

## Objectives

- Lists, continued
  - Making copies
  - Passing as parameters

March 10, 2023

Sprenkle - CSCI111

1

1

## Review

- What is a list?
- What is the syntax for a list?
- How can we make a list of numbers with a fixed step quickly?
- How are lists and strings similar?
  - What similar things can we do to lists and strings?
- How are they different?
  - What are the implications of those differences?
- What does `None` mean? When does it come up?

March 10, 2023

Sprenkle - CSCI111

2

2

## Review: Lists and Strings in Common

Concatenation	<code>&lt;seq&gt; + &lt;seq&gt;</code>
Repetition	<code>&lt;seq&gt; * &lt;int-expr&gt;</code>
Indexing	<code>&lt;seq&gt;[&lt;int-expr&gt;]</code>
Length	<code>len(&lt;seq&gt;)</code>
Slicing	<code>&lt;seq&gt;[:]</code>
Iteration	<code>for &lt;var&gt; in &lt;seq&gt;:</code>
Membership	<code>&lt;expr&gt; in &lt;seq&gt;</code>

March 10, 2023

Sprenkle - CSC1111

3

3

## Review: Strings vs. Lists

- Strings are **immutable**
  - Can't be mutated?
  - Err, can't be modified/changed
    - A change requires recreation
- Lists are **mutable**
  - Can be changed
    - Called "change in place"
  - Changes how we call/use methods

```
groceryList=["milk", "eggs", "bread", "Doritos", "OJ", "sugar"]
```

```
groceryList[0] = "skim milk"  
groceryList[3] = "popcorn"
```

```
groceryList is now ["skim milk", "eggs", "bread", "popcorn", "OJ", "sugar"]
```

March 10, 2023

Sprenkle - CSC1111

4

4

## Review: Strings vs. Lists

### Strings

- Methods that are meant to change a string return a *changed copy* of the String
- Consequence: Call the method and assign that to a variable
- Example use:
  - `upper = mystr.upper()`

### Lists

- Methods that are meant to change a list change the list *in place*
  - *Don't return anything*
- Consequence: Call the method but don't assign it to a variable
- Example use:
  - `myList.sort()`

March 10, 2023

Sprenkle - CSCI111

5

5

## Review: Special Value: **None**

(Similar to `null` in Java)

- Special value we can use
  - E.g., Return value from function/method when there is an error
  - Or if function/method does not return anything
- If you execute 

```
myList = myList.sort()
print(myList)
```

  - Prints `None` because `myList.sort()` does **not return anything**

How do you fix the above code to do what you want?

March 10, 2023

6

6

## Lists vs. Arrays

- Briefly, lists are similar to arrays in other languages
  - More similar to *Vectors* in C++ and *ArrayLists* in Java
- Typically, arrays have **fixed** lengths
  - Can't insert and remove elements from arrays so that the length of the array changes
  - Need to make the array as big as you'll think you'll need

March 10, 2023

Sprenkle - CSC1111

7

7

## Fibonacci Sequence: Array-like Implementation

- Create a list of the 1st 20 Fibonacci numbers
  - $F_0 = F_1 = 1$ ;  $F_n = F_{n-1} + F_{n-2}$

- Create whole list
- Update values

```
fibs = [0]*20 # creates a list of size 20,  
             # containing all 0s  
fibs[0] = 0  
fibs[1] = 1
```

March 10, 2023

Sprenkle - CSC1111

[fibs2.py](#)

8

8

## Fibonacci Sequence: Array-like implementation

- Create a list of the 1st 20 Fibonacci numbers

➤  $F_0 = F_1 = 1$ ;  $F_n = F_{n-1} + F_{n-2}$

- Create whole list
- Update values

```
fibs = [0]*20      # creates a list of size 20,  
                  # containing all 0s  
fibs[0] = 0  
fibs[1] = 1  
  
for x in range(2, len(fibs)):  
    newfib = fibs[x-1] + fibs[x-2]  
    fibs[x] = newfib  
  
for num in fibs:  # print each num in list on sep lines  
    print(num)
```

March 10, 2023

Sprenkle - CSC1111

fibs2.py

9

9

## Understanding Lists

- What does the following code display?

```
x = [1, 2, 3]  
y = x  
y[0] = -1  
print(y)  
print(x)
```

- Run in Python interpreter

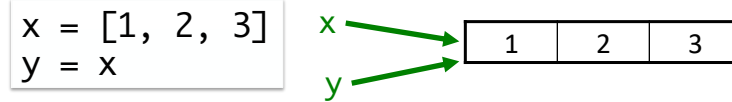
March 10, 2023

Sprenkle - CSC1111

10

10

## List Identifiers are Aliases



- `y` is **not** a copy of `x`
- `y` is another alias to that list/object
  - `y` points to what `x` points to

March 10, 2023

Sprenkle - CSC1111

11

11

## Understanding Lists

- What does the following code display?

```
x = [1, 2, 3]
y = x
y[0] = -1
print(y)
print(x)
```

- View in Python visualizer

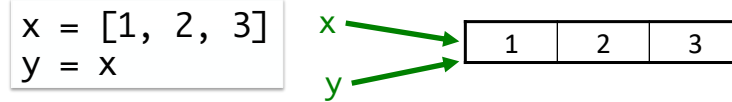
March 10, 2023

Sprenkle - CSC1111

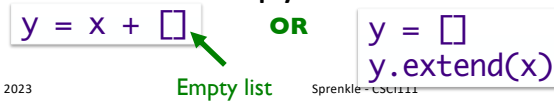
12

12

## List Identifiers are Aliases



- **y** is **not** a copy of **x**
- **y** is another alias to that list/object
  - **y** points to what **x** points to
- How to make a copy of **x**?



March 10, 2023

Sprenkle - CSCI111

13

13

Immutable vs Mutable Parameters

## PASSING PARAMETERS

March 10, 2023

Sprenkle - CSCI111

14

14

## Passing Parameters

- Only *copies* of the actual parameters are given to the function
  - For **immutable** data types
- The **actual** parameters in the calling code do not change
- **Swap example:**
  - Swap two values in script
  - Then, put into a function

Which are?



March 10, 2023

Sprenkle - CSC1111

15

15

## Recall: Immutable Data is Passed by Value

```
def main():
    x = 5
    y = 7

    swap(x, y)

    print("x =", x)
    print("y =", y)

def swap(a, b):
    tmp = a
    a = b
    b = tmp
    print(a, b)

main()
```

This code does not have the desired effect in that `x` and `y` are not swapped.

Since integers are passed **by value**, the values of `x` and `y` are not changed by the call to the `swap` function.

March 10, 2023

Sprenkle - CSC1111

swap.py

16

16



## Lists as Parameters to Functions

- Lists are not passed-by-value/copied
- Different from immutable types (e.g., numbers, strings)
- Function parameter is actually an *alias* to the list in memory

Impact: If a list that is passed as a parameter into a function is **modified in the function**, the list is **modified outside the function**

March 10, 2023

Sprenkle - CSC1111

17

17

### Problem:

Sort a list of 3 numbers, in descending order

```
# order list such that list3[0] >= list3[1] >= list3[2]
def descendSort3Nums( list3 ):
```

Called as:

```
myList = ...
descendSort3Nums(myList)
print(myList)
```

How can you implement with list methods?  
Can we do this using only 3 comparisons?

March 10, 2023

Sprenkle - CSC1111

[descendSort.py](#)

18

18

## Descend Sort a List w/ 3 elements

```
def descendSort3Nums(list3):  
    if list3[1] > list3[0]:  
        # swap 'em  
        tmp = list3[0]  
        list3[0] = list3[1]  
        list3[1] = tmp  
  
    if list3[2] > list3[1]:  
        tmp = list3[1]  
        list3[1] = list3[2]  
        list3[2] = tmp  
  
    if list3[1] > list3[0]:  
        tmp = list3[0]  
        list3[0] = list3[1]  
        list3[1] = tmp
```

```
def main():  
    list = [1,2,3]  
    descendSort3Nums(list)  
    print(list)
```

Function does **not** return anything.  
Simply modifies the `list3` parameter.

March 10, 2023

Sprenkle - CSCI111

19

19

## Your Next CS Class

- CSCI-112: Fundamentals of Programming II
  - 2 sections available
- Heart breaker (or deal maker?): I am not teaching CSCI-112 in the fall
  - I'm listed because I'm the department head

March 10, 2023

Sprenkle - CSCI111

20

20

# Broader Issues

The Potentially Inappropriate  
Memebrary for Historians and Literaries  
Join  
Robert Kirschmann · Mar 8, 2022 ·

Teaching STEM without teaching the Humanities is how you get Spider-Man villains.

Teaching Theater without teaching the Humanities is how you get Batman villains.

<p>Alicia Elias Justin Sambridhi Tyler</p>	<p>Amanda Claire Elizabeth Winter</p>	<p>Brian Ethan Kyle Tim</p>	<p>David Harrison Michelle Sam</p>	<p>Charlie Jackson Matt Micah</p>
--	---	---	--	---

March 10, 2023 Sprenkle - CSCI111 21

21

# Broader Issue

- What is public-key encryption?
- What is cryptocurrency?
- In the movie *Tombstone*, Ike Clanton says, “Listen, Mr. Kansas Law Dog. Law don't go around here. Savvy?”
  - Compare the privacy concerns in the articles and the role of the government/the law
  - How has the role of government/law changed over time? Where do we go from here?

March 10, 2023 Sprenkle - CSCI111 22

22

## Update

- “A US judge sentenced BitMEX co-founder Samuel Reed to 18 months of probation, saying he had a lesser role than two other co-founders in the offenses at the cryptocurrency derivative exchange.”

March 10, 2023

Sprenkle - CSCI111

23

23

## Looking Ahead

- Pre Lab 8 due before lab on Tuesday
  - Much of List chapter, but skipping some sections

March 10, 2023

Sprenkle - CSCI111

24

24