# Understand modules in python to increase your code’s productivity and readability

# Overview

In python, we have modules that are files with a .py extension. It contains functions/classes/variables that we may want to use again and again. So instead of redefining them for the umpteenth time in our program files, we use modules.

# Scope of article

1. The article defines modules and how to use them
2. We will learn different ways to import a module
3. We will also learn some important built-in modules

## Table of content:

1. What are modules in python
2. Types -
   1. Built-in
   2. User-defined
3. Create a module
4. How to use a module
5. Variables in module
6. The import statement
7. How to import modules
8. Python module search path
9. Naming a module
10. Re-naming a module
11. Reloading a module
12. The dir() built-in function
13. Examples

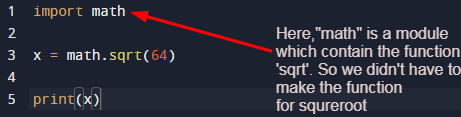
## What are modules in python?

### **TLDR: Modules are files with extensions .py. They can contain functions,class,variables,etc.**

Similar to a modular kitchen that is divided into small compartments, an efficient program in python consists of modules. **Modules are handleable units of code written in a separate python file.** This code can include functions,class,variable,etc.

After the module performs its task, the control return back to the calling statement.

They are also called libraries. Modules make the programs less complex, reduce redundancy, and increase readability.



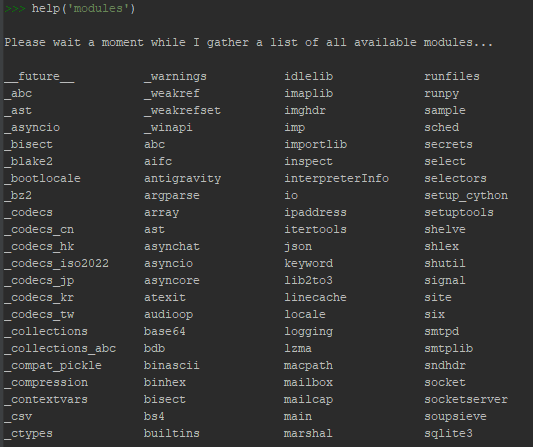
## Types of modules in python

**TLDR: Types of modules in python are built-in and user-defined.Some Important built-in modules are math. py,cmath.py, random.py, and datetime.py.**

In python modules are divided into two types:

1. Built-in modules
2. User-defined modules

Built-in modules: Modules that are pre-available in python are known as built-in modules. They are also known as standard library modules. Here is the list of some built-in modules of python. They need to be imported before use. To import them, write keyword **import** followed by module name.



Here are some important built-in modules in python with their definitions



User-defined modules: These modules are created by the users. User should name their modules while adhering to the following suggestions:

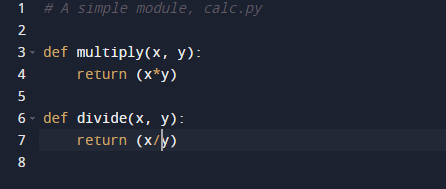
1. It should be short and, lowercased.
2. Avoid using special symbols like( \_,?,., etc). If you name your module “my.module.py” the python expects to find the module.py file in the folder named “my” which is not the case.

## 

## Create a module

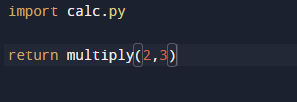
To create a module in python write the code in a file with a .py extension.

Example: Here we have created a module named “calc.py” which returns the multiplication and division of two numbers.



## How to use a module

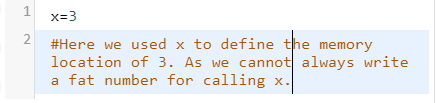
To use the module file use the keyword “import” followed by the module name



## Variables in module

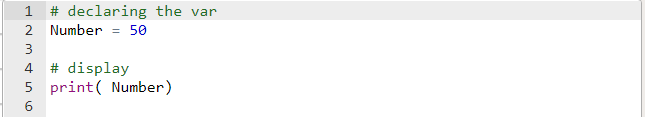
Variables in python tell the computer about the memory location.

Module in python can consist of functions and variables. These variables can be of type arrays, dictionary, objects, etc.



In python, we don’t need to define the datatype of variable used

To declare a variable simply put your variable name which will store the value in L.H.S and enter the value that you want to store in R.H.S.



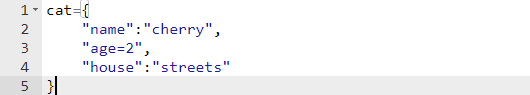
## Rules for naming a variable in python:

1. Variable name must start with a letter or underscore.
2. A variable name cannot begin with a number
3. It can only contain alphanumeric characters or an underscore
4. VAR, var, Var are three different variables as variables in python are case sensitive
5. Keywords reserved in python cannot be used to name the variable.

### How to use variables in modules

Example

Save this code in the file as mymodule.py



Import the module mymodule.py and access the cat dictionary:

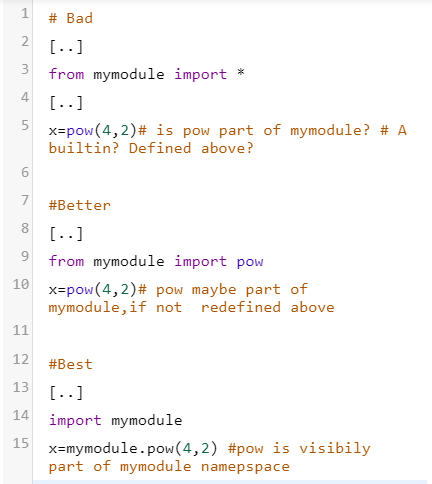


## 

## 

## The import statement

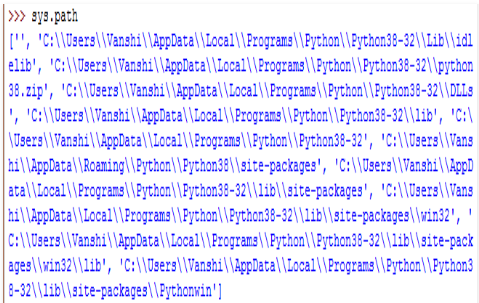
TLDR:



Python import statement allows you to import a module into your code. They are similar to #include header files in C/C++.

When the python interpreter comes across an import statement, it imports the module if it finds the module in the search path.

Here search path means the list of directories that the interpreter searches through before importing the modules.



The above image shows the list of directories that the interpreter went through before importing the module.

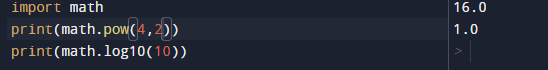
## How to import a module

Following are the ways to import a module:

To import the entire module:

Use statement import *module name*

Example



The import math statement will make all the functions under the math module available to the program

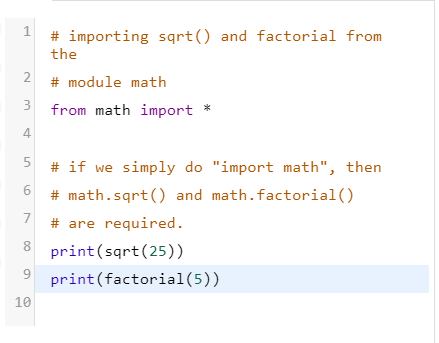
Here we are accessing the module “math”

The functions pow() and log() are defined in the math module. Dot operator is used here

To import the entire module you can also use

Use statement from module\_name import \*

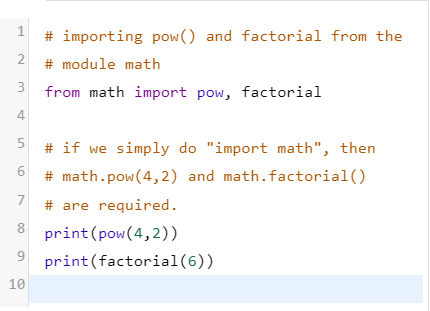
from module\_name import\* way is generally not used by the coders as it makes it harder to track down a problem. Because we won’t be able to know whether the function is imported and where it came from.

****

Here, no dot operator has been used as done in the former. We only wrote function names before arguments.

To import specific functions/variables/classes

Use statement from module\_name import function\_name/class\_name/variables\_name



Here also we didn’t use the dot operator. And simply called the mathematical functions with arguments.

## 

## Python module search path

By default python interpreter searches through the following directories when a module is imported:

1. Current directory
2. Environment variable path
3. Directories installed from 3rd parties

The interpreter will throw an error if the imported module is not found in any of the above directories.

If you do not want to put your module file in the current directory then you can find the path by going to advanced system *<setting><environment variables><path>*. You can add your module file in that path also

To find the module path write in your code

*sys(module\_name)*

## Naming a module

User should name their modules while adhering to the following suggestions:

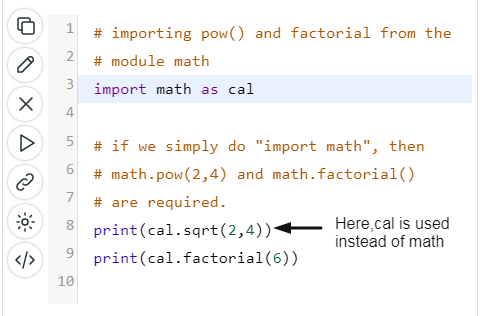
1. It should be short and, lowercased.
2. Avoid using special symbols like( \_,?,., etc). If you name your module “my.module.py” the python expects to find the module.py file in the folder named “my” which is not the case.

## Re-naming a module

If the module name is too long then we can use the keyword as to rename the module.

Syntax :import module\_initial name as module\_updated name

Example

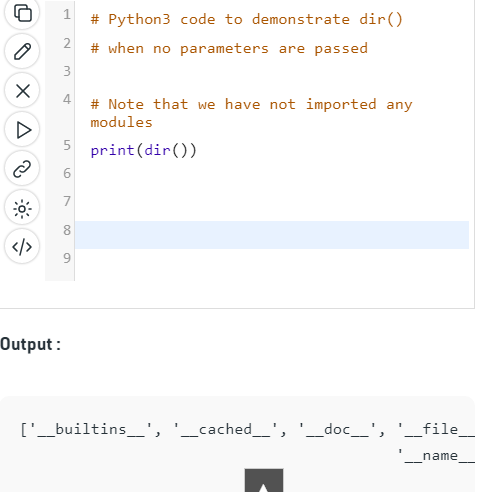


## The dir() built in function

*dir([object])*

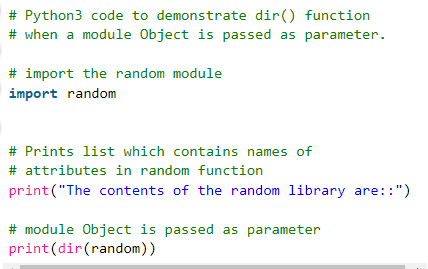
The dir() function returns attributes & values in local scope.

If it is used without any arguments, then it returns all the attributes and values in the local scope.



If a module is passed in the arguments then it will return everything that exists inside that module.

Example

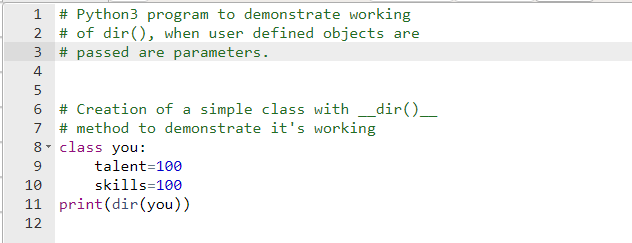


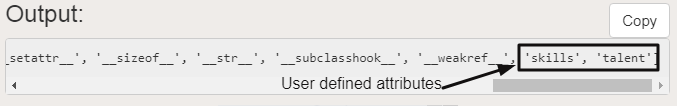


To see what attributes exist in a user-defined class.

Create a class and then attributes in it

Write dir(class\_name) to see the list of attributes present in the class.





## Summary