

# Procedures Redux

We looked briefly at procedures before. Now we need to get our hands a tad dirtier and understand procedures better. But, for now, let's forego any more discussion of their purposes and uses in the abstract. Let's just write some, but with some explanation.

Let's say your program takes in user input, input that might vary but, in every case, you have to process the input and provide a message back to the user. Here, we have provided a program that converts celsius to fahrenheit, and so to the user it appears so:

```
>>> "Please enter the temperature you want to convert: "  
20 # user's input  
>>> "The temperature in celsius of 20 degrees is equivalent to 68.0 degrees in  
fahrenheit."
```

But in the background you have to do something like this:

```
# [1] take in user input  
userinput = input("Please enter the temperature you want to convert: ")  
  
# convert input to integer  
userinput_to_int = int(userinput)  
  
# calculate the temperature in celsius  
celsius_temp = userinput_to_int * 9 / 5 + 32  
  
# return information to user  
print(f"The temperature in celsius of {userinput_to_int} degrees is equivalent  
to {celsius_temp} degrees in fahrenheit.")
```

If your response to this is, "So what?", then you fail to understand how extensive and in-depth code can be. For example, what if the user also wants to get celsius from fahrenheit? Rankine to fahrenheit? Kelvin to fahrenheit?

And that's just for starters.

Not only is creating a single module for each eventuality a mug's game, but you also have to control how you direct the program from one to the other.