

python

python ~

py

python

pyc

bytes code

import python py pyc import

pyo

python -O pyc pyo -O

*-O : optimize generated bytecode slightly; also PYTHONOPTIMIZE=x
-OO : remove doc-strings in addition to the -O optimizations*

python optimizer assert -O .pyc optimize bytes code -OO __doc__

python pyc pyo python

python pyc pyo

pyd

windows dll unix so pyd dll

*Is a *.pyd file the same as a DLL?*

Yes, .pyd files are dll's, but there are a few differences. If you have a DLL named `foo.pyd`, then it must have a function `PyInit_foo()`. You can then write Python "import foo", and Python will search for `foo.pyd` (as well as `foo.py`, `foo.pyc`) and if it finds it, will attempt to call `PyInit_foo()` to initialize it. You do not link your .exe with `foo.lib`, as that would cause Windows to require the DLL to be present.

Note that the search path for `foo.pyd` is `PYTHONPATH`, not the same as the path that Windows uses to search for `foo.dll`. Also, `foo.pyd` need not be present to run your program, whereas if you linked your program with a dll, the dll is required. Of course, `foo.pyd` is required if you want to say `import foo`. In a DLL, linkage is declared in the source code with `__declspec(dllexport)`. In a .pyd, linkage is defined in a list of available functions.

- <https://stackoverflow.com/questions/8822335/what-do-the-python-file-extensions-pyc-pyd-pyo-stand-for>