Coding Your First Program

by Sophia

WHAT'S COVERED

In this lesson, you will learn how to think like a developer and review what's involved in the program development process. Specifically, this lesson covers:

1. Mentalism Program
2. Validating and Debugging

1. Mentalism Program

For this first program, we'll make use of the various parts of a program that we've looked at so far. Our goal will be to create a program that can perform a mentalism magic trick in Python. Let's first try it out on you—no programming yet!

> Hello! What is your name?
> Now, we'll perform some mind-reading on you.
> First, think of a number between 1 and 10.
> Multiply the result by 2.
> Now, add 8 to it.
> Now, divide the number by 2.
> Now, subtract the original number that you thought about.
> Alright, let's do some processing now.
> Is the number that you now have a 4?

Wow!

It's important to note that the number that you originally chose doesn't have any effect on the result at all. The only number that mattered was the number that was added (the number 8 in our trick), as the result is half that number (the number 4). To ensure that we have nice round numbers, the number that was added must be an even number. All the rest of the calculation was an illusion to hide what was being done in the program. Viewing this in equation form may also help to reveal the "magic." With x as the number guessed, the math above yields:
((((x*2) + 8)/2) – x), and this simplifies to ((x + 4) – x), which equals 4.

What inputs do we have? We want to ask the person's name. In this case, for our code, we don't need to ask the person for the number that they are guessing since that number doesn't affect us directly. What we do need to do is generate a random even number for the person to add.

Our processing will involve random number generation. For simplicity, we will first generate a random number between 1 and 5. That will be the final answer. The number that the person will have to add (to the number they are thinking of) will be that number multiplied by 2, as it will need to be even.

Our output screen will be letting the user know what number they have at the end of the calculations. Let's see what that program would look like:

TRY IT

Directions: Let's try to add the following code snippets to the IDE. We will explain what is happening before each snippet.

In this first line, we have a comment describing what the code is meant to do. This is good practice to start adding in these comments to define what the code represents. We can tell it's a comment with the # that starts the line.

```
#We are importing a module that we need to be able to generate random numbers
```

This next line is meant to import a specific module called random. There are some modules or pieces of code that are created for you that you can make use of. The random module is one that allows you to generate pseudo-random numbers, which we'll use. Notice that the code and the comments will be in different colors in the editor. That may be different depending on your settings in the IDE.

```
import random
```

In this next set of lines, we'll describe what it is that we're intending to do as part of the program. Remember that the lines with the # are comments which are not run by the program. Rather, they are for you as the programmer to explain what the code is doing.

```
#We are creating a random even number between 2 and 10 by
#first randomizing an integer between 1 and 5. This will be our
#final number. The number to add will take this final number and multiply it by 2.
```

The next two lines define two different variables, `randomFinalNumber` and `numberToAdd`, that we'll be using as part of the program. The `randomFinalNumber` is taking a random number between 1 and 5 and assigning it to `randomFinalNumber`. This is something that is part of the random module that we imported previously. The `numberToAdd` is taking the value of `randomFinalNumber` and multiplying it by 2. So, if `randomFinalNumber` was set to 3, `numberToAdd` is set to 6.

```
randomFinalNumber = random.randrange(1, 5)
numberToAdd = randomFinalNumber * 2
```

Next, we'll define another variable called `name`. We'll use the function input to wait on input from the user and have it set to the variable `name`. Prior to that, it'll prompt the user (without quotes) "Hello! What is your name?". Then, it will wait for the user to enter a response. The response is then stored in the variable name that we've declared.

```
#Asking the user to enter in their name
name = input("Hello! What is your name? ")
```

Next, we'll have a series of output and input statements. The print statements will simply output the text to the screen. Notice that literal strings have double quotes around them. We can add in the variable values within them by having them in the print lines as well.

```
#Script to walk through each of the steps
print("Welcome " +name +", we'll perform some mind reading on you.")
print("First, think of a number between 1 and 10.")
answer = input("Ready for the next step? ")
print("Multiply the result by 2.")
answer = input("Ready for the next step? ")
print("Now, add...let's see...")
print(numberToAdd)
answer = input("Ready for the next step?")
print("Now, divide the number you have by 2.")
answer = input("Ready for the next step? ")
print("Now, subtract the original number that you thought about.")
answer = input("Ready for the last step? ")
```

Lastly, we'll print the `randomFinalNumber` to the screen and watch the astonished user compare the same number, had they been following the steps of the program.

```
#Guessing the number
print("Well " +name +", let me read your mind...The number that you have right now is a....")
print(randomFinalNumber)
```

Let's walk through the algorithm of the program to see what happens. After each step, we ask the user to enter something so that they can move to the next step. It doesn't matter what they enter, though, as we don't make use of the response. Here is a sample run:

```
Hello! What is your name? Sophia
Welcome Sophia, we'll perform some mind reading on you.
First, think of a number between 1 and 10.
Ready for the next step?
Multiply the result by 2.
Ready for the next step?
Now, add...let's see...
8
Ready for the next step?
Now, divide the number you have by 2.
Ready for the next step?
Now, subtract the original number that you thought about.
```

```
Ready for the last step?
Well Sophia, let me read your mind...The number that you have right now is a....
4
```

2. Validating and Debugging

This is nice, but it would also be useful if we can validate that the calculations are correct. Let's add some functionality to the program to actually perform the calculations so you can ensure that it works. We'll store this in a new variable called `enteredNumber` and output the result after each step. Remember, the process of testing and fixing those errors is called debugging. Testing and debugging are two terms that you'll hear used quite often. Testing normally just identifies if there are bugs, whereas debugging is going through the process of fixing those bugs. Outputting the variables as you progress is a very useful way to debug your code.

TRY IT

Directions: Let's try this again. Enter the following code into the IDE. Updates/changes for lines of code have comments after them.

```
#We are importing a module that we need to be able to generate random numbers
import random

#We are creating a random even number between 2 and 10 by
#first randomizing an integer between 1 and 5. This will be our
#final number. The number to add will take this final number and multiply it by 2.
randomFinalNumber = random.randrange(1, 5)
numberToAdd = randomFinalNumber * 2

#Asking the user to enter in their name
name = input("Hello! What is your name? ")

#Script to walk through each of the steps
print("Welcome " +name +", we'll perform some mind-reading on you.")
print("First, think of a number between 1 and 10.")
```

This is a new line here that we'll use to prompt the user to enter a number. Notice, though, that we have `int()` that wraps around the input statement. This converts the string that is read in to an `int` data type. We'll get into more of these statements and what they do later in the course.

```
enteredNumber = int(input("Enter in a number between 1 and 10: ")) # this is a new line
print("Multiply the result by 2.")
```

The next new line we are adding starts the calculation. Here, we're taking the `enteredNumber` that the user entered in and multiplying it by 2 and setting it to the variable `userNumber`. Then, in the following line, we'll be

outputting to the screen what that `userNumber` is currently set to.

```
userNumber = enteredNumber * 2 # this is a new line
print(">> userNumber at this step = " + str(userNumber)) # this is a new line
answer = input("Ready for the next step? ")
print("Now, add...let's see...")
print(numberToAdd)
```

The next two lines add the `numberToAdd` to the `userNumber` variable, then again print out to the screen what `userNumber` now is currently set to.

```
userNumber = userNumber + numberToAdd # this is a new line
print(">> userNumber at this step = " + str(userNumber)) # this is a new line
answer = input("Ready for the next step? ")
print("Now, divide the number have by 2.")
answer = input("Ready for the next step? ")
```

Next, we will take the `userNumber`, divide it by two, and set it back to `userNumber`. Then, we'll be outputting to the screen what that `userNumber` is currently set to.

```
userNumber = userNumber / 2 # this is a new line
print(">> userNumber at this step = " + str(userNumber)) # this is a new line
print("Now, subtract the original number that you thought about.")
answer = input("Ready for the last step? ")

#Guessing the number
print("Well " +name +", let me read your mind...The number that you have right now is a....")
print(randomFinalNumber)
```

Finally, these last two lines of code will deduct the initial number that the user entered at the beginning of the code. Then, we'll be outputting to the screen what that `userNumber` is currently set to.

```
userNumber = userNumber - enteredNumber # this is a new line
print(">> userNumber at this step = " + str(userNumber)) # this is a new line
```

Outputting the variables as you program for testing purposes can be very helpful for debugging the logic as you go. At each step, we output the current calculation so that we can verify at the end that the logic is correct.

```
Hello! What is your name? Sophie
Welcome Sophie, we'll perform some mind reading on you.
First, think of a number between 1 and 10.
Enter in a number between 1 and 10: 5
Multiply the result by 2.
>> userNumber at this step = 10
Ready for the next step? y
```

```
Now, add...let's see...
8
>> userNumber at this step = 18
Ready for the next step?y
Now, divide the number you have by 2.
Ready for the next step? y
>> userNumber at this step = 9.0
Now, subtract the original number that you thought about.
Ready for the last step? y
Well Sophie, let me read your mind...The number that you have right now is a....
4
>> userNumber at this step = 4
```

Once we are done, we can simply comment out the lines of code that shouldn't be output to the screen.

TRY IT

Directions: Using the IDE, make sure you comment out the lines of code used for testing (see what is commented out below) so your testing variable does not show to the end-user.

```
#We are importing a module that we need to be able to generate random numbers
import random

#We are creating a random even number between 2 and 10 by
#first randomizing an integer between 1 and 5. This will be our
#final number. The number to add will take this final number and multiply it by 2.
randomFinalNumber = random.randrange(1, 5)
numberToAdd = randomFinalNumber * 2

#Asking the user to enter in their name
name = input("Hello! What is your name? ")

#Script to walk through each of the steps
print("Welcome " +name +", we'll perform some mind reading on you.")
print("First, think of a number between 1 and 10.")
#enteredNumber = int(input("Enter in a number between 1 and 10: "))
print("Multiply the result by 2.")
#userNumber = enteredNumber * 2
#print(">> userNumber at this step = " + str(userNumber))
answer = input("Ready for the next step? ")
print("Now, add...let's see...")
print(numberToAdd)
#userNumber = userNumber + numberToAdd
#print(">> userNumber at this step = " + str(userNumber))
answer = input("Ready for the next step?")
```

```
print("Now, divide the number you have by 2.")
answer = input("Ready for the next step? ")
#userNumber = userNumber / 2
#print(">> userNumber at this step = " + str(userNumber))
print("Now, subtract the original number that you thought about.")
answer = input("Ready for the last step? ")

#Guessing the number
print("Well " +name +", let me read your mind...The number that you have right now is a....")
print(randomFinalNumber)
#userNumber = userNumber - enteredNumber
#print(">> userNumber at this step = " + str(userNumber))
```

Another approach to debugging and testing could be to add the code at the bottom of the program so that the code is all combined together.

TRY IT

Directions: Try removing the testing code from the main program and add it to the bottom like you see in the program below.

```
#We are importing a module that we need to be able to generate random numbers
import random

#We are creating a random even number between 2 and 10 by
#first randomizing an integer between 1 and 5. This will be our
#final number. The number to add will take that and multiply it by 2.
randomFinalNumber = random.randrange(1, 5)
numberToAdd = randomFinalNumber * 2

#Asking the user to enter in their name
name = input("Hello! What is your name? ")

#Script to walk through each of the steps
print("Welcome " +name +", we'll perform some mind-reading on you.")
print("First, think of a number between 1 and 10.")
print("Multiply the result by 2.")
answer = input("Ready for the next step? ")
print("Now, add...let's see...")
print(numberToAdd)
answer = input("Ready for the next step? ")
print("Now, divide the number have by 2.")
answer = input("Ready for the next step? ")
print("Now, subtract the original number that you thought about.")
answer = input("Ready for the last step? ")
```

```
#Guessing the number
print("Well " +name +", let me read your mind...The number that you have right now is a....")
print(randomFinalNumber)

#Validating the results
enteredNumber = int(input(">> Enter in the number the individual guessed between 1 and 10: "))
userNumber = enteredNumber * 2
print(">> Muliplied by 2 = " + str(userNumber))
userNumber = userNumber + numberToAdd
print(">> Told to add "+ str(numberToAdd) + " = " + str(userNumber))
userNumber = userNumber / 2
print(">> Divided by 2 = " + str(userNumber))
userNumber = userNumber - enteredNumber
print(">> Subtracted the original number "+ str(enteredNumber) +" = " + str(userNumber))
```

The final output should look like the following:

Note: Remember this program is using the random number generator so the numbers in the output below can be different than what you see when you run the program (it was 6 for this test run). However, the final number should be correct based on the random number chosen at the beginning.

```
Hello! What is your name? Sophie
Welcome Sophie, we'll perform some mind reading on you.
First, think of a number between 1 and 10.
Multiply the result by 2.
Ready for the next step?
Now, add...let's see...
4
Ready for the next step?
Now, divide the number you have by 2.
Ready for the next step?
Now, subtract the original number that you thought about.
Ready for the last step?
Well Sophie, let me read your mind...The number that you have right now is a....
2
>> Enter in the number the individual guessed between 1 and 10: 6
>> Muliplied by 2 = 12
>> Told to add 4 = 16
>> Divided by 2 = 8.0
>> Subtracted the original number 6 = 2.0
```

To see the final version of this program visit **Sophia's Python code page**

SUMMARY

In this final lesson of Challenge 1.2, we had the chance to think like a developer while we reviewed the first coding program, the **Mentalism Program**. We added code snippets of the program into the IDE to test individually. We also had a chance to **validate and debug** the program by outputting variables to the screen as the program progressed through the code.

Best of luck in your learning!