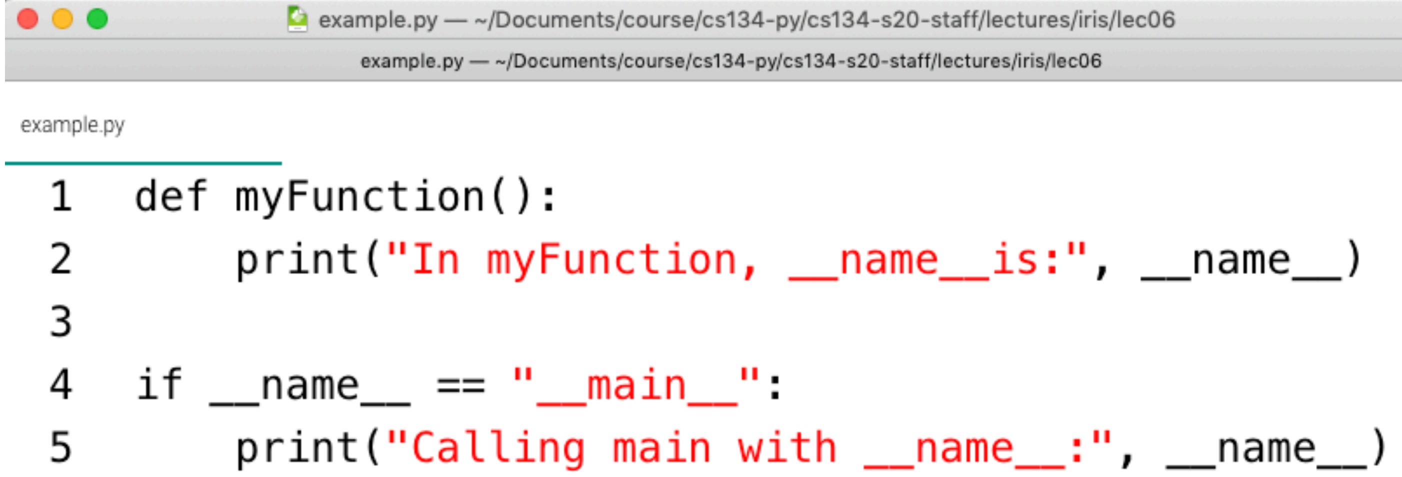
# Much Improved…

Inside example.py:

- `if __name__ == “__main__”:`
  - `print(“Inside main function”)`

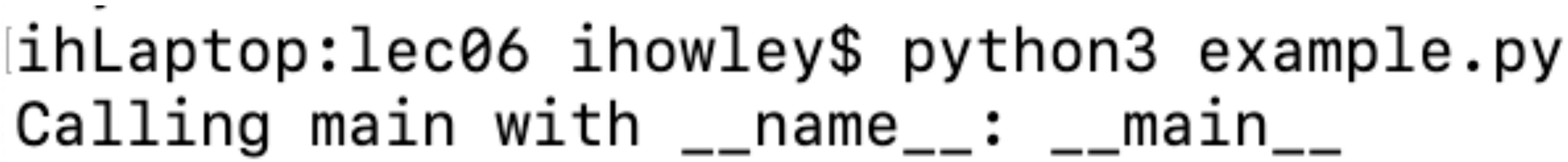
**This only happens when run as a script**

# What is __name__?


example.py — ~/Documents/course/cs134-py/cs134-s20-staff/lectures/iris/lec06
example.py — ~/Documents/course/cs134-py/cs134-s20-staff/lectures/iris/lec06
example.py

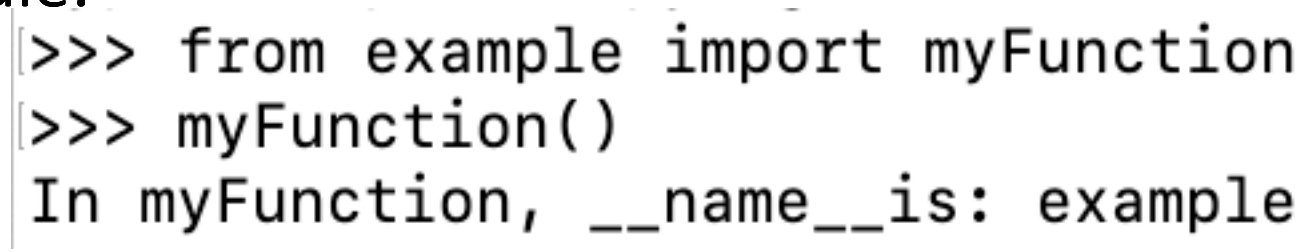
```python
def myFunction():
    print("In myFunction, __name__is:", __name__)

if __name__ == "__main__":
    print("Calling main with __name__:", __name__)
```

- Run as a script:



```
ihLaptop:lec06 ihowley$ python3 example.py
Calling main with __name__: __main__
```

- Imported as a module:



```
>>> from example import myFunction
>>> myFunction()
In myFunction, __name__is: example
```

- __name__ changes depending on if the program is run as a script or as a module.
- Only the program being run as a script gets the name __main__

# `__all__` special variable

- If the variable starts/ends with “_ _” it’s a special python variable
- We saw this with `__name__`

- `__all__` is another special variable
- Whatever is stored in `__all__` is imported when the user types:
  - `from ____ import *`
- Any function/variable/etc. that’s not included in __all__ can be imported explicitly
  - `from <module name> import <not-included-in-star-variable/function>`

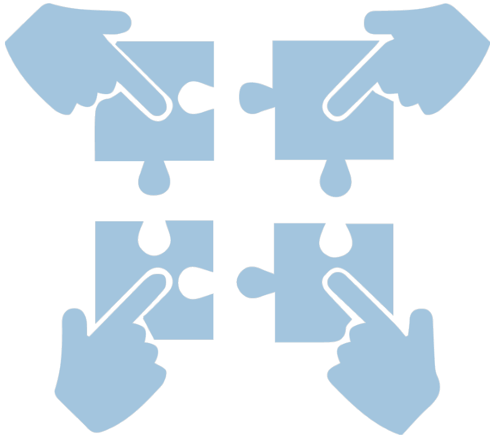
# TODAY'S LESSON

## Lists hold many objects

(Great for when you have large quantities of items.)

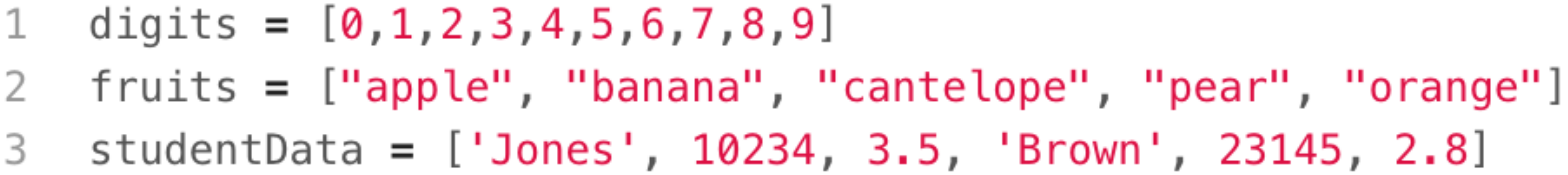
# POGIL – Activity 16: Lists

- Look at Python Activity 16, Questions 1-3c
- Find a partner and talk through the questions together

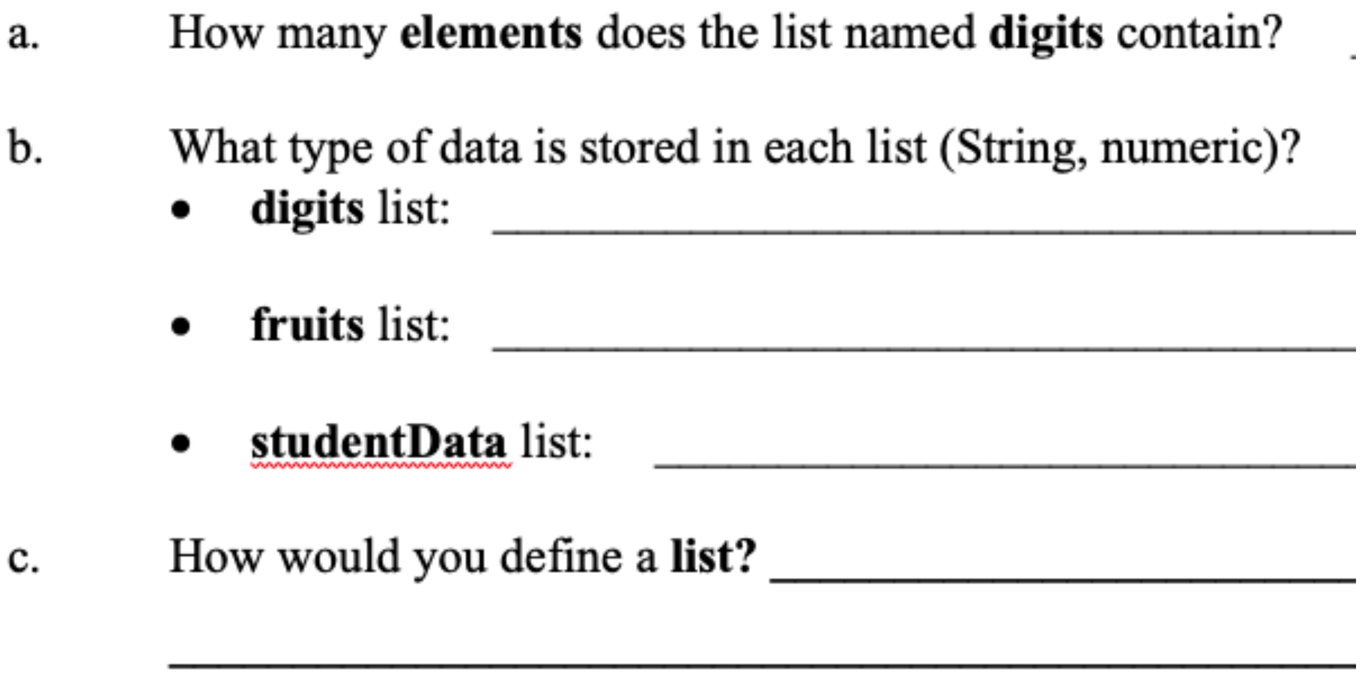


# POGIL – Activity 16: Question 1

- Look at POGIL 16 – Lists, Question 1



```
digits = [0,1,2,3,4,5,6,7,8,9]
fruits = ["apple", "banana", "cantelope", "pear", "orange"]
studentData = ['Jones', 10234, 3.5, 'Brown', 23145, 2.8]
```



a. How many **elements** does the list named **digits** contain?

b. What type of data is stored in each list (String, numeric)?

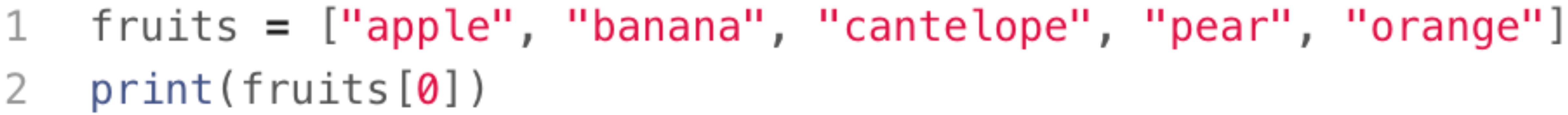
- **digits** list: ____________________
- **fruits** list: ____________________
- **studentData** list: ____________________

c. How would you define a **list?** ____________________

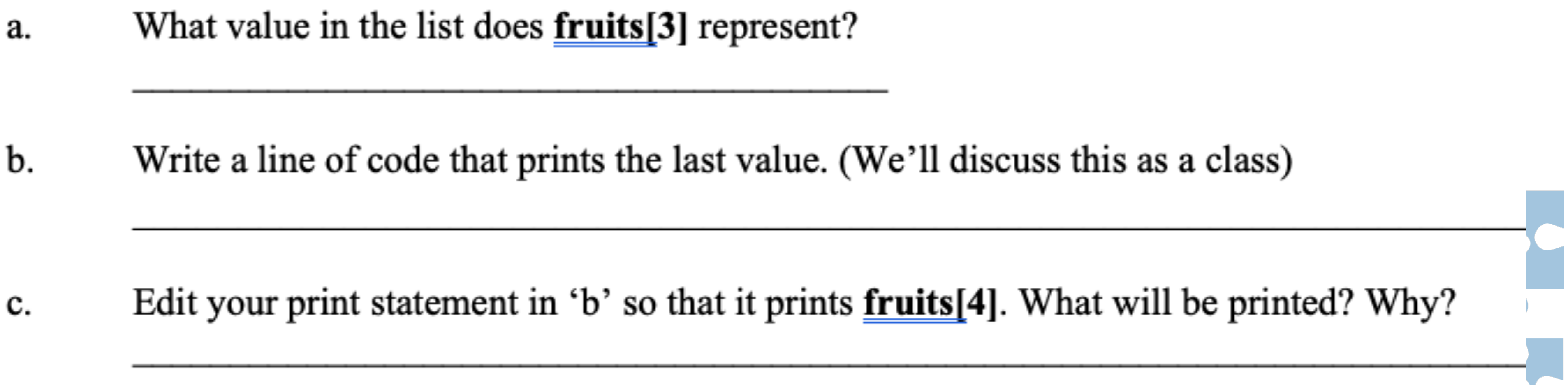
____________________

# POGIL – Activity 16: Question 2

- Look at POGIL 16 – Lists, Question 2



```python
fruits = ["apple", "banana", "cantelope", "pear", "orange"]
print(fruits[0])
```



a. What value in the list does **fruits[3]** represent?

______________________________________________

b. Write a line of code that prints the last value. (We’ll discuss this as a class)

______________________________________________

c. Edit your print statement in ‘b’ so that it prints **fruits[4]**. What will be printed? Why?

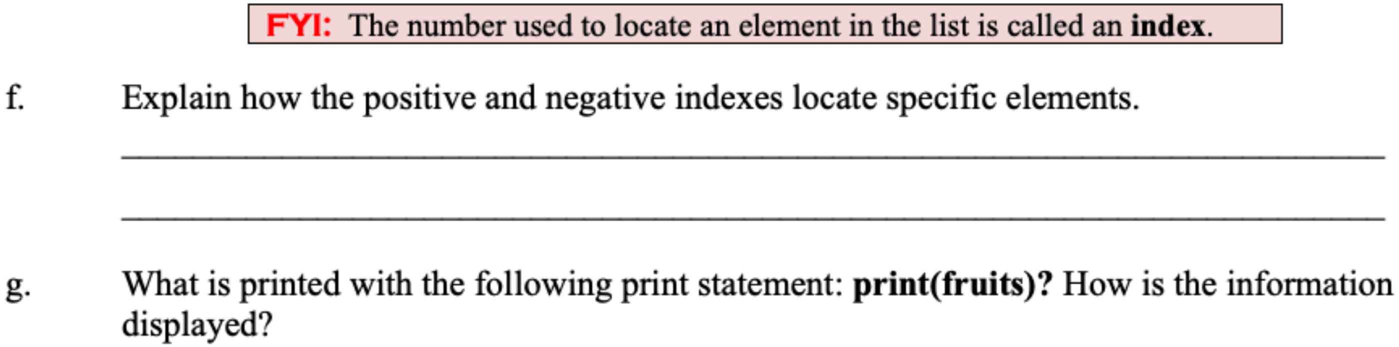
______________________________________________

# POGIL – Activity 16: Question 2

- Look at POGIL 16 – Lists, Question 2

```
fruits = [‘apple’, ’banana’, ’cantelope’, ’pear’]
print(fruits[0])
```

- d. What will fruits[-1] print?
  - ‘pear’ Why?
- e. What about fruits[-2]?
  - ‘cantelope’ Why?



**FYI:** The number used to locate an element in the list is called an **index**.

f. Explain how the positive and negative indexes locate specific elements.

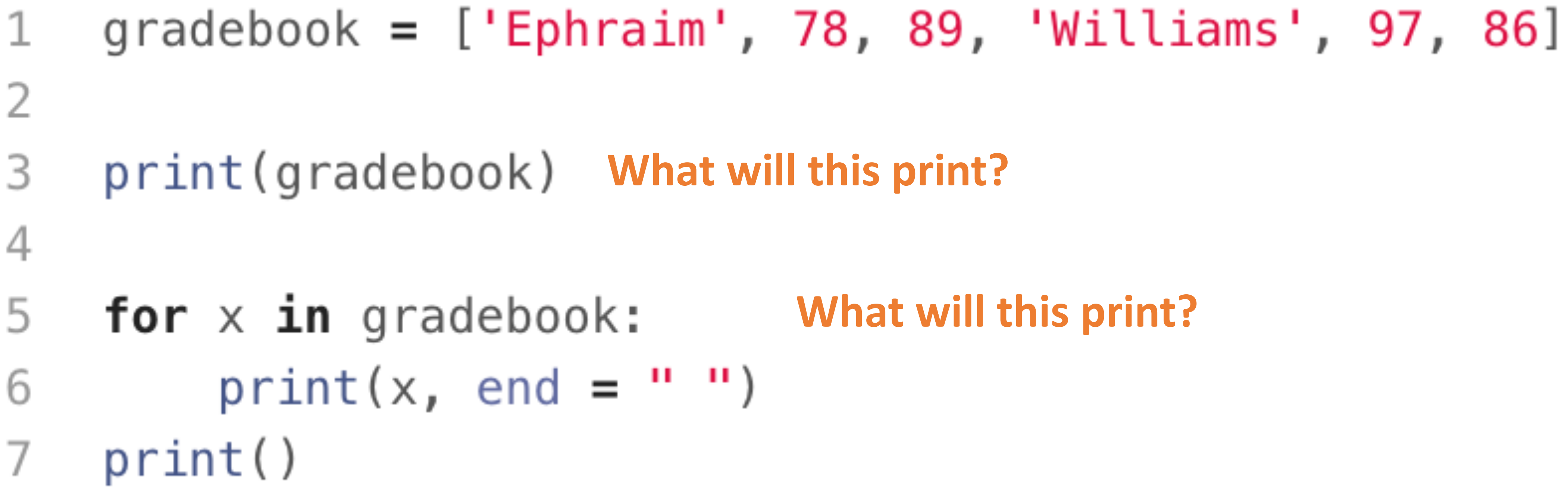
________________________________________________________________

________________________________________________________________

g. What is printed with the following print statement: **print(fruits)?** How is the information displayed?

# POGIL – Activity 16: Question 3

- Look at POGIL 16 – Lists, Question 3



```python
gradebook = ['Ephraim', 78, 89, 'Williams', 97, 86]

print(gradebook)

for x in gradebook:
    print(x, end = " ")
print()
```

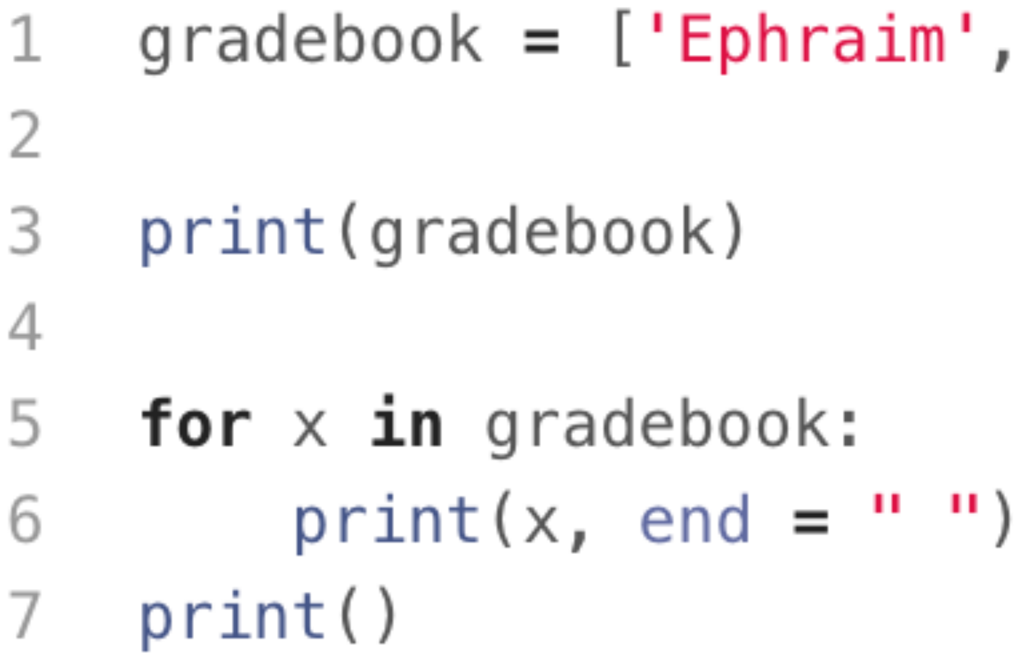
What will this print?

What will this print?

**How does the output differ?**

# POGIL – Activity 16: Question 3

- Look at POGIL 16 – Lists, Question 3



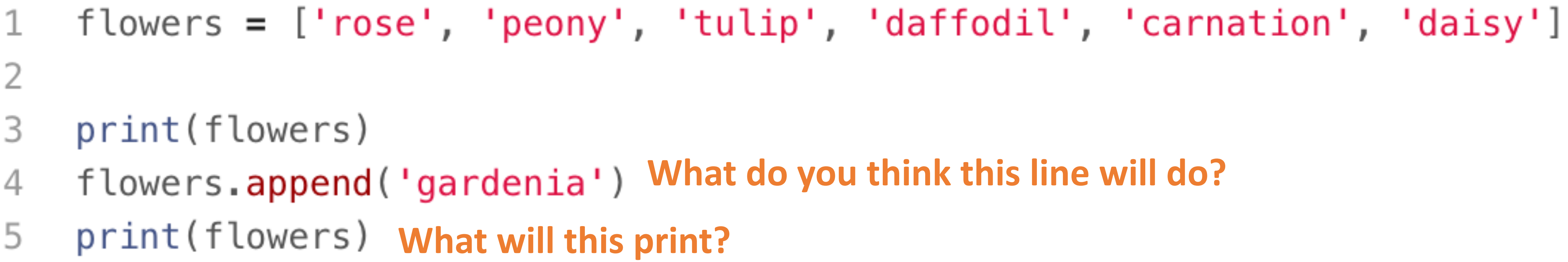
```python
gradebook = ['Ephraim', 78, 89, 'Williams', 97, 86]

print(gradebook)

for x in gradebook:
    print(x, end = " ")
print()
```

**When iterating over a sequence, we start with the first element in the sequence, and end with the last (unless otherwise stated)**

- This for loop doesn’t use range(..)
- How does it know where to start and when to stop?

# POGIL – Activity 16: Question 4

- Look at POGIL 16 – Lists, Question 4



```
flowers = ['rose', 'peony', 'tulip', 'daffodil', 'carnation', 'daisy']

print(flowers)
flowers.append('gardenia')
print(flowers)
```

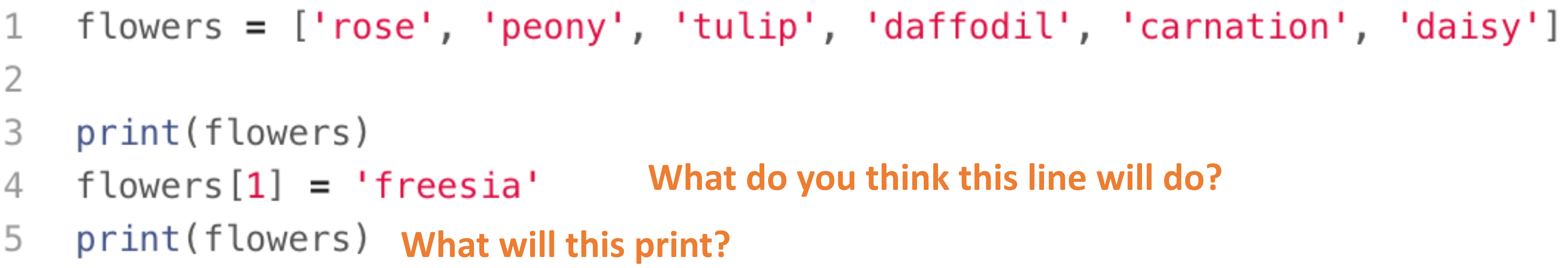
What do you think this line will do?

What will this print?

**We can add elements to the end of a list with the .append(..) method**

# POGIL – Activity 16: Question 8

- Look at POGIL 16 – Lists, Question 8



```python
flowers = ['rose', 'peony', 'tulip', 'daffodil', 'carnation', 'daisy']

print(flowers)
flowers[1] = 'freesia'
print(flowers)
```

What do you think this line will do?

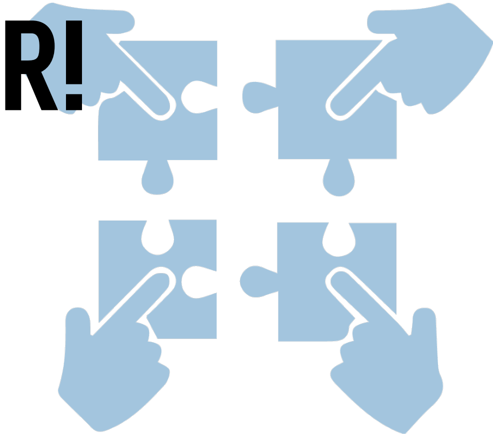
What will this print?

**We can reassign values in a list by placing the list with index on the left hand side of an assignment operator.**

# YOU SHOULD COMPLETE THE REST OF ALL POGILS OUTSIDE OF CLASS.

BEST DONE WITH A PARTNER OR STUDY GROUP.

CHECK YOUR ANSWERS ON A COMPUTER!



# Using Lists

**Note the square brackets indicate it’s a list!**

```
myList = [’apples’, ‘oranges’, 55]
```

```
myList[0]
   ‘apples’
```

**We can access elements in the list by using square brackets with the desired element’s index between**

```
myList[2]
   55
```

**Indices start at 0!**

```
myList[-1]
   55
```

**Negative index starts at end of the list**

```
myList[-2]
   ‘oranges’
```

# QUESTIONS?

# Leftover Slides