

Let's say you have a library that may be called from python 2.x or 3.x, but you don't know which one you'll have to use yet. And obviously let's say there is NO way to make a lib compatible with both (because of metaclasses for example). You have a directory structure thus laid out: `<code>. +--libs2x | +-- mylib | +-- __init__.py +--libs3x | +-- mylib | +-- __init__.py +--my_module | +-- __init__.py </code>` In `"%%./my_module/__init__.py%%"`, you can use the following Python code to import the right lib depending on the Python version: `<code python>from __future__ import print_function import sys, os if sys.version_info.major == 3: dir_name = "libs_3x" elif sys.version_info.major == 2: dir_name = "libs_2x" else: raise ImportError lib_path = os.path.dirname(__file__) # Start from the file and lib_path = os.path.dirname(__path__) # go up two dirs. Do whatever you have to do to get to the libs path. lib_path = os.path.join(lib_path, dir_name) # Go into the folder corresponding to the right version sys.path.append(lib_path) import mylib # DO SOMETHING</code>` You can then import mylib from python 2 or 3.

From:

<https://les-fees-speciales.coop/wiki/> - **Les Fées Spéciales**

Permanent link:

https://les-fees-speciales.coop/wiki/lib_compatibility

Last update: **2015/08/20 09:56**

