

Programming Project #1

Assignment Overview

This assignment involves coding and testing of a program based on the “Hello World” program from the first lab, and then using the **handin** program to hand it in.

You can turn in programs using **handin** as often as you want. Try turning in the “Hello World” program (or the completed project) during lab so if there is a problem, the TA can help you solve it. During lab you should create the project so that if you need help, you are still in lab with the TA available. Also, submit the program (even if it isn't done, you can always resubmit it later) to make sure that you are enabled for handin. Complete the project and hand it in again when done.

The basic design of the first programs that you construct in this class consists of a prompt for information, receiving information, processing that information, and then displaying the results.

This assignment is worth 10 points (1% of course grade), and must be completed before 11:59 PM on Monday, January 14th. Projects are typically due on Mondays.

Background

This programming project will use the `input` and `print` functions along with some simple mathematics for conversion. The important part of the project is to learn the skills needed to access the class web site to download a project description, create a new program in Python and finally to hand it in.

The Richter Scale

The Richter scale is a way to measure and relate the magnitude of earthquakes. It is a base-10 logarithmic scale. The magnitude is defined as the logarithm of the ratio of the amplitude of waves measured by a seismograph to an arbitrary small amplitude. An earthquake that measures 5.0 on the Richter scale has a shaking amplitude 10 times larger than one that measures 4.0, and corresponds to a 31.6 times larger release of energy. [Rephrased from Wikipedia]

You are going to relate a Richter scale measurement to the amount of energy released by an earthquake. You will do so in two ways: the number of joules (a standard physics measure of energy) released, and the equivalent amount of energy as related to an amount of TNT exploded. Finally, you will list a number of Richter measurements and their energy equivalents.

Formulae

The energy in joules released for a particular Richter scale measurement is provided by the formulae:

$$Energy = 10^{(1.5 * richter) + 4.8}$$

Where *Energy* is measured in joules and *richter* is the Richter scale measurement (typically on a scale from 1-10 as a floating point number).

One ton of exploded TNT yields 4.184×10^9 joules. Thus you can relate the energy released in joules to tons of exploded TNT.

Program Specifications

Your program will prompt the user for a **floating point** number representing a Richter scale measure. Your program will then print the following information:

- The original Richter scale measure provided by the user
- The energy, in joules, released by that measure
- The equivalent of that energy measure in tons of TNT exploded

Your program will then print both the joule measure and the tons of TNT measure for the following Richter scale measures:

- 1.0
- 5.0
- 9.1 (Indonesia earthquake, 2004)
- 9.2 (Alaska earthquake, 1964)
- 9.5 (Chile earthquake, 1960. Largest measured).

You can use the website http://www.convertalot.com/earthquake_power_calculator.html to check your work.

Deliverables

proj01.py -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the **specified file name**, i.e. "proj01.py"
2. Save a copy of your file in your CSE account disk space (H drive on CSE computers).
3. You will electronically submit a copy of the file using the "handin" program:
<http://www.cse.msu.edu/handin/webclient>

Assignment Notes:

- To input the numbers it is necessary to use the `input` function. The `input` function takes a string, a sequence of characters between quotes, as a prompt to print to the user. It then waits until the user types a response, terminated by the user typing the Enter key. A string, again as a sequence of characters, is returned.
- The returned string must be converted to a number. In this assignment we are strictly working with floating point numbers, a string is converted to a float using the `float` function. The `float` function takes as an argument a single string and returns the floating point number the string represents. A typical interaction would be something like:

```
num_str = input("Please enter a number: ")
num_float = float(num_str)
```

- `print` is a command that will print on the output window any combination of variables, values and strings. Each item to be printed must be separated from other items by a comma. All the items will be printed together, followed by a new line. For example:

```
bills_float = 123.456
print("Number ",bills_float," times two: ", bills_float*2)
```

- This command has 4 items to print: a string ("Number "), the value in the variable `bills_float` (123.456), another string (" times two: ") and the result of evaluating an expression (246.912). It will print:

```
Number 123.456 times two: 246.912
```

- Look at the program *numInput* in the 231 Examples directory (<http://www.cse.msu.edu/~cse231/Examples/CoursePack/Python/>) as an example of using `input`, `print` and `float` (also `int` for integer numbers).
- Once converted to numbers, the operations on these numbers are, respectively: `+` (sum), `-` (difference), `*` (product), `/` (division), `//` (integer division) and `%` (integer remainder).
- In Python, no matter the kinds of numbers involved (integer or float), the result of division (`/`) is a floating point number. For example:

```
>>> 3/4
0.75
```
- To raise a number to a power, use the `**` operator. For example:

```
print("10 raised to the power 1.5 is ", 10**1.5)
```

which results in:

```
10 raised to the power 1.5 is 31.622776601683793
```
- To indicate scientific notation in Python, use the format `xey`, where *x* and *y* denote numbers (`float` or `int`), as in `4e5` to indicate 4×10^5
- Python has many facilities to make the output look nice, but we have not studied them yet. Although not required, you can do simple alignment (as in the provided snapshot) by placing strings of spaces (" ") of particular lengths in the print statement. It will take some trial and error to get it to look better.

To clarify the problem specifications, we provide at the end of this document a snapshot of interaction with the already written program.

Getting Started

1. Using IDLE create a new program.
2. If you are in a CSE lab, select the H: drive as the location to store your file. And be sure to log out when you leave the room or the next person to sit down at the machine will be able to copy your program.

3. Save the name of the project: proj01.py
4. Using the example from numberInput.py, write the code. Track down any errors (shouldn't be any at first).
5. Run the program
6. Use the handin web site to hand in the program (to make sure you can do it)
7. Edit the program
8. Now you enter a cycle of edit-run to incrementally develop your program.
9. Use handin to hand in your final version.

*Be sure to save a copy of your completed program in your CSE file space (H: drive on the lab machines) **before** the deadline by which the program needs to be handed in. If you write your program at home and turn it in from home, you will need to copy it over **before** the deadline. In case of problems with electronic submission, an archived copy on the CSE disk is the only acceptable evidence of completion.*

Questions for you to consider (not hand in)

1. What happens if you enter a negative number?
2. What happens when you enter a letter instead of a number at the prompt?

Sample Interaction

```
>>> ===== RESTART =====
>>>
Please give me a Richter scale measure: 3.4
Richter measure: 3.4
Equivalence in joules: 7943282347.242789
Equivalence in Tons of TNT: 1.8984900447521007

Richter      Joules      TNT
1            1995262.3149688789      0.00047687913837688307
5            1995262314968.8828      476.87913837688404
9.1          2.818382931264449e+18      673609687.2046962
9.2          3.981071705534953e+18      951498973.5982201
9.5          1.1220184543019653e+19      2681688466.3048882
>>>
```