input function in Python 2.7, evaluates whatever your enter, as a Python expression. If you simply want to read strings, then use raw\_input function in Python 2.7, which will not evaluate the read strings.

If you are using Python 3.x, raw\_input has been renamed to input. Quoting the [Python 3.0 release notes](https://docs.python.org/3.0/whatsnew/3.0.html#builtins),

raw\_input() was renamed to input(). That is, the new input() function reads a line from sys.stdin and returns it with the trailing newline stripped. It raises EOFError if the input is terminated prematurely. To get the old behavior of input(), use eval(input())

**In Python 2.7**, there are two functions which can be used to accept user inputs. One is [input](http://docs.python.org/2/library/functions.html#input) and the other one is [raw\_input](http://docs.python.org/2/library/functions.html#raw_input). You can think of the relation between them as follows

input = eval(raw\_input)

Consider the following piece of code to understand this better

>>> dude = "thefourtheye"

>>> input\_variable = input("Enter your name: ")

Enter your name: dude

>>> input\_variable

'thefourtheye'

input accepts a string from the user and evaluates the string in the current Python context. When I type dude as input, it finds that dude is bound to the value thefourtheye and so the result of evaluation becomes thefourtheye and that gets assigned to input\_variable.

If I enter something else which is not there in the current python context, it will fail will the NameError.

>>> input("Enter your name: ")

Enter your name: dummy

Traceback (most recent call last):

File "<input>", line 1, in <module>

File "<string>", line 1, in <module>

NameError: name 'dummy' is not defined

**Security considerations with Python 2.7's input:**

Since whatever user types is evaluated, it imposes security issues as well. For example, if you have already loaded os module in your program with import os, and then the user types in

os.remove("/etc/hosts")

this will be evaluated as a function call expression by python and it will be executed. If you are executing Python with elevated privileges, /etc/hosts file will be deleted. See, how dangerous it could be?

To demonstrate this, let's try to execute input function again.

>>> dude = "thefourtheye"

>>> input("Enter your name: ")

Enter your name: input("Enter your name again: ")

Enter your name again: dude

Now, when input("Enter your name: ") is executed, it waits for the user input and the user input is a valid Python function invocation and so that is also invoked. That is why we are seeing Enter your name again: prompt again.

So, you are better off with raw\_input function, like this

input\_variable = raw\_input("Enter your name: ")

If you need to convert the result to some other type, then you can use appropriate functions to convert the string returned by raw\_input. For example, to read inputs as integers, use the int function, like shown in [this answer](http://stackoverflow.com/a/20449433/1903116).

**In python 3.x**, there is only one function to get user inputs and that is called [input](http://docs.python.org/3/library/functions.html#input), which is equivalent to Python 2.7's raw\_input.