Introduction to PYTHON – Part 3

Python 020 – Constants I

Some programming languages allow you to create a "constant", which is like a variable but can never be changed in value. Python does not, so your only choice is to create a variable and then never modify it!

An example may be:

```
taxRate = 0.14
```

Assignment

Write a program that initializes a variable HST with the current Ontario value, then prompts for the price of an object on sale, prints the tax, and the total price in a neatly formatted receipt format.

Save as "020.py".

Python 020a – Constants II

Assignment

Manitoba currently has not adopted the HST tax system that Ontario uses. Investigate the tax rates (RST, GST) used in Manitoba, and write a program that prompts for the price of an object on sale, prints the individual taxes, the total tax, and the total price in a neatly formatted receipt format.

Save as 020a.py.

Python 021 – Variables I

Remember that variables can be used to represent text (called “strings” in programming context) as well as numbers.

The following example assigns the string value "Mary" to a variable called “name”:

```
name = "Mary"
```

Note the use of quotation marks around the string.

Assignment

Write a program that includes two variables called "FName" and "LName". Assign your first name and last name to these variables, then print your name to the screen.

Save as "021.py".

Python 021a – Variables I

Assignment

Write a program that prompts for and prints your name, your address, your city, and your phone number. Use appropriate variable names for all variables.

Save as "021a.py".

Python 021b – Variables II

Assignment

Write a program that prompts for a total of 10 variables, using a combination of integer, real number, and string types. Output the variables in sentence format using no more than five sentences, similarly to "Your favourite colour is red and favourite number is 7.", where "red" and "7" are the values that were entered.

Save as "021b.py."

Python 022 – Variables III

Assignment

Write a program that prompts for and calculates the product of three integers. The output should look similar to "The product of __, __, and __ is __. "

Save as "022.py".

Python 022a – Variables IV

Assignment

Complete the programming challenge "Transistor Current Gain".

Program Requirements

A transistor's current gain is calculated using the formula $\beta = I_C / I_B$, where β (Beta) is the current gain, I_C is the collector current, and I_B is the base current.

Write a program that prompts for Base and Collector currents, and outputs the current gain.

Example Run

```
Enter the amount of collector current (in Amperes): 0.1
Enter the amount of base current (in Amperes): 0.001
The transistor current gain is 100.
```

Supplemental Exercise

Modify the program so it continuously runs until 0 (zero) is entered for the collector current.

Save as "022a.py".

Python 023 – Counters & While Loops I

Counters are integer variables used to keep track of how often a programming event occurs. For example, a counter may be used to keep track of how many times the user has tried to win a game.

A loop is a segment of program code that is repeated until a specific event occurs. Loops are often used in conjunction with counters, as the following code demonstrates:

```
# counter
i = 0

# while loop
while i < 5:
    print i
    i = i + 1

# let us know we're finished
print "Finished!"
```

In this example, the variable "i" is used to count the number of times the loop is executed. You will notice that "i" is used often as a loop variable, as well as "j" and sometimes "k" for nested loops (loops within loops).

The loop that is shown here is a "while" loop. While loops are used when we don't necessarily know how many times the loop will execute. We will also learn about "for" loops in a later lesson, which are used when we know exactly how many times a loop must execute.

Note that the lines below the while statement are indented. This tells Python which lines are to be run within the loop.

The statement "`i = i + 1`" is used to increment the loop counter. This value is tested after each iteration of the loop code to see if the condition (`i < 5`) is still true. As long as it's true, the loop will continue executing.

Assignment

Modify the example program so it prints all even numbers from 100 to 120 (inclusive).

Save as "023.py".

Additional Notes

In the above example, we used the "<" (less than) operator. Any of the following operators are legal:

```
<  less than
>  greater than
== equal to
!= not equal to
>= greater than or equal to
<= less than or equal to
```

Python 023a – Counters & While Loops II

Assignment

Modify 023.py so it prompts for a starting number and ending number.

Save as "023a.py".

Python 023b – Counters & While Loops II

Assignment

Modify 023.py by replacing the "`while i < 5`" statement with "`while 1`" (or "`while True`"). What happens to your program?

Do you understand why this is called an infinite loop?

(You do not have to save this program.)

Python 024 – For Loops I

Another kind of loop is the "for" loop, which, as previously mentioned, is used when you know exactly how many times you need the loop to execute. The number of times, and the values of the loop counter, are specified as shown below:

```
for i in range (1,5):
    print "The index counter is", i
print "All done!"
```

The output of this example looks like this:

```
The index counter is 1
The index counter is 2
The index counter is 3
The index counter is 4
All done!
```

Just like with the while loop, the lines that are executed in the loop are determined by which lines are indented. In the example above, the the loop counter ("**I**") equals "1" in the first iteration (as designated by the first argument in the for loop), then 2, then 3, then 4. Note that the loop then stops; it does not execute the loop for "5", as you would first think by the example. Think to yourself that the loop starts at 1 and runs as long as it is less than 5.

Here's another example, but this time only printing every 2nd number:

```
for i in range (1,10,2):
    print "The index counter is", i
print "All done!"
```

The third argument in the for loop (the "2") is the step size, which tells Python to iterate the loop by 2's. The output from this loop looks like this:

```
The index counter is 1
The index counter is 3
The index counter is 5
The index counter is 7
The index counter is 9
All done!
```

Assignment

Modify program 023 so that it uses a for loop instead of a conditional loop.

Save as "024.py".

Python 024a – For Loops II

Assignment

Modify program 023a so that it uses a for loop instead of a conditional loop.

Save as "024a.py".

Python 024b – For Loops III

Assignment

Write a program that prompts for an initial amount of money and an annual interest rate, then calculates how much money will be accumulated in 20 years.

Save as "024b.py".

Python 024c – For Loops IV

Assignment

Write a program that prompts for a starting temperature *temp1* and ending temperature *temp2*, then prints a neatly formatted Celsius to Fahrenheit conversion table, printing one conversion every 5 degrees.

Save as "024c.py".

Python 024d – For Loops V

Assignment

Modify the above temperature conversion program so it converts from Fahrenheit to Celsius.

Save as "024d.py".

Python 024e – For Loops VI

Assignment

Write a program that prompts for a number *n*, then prints the factorial (n!) of that number.

Factorial Formula: http://easycalculation.com/statistics/learn-factorial.php

Save as "024e.py".

Python 024f – For Loops VII

Supplemental Assignment

Write a program that prompts for a number *n*, then prints the *n*th Fibonacci number.

Fibonacci Number: http://mathworld.wolfram.com/FibonacciNumber.html

Save as "024f.py".