
Trick: importing from a path

- ❖ There is no path-based ``import`` (e.g. ``require_relative``)
- ❖ Useful in a few cases:
 - ❖ Importing “plugins” from a folder
 - ❖ Testing systems to control versions imported
 - ❖ Importing from neighboring modules without install
 - ❖ Useful in ``__main__.py`` to get own module loaded

Trick: importing from a path

- ❖ Inline code:

```
import importlib.machinery
import sys

spec = importlib.machinery.PathFinder.find_spec('foo', ['/tmp'])
foo = importlib.util.module_from_spec(spec) # make it a module
spec.loader.exec_module(foo) # let the module run
sys.modules['foo'] = foo # make future imports sane
```

- ❖ `importlib.util.spec_from_file_location` also works
 - ❖ It requires knowing packages from files (`__init__.py`)
- ❖ This is different in Python2.x