Python for Class XII _ - Part 14

Topics:
Using Python libraries: create and import Python libraries.
(Continued)

Accessing Modules from Another Directory

Modules may be useful for more than one programming project, and in that case it makes less sense to keep a module in a particular directory that's tied to a specific project.

If you want to use a Python module from a location other than the same directory where your main program is, you have a few options.

Appending Paths

One option is to invoke the path of the module via the programming files that use that module. This should be considered more of a temporary solution that can be done during the development process as it does not make the module available system-wide.

To append the path of a module to another programming file, you'll start by importing the `sys` module alongside any other modules you wish to use in your main program file.

The `sys` module is part of the Python Standard Library and provides system-specific parameters and functions that you can use in your program to set the path of the module you wish to implement.

For example, let's say we moved the `circle.py` file and it is now on the path `c:\PythonProgs\modules` while the `main_v4.py` file is in another directory. In our `main_v4.py` file, we can still import the `hello` module by importing the `sys` module and then appending `c:\PythonProgs\modules` to the path that Python checks for files: `sys.path.append('C:\PythonProgs\modules')`

```
# main_v4.py
import sys
sys.path.append('C:\PythonProgs\modules')

# import module circle
import circle

r = float(input("Enter the radius of the circle: "))
print("Area of the circle(in sq. units) = %.2f" %circle.area(r))
print("Circumference of the circle = %.2f" %circle.circumference(r))
```

Run:
python main_v4.py

OUTPUT:
Enter the radius of the circle: 8.4
Area of the circle(in sq. units) = 221.67
Circumference of the circle = 52.78

As long as you correctly set the path for the `circle.py` file, you'll be able to run the `main_v4.py` file without any errors and receive the same output as above when `circle.py` was in the same directory.

Adding the Module to the Python Path

A second option that you have is to add the module to the path where Python checks for modules and packages. This is a more permanent solution that makes the module available environment-wide or system-wide, making this method more portable.

To find out what path Python checks, run the Python interpreter from your programming environment:

```
python
```

Next, import the `sys` module:

```
import sys
```

Then have Python print out the system path:

```
print(sys.path)
```

Here, you'll receive some output with at least one system path. If you're in a programming environment, you may receive several. You'll want to look for the one that is in the environment you're currently using, but you may also want to add the module to your main system Python path. What you're looking for will be similar to this:

$ python

Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import sys

>>> print(sys.path)

['', 'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38\\python38.zip', 'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38\\DLLs', 'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38\\lib', 'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38', 'C:\\Users\\Vinod Nayak\\AppData\\Roaming\\Python\\Python38\\site-packages', 'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38\\lib\\site-packages']

>>>

Here is the directory we are searching for:

'C:\\Users\\Vinod Nayak\\AppData\\Local\\Programs\\Python\\Python38\\lib\\site-packages'

(Note: it will be different on different installation)

Now you can move your circle.py file into that directory. Once that is complete, you can import the `circle` module as usual:

```
# main_v4.py
# import module circle
import circle
```

```
r = float(input("Enter the radius of the circle: "))
print("Area of the circle(in sq. units) = %.2f" %circle.area(r))
print("Circumference of the circle = %.2f" %circle.circumference(r))
```

When you run your program, it should complete without error.

```
$ python main_v4.py
Enter the radius of the circle: 7
Area of the circle(in sq. units) = 153.94
Circumference of the circle = 43.98
```

Modifying the path of your module can ensure that you can access the module regardless of what directory you are in. This is useful especially if you have more than one project referencing a particular module.