



Python Programming

Lesson 3

Let's Talk with the Program

Presented by Advaspire Team



User Input

Up to this point of lessons, you have assigned different values for variables in program. Now, we will level up with using input to control your program interactively in the terminal.

The image shows a login form with a red and blue border. On the left, there are three buttons: 'Login with Facebook' (blue), 'Login with Twitter' (light blue), and 'Login with Google+' (red). In the center, there is a vertical line with a circle containing the text 'OR'. On the right, under the heading 'Sign in manually', there are two input fields: 'Username or email' and 'Password'. Below these fields are a 'Remember me' checkbox and a red 'LOGIN' button. At the bottom right, there are links for 'Register now' and 'Forgot password?'.

Program input is what enable most program to receive and return a value to the user. It is kind of like calculator work, where we need to insert the equation for it to solve and return back an answer to us.

Another example that most of us have been using is the logging in to social media or online game in general. However, it have a bit different mechanics behind it but that will be explored in another lesson.



Accepting User Input

To use input, first we will need to create some variables. This will let the information received from the user being stored and can be used in the next block of code. Do bear in mind that the input will only be stored for that particular run only. Once you run it again, you will need to re-enter the info.

```
name = input("Enter your name: ")  
  
print("Hello," + name + ". Nice to meet you")
```

```
general use x  
C:\Users\haziq\PycharmProjects\pythonProject\venv>  
Enter your name: Jeff  
Hello,Jeff. Nice to meet you
```

Once variable is created, type the code like in the example here. The input is the function so it will be first outside the bracket. Inside the bracket will be the question or anything to tell the user to give feedback about. This part needs to be in string because you want it displayed.

Because the input functions using variable, you can print it just like when you were doing the concatenation or combine together strings and variables.



Print Multiple Statement

You can also create more than one input function at a time. You can be really creative with this one. Like in the example, there are two variable and I want to print 2 different concatenation. By doing it like that, you can print the combination one after another. The position for the code will determine which one will be printed first.

```
Name = input('Enter your name: ')
age = input('Enter your age: ')
print('Hello, ' + name + '!')
print('You are ' + age + ' years old!')
```

The input stored can also be used multiple time in a different line of code. Just make sure the variable names are correctly used.

```
name = input('Enter your name: ')
age = input('Enter your age: ')

print('Hello, ' + name)
print(name + " you are " + age + " years old")

general use x
C:\Users\haziq\PycharmProjects\pythonProject\
Enter your name: Haziq
Enter your age: 23
Hello, Haziq
Haziq you are 23 years old
```



Build a Simple Calculator

Now that we have learned how to use the input function, let's build a calculator. We are going to create three variable, 2 of which will request the user input, and the last one will contain the formula.

To start, create 2 variable called num1 and num2. The input will be 'Enter a number' and 'Enter another number'. In the 3rd variable called result, insert both num1 and num2 with your chosen operator like in the example. Once done, print the result variable and run to see what happen.

```
num1= input('Enter a number: ')\nnum2= input('Enter another number: ')\nresult= num1+num2\nprint(result)
```



Number in Python

Why did the calculator give the answer like that?

```
Enter a number:25
Enter another number: 25
2525
```

Number is one of many types of data we use in python. Because it is a data type, python will by default consider it as a string when you type it in the code.

That's what happen with our calculator just now. Because both number is considered a string, when the + is used, it became a concatenation.

So the solution so that it is calculated like a calculator is by converting the number from a string to integer.



Integer

What is integer?

Integer is basically a whole number, normal number like 1,3,-5,19 and so on. So by using the int function, it will convert the user input as an integer and not string. To use it, just add the int() with variable inside the bracket.

Now try running it again and see if it calculate correctly or not.

```
num1= input('Enter a number: ')
num2= input('Enter another number: ')
result= int(num1)+int(num2)
print(result)
```

```
Enter a number:25
Enter another number: 25
50
```

Right now we are just using whole number for the calculation. But what if you need to use number with decimal point like 3.14? Try it to find out what happen.



What happen?

This was what happened when we tried to use decimal numbers with the int function. Integer is a whole number, so it will be conflicted with the decimal number.

The only way to use decimal number in the calculator is by converting the input to floating number.

```
Enter a number: 2.5
Enter another number: 3.2
Traceback (most recent call last):
  File "C:\Users\haziq\PycharmProjects\pythonProject\main.py", line 3, in <module>
    result= int(num1)+int(num2)
ValueError: invalid literal for int() with base 10: '2.5'
```




Floating Number

What is floating number?

Floating number are numbers that have decimal point in it. Numbers like 3.14, 2.15848 and -2.686 are called floating number. In python, its just called as float.

```
num1= input('Enter a number: ')\nnum2= input('Enter another number: ')\nresult= float(num1)+float(num2)\nprint(result)
```

```
Enter a number: 3.14\nEnter another number: 2.32\n5.46
```

So what's the different between (float) and (INT)? They are actually similar but float can process both whole number and float, just for whole number they will add .0 at the end.

In general, use int only when the input or data you use involve whole number. If it involve both kind of number, just use float

```
Enter a number: 2\nEnter another number: 3\n5.0
```



Combine String with Number

Now that we have done our calculator, how about we change the answer section for a bit? Based on the previous code, the calculator will just directly give you the answer. What if you want to add some sentence before the answer? Let's try using concatenation like we learned previously.

```
num1 = input('Enter a number:')  
num2 = input('Enter another number: ')  
result = int(num1)+int(num2)  
print("The answer for that equation is : " + result)
```



Combine String with Number

Did an error occurred like in the example? This happens because in a concatenation, only strings are accepted. Because the answer in the calculator is already converted into float in the code, it cannot match with string. So now you need to convert the float back to string. But how?

```
Enter a number:25
Enter another number: 25
Traceback (most recent call last):
  File "C:\Users\haziq\PycharmProjects\pythonProject\general use.py",
    print ("The answer for that equation is : " + result)
TypeError: can only concatenate str (not "int") to str
```



Convert Back to String

There are 2 way to this. Both will require the use of string function, or str in python.

First method is by converting it to string directly inside the print block like in the example here.

```
num1 = input('Enter a number:')  
num2 = input(' Enter another number: ')  
result = int(num1)+int(num2)  
print("The answer for that equation is : " + str(result))
```

```
general use x  
C:\Users\haziq\PycharmProjects\pythonProject\venv\Scripts\py  
Enter a number:25  
Enter another number: 25  
The answer for that equation is : 50
```



Convert Back to String

Second method is by creating a variable to convert the 'result' variable to string. This is easy, just do it like in the example.

```
num1 = input('Enter a number:')  
num2 = input(' Enter another number: ')  
result = int(num1)+int(num2)  
answer = str(result)  
print("The answer for that equation is : " + answer)
```

Now you can make a concatenation with numbers!



Task

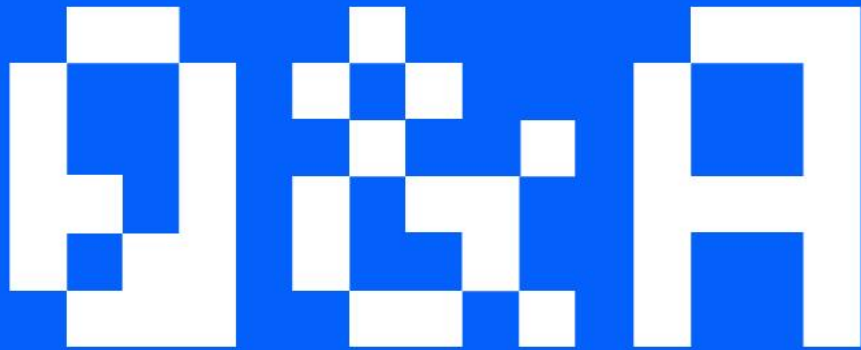
Using the Input function as our base, create a code that will ask the user questions which will then be combined to form a sentence.

1. Create 2 variable for the question. The theme for the question is hobby.
2. For print, write the concatenation for strings and variable so it will form a complete sentence when printed
3. Refer to the example for as reference for the sentence.

```
What's the weather? a bit cold
Where are you at? Semenyih
The weather is a bit cold in Semenyih
```



You can direct message your teacher and ask your question through [Slack Robotene Community](#) or arrange a [One-to-One Consultation](#) with your teacher.



Any Questions?



Thank you :)