Difference between Module and Package

All Versions

Examples

Modules

A module is a single Python file that can be imported. Using a module looks like this:

`module.py`

```
def hi():
    print("Hello world!")
```

`my_script.py`

```
import module
module.hi()
```

in an interpreter

```
>>> from module import hi
>>> hi()
# Hello world!
```

Packages

A package is made up of multiple Python files (or modules), and can even include libraries written in C or C++. Instead of being a single file, it is an entire folder structure which might look like this:

Folder `package`

- `__init__.py`
- `dog.py`
- `hi.py`

`__init__.py`

```
from package.dog import woof
from package.hi import hi
```

`dog.py`

```
def woof():
    print("WOOF!!!")
```

`hi.py`

```
def hi():
    print("Hello world!")
```

All Python packages must contain an `__init__.py` file. When you import a package in your script (`import package`), the `__init__.py` script will be run, giving you access to the all of the functions in the package. In this case, it allows you to use the `package.hi` and `package.woof` functions.

Syntax

Parameters

Remarks

It is possible to put a Python package in a ZIP file, and use it that way if you add these lines to the beginning of your script:

```
import sys
sys.path.append("package.zip")
```