

Import Python Module by String Name

Table of Contents

- Introduction
- Use `importlib` to programmatically import modules
- Conclusion

Introduction

In Python, when building a framework or code generation tool, you might need to programmatically build a string representing the name of a module you want to import. How would you import a module if you have its name as a string? Here's how to do it with [importlib](#)! Note that this feature was introduced in Python 3.1 but there is a 2.7 version of importlib though less featured.

Use importlib to programmatically import modules

The **importlib** provides a simple **import_module** function that accepts a string name with dot separators. It works just like a normal import except it uses a string to store the name. On import, the file is executed and the module object is returned. If you only need the code run on import it is not necessary to store the returned module.

```
import importlib

# Contrived example of generating a module named as a string
full_module_name = "mypackage." + "mymodule"

# The file gets executed upon import, as expected.
mymodule = importlib.import_module(full_module_name)

# Then you can use the module like normal
mymodule.func1()
mymodule.func2()
```

Conclusion

You should now know how to dynamically import modules using a module name stored in a string.