Namespace and Executable Packages

Austin Bingham
COFOUNDER - SIXTY NORTH
@austin_bingham sixty-north.com

Overview

Introduce namespace packages

Demonstrate executable directories

Execute code from zipped directories

Make executable packages

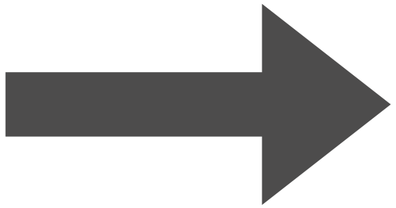
Namespace Packages

Packages in Multiple Directories

Multiple Distinct Directories

```
path1/
  package_name/
    module_a.py
```

```
path2/
  package_name/
    module_b.py
```



Imported as

Single Logical Package

```
package_name
```

```
package_name.module_a
```

```
package_name.module_b
```

PEP 420: Implicit Namespace Packages

"Namespace packages are a mechanism for splitting a single Python package across multiple directories on disk."

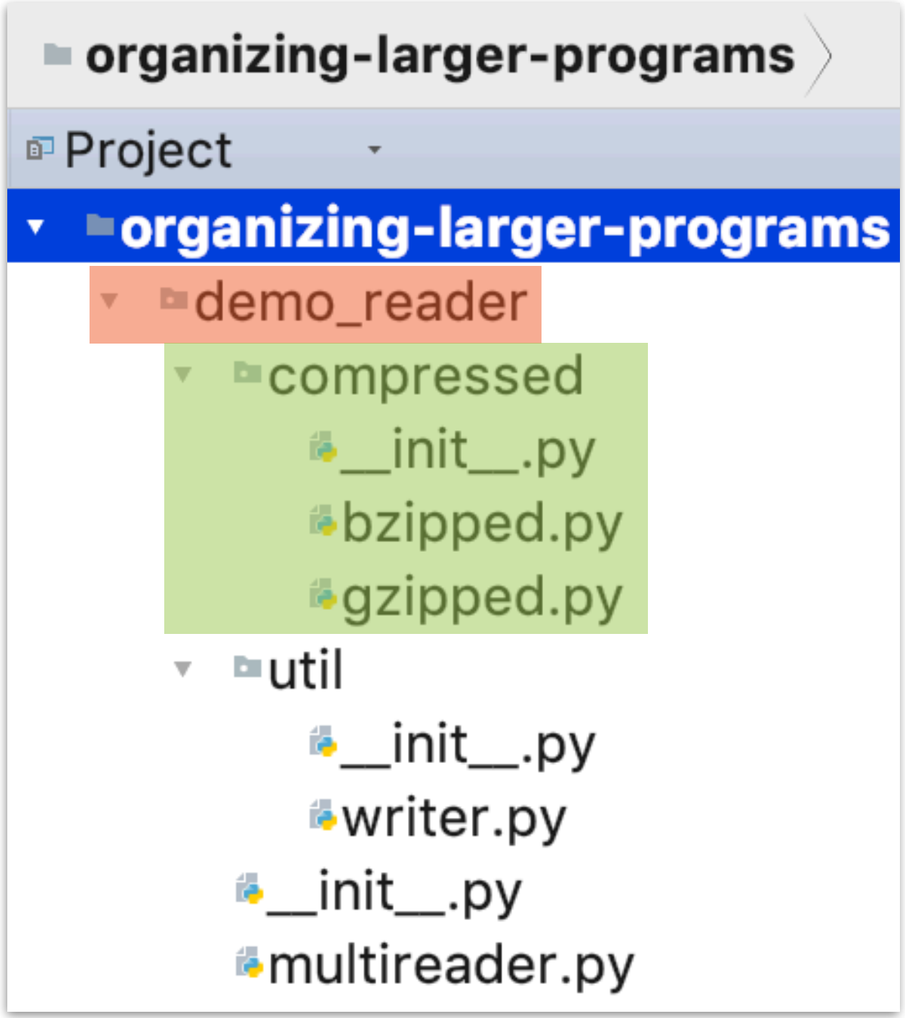
python.org/dev/peps/pep-0420/

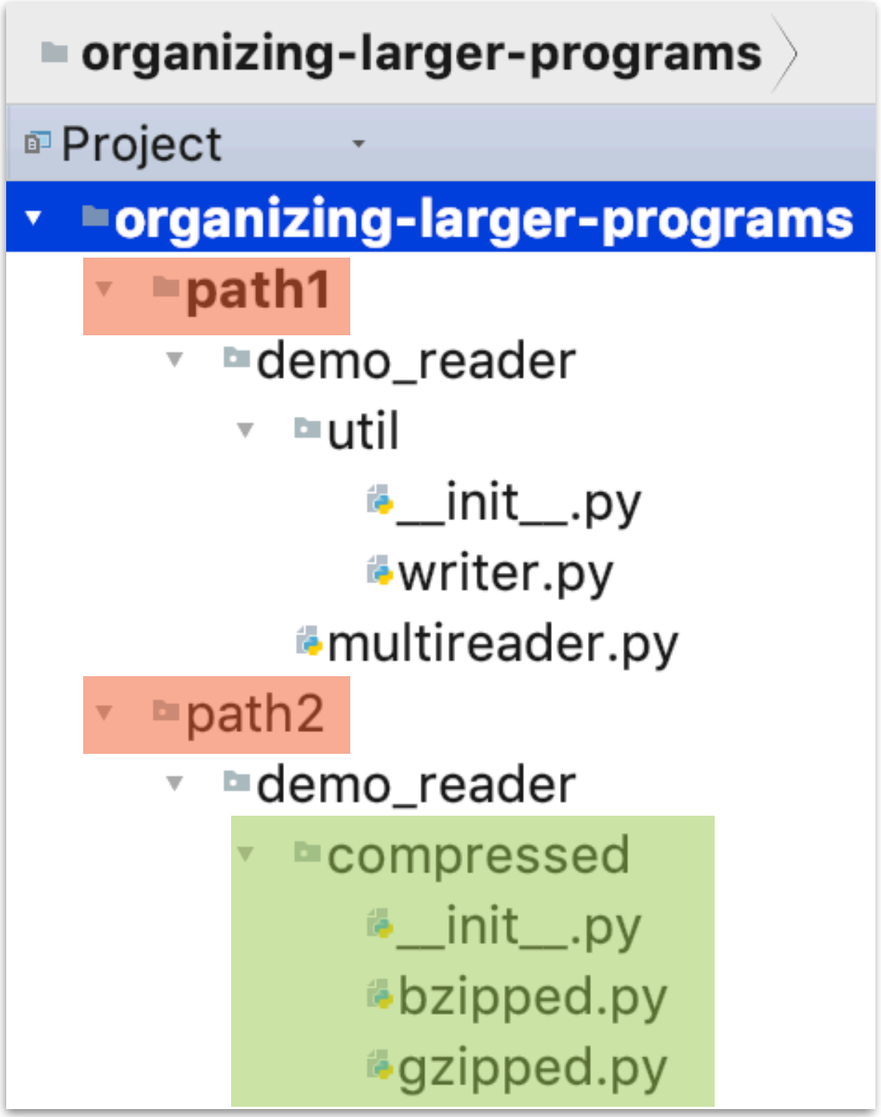
Namespace packages may not have `__init__.py`

Namespace Package Discovery Algorithm

- **Scan each directory in `sys.path`**
- **Import standard package if found**
- **Import standard module if found**
- **Otherwise, all matching directories contribute to a namespace package**

Conversion to Namespace Package





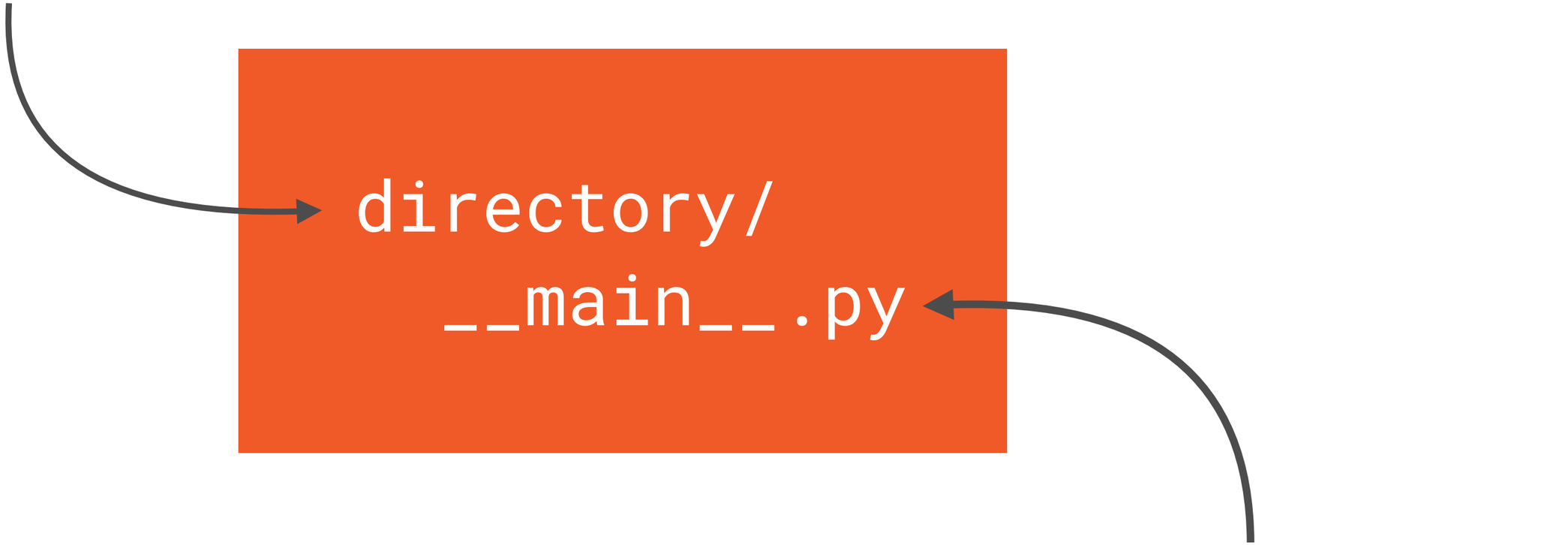
Executable Directories

You can execute a directory if it contains `__main__.py`

`__main__.py` will be executed

Using `__main__.py` and `sys.path`

1. Added to `sys.path`



2. Executed by “`python directory`”

Using __main__.py and sys.path

1. Added to sys.path

2. Executed by “python program”

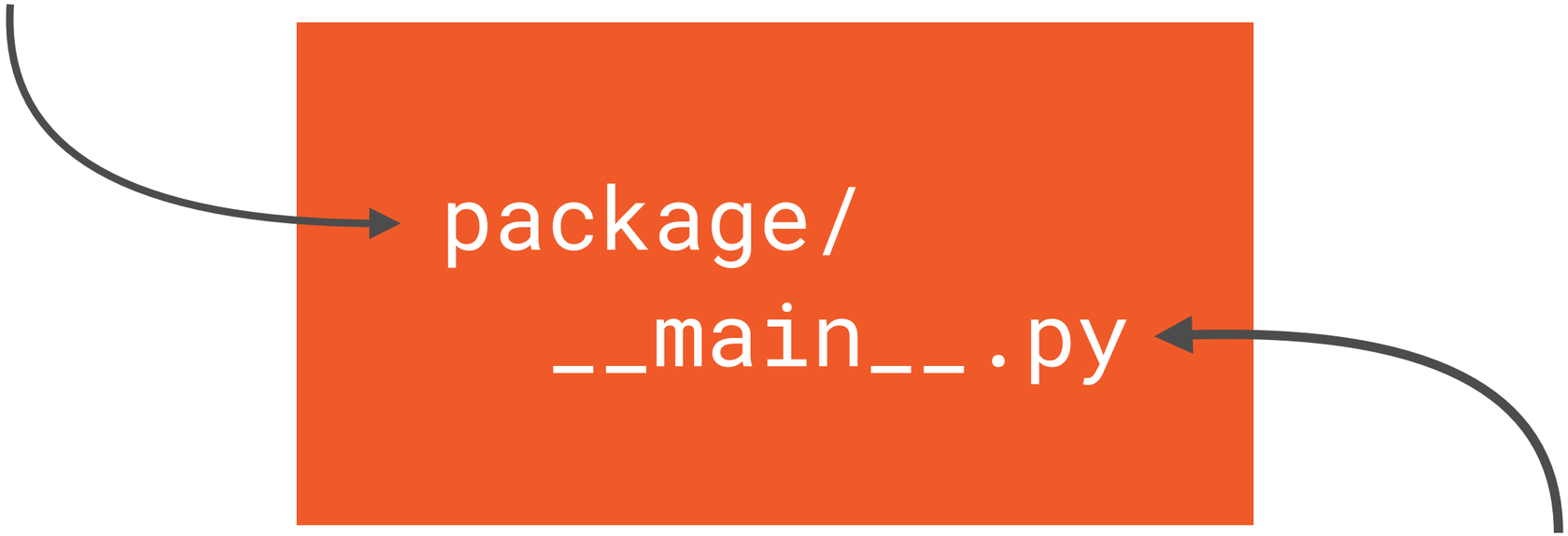
Executable Zip Files

The zip file contains the same contents as the directory, *not the directory itself*

Executable Packages

Using `__main__.py` in Packages

1. Must be accessible as package



2. Executed by “`python -m package`”

Executing Directories vs. Packages

`python directory`

Executing a *directory*

“`directory`” added to `sys.path`

“`directory/__main__.py`” is not in a package

`python -m directory`

Executing a *package*

- The “-m” tells Python to treat it as a module

“`directory`” treated as a package

“`directory/__main__.py`” is a submodule of the `directory` package

`__init__.py` vs. `__main__.py`

`__init__.py` can execute any code it likes on import . . .

. . . but only a package with `__main__.py` can be executed

Summary

Construct packages from multiple directories with *namespace packages*

Namespace packages cannot contain `__init__.py`

Directories can be made executable with `__main__.py`

Python can execute zip files like directories

Packages can be both importable and executable with `__main__.py`