

## COMP 141

Lists III



1

## Announcements

- Reminder – Program 7 due Sunday
- Solutions to Practice from last time in Box folder
  - list2practiceCompleted.py

2

## Copying Lists

- **To make a copy of a list you must copy each element of the list**
  - Two methods to do this:
    - Creating a new empty list and using a `for` loop to add a copy of each element from the original list to the new list
    - Creating a new empty list and concatenating the old list to the new empty list

3

## Copying Lists (cont'd.)

```
list1 = [1,2,3,4]
list2 = list1
```

Figure 8-4 list1 and list2 reference the same list



```
list1 = [1,2,3,4]
list2 = list1
print("List 1:", list1)
print("List 2:", list2)
list1[0] = 99
print("List 1:", list1)
print("List 2:", list2)
```

### Output

```
List 1: [1, 2, 3, 4]
List 2: [1, 2, 3, 4]
List 1: [99, 2, 3, 4]
List 2: [99, 2, 3, 4]
```

4

3

4

## Code to Copy List

```
list1 = [1,2,3,4]
list2 = []
for item in list1:
    list2.append(item)
print("List 1:", list1)
print("List 2:", list2)
list1[0] = 99
print("List 1:", list1)
print("List 2:", list2)
```

### Output

```
List 1: [1, 2, 3, 4]
List 2: [1, 2, 3, 4]
List 1: [99, 2, 3, 4]
List 2: [1, 2, 3, 4]
```

5

## Saving Lists to a File

- To save the contents of a list to a file:
  - Use a `for` loop to write each element and `'\n'`
  - DON'T just typecast list to a string and write the string.

6

5

6

## Writing a List to a File

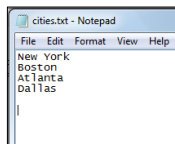
```
# This program saves a list of strings to a file.
def main():
    # Create a list of strings.
    cities = ['New York', 'Boston', 'Atlanta', 'Dallas']

    # Open a file for writing.
    outfile = open('cities.txt', 'w')

    # Write the list to the file.
    for item in cities:
        outfile.write(item + '\n')

    # Close the file.
    outfile.close()

# Call the main function.
main()
```



7

## Reading Files into Lists

- Don't read a file into a list more than once per program.
  - This is the true power of lists – you only need to create them once and then you can use them over and over.

8

7

8

```
# This program reads a file's contents into a list.

def main():
    # Open a file for reading.
    infile = open('cities.txt', 'r')

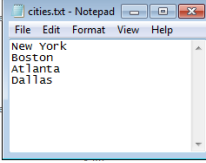
    cities = []

    # Read the contents of the file into a list.
    for line in infile:
        line = line.rstrip()
        cities.append(line)

    # Close the file.
    infile.close()

    # Print the contents of the list.
    print(cities)

# Call the main function.
main()
```



```
Output
['New York', 'Boston', 'Atlanta', 'Dallas']
```

9

## Comparing Lists

```
>>> list1 = ['green', 'red', 'blue']
>>> list2 = ['red', 'blue', 'green']
>>> list1 == list2
False
>>> list1 != list2
True
>>> list1 < list2
True
>>> list1 <= list2
True
>>> list1 > list2
False
>>> list1 >= list2
False
```

- To compare two lists, they must contain the same type of elements.
- The comparison uses alphabetical ordering.
  - Compares first element of each list and if they differ, this determines the outcome of the comparison.

10

## Parallel Lists

Parallel lists are two or more lists of the same length, where there is a relationship between `list1[p]` and `list2[p]` (for any index/position `p`).

	0	1	2
list1 =	"hello"	"my"	"friend"
list2 =	"ahoy"	"me"	"matey"

11

## Using Parallel Lists

```
students = []
grades = []
keep_going = 'Y'

while(keep_going == 'Y'):
    student = input("Enter student's name: ")
    grade = input("Enter the student's grade: ")
    students.append(student)
    grades.append(grade)
    keep_going = input("Would you like to enter another student/grade? (Y or N) ")

print(students)
print(grades)
```

```
Enter student's name: Mary
Enter the student's grade: 99
Would you like to enter another student/grade? (Y or N) Y
Enter student's name: Dave
Enter the student's grade: 78
Would you like to enter another student/grade? (Y or N) Y
Enter student's name: Kate
Enter the student's grade: 86
Would you like to enter another student/grade? (Y or N) Y
Enter student's name: Todd
Enter the student's grade: 92
Would you like to enter another student/grade? (Y or N) N
['Mary', 'Dave', 'Kate', 'Todd']
['99', '78', '86', '92']
```

12

## Using Parallel Lists with Files

```
def main():
    infile = open("songs.txt", 'r')

    titles = []
    artists = []
    prev = []
    num_weeks = []

    for line in infile:
        line = line.rstrip()
        title, artist, prev_pos, weeks = line.split(':')
        titles.append(title)
        artists.append(artist)
        prev.append(int(prev_pos))
        num_weeks.append(int(weeks))

    longest = max(num_weeks)
    indLongest = num_weeks.index(longest)

    print("Artist: ", artists[indLongest], "\nTitle: ", titles[indLongest])

    print(artists)
    prev = artists[0]
    for i in range(1, len(artists)):
        current = artists[i]
        if prev == current:
            print(current)
        prev = current

main()
```

13

## Talk Like a Pirate Lab

Spend at least 30 minutes working on this lab.

It will help you practice the parallel lists idea, which will be key for Program 8.

14