# PA3: Unit Converter

Write a Unit Converter for many kinds of units.

## Requirements

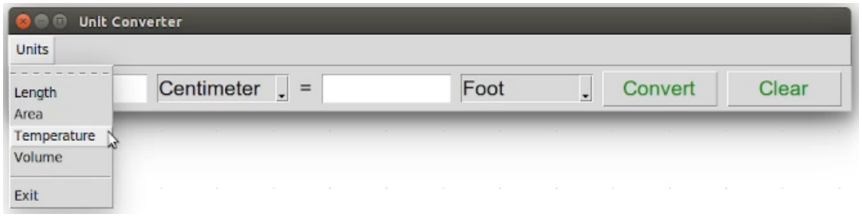
1. The application can convert at least 4 different types of units, including Length and Temperature.

2. The ConverterUI and UnitConverter classes **do not depend on any actual unit types**.

   The symbols UnitType.LENGTH, UnitType.TEMPERATURE, etc., must not appear anywhere in ConverterUI or UnitConverter source code!

3. Use main.py to load the initial units into the ConverterUI. OK to reference a specific unit type in main.py.

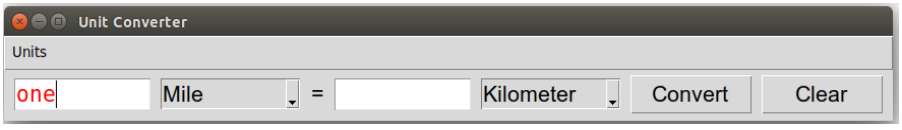
4. There is a menu that shows the available unit types and an "Exit" item. The ConverterUI gets the available unit types from the UnitType enum.




5. The Comboboxes of units must show actual unit names when the app is loaded or the user selects a new unit type (using the menu). Never display an empty combobox.

6. The application should never crash and never print on the console.

7. If the user enters an invalid value, change the text color to **red** and clear the other input box so it is clear that there is an input error. Remember to change colors back to the normal color the next time he performs a conversion or presses Clear.
*Better*: also set the focus on the input field where the error occurred. This will cause the mouse pointer to jump to that input field.




## The Polymorphic Acid Test

When your code is complete, you should be able to add a new unit type to unittype.py and some new units in units.py, and they *automatically* appear in the application. No changes to converter.py or converter_ui.py are required.

## Programming Hints

The Unit class in the starter code has an optional 4th attribute named offset, with a default value of 0. You can use this for units like temperature that have some offset from the base value. It's up to you to decide what is the reference or "standard" unit, which will determine the offsets.

For example, using Kelvin as the "standard" or reference temperature scale, then

T(Celsius) = 1.0 * T(Kelvin) - 273.15

but using Fahrenheit as the "standard" or reference temperature scale (I don't recommend this):

T(Celsius) = 0.555555 * T(Fahrenheit) - 17.77778

## Dynamic Unit Conversion using a Web Service

If you are ambitious, you can implement a unit type that uses a web service to perform the conversion, such as converting currencies. In this case, modify the UnitType enum to include a function reference to a converter function (that you write) that calls the web service. That is, each UnitType object can have its own converter function.

Modify the convert method in the UnitConverter class to use the converter function from the UnitType object, instead of trying to always do the conversion itself.