

Unit 1: Getting Started With Python

Skill Builder 3: Introducing the Python Function

In this lesson, you will define a function and use the function to evaluate expressions. You will experience the purpose of indentation in Python Editor.

Objectives:

- Define a function
- Use the function in evaluating expressions
- Use the input() function

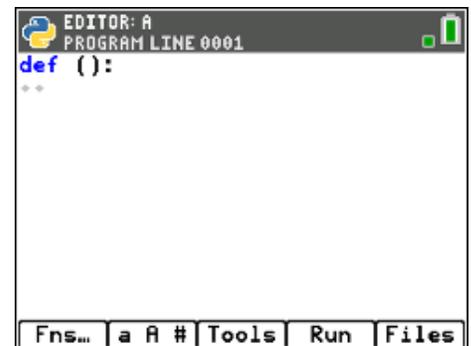
Functions play a big role in Python programming as in mathematics. Functions can be used to generate many different kinds of values, and functions can serve as Python subroutines which help to break a complex process into smaller, manageable parts.

A program is a step-by-step recipe for solving a problem: an **algorithm**. All programs are algorithms underneath.

1. Write a program that lets the user enter a number for x, and the program will use that value to evaluate the function $f(x)=x^2 + 3x - 1$.

Start a new blank Python program. Ours is simply called "A."

Define a function: Select **<Fns...>** **def function()**. The function *template* appears in the Editor. You must supply the *name* of the function, the *argument(s)* for the function and the *code* for the function block.



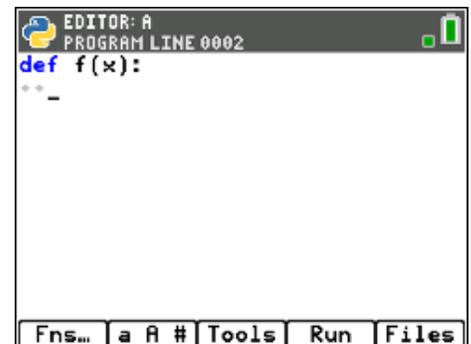
2. The name of the function goes to the left of the parentheses. Leave a space after the keyword **def**.

The argument(s) of the function go inside the parentheses. If there is more than one, use commas to separate them. There may also be no arguments.

The image shows the function defined as:

def f(x):

♦♦



Important tip: The colon (:) at the end of the line indicates that the next section is the function block and the next line is indented two spaces.

Indentation is critical in Python programs. This is how Python interprets function, loop, and **If** blocks. This Editor uses light gray diamond symbols (♦♦) to help with proper indentation.

3. With the cursor after the two diamonds, select **<Fns...>** and use the **return** statement.



TI-84 PLUS CE PYTHON

STUDENT ACTIVITY

Type the expression:

```
return x**2 + 3*x - 1
```

remember to use the [^] key for ** (the “raise to power” operation) and remember to use the multiplication sign in 3*x. 3x is an error.

We have finished defining the function. When you run this program, this function definition is not executed right away but the Shell “knows” that it exists.



4. Now write a “main program”:

Statements that allow the user to enter a number and produce (print) the function’s value for that number.

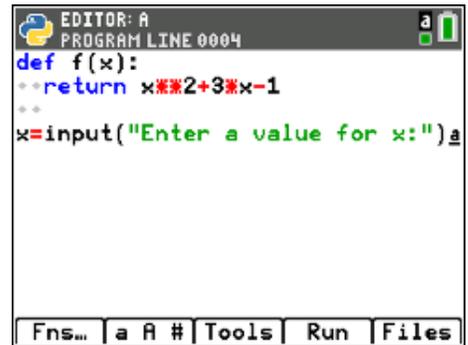
Press [enter] and [del] the indent spaces on your new line.

Create the statement:

```
x = input(“Enter a value for x: “)
```

Begin by typing the letter x and use [sto] for the “equals” sign.

Select <Fns...> and then right-arrow to the I/O menu (across the top line of the screen).



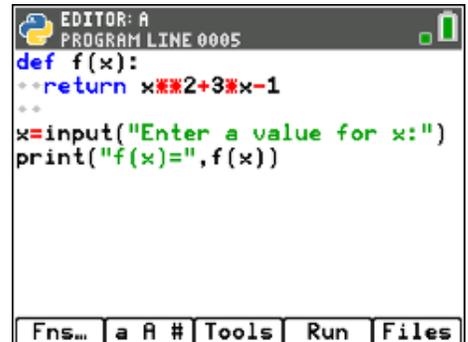
For the text in quotes, turn on alpha-lock and use the keypad to type your text. The quotation mark (”) is on the [+] key. [space] is on the [0] key, and the colon (:) is on the [.] (decimal) key.

5. Press [enter] after the closing parenthesis and, on the new line, write the statement:

```
print(“f(x) = “, f(x))
```

Recall that print() is on <Fns...> I/O.

The text gets tricky because you have to turn alpha-lock off to type the parentheses and = sign inside the quotes. Another option is to use the on-screen text selector found by selecting the menu <a A #>.



Note: Your print() statement actually contains f(x) twice: one in quotes (green) and one at the end (black). Both are typed by you, but the first one (green) is displayed as is. The second one, f(x), calls the function at the top of your code and processes the function’s expression to be displayed here.

- <Run> your program. At the prompt, enter a number for the variable **x** and press [enter].

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running A
>>> from A import *
Enter a value for x:|
Fns... | a A # | Tools | Editor | Files
    
```

- Welcome to (possibly) your first “runtime” error. The message is lengthy but the important part is in the last two lines. Note the last line number and the **TypeError: “can’t convert...”**.

The error happens because the **input()** function returns a string, not a numeric value. The programmer (that’s you!) must convert the string to a numeric value.

There are five simple types of data in Python: **int** (integer), **str** (string), **float** (numbers with decimals), **complex** (complex numbers) and **bool** (Booleans are either **True** or **False**).

```

PYTHON SHELL
Enter a value for x:5
Traceback (most recent call last
):
  File "<stdin>", line 1, in <mo
dule>
  File "A.py", line 5, in <modul
e>
  File "A.py", line 2, in f
TypeError: can't convert 'int' o
bject to str implicitly
>>> |
Fns... | a A # | Tools | Editor | Files
    
```

*Note: **True** and **False** are the only Python keywords that are **Capitalized**.*

- Fixing the error:** Return to <Editor>. There are several convert functions built into Python. They are found on <Fns...> **Type**. They are **int()**, **float()**, **str()** and **complex()**.

```

EDITOR: A
Func Ctl Ops List Type I/O Modul
1:int()
2:float()
3:round()
4:str()
5:complex(real, imag)
6:type()
Esc
    
```

9. To convert the string `x` to a number, right after the input statement on a new blank line write

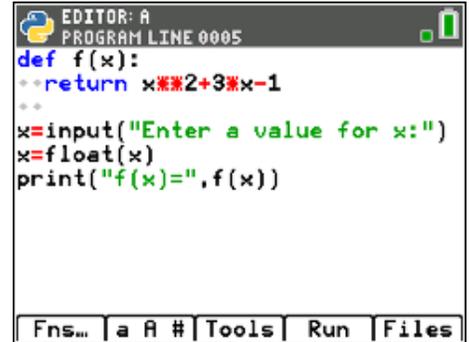
`x=float(x)`

or

`x=int(x)`

Try both and see the difference.

We chose **`float()`** in this example. Why?

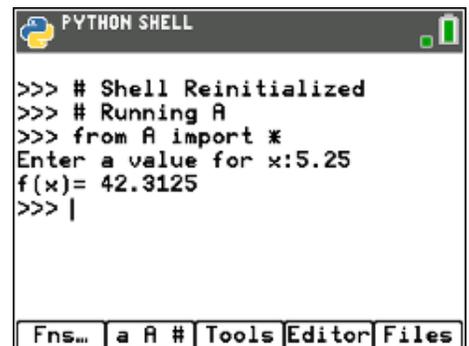


```

EDITOR: A
PROGRAM LINE 0005
def f(x):
    return x**2+3*x-1
x=input("Enter a value for x:")
x=float(x)
print("f(x)=",f(x))
    
```

10. **<Run>** the program again, and it will (should!) work as intended. Be sure to test your program with numbers for which you can confirm the calculation yourself!

Note: If it does not work, check your code! That's "debugging."

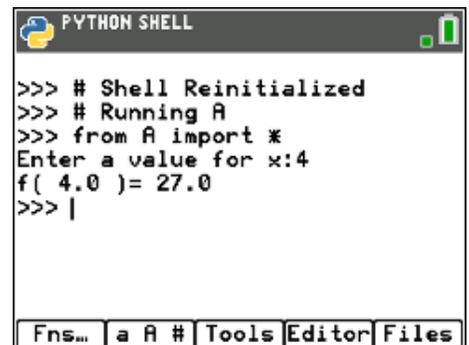


```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running A
>>> from A import *
Enter a value for x:5.25
f(x)= 42.3125
>>> |
    
```

11. **Challenge.** How would you change the code to get the value of `x` to appear inside the function's parentheses (instead of the letter `x`) like this:

`f(4.0) = 27.0`



```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running A
>>> from A import *
Enter a value for x:4
f( 4.0 )= 27.0
>>> |
    
```