

Assignment 3

Covering Basic Python

Due: Wednesday, February 11, before 5:00 PM. This assignment is an "individual" assignment (not group work), and each student shall do it independently and turn in a separate document.

Do the following exercises, giving your answers in a raw text (.txt) file.

a. In Python 2.5 (which our PixelMath Python uses) or Python 2.7, a long integer is a subtype of int that can store a number of arbitrarily large magnitude, either positive or negative. Using the factorial function defined in the built-in demo within PixelMath Python, experiment to find the smallest value of n for which factorial(n) returns an answer of type long integer.

b. Write a series of 3 lines of Python, such that the first two are assignment statements giving variable eurosPerDollar and the variable DollarsInTheBank the values 0.8 and 200, respectively. Then the third line should compute the product of these and assign the value to a variable euroEquivalent.

c. Give Python code to compute both the integer quotient and remainder for division of 1000 by 17.

Hint: see the list of available arithmetic operators at

<https://docs.python.org/2/library/stdtypes.html#numeric-types-int-float-long-complex>

d. What might be a good experimental strategy for finding the largest float that PixelMath Python can represent?

e. Use your strategy from d to find that number. That number should be a finite number. (If you are within 1 percent of the correct answer, that will be fine.)

f. Use a similar strategy to find the smallest number greater than 0.0 that can be represented.

e. Come up with a way to assign to a variable s the following string, such that the number of characters on the line is as small as you can, and it's all on one line of Python code. Here is the string (shown on multiple lines here, but assume it is a one-line string):

```
He said, "It's clear you can't do it. But it's clear Kearse caught it."
and he repeated, "It's clear".
```

f. How many characters did you use?

g. Now, assuming you are allowed to perform multiple assignments, come up with code that achieves the result using fewer characters of code, and still on one line.

h. How many characters did you use in your new version?

i. Starting with the string variable `misp = "Misp"`, build an expression, without using any other string literals, that evaluates to "Mississippi". (The expression should use slices of `misp` and string concatenation.)

j. Write a couple lines of Python code that defines a list of the months of the year (in order from January to December) but then prints out the list in alphabetical order.

k. Write code that sets up a dictionary with nine planet names as keys, starting with "Mercury" and which includes as their values, their diameters, in kilometers. Then write code that will list and print out all these pairs, in order of largest planet first. The list should be constructed, by your code, from the dictionary.

l. Consider the following conditions that can be true or false. For now, they are shown as all true:

```
raining = True
away = True
injuries = True
hawks = True
```

Give a Python assignment statement with boolean expression such that `favor` is true provided that either it is not raining and the team is not away and injuries is not true, or it's the hawks and injuries is not true. (Otherwise, `favor` is false).

m. Give a function definition for taking the cube root of a number. The definition should start out:
`def cube(n):`

n. Give a recursive definition for a function that takes a list of names and returns a list of initials. For example:
`list_of_initials(["Amy", "Bob", "Cath", "Danny"])`
would return
`["A", "B", "C", "D"]`

o. Give a recursive function definition for a function that accepts a list of integers and returns a new list of integers, in which each element of the original is repeated in the result:

For example:
`repeat([3, 1, 4, 1, 5])`
should evaluate to
`[3, 3, 1, 1, 4, 4, 1, 1, 5, 5]`

p. Write a function definition for `ask_the_user(prompt)` that when called, uses the `raw_input` method to ask the user the question given in the `prompt` argument. When the user responds with Yes or No (not case-sensitive), it should return `True` (for yes) or `False` (for no). If the user gives some other answer, then your function should reprompt them with the additional prefix to the prompt: "You must answer Yes or No to the question, " and followed by the question again. It should keep doing this until the user does respond with either Yes or No.

Example:

```
ask_the_user("Do you like oysters?")
The function prompts the user: Do you like oysters?
  (user replies Maybe)
the system re-prompts the user: You must answer Yes or No to the
question, Do you like oysters?
  (user replies no)
False is returned.
```

q. Define a function `guess_my_number(max_num, max_tries)` that starts a game of "Guess My Number" and plays it with the user.

For example,

```
guess_my_number(100, 8)
should welcome the user with a message such as
  I am thinking of a number between 0 and 100.
  You have 8 questions to ask. Each question should be of one
  of the following forms.
    Is your number less than 50?
    Is your number greater than 15?
    Is your number 23?
```

The user wins if the number is guessed using only 8 or fewer questions. Otherwise, the computer agent (you can name it) wins.

After each question, the computer agent should answer either Yes, No, Yes you got it! or Sorry but no. These last two answers are for the third kind of question.

Turn in your work by 5:00 PM, Wednesday, Feb. 11.

Submit the document electronically that contains your answers to the Assignment 3 DropBox at Catalyst CollectIt. <https://catalyst.uw.edu/collectit/dropbox/tanimoto/34051>