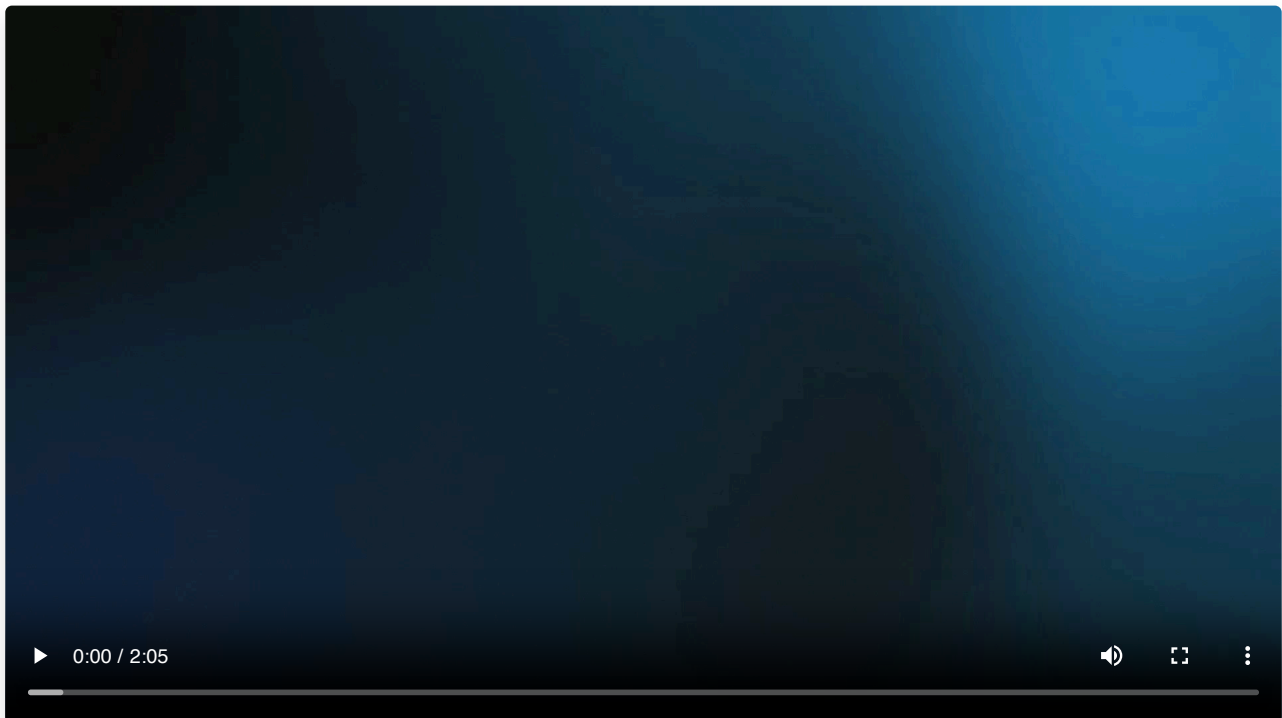


If Statement in a For Loop

Textbook

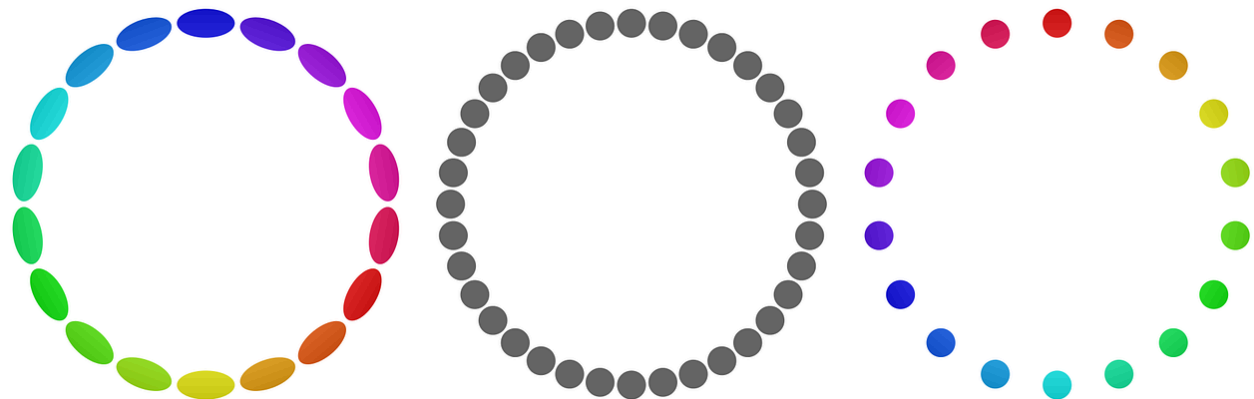
If Statement in a For Loop

The Autograder



If Statement in a For Loop

You can put any code you want inside a for loop. For loops get really fun when we add in an if statement. The loop moves through a list and checks to see if each item in the list fits a conditional. According to your code, each item in the list will do something different.



```
1 student_ages = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in student_ages:
4     if x >= 16:
5         print("This student is old enough to drive.")
6
7     else:
8         print("This student is not old enough to drive.")
```

Try it!

What's going on here? Well, the `for x in student_ages:` is the [for loop](#). The x means each item in the list will be analyzed.

Each item in the list will run through the code that is below the for loop. **EACH item in the list will be tested to see if it is greater than or equal to 16 (x >= 16).** If it is, the line that says they are old enough to drive will print.

ELSE, they are NOT greater than or equal to 16, and the line that says they are not old enough to drive will print.

Indentation

Notice the [indentation](#) in this example! Each line that has one step of indentation after the for loop is responding to the [for loop](#). In the example above, the `if x >= 16:` code runs for each item in `student_ages`.

Then, the lines that are indented one step of indentation after the if statement are responding to the if statement. In the example above, the print statement `"This student is old enough to drive."` only runs `if x >= 16`.

Can you see how important indentation is?

Range

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.

Updating Variables

Often for loops are useful because they can update variables with each loop. Here is an example of a piece of code that will update the variable named "total" with each loop.

```
total = total + x
```

```
1 my_list = [2, 5, 8, 10]
2
3 total = 0
4
5 for x in my_list:
6     total = total + x
7
8 print(total)
9
10
11
```

Try it!

This code will print out `25`.

Notice in this code example that the variable named `total` is originally equal to 0. Each time the for loop runs, the variable named total gets updated to add another item in the list.

At the end, after the loop has run, the print statement prints out the new value for the variable named "total."

Break

The `break` statement will immediately cause the loop to stop running. The `break` statement can be used inside the loop's body only.

```
1 student_ages = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in student_ages:
4     if x >= 16:
5         print("This student is old enough to drive.")
6         break
7     else:
8         print("This student is not old enough to drive.")
```

Try it!

If the loop also has an `else` branch, it will not run. For example, the print statement `This student is not old enough to drive` won't print.

Continue

The `continue` statement causes the loop to immediately go to the next iteration of the loop.

```
1 student_ages = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in student_ages:
4     if x >= 16:
5         print("This student is old enough to drive.")
6         continue
7         print("finally")
8     else:
9         print("This student is not old enough to drive.")
```

Try it!

In this loop, if a value in the list is greater than 16, it will print a statement and then immediately go to the next item in the list. Any code written below it will not run. Notice how the word `finally` never prints.

Challenge Help

To create a list of strings, use the following code:

```
my_list = input().split()
```

Remember that inputs are automatically accepted as strings.

The input should look like words separated by spaces. No commas, parenthesis, or brackets are needed. For example: `apple banana strawberry` would be an acceptable input.

This would create a python list: `["apple", "banana", "strawberry"]`

These challenges may require putting a bunch of numbers as input into a list. 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

If Statement in a For Loop

1. Create a list of the grossest foods ever and assign it to a variable named foods.
2. Loop through the list named foods using a for loop.
3. For each item in the list, use an if statement to check if the item is equal to the string " `mushrooms` ". If it is, print a statement like ____ + " `are gross` ".
4. Use an elif statement to check and see if each item is equal to the string " `broccoli` ". If it is, print a statement like ____ + " `is gross` ".
5. Use an elif statement to check and see if each item is equal to the string " `fish` ". If it is, print a statement like, ____ + " `is gross` ".
6. Create a different print statement to **concatenate** the food into a sentence for each conditional.
7. Create an else statement that prints a string if something in your list does not fit in the above categories.

Hint: Don't forget to use **double equal signs == for your conditionals**. If ____ == "broccoli." One equal sign = is used to declare a variable. Two equal signs == are used for comparisons.

Requirements:

- Create a list and assign it to a variable
- Create a For loop to loop through your list
- Use an If statement that prints a concatenated sentence
- Create else if statements that print a concatenated sentence
- Create an else statement at the end of your if statement.

Questions (8)

1. True or False: Indentation matters in Python

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

2. How many times will a FOR loop run in Python?

Choose the correct answer:

- A. Until the loop has addressed each item in the data set
- B. Until a condition is not met
- C. The default number of loops runs is 10
- D. Until the condition kicks to the ELSE statement

3. What will the following code print out?

```
colors = ["red", "blue", "purple", "green", "pink", "yellow", "orange"] print(colors[3:6])
```

Choose the correct answer:

- A. ['green', 'pink', 'yellow']
- B. ['red', 'blue', 'purple']
- C. ["red", "blue", "purple", "green", "pink", "yellow", "orange"]
- D. green, pink, yellow

4. What will the following code print out?

```
my_list = [3, 1, 10, 4] total = 0 for x in my_list: total = total + x print(total)
```

Choose the correct answer:

- A. 18
- B. 10
- C. [3, 1, 10, 4]
- D. x

5. What can you put inside a for loop?

Choose the correct answer:

- A. only print statements
- B. only if statements
- C. any code you want
- D. only math statements

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

DEBUG CODE

Code to Debug:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterfly:
4     if x >= 12:
5         print("You have many butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

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

DEBUG CODE

Code to Debug:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterflies:
4     if x >= 14:
5         print("You have a lot of butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

8. True or False: You cannot update variables in Python.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

Challenges (5)

1. Count the Zeros

How many zeros are in the list? Create a program that will count them for you.

1. Create an input that will generate a list.

Here's a reminder for how to do that:

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

2. Make sure the user inputs a bunch of numbers that has several instances of the number "0".
3. The number "50" has a zero in it but won't be counted.
4. Only the number "0" should be counted.
5. **Print the number of zeros** in the input.

For example:

Input: 2 0 9 0 0 80 21 0

Output: 4

Another example:

Input: 4 0 0 0 3

Output: 3

2. Find the Even Numbers

Write a program that takes a list of integers as the input.

Your program should print all of the even integers in the list, with each integer printed on a separate line.

For example:

Input: 1 2 3 4 5

Output:

2

4

Another example:

Input: 3 2 9 8 10

Output:

2

8

10

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

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


3. World Travel

Check to see if the letter "p" is in a list of countries.

1. Create an input that will generate a list of countries that you would like to visit someday.
2. Here's a hint for how to generate a list of strings from an input: `my_list = input().split()`
3. If the country has the letter "p" in it, print it out.
4. Print out countries that have the letter "p" in it.
5. Include countries with either lowercase or uppercase letter p.

For example:

Input: `Spain Australia Japan Brazil Peru`

Output: `Spain Japan Peru`

Another example:

Input: `Ecuador Madagascar Portugal Philippines`

Output: `Portugal Philippines`

4. Better than Average

Create a program that will print out the numbers that are greater than the average of a list. Create an input that generates a list of integers. Print out the numbers that are greater than the average of the list.

Reminder: the average is the sum of all the numbers in the list divided by how many items are in the list.

For example:

Input: `2 2 1 5 10`

Output: `5 10`

Another example:

Input: `10 20 14 18 12`

Output: `20 18`

Hint: Use one for loop to find the total. Use another loop to see if each item in the list is greater than the average or not.

5. Color Inspector

Consider the following list: `my_list = ["red", "orange", "yellow", "green", "blue", "purple", "black", "white", "gray", "pink", "indigo", "brown", "tan", "gold", "silver"]`

1. Create a program that accepts a letter as an input.
2. The program then prints out how many items in the list have that letter in it.
3. The print statement needs to be in this format: " `There are ____ items in the list that have the letter ____ in it.` "

For example:

Input: `r`

Output: `There are 7 items in the list that have the letter r in it.`

Another example:

Input: `e`

Output: `There are 8 items in the list that have the letter e in it.`

Answer Keys & Solutions

Checkpoint Solutions

If Statement in a For Loop

```
1 foods = ["fish", "mushrooms", "liver", "cabbage"]
2
3 for x in foods:
4     if x == "mushrooms":
5         print(x + " are slimy")
6     elif x == "broccoli":
7         print(x + " is stinky")
8     elif x == "fish":
9         print(x + " is fishy")
10    else:
11        print(x + " are also gross")
```

Questions

1. True or False: Indentation matters in Python

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Code will break if it is not indented correctly.

2. How many times will a FOR loop run in Python?

MULTIPLE CHOICE

Correct Answer:

A. Until the loop has addressed each item in the data set

✓ Correct

B. Until a condition is not met

✗ Incorrect

C. The default number of loops runs is 10

✗ Incorrect

D. Until the condition kicks to the ELSE statement

✗ Incorrect

Explanation:

A for loop runs for each item in a data set.

3. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. ['green', 'pink', 'yellow'] | ✓ Correct |
| B. ['red', 'blue', 'purple'] | ✗ Incorrect |
| C. ["red", "blue", "purple", "green", "pink", "yellow", "orange"] | ✗ Incorrect |
| D. green, pink, yellow | ✗ Incorrect |

Explanation:

The numbers indicate where the list will start and stop slicing.

4. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|------------------|-------------|
| A. 18 | ✓ Correct |
| B. 10 | ✗ Incorrect |
| C. [3, 1, 10, 4] | ✗ Incorrect |
| D. x | ✗ Incorrect |

Explanation:

The variable named total gets updated each time the loop runs.

5. What can you put inside a for loop?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--------------------------|-------------|
| A. only print statements | ✗ Incorrect |
| B. only if statements | ✗ Incorrect |

C. any code you want

✓ Correct

D. only math statements

✗ Incorrect

Explanation:

Loops can contain a variety of code.

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

DEBUG CODE

Incorrect Code:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterfly:
4     if x >= 12:
5         print("You have many butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

Correct Solution:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterflies:
4     if x >= 12:
5         print("You have many butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

Explanation:

The list name it is looping for doesn't exist.

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

DEBUG CODE

Incorrect Code:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterflies:
4     if x >= 14:
5         print("You have a lot of butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

Correct Solution:

```
1 butterflies = [14, 17, 12, 14, 15, 18, 19, 10]
2
3 for x in butterflies:
```

```
4     if x >= 14:
5         print("You have a lot of butterflies.")
6     else:
7         print("You don't have many butterflies.")
```

Explanation:

Check quotation marks.

8. True or False: You cannot update variables in Python.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Variables often get reassigned values

Challenges

1. Count the Zeros

Solution:

```
1 my_list = [int(n) for n in input("Input a list of numbers").split()]
2
3 total = 0
4
5 for x in my_list:
6     if x == 0:
7         total = total + 1
8
9 print(total)
```

2. Find the Even Numbers

Solution:

```
1 my_list = [int(n) for n in input().split()]
2
3
4 for x in my_list:
5     if x % 2 == 0:
```

```
6 print(x)
```

3. World Travel

Solution:

```
1 my_list = input().split()
2
3 for x in my_list:
4     if "p" in x or "P" in x:
5         print(x)
```

4. Better than Average

Solution:

```
1 my_list = [int(n) for n in input().split()]
2
3 total = 0
4
5 for x in my_list:
6     total = total + x
7
8 average = total/len(my_list)
9
10
11 for x in my_list:
12     if x > average:
13         print(x)
```

5. Color Inspector

Solution:

```
1 choice = input("Choose a letter")
2
3 my_list = ["red", "orange", "yellow", "green", "blue", "purple", "black", "white",
4 "gray", "pink", "indigo", "brown", "tan", "gold", "silver"]
5
6 total = 0
7
8 for x in my_list:
9     if choice in x:
10         total = total + 1
11
12 print("There are " + str(total) + " items in the list that have the letter " + choice
13 + " in it.")
```

