

Python Lists Continued

Textbook

Python Lists Continued

There are many different things we can do with lists in Python! Check out some of the awesome things we can do.



Range Review

The [range](#) of a list includes all the index spots between two numbers. The first number is separated from the second number with a colon.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies"]  
2  
3 print(smells[2:5])
```

Try it!

This will print out `["rain", "ocean", "garbage"]`. **Notice that it does not include the index number 5.** The range goes up to the second number after the colon, but does not include it.

Step

You can also print every other value. This is called a step and you can see how this works in the example below.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies"]
2
3 print(smells[0:6:2])
```

Try it!

Steps can also be values other than just 2, as seen in the example below.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies", "roses",
2 "perfume", "grass", "grapes"]
3 print(smells[0:6:5])
```

Try it!

Replacement

You can even replace an item in a list with something new. The index value will determine which item gets replaced.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies"]
2
3 smells[3] = "perfume"
4
5 print(smells)
```

Try it!

This will print out `["skunk", "lilac", "rain", "perfume", "garbage", "cleaner", "cookies"]` .

Length

You can also find out how many items are in the list. This is done by using `len(listName)` .

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies"]
2
3 print(len(smells))
```

Try it!

This will print out 7.

Notice that with [indexing](#), you start counting at 0, so the last value in the list has an index value of 6. But when counting how many items are in a list, you would start counting at 1, so there are 7 items in the list.

Sort

Sorting lists is very simple with the `sort()` method. Here is an example of sorting alphabetically.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies"]
2 smells.sort()
3 print(smells)
```

Try it!

Here is an example of sorting numerically.

Sort in Reverse

To sort the list in reverse order, just add the code `reverse=True` into the code.

```
1 cookies = [3, 55, 9, 12, 13]
2 cookies.sort(reverse=True)
3 print(cookies)
```

Try it!

Copy a List

We can also copy a list and put it into a new variable.

```
1 cookies = [3, 55, 9, 12, 13]
2
3 copied_list = cookies.copy()
4
5 print(copied_list)
```

Try it!

Slicing a List

Something to note is when you slice a list, you are creating a new list to use.

```
1 smells = ["skunk", "lilac", "rain", "ocean", "garbage", "cleaner", "cookies", "roses",
2 "perfume", "grass", "grapes"]
3 new_list = smells[:]
4
5 new_list.append("cinnamon")
6
```

```
7 print(smells)
8
```

Try it!

This is why the above code will not print the string cinnamon. The new_list is separate from the original list.

Maximum or Minimum Item in a List

You can use max() to find the largest number in a list.

```
1 mylist = [4, 56, 32, 100, 457, 8, 12]
2
3 print(max(mylist))
```

Try it!

This will print out 457 .

min() will also work to find the smallest number.

```
1 mylist = [4, 56, 32, 100, 457, 8, 12]
2
3 print(min(mylist))
```

Try it!

If the list has strings, it will print out the item with the most or least characters.

```
1 mylist = ["bob", "jack", "aimee", "savannah", "fred", "sam"]
2
3 print(max(mylist))
```

Try it!

This will print out savannah .

Challenge Help

Creating A List of Integers with Input

These challenges will require putting the input into a list of integers. To do this, use this code:

```
my_list = [int(n) for n in input().split()]
```

This code creates a list called "my_list" and the input().split() command breaks up the input into each individual integer. It uses a for loop to assign all the individual inputs to their index in the list.

For this code, put your input question right into the input() above. For example:

```
my_list = [int(n) for n in input("Input a list of numbers").split()]
```

The input should look like numbers separated by spaces. No commas, parenthesis, or brackets are needed. For example: `2 6 8 33 24 2` would be an acceptable input.

This would create a python list: `[2, 6, 8, 33, 24, 2]`

Checkpoint

Python Lists Continued

1. Create a list that has **7 strings** in it.
2. From your list, select the **first 3 values** using a range statement and print them.
3. From your list, **replace the last item** in the list with a different value and print the new list, with the replaced final item.
4. Print the length of the list.

Requirements:

- Create a list with 7 strings in it.
- Use a slice statement to print the first 3 values.
- From your list, replace the last item in the list with a different value and print the new list, with the replaced final item.
- Print the length of the list

Questions (14)

1. When counting how many items are in a list (not indexing), what number do you start counting at?

MULTIPLE CHOICE

Choose the correct answer:

- A. 2
- B. 3
- C. 1
- D. 0

2. Given the list `veggies = ["carrots", "corn", "lettuce", "broccoli", "beans", "spinach"]`, how would you use range to print "corn", "lettuce", "broccoli" ?

MULTIPLE CHOICE

Choose the correct answer:

- A. `veggies[2:4]`
- B. `veggies[1:4]`
- C. `veggies[2:4]`
- D. `veggies[1:3]`
- E. `veggies[0:3]`

3. With indexing a list, what number do you start at?

Choose the correct answer:

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

4. What method do you use to sort a list in Python?

Choose the correct answer:

- A. `sort()`
- B. `len()`
- C. `step()`
- D. `range()`

5. What method do you use to find the length of a list in Python?

Choose the correct answer:

- A. `len()`
- B. `num()`
- C. `sort()`
- D. `step()`
- E. `total()`

6. What will the following code print out?

```
flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"] print(flowers[0:6:3])
```

Choose the correct answer:

- A. `['rose', 'dandelion']`
- B. `['rose', 'daffodil', 'sunflower', 'lotus']`
- C. `"daffodil"`
- D. `['dandelion', 'sunflower', 'lily', 'lotus']`

7. What will the following code print out?

```
flowers = ["rose", "tulip", "lily", "lotus"] flowers[0] = "violet" print(flowers)
```

Choose the correct answer:

- A. ['violet', 'tulip', 'lily', 'lotus']
- B. ["rose", "tulip", "lily", "lotus"]
- C. ["rose", "violet", "lily", "lotus"]
- D. flowers

8. What will the following code print out?

```
flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"] print(len(flowers))
```

Choose the correct answer:

- A. 6
- B. 7
- C. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
- D. "lotus"

9. What will the following code print out?

```
flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"] flowers.sort() print(flowers)
```

Choose the correct answer:

- A. ['daffodil', 'dandelion', 'lily', 'lotus', 'rose', 'sunflower', 'tulip']
- B. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
- C. 7
- D. ["tulip", "sunflower", "rose", "lotus", "lily", "dandelion", "daffodil"]

10. What will the following code print out?

```
flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"] flowers.sort(reverse=True) print(flowers)
```

Choose the correct answer:

- A. ['tulip', 'sunflower', 'rose', 'lotus', 'lily', 'dandelion', 'daffodil']
- B. ['daffodil', 'dandelion', 'lily', 'lotus', 'rose', 'sunflower', 'tulip']
- C. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
- D. 7

11. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(flowers[0:6:3])
```

12. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 flowers = ["rose", "tulip", "lily", "lotus"]
2
3 flowers[0] = "violet"
4
5 print(flowers)
```

13. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(len(flowers))
```

14. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 flowers.sort(reverse=True)
4 print(flowers)
```


Challenges (5)

1. Find the Largest Value

1. Write a program that gets a list as the input.

Use the Challenge Help section of the lesson if you need help with this.

2. Find the **largest number in your list**.
3. Print the largest number.

For example:

Input: 2 8 9 3 5 2

Output: 9

Another example:

Input: 10 5 8 1

Output: 10

Reminder: this is how to create a list of integers from an input:

```
my_list = [int(n) for n in input().split()]
```

2. Greater than left index

Write a program that takes in a list as input.

Use the Challenge Help section of the lesson if you need help with this.

Print all elements that are **greater than the value to their left**.

Reminder: this is how to create a list of integers from an input:

```
my_list = [int(n) for n in input().split()]
```

For example:

Input: 6 4 3 8 6 5 10

Output: 8 10

Another example:

Input: 1 3 2 1 2

Output: 3 2

Hint: Create a variable named `current` and assign it to the first item in the list. Then use a `for loop` to loop through the inputted list. Then (still inside the loop) update the variable named `current` to be the next item in the list.

3. Unique Numbers

Write a program that takes in a list as input and prints the number of unique elements on the list.

Use the Challenge Help section of the lesson if you need help with this.

The numbers will always be sorted in order from least to greatest.

For example

Input: 2 2 5 7

Output: 3

Another example:

Input: 1 2 3 3 7 7 2

Output: 4

Reminder: this is how to create a list of integers from an input:

```
my_list = [int(n) for n in input().split()]
```

Hint: Try using `not in`

4. Swap Neighbors

Write a program that takes in a list as input and swaps each adjacent pair. Print out a list with the numbers swapped.

If there are an odd number of elements, leave the final element at the end of your list.

For example:

Input: 1 2 3 4 5 6 7 8 9

Output: [2, 1, 4, 3, 6, 5, 8, 7, 9]

Another example:

Input: 0 3 1 6 9 3 10

Output: [3, 0, 6, 1, 3, 9, 10]

Reminder: this is how to create a list of integers from an input: `my_list = [int(n) for n in input().split()]`

Hint: Try out this code syntax to swap variables

```
a = 43
```

```
b = 56
```

```
a, b = b, a
```

Hint: You can put all kinds of things inside the parentheses of `range()`

5. Greater than Neighbors

Write a program that takes in a list of numbers as input.

Use the Challenge Help section of the lesson if you need help with this.

Print the number of elements on the list that are **larger than both of their neighbors**.

Reminder: this is how to create a list of integers from an input:

```
my_list = [int(n) for n in input().split()]
```

For example:

Input: 2 4 6 4 3 100 3

Output: 2

This is the output because only two elements are larger than both their neighbors.

Another example:

Input: 3 4 3 4 3 5

Output: 2

This is the output because only two elements are larger than both their neighbors.

Answer Keys & Solutions

Checkpoint Solutions

Python Lists Continued

```
1 activities = ["movies", "skating", "bowling", "laser tag", "escape room", "trampoline  
park", "library"]  
2  
3 print(activities[0:3])  
4  
5 activities[6] = "water park"  
6  
7 print(activities)  
8  
9 print(len(activities))
```

Questions

1. When counting how many items are in a list (not indexing), what number do you start counting at?

MULTIPLE CHOICE

Correct Answer:

- A. 2 ✗ Incorrect
- B. 3 ✗ Incorrect
- C. 1 ✓ Correct
- D. 0 ✗ Incorrect

Explanation:

Counting how many items in a list starts counting at 1.

2. Given the list `veggies = ["carrots", "corn", "lettuce", "broccoli", "beans", "spinach"]`, how would you use `range` to print "corn", "lettuce", "broccoli" ?

MULTIPLE CHOICE

Correct Answer:

- A. `veggies[2:4]` ✗ Incorrect

B. `veggies[1:4]`

✓ Correct

C. `veggies[2:4]`

✗ Incorrect

D. `veggies[1:3]`

✗ Incorrect

E. `veggies[0:3]`

✗ Incorrect

Explanation:

Remember indexing starts counting at 0. The first number is the start of the range and the second number is the end of the range

3. With indexing a list, what number do you start at?

MULTIPLE CHOICE

Correct Answer:

A. 0

✓ Correct

B. 1

✗ Incorrect

C. 2

✗ Incorrect

D. 3

✗ Incorrect

E. 4

✗ Incorrect

Explanation:

You don't start counting at 1 when we are indexing.

4. What method do you use to sort a list in Python?

MULTIPLE CHOICE

Correct Answer:

A. `sort()`

✓ Correct

B. `len()`

✗ Incorrect

C. `step()`

✗ Incorrect

D. `range()`

✗ Incorrect

Explanation:

sort() will sort a list

5. What method do you use to find the length of a list in Python?

MULTIPLE CHOICE

Correct Answer:

- | | |
|-------------|-------------|
| A. len() | ✓ Correct |
| B. num() | ✗ Incorrect |
| C. sort() | ✗ Incorrect |
| D. step() | ✗ Incorrect |
| E. total() | ✗ Incorrect |

Explanation:

len() is short for length

6. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--|-------------|
| A. ['rose', 'dandelion'] | ✓ Correct |
| B. ['rose', 'daffodil', 'sunflower', 'lotus'] | ✗ Incorrect |
| C. "daffodil" | ✗ Incorrect |
| D. ['dandelion', 'sunflower', 'lily', 'lotus'] | ✗ Incorrect |

Explanation:

The third value shows how many items to skip.

7. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. ['violet', 'tulip', 'lily', 'lotus'] | ✓ Correct |
| B. ["rose", "tulip", "lily", "lotus"] | ✗ Incorrect |

C. ["rose", "violet", "lily", "lotus"]

✗ Incorrect

D. flowers

✗ Incorrect

Explanation:

The list named flowers will have a different value at index value of 0.

8. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. 6

✗ Incorrect

B. 7

✓ Correct

C. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]

✗ Incorrect

D. "lotus"

✗ Incorrect

Explanation:

len() is for a length

9. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. ['daffodil', 'dandelion', 'lily', 'lotus', 'rose', 'sunflower', 'tulip']

✓ Correct

B. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]

✗ Incorrect

C. 7

✗ Incorrect

D. ["tulip", "sunflower", "rose", "lotus", "lily", "dandelion", "daffodil"]

✗ Incorrect

Explanation:

sort() will sort items alphabetically

10. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. ['tulip', 'sunflower', 'rose', 'lotus', 'lily', 'dandelion', 'daffodil']

✓ Correct

B. ['daffodil', 'dandelion', 'lily', 'lotus', 'rose', 'sunflower', 'tulip']

✗ Incorrect

C. ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]

✗ Incorrect

D. 7

✗ Incorrect

Explanation:

`sort()` will sort items alphabetically

11. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(flowers[0:6:3])
```

Correct Solution:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(flowers[0:6:3])
```

Explanation:

Check for opening and closing quotation marks on every string.

12. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 flowers = ["rose", "tulip", "lily" "lotus"]
2
3 flowers[0] = "violet"
4
5 print(flowers)
```

Correct Solution:

```
1 flowers = ["rose", "tulip", "lily", "lotus"]
2
3 flowers[0] = "violet"
4
5 print(flowers)
```


Explanation:

Check for commas in all the right places.

13. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(len(flowers)
```

Correct Solution:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 print(len(flowers))
```

Explanation:

Check for all opening and closing parentheses.

14. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 flowers.sort(reverse=true)
4 print(flowers)
```

Correct Solution:

```
1 flowers = ["rose", "tulip", "daffodil", "dandelion", "sunflower", "lily", "lotus"]
2
3 flowers.sort(reverse=True)
4 print(flowers)
```

Explanation:

Booleans need to be capitalized

1. Find the Largest Value

Solution:

```
1 my_list = [int(n) for n in input().split()]
2 biggest = 0
3 for x in my_list:
4     if x > biggest:
5         biggest = x
6
7 print(biggest)
8
9 # ALTERNATE SOLUTION
10 # my_list = [int(n) for n in input().split()]
11 # my_list.sort(reverse=True)
12
13 # print(my_list[0])
```

2. Greater than left index

Solution:

```
1 my_list = [int(n) for n in input().split()]
2
3 current = my_list[0]
4
5 for x in my_list:
6     if x > current:
7         print(x)
8     current = x
```

3. Unique Numbers

Solution:

```
1 my_list = [int(s) for s in input().split()]
2
3 new_list = []
4
5 total = 0
6
7 for x in my_list:
8     if x not in new_list:
9         new_list.append(x)
10
11 print(len(new_list))
12
13 # ALTERNATE ANSWER SUGGESTION
14 # my_list = [int(n) for n in input().split()]
15 # my_list.sort()
16 # num = []
17 # total = 0
18
```

```
19 # for x in my_list:
20 #     if x not in num:
21 #         num.append(x)
22 #         total += 1
23
24 # print(total)
```

4. Swap Neighbors

Solution:

```
1 my_list = [int(n) for n in input().split()]
2
3 for i in range(0, len(my_list) - 1, 2):
4     my_list[i], my_list[i + 1] = my_list[i + 1], my_list[i]
5 print(my_list)
```

5. Greater than Neighbors

Solution:

```
1 my_list = [int(s) for s in input().split()]
2
3 counter = 0
4
5 for i in range(1, len(my_list) - 1):
6     if my_list[i - 1] < my_list[i] > my_list[i + 1]:
7         counter += 1
8 print(counter)
```