# Input and Output

## The input Function

The hello program of [The Classic First Program](http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/execution.html#the-classic-first) always does the same thing. This is not very interesting. Programs are only going to be reused if they can act on a variety of data. One way to get data is directly from the user. Modify the hello.py program as follows in the editor, and save it with File ‣ Save As....`, using the name hello\_you.py.

person = input('Enter your name: ')

print('Hello', person)

Run the program. In the Shell you should see

Enter your name:

Follow the instruction (and press Enter). Make sure the typing cursor is in the Shell window, at the end of this line. After you type your response, you can see that the program has taken in the line you typed. That is what the built-in function input does: First it prints the string you give as a parameter (in this case 'Enter your name: '), and then it waits for a line to be typed in, and returns the string of characters you typed. In the hello\_you.py program this value is assigned to the variable person, for use later.

The parameter inside the parentheses after input is important. It is a prompt, prompting you that keyboard input is expected at that point, and hopefully indicating what is being requested. Without the prompt, the user would not know what was happening, and the computer would just sit there waiting!

Open the example program, interview.py. Before running it (with any made-up data), see if you can figure out what it will do:

*'''Illustrate input and print.'''*

applicant = input("Enter the applicant's name: ")

interviewer = input("Enter the interviewer's name: ")

time = input("Enter the appointment time: ")

print(interviewer, "will interview", applicant, "at", time)

The statements are executed in the order they appear in the text of the program: sequentially. This is the simplest way for the execution of the program to flow. You will see instructions later that alter that natural flow.

If we want to reload and modify the hello\_you.py program to put an exclamation point at the end, you could try:

person = input('Enter your name: ')

print('Hello', person, '!')

Run it and you see that it is not spaced right. There should be no space after the person’s name, but the default behavior of the print function is to have each field printed separated by a space. There are several ways to fix this.

|  |  |
| --- | --- |
|  | The + operation on strings adds no extra space. |

## Print with Keyword Parameter sep

One way to put punctuation but no space after the person is to use the plus operator, +. Another approach is to change the default separator between fields in the print function. This will introduce a new syntax feature, keyword parameters. The print function has a keyword parameter named sep. If you leave it out of a call to print, as we have so far, it is set equal to a space by default. If you add a final field, sep='', in the print function

*'''Hello to you! Illustrates sep with empty string in print.*

*'''*

person = input('Enter your name: ')

print('Hello ', person, '!', sep='')