Announcements

Guest Lecture
Bill Berghel
M.S., Georgia Tech
B.S., Washington & Lee
Retired data scientist with FedEx

Slides source:
Catie Welsh’s Fall 2016-2017 Lecture 25 slides for COMP141

Other Announcements

Introduction to Lists

**List**: an object that contains multiple data items
- **Element**: An item in a list
- **Format**: `list = [item1, item2, etc.]`
- Can hold items of different types

*print* function can be used to display an entire list

*list()* function can convert certain types of objects to lists

A list of integers:

```
 even_numbers = [2, 4, 6, 8, 10]
```

A list of strings:

```
 Name=['Emma', 'Sophia', 'Isabella', 'Emily']
```

A list holding different types:

```
 fastfood = ['egg mcmuffin', 290, 2.79]
```
Example Using Lists

```
def main():
    # Create a list with some items.
    food = ['burger', 'fries', 'drink']
    # Display the list.
    print("Here are the items in the food list.")
    print(food)
    # Call the main function
    main()
```

```
Run
```

Why Use Lists?

Lists exist so that programmers can store multiple related variables together.

Useful when we don’t know ahead of time how many items we are going to store.

- Lists solve this problem because a single list can hold from zero to practically any number of items in it.

Basic List Operations

Lists are created using square brackets around items (elements) separated by commas.

- `mylist = [1, 2, 3]`
- `numbers = [-9.1, 4.77, 3.14]`
- `fedexsp = ['people', 'service', 'profit']`

Lists are accessed using indices/positions just like strings.

Most — but not all — string functions also exist for lists.
One Important Difference

Strings are immutable.
- You can't change a string without making a copy of it.
  \[s = 'abc'\]
  \[s[0] = 'A' \# definitely not legal!\]
  \[s = 'A' + s[1:] \# legal\]

Lists are mutable.
- You can change lists in-place without explicit copying.
  \[L = [2, 4, 6, 8, 10]\]
  \[L[0] = 15 \# legal\]
  \[L.append(26) \# legal\]

Compare Immutable and Mutable

How can we switch the first and last letter in a string?

How can we switch the first and last items in a list?
Compare Immutable and Mutable

How can we switch the first and last letter in a string?

```python
ltrs = 'Annoy'
print('original string is:', ltrs)
ltrs = ltrs[-1] + ltrs[1:] + ltrs[0]
print('new string is:', ltrs)
```

How can we switch the first and last items in a list?

```python
frobzoo = ['profit', 'service', 'people']
print('wrong list is:', frobzoo)
temp = frobzoo[0]
frobzoo[0] = frobzoo[len(frobzoo)-1]
frobzoo[len(frobzoo)-1] = temp
print('corrected list is:', frobzoo)
```  

**We do not need to look at the rest of the list.**

Three Common Ways to Make a List

Make a list that already has the elements in it:

```python
lst = [4, 7, 3, 8]
```

Make a list of a certain length and prepopulate the same element in all positions:

```python
lst = [8] * 4 # makes the list [8,8,8,8]
```

- Use when you need a list of a certain length ahead of time.
- Note the repetition operator, similarly to strings.

Make an empty list:

```python
lst = []
```

- Common when you're going to put things in the list coming from the user or a file.

Examples of Concatenation

```python
a = [1,2,3]
b = [4,5,6]
c = a + b
print(c) # prints [1, 2, 3, 4, 5, 6]
```

```python
mylist = ['a','b','c']
other = ['d','e','f']
print(mylist + other) # ['a', 'b', 'c', 'd', 'e', 'f']
```
Simple List Problems

How would we write a function to convert a number from 1-12 into the corresponding month of the year as a string?

```python
def getmonth(month):
    # Code to convert month number to string
```

Simple List Problems

What does this code do?

```python
lst1 = [2] * 3
lst2 = [4] * 2
lst3 = lst1 + lst2
for x in range(0, len(lst3), 2):
    lst3[x] = -1
```

Simple List Problems

What does this code do?

```python
lst1 = [2] * 3
lst2 = [4] * 2
lst3 = lst1 + lst2
for x in range(0, len(lst3), 2):
    lst3[x] = -1
print('lst1 is', lst1)
print('lst2 is', lst2)
print('lst3 is', lst3)
```
Examples of List Slices

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

numbers[2:] # [3, 4, 5, 6, 7, 8, 9, 10]
numbers[1:-2] # [1, 2, 3, 4, 5, 6, 7, 8]
numbers[1:8:2] # [2, 4, 6, 8]
numbers[5:1] # [6, 5, 4, 3, 2, 1]
numbers[::1] # [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Problem – Total of the Values in a List

```python
# Program total_list.py
# This program calculates the total of the values in a list.
def main():
    numbers = [2, 4, 6, 8, 10]
    total = 0
    for value in numbers:
        total += value
    print('The total of the elements is', total)

main()
```

The total of the elements is 30

Problem – Total of Sales Data

```python
def main():
    # The NUM_DAYS constant holds the number of days
    # for which we will gather sales data.
    NUM_DAYS = 5

    def main():
        # create a list to hold the sales for each day.
        sales = [0] * NUM_DAYS
        # create a variable to hold an index.
        index = 0
        print('Enter the sales for each day.

# Get the sales for each day.
while index < NUM_DAYS:
    print('Day #', index + 1, '(', end='')
    sales[index] = int(input())
    index += 1
# Display the values entered.
print('The sales you entered:

for value in sales:
    print(value)

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