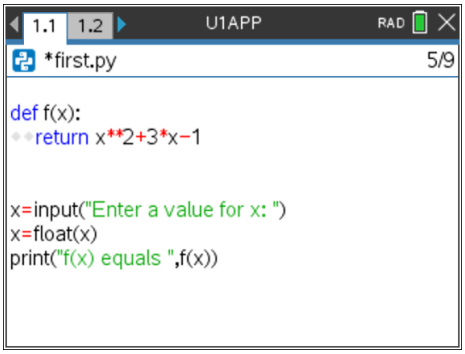
\Unit 1: Getting Started with Python | **Application: Two Functions Are Better Than One**

In this application you will make a second function and use both functions to investigate some of the mathematical properties of functions.

Objectives:

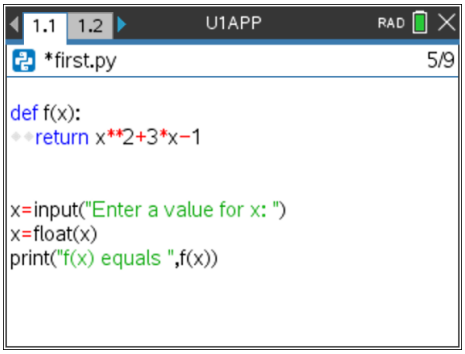
- Editing a Python file
- Copying a Python file
- Adding a function
- Evaluating function expressions in the Shell
- Creating an inverse function.

In the last lesson (**Unit 1, Skill Builder 3**) you defined a function f(x). In this application you will add another function to that file and then evaluate some expressions using those functions.

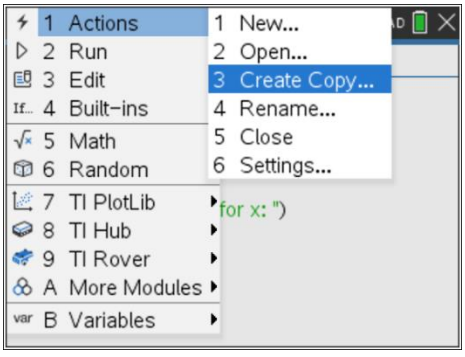


Teacher Tip: This application is an extension of Skill Builder 3 and reinforces some mathematics regarding arithmetic expressions with functions and making an inverse function.

1. Begin with the TI-Nspire document with the Python program you used in Skill Builder 3. See the image.
   Save the TI-Nspire document using a different name by pressing **doc > File > Save As…** and use a different name. The title bar in the image shows the new name U1APP.



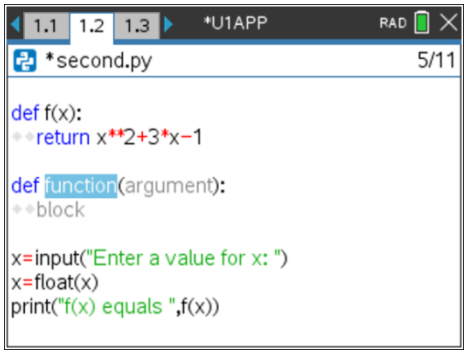
2. Make a *copy* of the Python program by pressing **menu > Actions > Create Copy…** in the Python Editor.
   Give the copy another name (the default adds a 1 to the end of the name).
   (if **Create Copy…** is unavailable, press **ctrl+B** in the program to store it. There should be no asterisk in front of the Python filename on the top of the Editor).
   This creates another Python Editor app containing the duplicate code in the document.



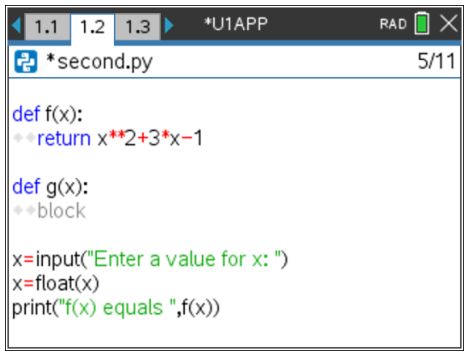
Teacher Tip: Python files are stored in the TI-Nspire document in which they are created. All Python files in the TI-Nspire document are available in *any* **Editor** in the document, *regardless* of the Problem number. When you insert a page (**ctrl+doc**) and select **Add Python** you can **Open** *any* Python file that exists in the document.

The Python **Shell**, however, *is* problem-based: Each problem has its own Shell environment. Restarting one Shell in a Problem resets *all* Shells in the problem.

3. Our new Python filename is **second.py**.
   Now add a second function template below the function f(x):
   On a blank line press **menu > Built-ins > Functions** and select **def function()**.
   Again, the syntax for the function structure includes a colon (**:**) at the end of the **def** line. This indicates that the following code is the definition of the function and the block is indented.



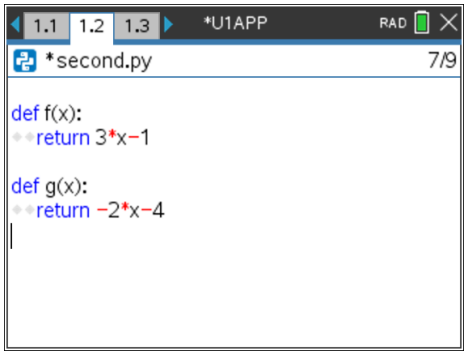
4. Name this second function **g**(x).



5. Change f(x) by removing **x**2+** from the function so that **f(x) = 3 * x - 1**.
   Define g(x):

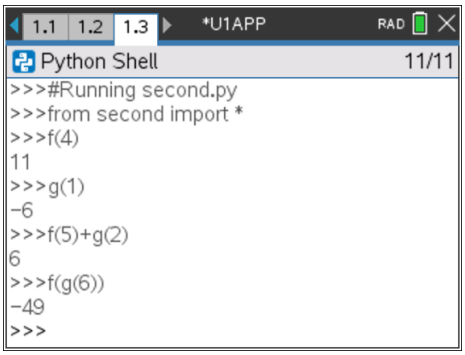
```
def g(x):
    return -2*x - 4
```

   Delete the 3 lines of code at the bottom of the program leaving only the two functions.



Teacher Tip: Working with two *linear* functions affords the opportunity to explore inverse functions later in this lesson.
Both negation and subtraction keys on the keypad do the same thing in the Python Editor: either negation or subtraction depending on the context.

6. Press **ctrl+R** and enter the expression f(1)+g(1).



Try other expressions using both functions like:

f(4)+g(1), f(5)+g(2), (f+g)(4), f(g(6))

Teacher Tip: To clear the Shell text (history), in the Shell press **menu > Tools > Clear History**.

(f+g)(x) is not a valid Python expression and creates an error message!

Shell histories in a TI-Nspire document are *not* saved with the document. Closing the document clears all Shell text. If you need to keep a copy of a Shell's history, select all the text (**ctrl+A, ctrl+C**) that you want to keep and paste it (**ctrl+V**) into a Notes app in the document.

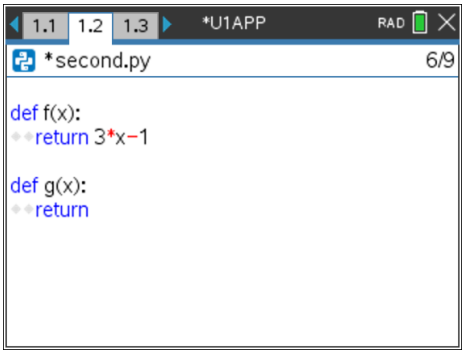
The next and final step might be too advanced for early learners as it involves function composition and inverse functions. But for students familiar with the topics it is a rewarding exercise.

7. **Other functions**

Are you ready for this? Back in the Editor, change **g(x)** so that

$$f(g(x)) = g(f(x)) = x$$

regardless of the value of x. Test your functions carefully.



Challenges: create a function that is parallel to or perpendicular to f(x).

Teacher Tip: Step 7 requires writing the inverse function of f(x). For the example cited, this is g(x)=(x + 1) / 3.