Chapter 5 - Lists

Lists

Lab - Lists

Lab 1

Odd or Even

Here is how you can iterate through a list, determine if the number is odd or even, and then put that number in a list of odd or even numbers. After the program has run, all of the odd numbers will be in the `odd` list and all of the even numbers will be in the `even` list.

Random Numbers

To make this project more interesting, we are going to use a list of randomly generated ints. First, import the `random` library into your program.

```
import random
```

Next, we are going to create a list called `numbers` and fill it with twenty random ints. Start my declaring the list `numbers` and make it an empty list (it has no elements). Then write a for loop that runs twenty times. Each time the loop runs, generate a random number and add it to the list.

```python
numbers = []
for i in range(20):
    numbers.append(random.randint(0, 100))
```

▼ **The Random Library**

To use the `random` library, you first import it with `import random`. To access any of the features of the library, start with `random.` and then add the method. Since we want a random integer, we will use `random.randint`. `randint` takes two numbers as parameters, the start and stop for the random number choice. We want a random integer between 0 and 100, so we use `random.randint(0, 100)`.

Odd and Even Lists

We are going to sort all of the random numbers into either the `even` list or the `odd` list. Create these lists and make them empty.

```python
odd = []
even = []
```

Iterating Over `numbers`

Now we are going to iterate over the `numbers` list and determine if the number is even or odd.

```python
for number in numbers:
    if number % 2 == 0:

    else:

```

Sorting the Elements

Once you have determined if a number is odd or even, use the `append` method to place it in the appropriate list.

```
for number in numbers:
    if number % 2 == 0:
      even.append(number)
    else:
      odd.append(number)
```

Printing the Lists

Let’s print the three lists to see if our program worked. Don’t forget to add some context to the lists. And to check our work, add together the lengths of `odd` and `even` to make sure they total 20.

```
print("The odd numbers: ", odd)
print("The even numbers: ", even)
print(len(odd) + len(even))
```

▼ **Code**

```
import random

numbers = []
for i in range(20):
    numbers.append(random.randint(0, 100))

odd = []
even = []

for number in numbers:
    if number % 2 == 0:
        even.append(number)
    else:
        odd.append(number)

print("The odd numbers: ", odd)
print("The even numbers: ", even)
print(len(odd) + len(even))
```

Lab 2

Coding Sum

Python already has a `sum` function, but we are going to write some code that manually calculates the sum of a list.

Setup

We are going to need a variable `total` that will be the sum. Set `total` to 0. Create an empty list called `numbers`. We will want the list to be randomly generated numbers, so import the `random` library as well.

```
import random

numbers = []
total = 0
```

Random Numbers

Next, lets add 20 random integers (from 0 to 100) to the list `numbers` with a for loop.

```
for i in range(20):
    numbers.append(random.randint(0, 100))
```

List Iteration

Now that the list of random numbers is complete, write a for loop to iterate over the list. Remember to use `number` as the iteration variable because it is the singular of the list `numbers`. Add each element of the list the the `total` variable.

```
for number in numbers:
    total += number
```

Answer and Checking our Work

Once this loop has finished iterating over the list `numbers`, the variable `total` should represent the sum of the list. Use a `print` statement to see the value of `total`. We are also going to print the sum of `numbers` using the `sum` function to check our work. If the numbers match, our code is good.

```python
print("The sum of numbers is ", total)
print(sum(numbers))
```

▼ **Code**

```python
import random

numbers = []
total = 0

for i in range(20):
    numbers.append(random.randint(0, 100))

for number in numbers:
    total += number

print("The sum of numbers is ", total)
print(sum(numbers))
```

Lab 3

Slicing a List

We are going to write a program that takes the middle third of a list and prints it. To keep things simple, we are going to work with list whose length is divisible by 3. The list will also be filled with random integers ranging from 0 to 100.

Setup

Start by importing the `random` library. Then declare `numbers` as an empty list.

```
import random

numbers = []
```

Random Numbers

Use a for loop that repeats 9 times to append a random integer to `numbers`.

```
for i in range(9):
    numbers.append(random.randint(0, 100))
```

More Variables

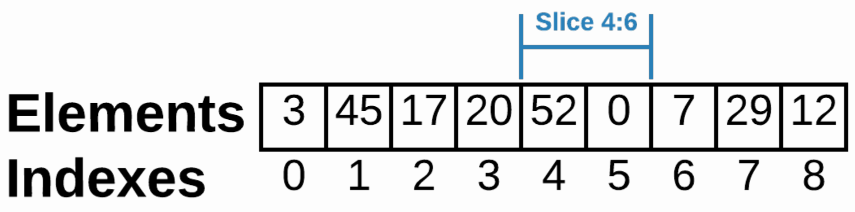
We are going to need some more variables. First, we need to know how long the list is. So create a variable `length` and set it to `len(numbers)`. The `slice` operator requires two numbers, one is the starting position of the slice and the other is the stopping position of the slice. Create two variables, `start` and `stop`, and set their values to 0 for now. Finally, we will also need a list variable called `middle`. Set `middle` to the slice of `numbers` from `start` to `stop`.

```
length = len(numbers)
start = 0
stop = 0
middle = numbers[start:stop]
```

Calculating the Start and Stop Values

If you want to find the middle of a list of 9 numbers, divide the length (9) by 3, which is 3. So `numbers` is composed of three groups of three elements. That would mean that the middle third is the elements 4, 5, and 6. Another way to state this is that `start` is `length // 3 + 1` and `stop` is `length // 3 * 2`. However, this will not give the correct slice.

▼ **Why use `//`?**
The `slice` operator requires integers for its parameters. If you use regular division (`/`), it will always return a floating point number. Floor division (`//`) will always return an integer.



Slice Example

Lists in Python start with the index 0. So calculating `start` with `length // 3 + 1` is off by 1. Change the declaration of `start` to be `length // 3`. The `slice` operator does not include the second parameter. So `stop` is not off by 1 like `start`. Change `stop` to be `length // 3 * 2`.

```
start = length // 3
stop = length // 3 * 2
```

Printing the Answer and Checking our Work

All of the calculations should be done. To check our work, start by printing the list `numbers`. Then print the value of `middle`. You should be able to check that `middle` actually is the middle third of `numbers`.

```
print(numbers)
print("The middle is ", middle)
```

▼ **Code**

```
import random

numbers = []
for i in range(9):
    numbers.append(random.randint(0, 100))

length = len(numbers)
start = length // 3
stop = length // 3 * 2
middle = numbers[start:stop]

print(numbers)
print("The middle is ", middle)
```

Lab 4

Cross-Referencing Colors

Write a program that cross-references a list of random colors with lists of warm, cool, and neutral colors. Keep track of how many warm, cool, and neutral colors are in the list of random colors. Also keep track of how many colors do not appear in any of the lists.

Setup

The lists `warm`, `cool`, and `neutral` are already provided. We will need a list `colors` filled with a randomly selected colors. We will also need the variables `warm_count`, `cool_count`, `neutral_count`, and `misc_count` to keep track of how often these colors appear. These variables can be set to 0.

```
colors = ["red", "black", "pink", "beige", "dark green", "azure", "amb

warm_count = 0
cool_count = 0
neutral_count = 0
misc_count = 0
```

Iterating

There are a couple of ways to solve this problem. You could iterate over the `warm` list and see if any of the elements in `colors` are present. You would repeat this with the `cool` and `neutral` lists as well. The problem arises when you try and figure out what are miscellaneous colors. There is no list for this. A miscellaneous color would any color that is not in the `warm`, `cool`, or `neutral` lists. Instead, iterate over the `colors` list and see if each element is in the `warm`, `cool`, or `neutral` lists. If an element is not in those three lists, then it is a miscellaneous color.

```
for color in colors:
```

Conditionals

The order of conditionals is not important. Let's start with the `warm` list. If the element is in this list, increment the `warm_count` variable. Use `elifs` for the other lists, and increment the respective variable.

```
for color in colors:
   if color in warm:
      warm_count += 1
   elif color in cool:
      cool_count += 1
   elif color in neutral:
      neutral_count += 1
```

Miscellaneous Colors

If all of these conditionals are false, then the color can be considered miscellaneous. Use an `else` statement to increment the `misc_count` variable.

```
for color in colors:
   if color in warm:
      warm_count += 1
   elif color in cool:
      cool_count += 1
   elif color in neutral:
      neutral_count += 1
   else:
      misc_count += 1
```

Printing the Answer and Checking our Work

Let's start by printing the length of `colors` with a message. Then print the values of `warm_count`, `cool_count`, `neutral_count`, and `misc_count` with a respective message. The sum of these variables should equal the length of `colors`.

```
print("The total # of colors is ", len(colors))
print("There are ", warm_count, " warm colors")
print("There are ", cool_count, " cool colors")
print("There are ", neutral_count, " neutral colors")
print("There are ", misc_count, " miscellaneous colors")
```

▼ **What Happened to the + Operator?**

You may have noticed that the string concatenation operator (+) is not used in the `print` statements. To use concatenation, you must have two strings. Which means you need to type cast the integer variables as strings. Python also lets you use a variable in a string by using a comma to separate the variable from the strings. No type casting is necessary. Both options do the same things.

▼ **Code**

```
colors = ["red", "black", "pink", "beige", "dark green", "azure", "a

warm_count = 0
cool_count = 0
neutral_count = 0
misc_count = 0

for color in colors:
    if color in warm:
        warm_count += 1
    elif color in cool:
        cool_count += 1
    elif color in neutral:
        neutral_count += 1
    else:
        misc_count += 1

print("The total # of colors is ", len(colors))
print("There are ", warm_count, " warm colors")
print("There are ", cool_count, " cool colors")
print("There are ", neutral_count, " neutral colors")
print("There are ", misc_count, " miscellaneous colors")
```

Lab Challenge

Changing a List

Write a program that takes a list of integers called `numbers` and will print a list with the elements `odd` or `even` based on the elements of `numbers`. For example, if `numbers = [1, 2, 3, 4]`, then your program will print `['odd', 'even', 'odd', 'even']`.

Important, do not edit the code in the top section. This code is necessary for the auto-grader to work. Add your code in the section below. Clicking the `TRY IT` button will test your code with `numbers = [1, 2, 3, 4]`.

▼ **Where is the code visualizer?**

Unfortunately, the code visualizer does not work with the statement `import sys`. Since importing the `sys` module is required for this problem, the code visualizer will not be available for this problem.