

modules

5.2.1. Regular packages

Python defines two types of packages, [regular](#) packages and [namespace](#) packages. Regular packages are traditional packages as they existed in Python 3.2 and earlier. A regular package is typically implemented as a directory containing an `__init__.py` file. When a regular package is imported, this `__init__.py` file is implicitly executed, and the objects it defines are bound to names in the package's namespace.

PEP 420 – Implicit Namespace Packages

`__init__.py`

It is required to make Python treat directories as packages. It can be simply empty file. This file is always executed when any part of module is imported.

`__main__.py`

Most commonly, the `__main__.py` file is used to provide a command-line interface for a package. `__main__.py` will be executed when the package itself is invoked directly from the command line using the `-m` flag. https://docs.python.org/3/library/__main__.html#main-py-in-python-packages

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

<https://niziak.spox.org/wiki/programming:python:modules>

Last update: **2024/02/23 20:15**

