COMP 141	
Lists II	
Rhodes_College	1



### 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?

def getmonth(month):

### **Examples of Concatenation**

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

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

### Rhodes College

### Simple list problems

• What does this code do? lst = [2] \* 3 lst2 = [4] \* 2 lst3 = lst + lst2 for x in range(0, len(lst3), 2): lst3[x] = -1

### **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[:-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]
Rhodes\_College

Pro	gram 8-8 (total_list.py)	
1	# This program calculates the total of the values	
2	# in a list.	
3		
4	def main():	
5	# Create a list.	
6	numbers = [2, 4, 6, 8, 10]	
7		
8	# Create a variable to use as an accumulator.	
9	total = 0	
10		
11	# Calculate the total of the list elements.	
12	for value in numbers: ← Can iterate by item in the list	
13	total += value	
14		
15	# Display the total of the list elements.	
16	print('The total of the elements is', total)	
17		
18	# Call the main function.	
19	main()	
Pro	gram Output	
The	total of the elements is 30	
		7



### Practice

Get the file Nov1.py from my Box.com code directory. It has the main function written for you and stubs for 2 other functions that you will need to write.

findAverage(numbers) – will return the average of all the numbers in the list

countNumbers(numbers, average) - will return 2 values; it counts the number of above average and below average numbers in a list



## Finding Items in Lists with the in Operator

- You can use the in operator to determine whether an item is contained in a list
  - General format: item in list
  - Returns True if the item is in the list, or False if it is not in the list
- Similarly you can use the **not in** operator to determine whether an item is not in a list





# List Methods and Useful Built-in Functions append (item): used to add items to a list - item is appended to the end of the existing list index (item): used to determine where an item is located in a list Returns the index of the first element in the list containing item Raises ValueError exception if item not in the list



12

### find() doesn't exist for lists

- list\_var.index(item)
- · Searches left to right, returns position where found, but crashes if not found.
- Let's build an algorithm that replicates find(), but works for lists (returns -1 if not found).

# **Example Using Append**

#### def main():

	<pre>infile = open("randomNums.txt", 'r') numbers = [] for line in infile:     numbers.append(int(line)) print(numbers) main()</pre>	
	Output	
	[62, 57, 35, 27, 45, 44, 46, 68, 86, 27, 88, 33, 11, 61, 64, 45,	
	56, 9, 33, 32, 56, 63, 24, 26, 100, 95, 62, 10, 87, 58, 69, 54, 75,	
	41, 22, 93, 82, 16, 92, 49, 6, 71, 85, 59, 56, 22, 3, 50, 1, 20, 54,	
	18, 27, 78, 17, 7, 41, 83, 92, 38, 5, 64, 60, 92, 15, 26, 57, 39,	
	80, 41, 67, 56, 24, 77, 28, 90, 24, 72, 2, 46, 75, 53, 58, 47, 50,	
<b>8</b>	18, 40, 65, 24, 58, 4, 58, 81, 40, 6, 77, 85, 86, 68, 63]	
Rhodes College		14

### List Methods and Useful Built-in Functions (cont'd.)

- insert(index, item): used to insert item at position index in the list
- sort(): used to sort the elements of the list in ascending order
- remove (item): removes the first occurrence of item in the list
- **reverse ()**: reverses the order of the elements in the list

15

X

Rhodes College

```
Program 8-5 (insert_list.py)
     # This program demonstrates the insert method.
    def main():
         # Create a list with some names.
         names = ['James', 'Kathryn', 'Bill']
         # Display the list.
print('The list before the insert:')
         print(names)
         # Insert a new name at element 0.
names.insert(0, 'Joe')
         # Display the list again.
         print('The list after the insert:')
print(names)
18 # Call the main function.
    main()
Program Output
The list before the insert:
['James', 'Kathryn', 'Bill']
The list after the insert:
```

['Joe', 'James', 'Kathryn', 'Bill']

16

18

20



### List Methods and Useful Built-in Functions (cont'd.)

- <u>del statement</u>: removes an element from a specific index in a list
  - General format: del list[i]
- <u>min and max functions</u>: built-in functions that returns the item that has the lowest or highest value in a sequence
  - The sequence is passed as an argument
- <u>sum function</u>: built-in functions that returns the total of all the values in a sequence
  - The sequence is passed as an argument

alpha\_list = ['a', 'b', 'c', 'd']
print("The lowest value is", min(alpha\_list))
print("The highest value is", max(alpha\_list))
# You cannot take the sum of a list that has strings in it
Output
Before Deletion: [5, 4, 3, 2, 50, 40, 30]
After Deletion: [5, 4, 2, 50, 40, 30]
The lowest value is 20
The highest value is 50
The sum of values in my list is 131

The lowest value is a The highest value is d

X

Rhodes College



Rhodes College

19

Rhodes College