Function Implementation

My function “convert.py” takes two inputs, a number and unit of length, and outputs two lists consisting of the conversion of the input length into different units of measurement. The function begins by establishing a “base” list of conversion values and their units, provided by Prof. Lee. This basis is defined by the conversion of 1 meter to the other units:

\[
\text{metn} = [0.000621, 1.093613, 3.28084, 39.370079] \\
\text{metu} = ['miles', 'yards', 'feet', 'inches']
\]

The code proceeds to define similar value and unit lists for the other measurements, based off of the values from the basis above. For instance, the lists for miles are:

\[
\text{miln} = [1/\text{metn}[0], (1/\text{metn}[0])*\text{metn}[1], (1/\text{metn}[0])*\text{metn}[2], (1/\text{metn}[0])*\text{metn}[3]] \\
\text{milu} = ['meters', 'yards', 'feet', 'inches']
\]

There are similar lists for yards, feet and inches.

The code then defines two empty lists, and initiates two indices to 0:

\[
\text{out} = [] \\
\text{out2} = [] \\
i = 0 \\
j = 0
\]

Finally, the code has an “if” statement for each possible unit input, and then runs through two loops: one to fill each list with the different conversions:

\[
\text{elif u == 'feet':} \\
\quad \text{while } i < \text{len(metn)}: \\
\quad \quad \text{conv} = "%.2f" % (\text{n*feen}[i], \text{feeu}[i]) \\
\quad \quad \text{out}.append(\text{conv}) \\
\quad \quad i = i+1 \\
\quad \text{print out} \\
\quad \text{while } j < \text{len(simetn)}: \\
\quad \quad \text{conv2} = "%.2f" % (\text{n*feen}[2]*\text{simetn}[j], \text{simetu}[j]) \\
\quad \quad \text{out2}.append(\text{conv2}) \\
\quad \quad j = j+1 \\
\quad \text{print out2}
\]
Examples

Running the code from the Terminal window using the “python” command will prompt both a length value and a unit input. Here’s an example converting 77 inches:

```
$ python convert.py
Input length (number): 77
Input units (miles, meters, yards, feet, inches): inches
['0.00121455179198 miles', '2.13888829128 yards', '6.4166682965 feet', '1.95579998709 meters']
['1955799987.09 nm', '1955799.98709 um', '1955.79998709 mm', '195.579998709 cm', '0.00195579998709 km']
```

Here’s an example converting 34 feet:

```
$ python convert.py
Input length (number): 34
Input units (miles, meters, yards, feet, inches): feet
['0.00643554699406 miles', '11.3333298789 yards', '10.3631996684 meters', '407.999989637 inches']
['10363199668.4 nm', '10363199.6684 um', '10363.1996684 mm', '1036.31996684 cm', '0.0103631996684 km']
```