

METHODS OF PYTHON

Variables

Using just literal constants can soon become boring - we need some way of storing any information and manipulate them as well. This is where *variables* come into the picture. Variables are exactly what the name implies - their value can vary, i.e., you can store anything using a variable. Variables are just parts of your computer's memory where you store some information. Unlike literal constants, you need some method of accessing these variables and hence you give them names.

Identifier Naming

Variables are examples of identifiers. *Identifiers* are names given to identify *something*. There are some rules you have to follow for naming identifiers:

- The first character of the identifier must be a letter of the alphabet (uppercase ASCII or lowercase ASCII or Unicode character) or an underscore (`_`).
- The rest of the identifier name can consist of letters (uppercase ASCII or lowercase ASCII or Unicode character), underscores (`_`) or digits (0-9).

- Identifier names are case-sensitive. For example, `myname` and `myName` are *not* the same. Note the lowercase `n` in the former and the uppercase `N` in the latter.
- Examples of *valid* identifier names are `i`, `name_2_3`. Examples of *invalid* identifier names are `2things`, `this is spaced out`, `my-name` and `>a1b2_c3`.

Data Types

Variables can hold values of different types called *data types*. The basic types are numbers and strings, which we have already discussed. In later chapters, we will see how to create our own types using classes.

Object

Remember, Python refers to anything used in a program as an *object*. This is meant in the generic sense. Instead of saying "the *something*", we say "the *object*".

Note for Object Oriented Programming users

NOTE

Python is strongly object-oriented in the sense that everything is an object including numbers, strings and functions.

We will now see how to use variables along with literal constants. Save the following example and run the program.

How to write Python programs

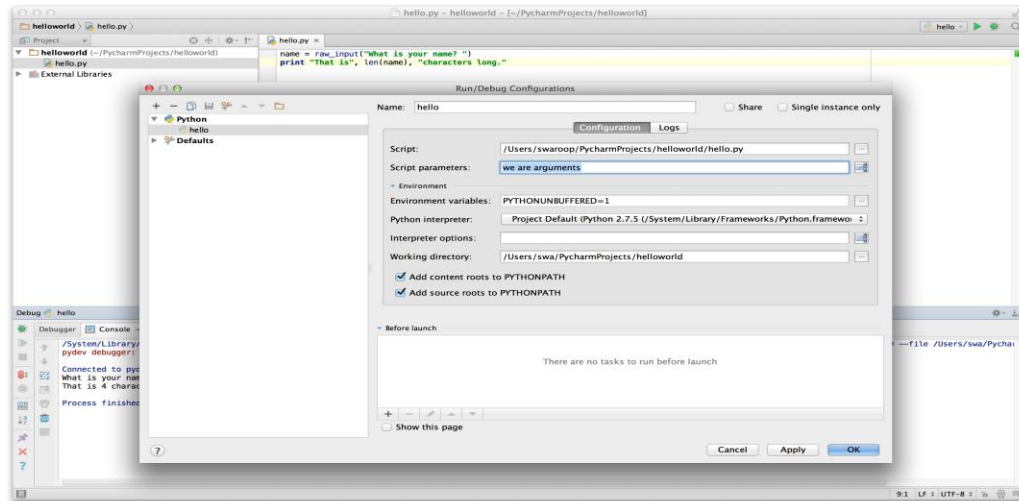
Henceforth, the standard procedure to save and run a Python program is as follows:

For PyCharm

1. Open PyCharm.
2. Create new file with the filename mentioned.
3. Type the program code given in the example.
4. Right-click and run the current file.

NOTE

Whenever you have to provide command line arguments, click on Run → Edit Configurations and type the arguments in the Script parameters: section and click the OK button:



Example: Using Variables And Literal Constants

Type and run the following program:

```
# Filename : var.py
```

```
i = 5
```

```
print i
```

```
i = i + 1
```

```
print i
```

```
s = "This is a multi-line string.
```

```
This is the second line."
```

```
print s
```

Output:

```
5
6
This is a multi-line string.
This is the second line.
```

How It Works

Here's how this program works. First, we assign the literal constant value 5 to the variable `i` using the assignment operator (`=`). This line is called a statement because it states that something should be done and in this case, we connect the variable name `i` to the value 5. Next, we print the value of `i` using the print statement which, unsurprisingly, just prints the value of the variable to the screen.

Then we add 1 to the value stored in `i` and store it back. We then print it and expectedly, we get the value 6.

Similarly, we assign the literal string to the variable `s` and then print it.

Note for static language programmers

NOTE Variables are used by just assigning them a value. No declaration or data type definition is needed/used.

Source: <http://www.swaroopch.com/notes/python/>