CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

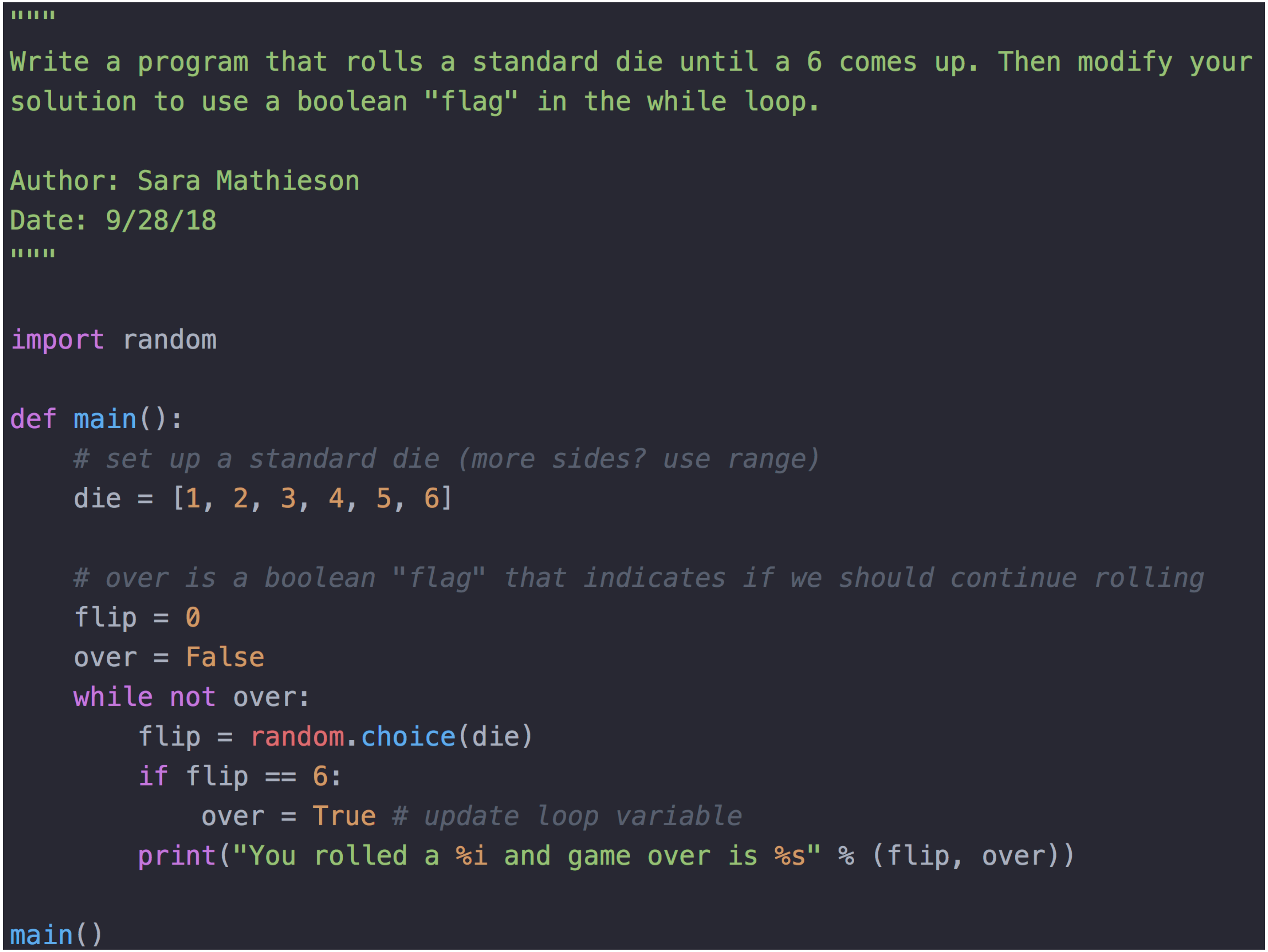
Outline Oct 1:

Notes

- **Lab 4** due **Saturday** night
- **Quiz 2** this **Friday** in class
- **Office Hours 3-5pm Friday (or by appointment)**

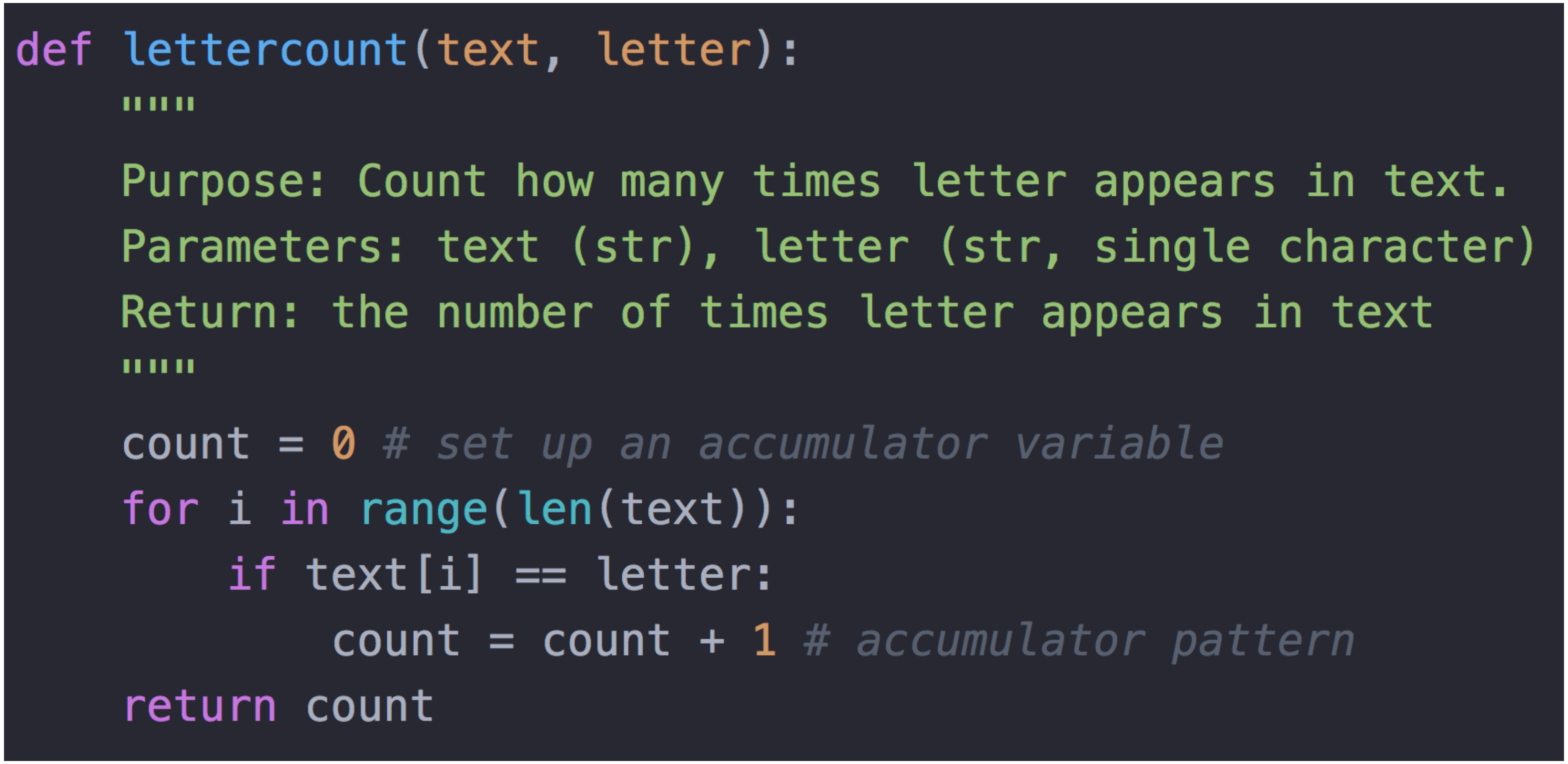
- While loops with boolean flags
- Function commenting
- Lists as a data structure
- Mutability and modifying lists
- Functions that modify lists
- List modifying practice: **build_list.py, shuffle_list.py**
- At end: notecard feedback (anonymous)
- Wednesday: stack diagrams with lists

While loops with boolean flags (example)



```python
"""
Write a program that rolls a standard die until a 6 comes up. Then modify your
solution to use a boolean "flag" in the while loop.

Author: Sara Mathieson
Date: 9/28/18
"""

import random

def main():
    # set up a standard die (more sides? use range)
    die = [1, 2, 3, 4, 5, 6]

    # over is a boolean "flag" that indicates if we should continue rolling
    flip = 0
    over = False
    while not over:
        flip = random.choice(die)
        if flip == 6:
            over = True # update loop variable
        print("You rolled a %i and game over is %s" % (flip, over))

main()
```

Commenting functions (example)



```python
def lettercount(text, letter):
    """
    Purpose: Count how many times letter appears in text.
    Parameters: text (str), letter (str, single character)
    Return: the number of times letter appears in text
    """
    count = 0 # set up an accumulator variable
    for i in range(len(text)):
        if text[i] == letter:
            count = count + 1 # accumulator pattern
    return count
```

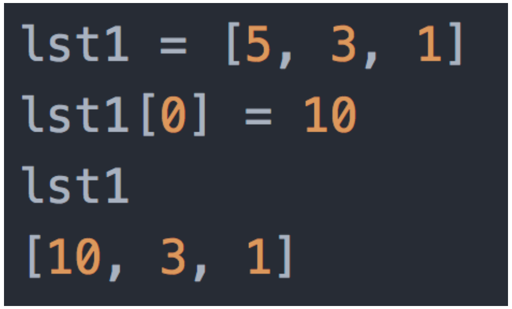
Lists

Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)

Lists as a mutable data structure

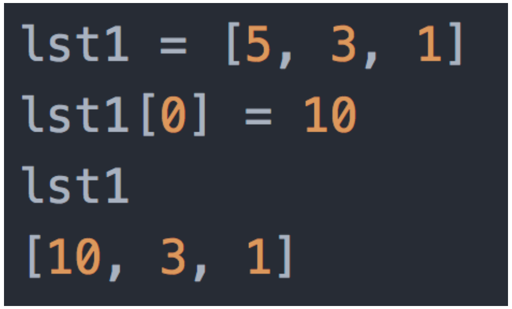
- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)



```
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

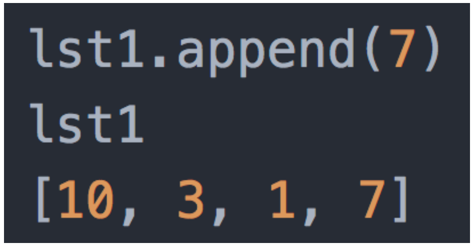
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)



```
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

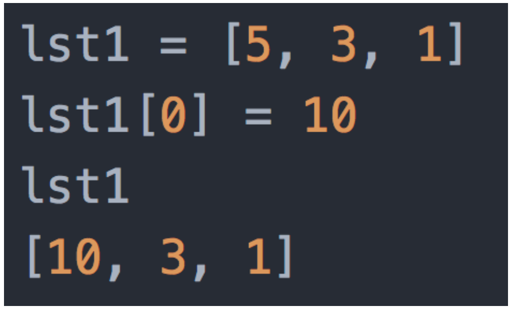
- We can add elements to a list



```
lst1.append(7)
lst1
[10, 3, 1, 7]
```

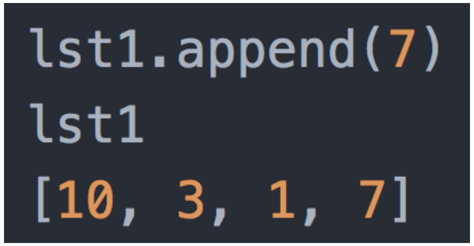
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)



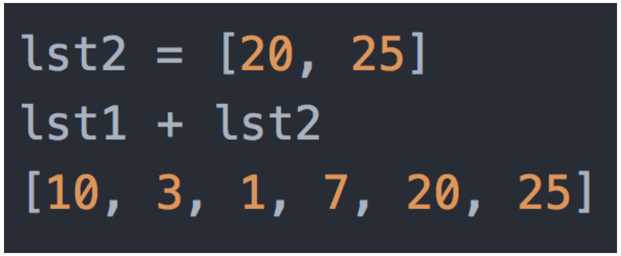
```
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

- We can add elements to a list

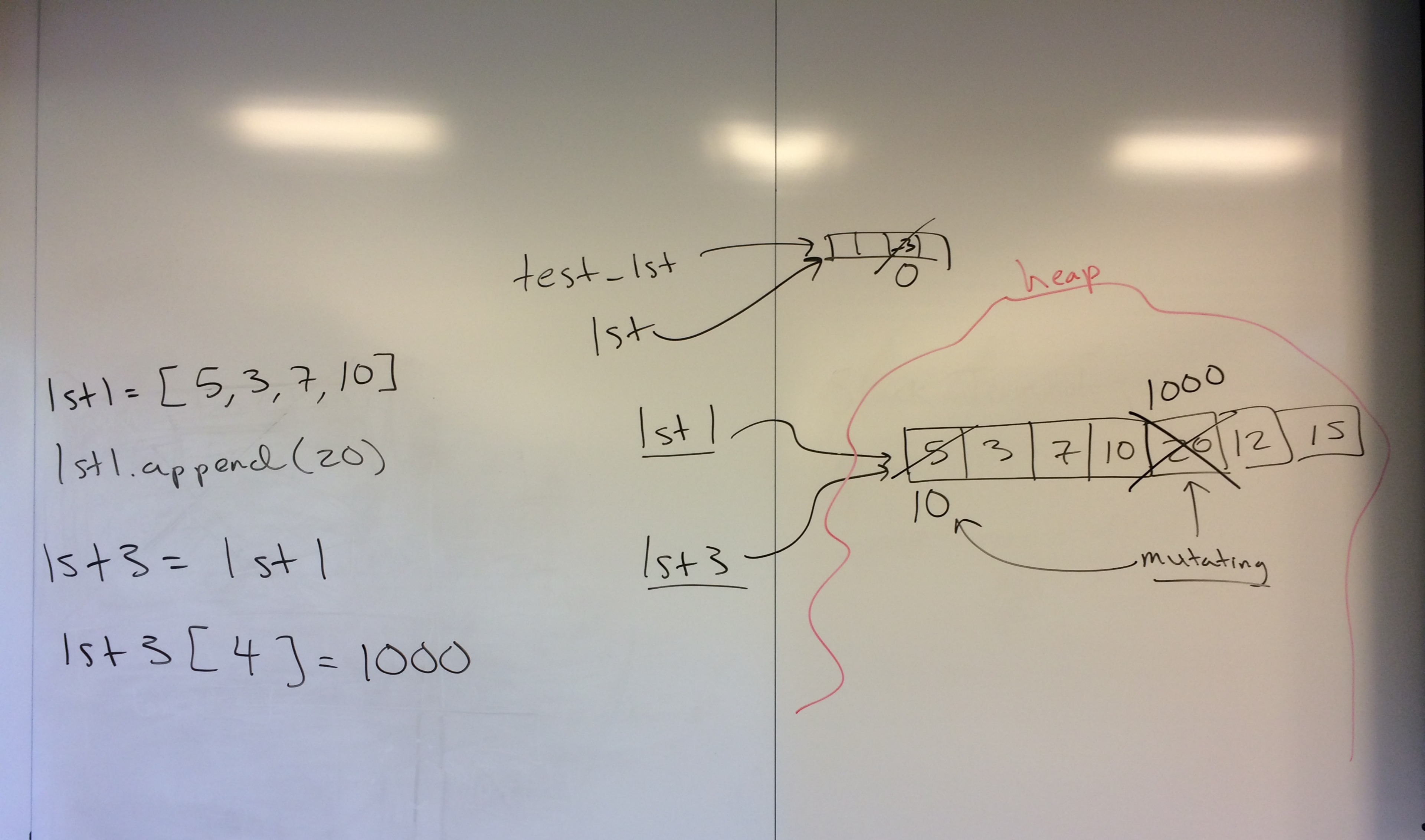


```
lst1.append(7)
lst1
[10, 3, 1, 7]
```

- Concatenating lists



```
lst2 = [20, 25]
lst1 + lst2
[10, 3, 1, 7, 20, 25]
```



```
lst1 = [5, 3, 7, 10]
lst1.append(20)

lst3 = lst1

lst3[4] = 1000
```

Mutating a list changes any variables that also point to the underlying data lists are **mutable!**

strings are **immutable!**

```
lst1 = [5, 3, 7, 10]

lst1.append(20)

lst3 = lst1

lst3[4] = 1000
```

List programs for today

- **build_list.py** (start together)
- **shuffle_list.py** (pair programming)
- Notecard feedback! On back board