Core Python: Organizing Larger Programs

NESTING MODULES WITH PACKAGES

Austin Bingham

COFOUNDER - SIXTY NORTH

@austin_bingham   sixty-north.com

Overview

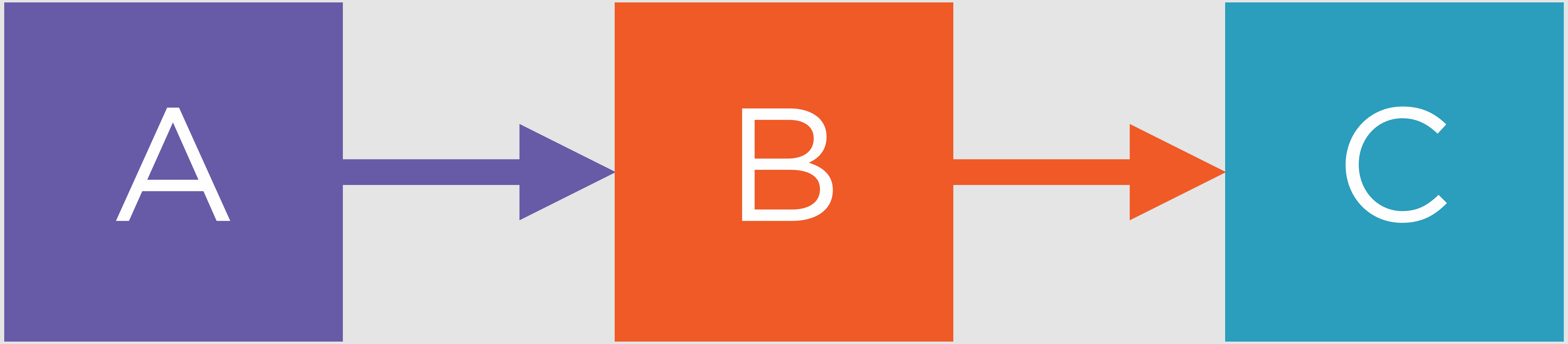
Review prerequisites

Imported nested packages

Packages are modules

Locating modules for import

Modularization



Prerequisites

Single-file Module Topics

Creating

Importing

Executing

Editing

Module-level attributes

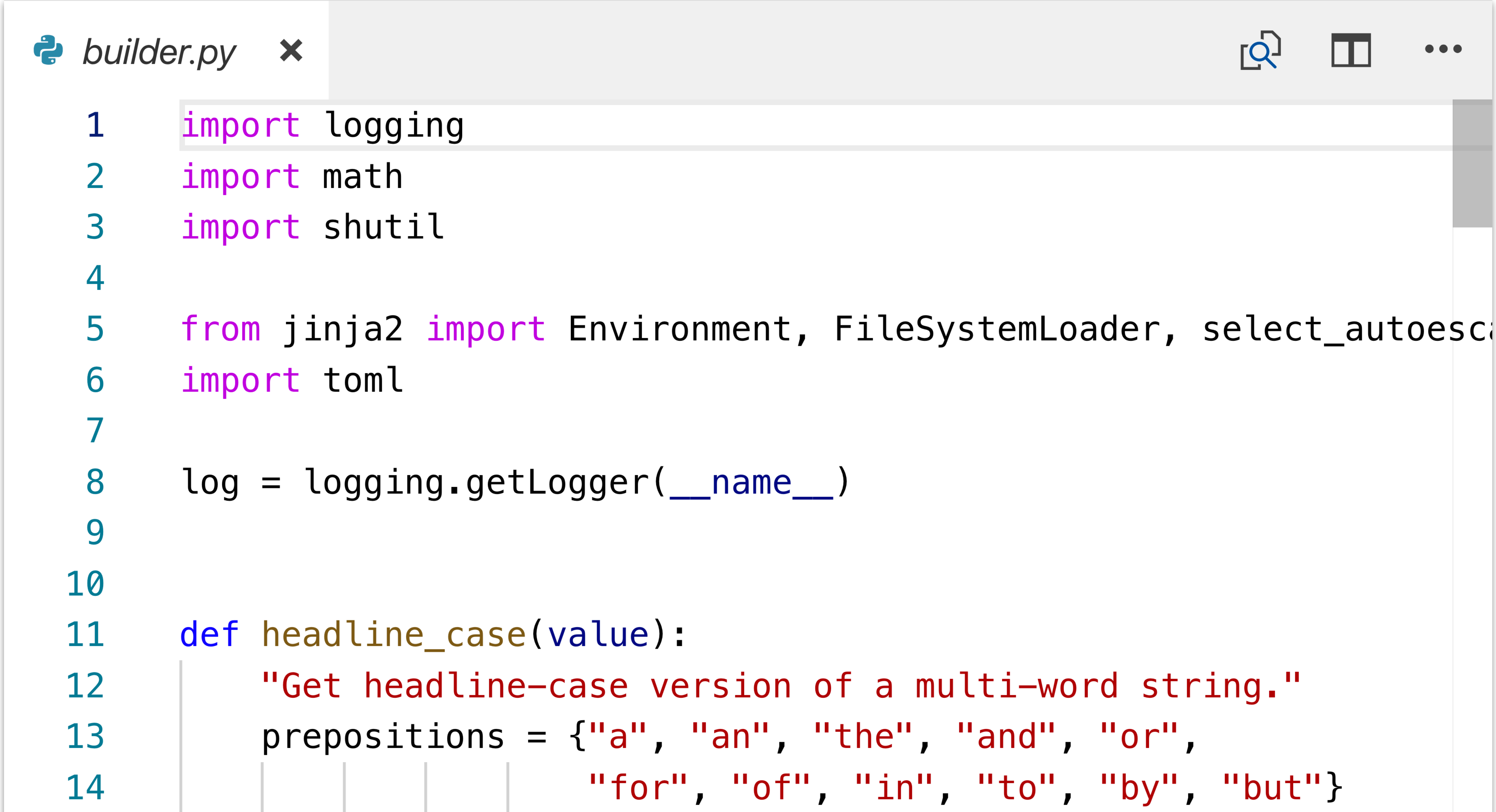
Importing Modules

```
import x

from x import y

from x import y as z
```

Create Python Modules



```python
import logging
import math
import shutil

from jinja2 import Environment, FileSystemLoader, select_autoesca
import toml

log = logging.getLogger(__name__)


def headline_case(value):
    "Get headline-case version of a multi-word string."
    prepositions = {"a", "an", "the", "and", "or",
                    "for", "of", "in", "to", "by", "but"}
```

Main block

```python
def main():
    "The main function for the program."
    return 42


# This is the "main block"
if __name__ == '__main__':
    main()
```

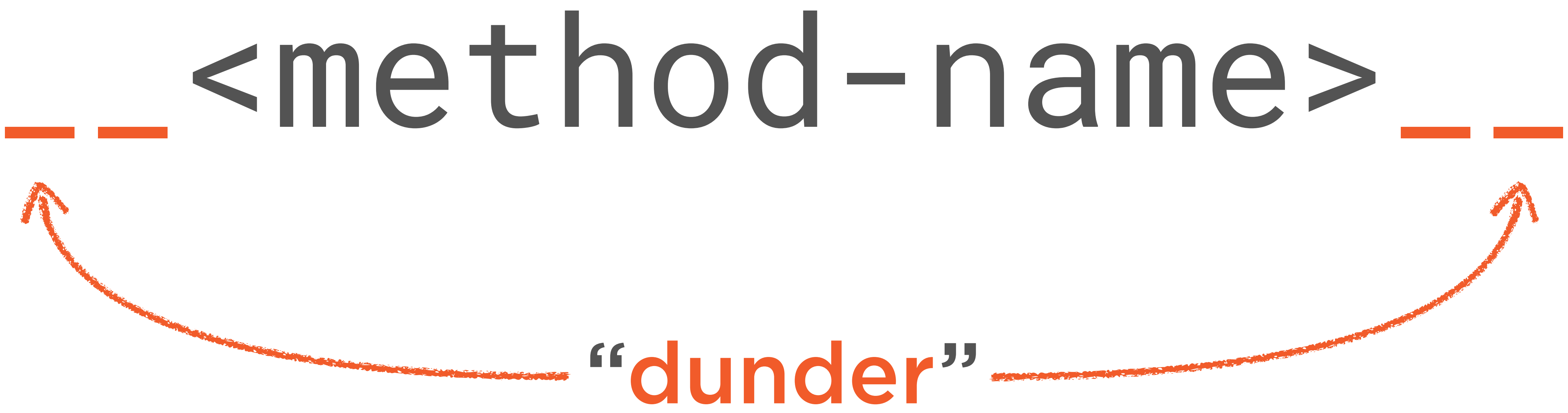
Python Fundamentals
on
Pluralsight

Working Python 3 Environment

python.org/downloads

```
>>> print(‘Hello, Programmer!’)
Hello, Programmer!
>>> import sys
>>> assert sys.version_info[0] == 3
```

Terminology for Python Special Methods



__<method-name>__

“dunder”

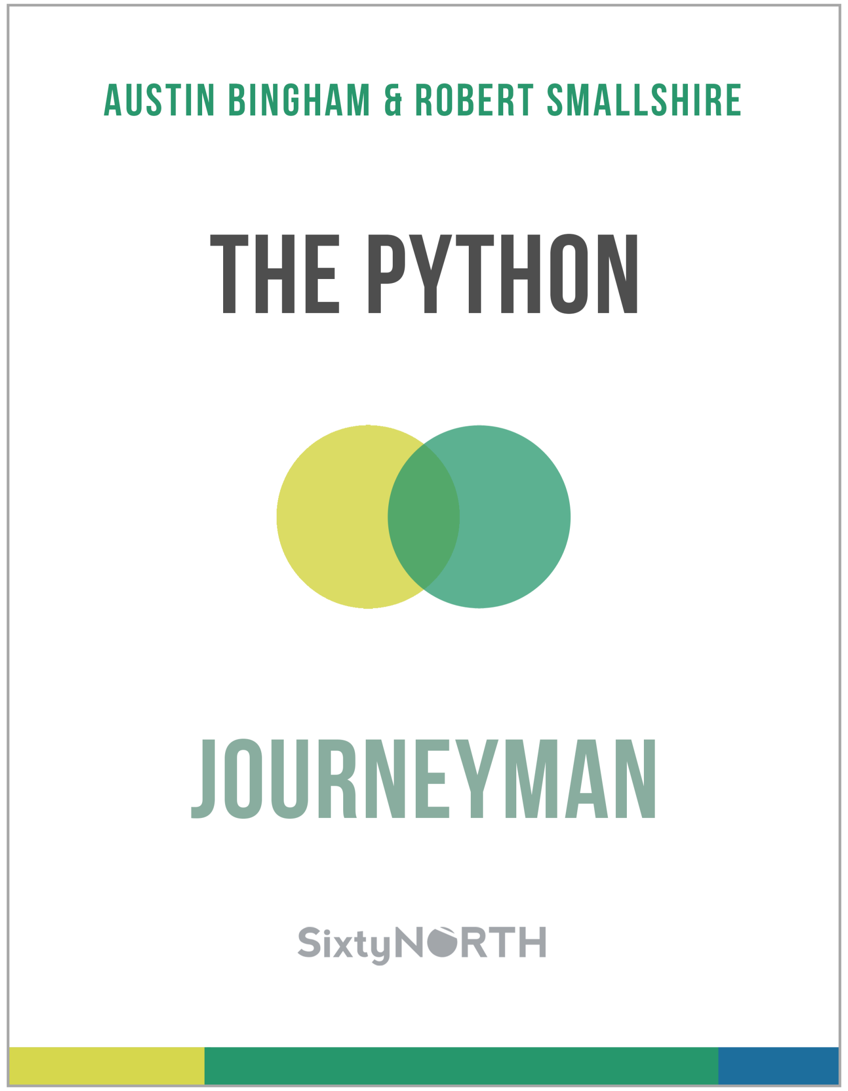
“double underscore”

Terminology for Python Special Methods

```
__len__
```

“dunder len”

Companion Python Craftsman Book Series



AUSTIN BINGHAM & ROBERT SMALLSHIRE

THE PYTHON

JOURNEYMAN

SixtyNORTH

https://leanpub.com
/python-journeyman
/c/pluralsight

Companion Python Craftsman Book Series

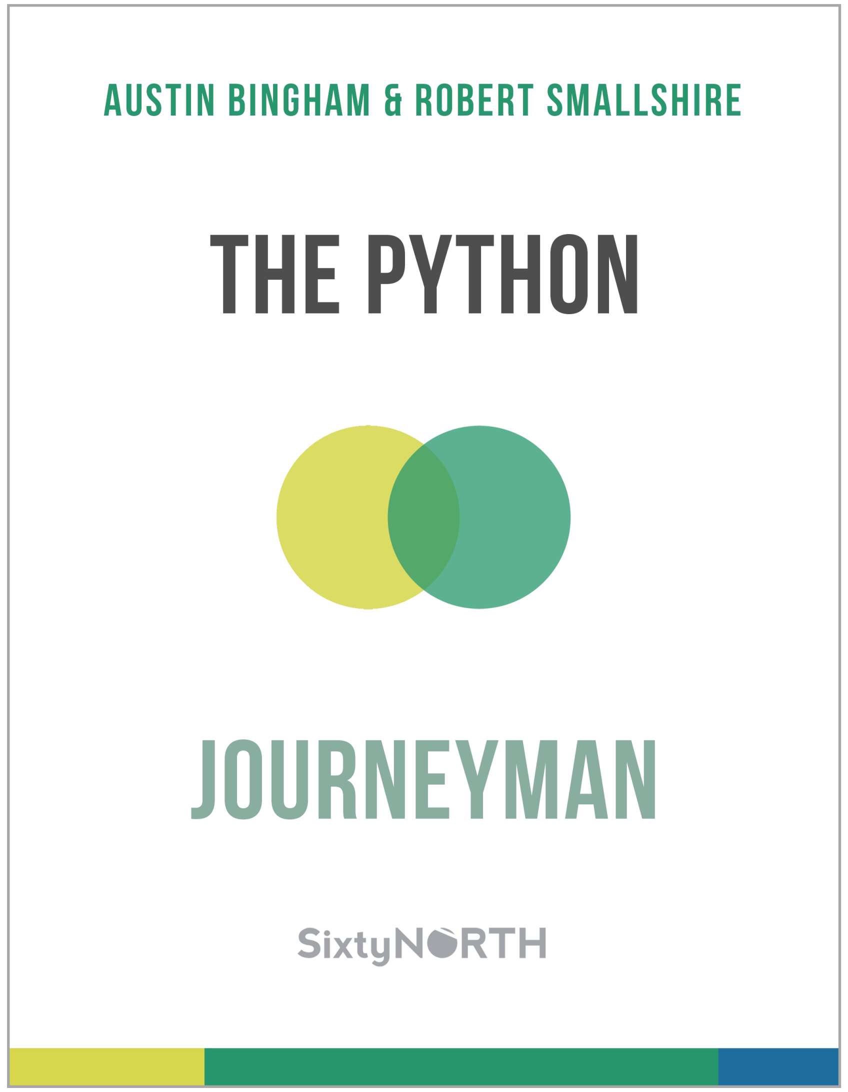
ROBERT SMALLSHIRE & AUSTIN BINGHAM

THE PYTHON

APPRENTICE

SixtyNORTH

https://leanpub.com
/python-apprentice
/c/pluralsight



AUSTIN BINGHAM & ROBERT SMALLSHIRE

THE PYTHON

JOURNEYMAN

SixtyNORTH

https://leanpub.com
/python-journeyman
/c/pluralsight

ROBERT SMALLSHIRE & AUSTIN BINGHAM

THE PYTHON

MASTER

SixtyNORTH

https://leanpub.com
/python-master
/c/pluralsight

Nesting Modules with Packages

Modules

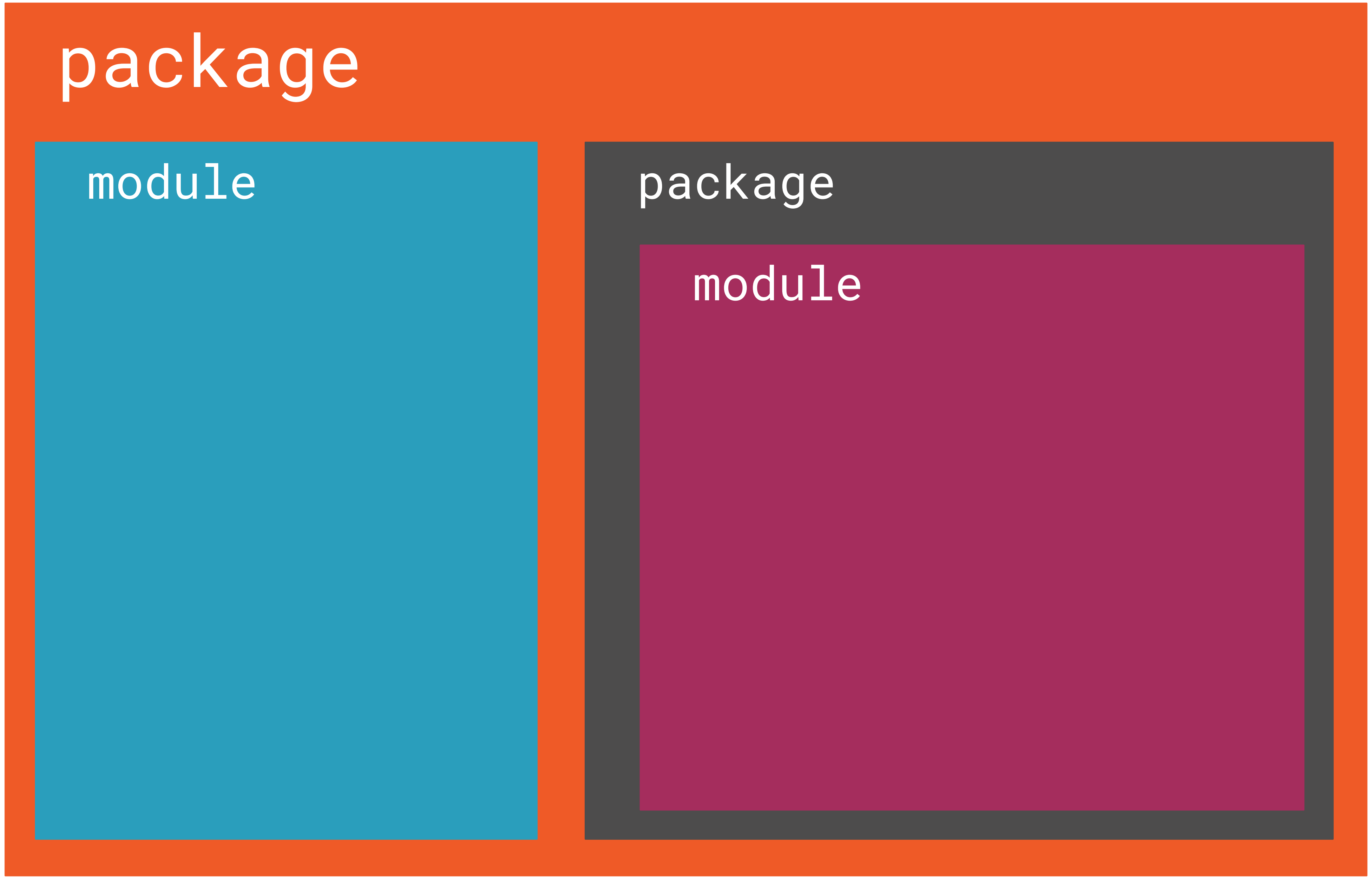
Python’s basic tool for organizing code

Normally a single Python source file

Load modules with `import`

Represented by `module` objects

Packages: Modules that Contain Other Modules



Package vs. Modules

```
package
└── module.py
```

Packages are generally directories

```
module.py
```

Modules are generally files

Python 3.3+: `__path__` is a list

Locating Modules

`sys.path`

List of directories

Searched in order in `import`

First match provides module

`ImportError` when there is no match

PYTHONPATH

Environment variable

List of paths added to `sys.path`

Windows

```
> set PYTHONPATH=path1;path2;path3
```

Linux/macOS

```
$ export PYTHONPATH=path1:path2:path3
```

Full Details

`sys.path`

docs.python.org/3/library/sys.html#sys.path

`PYTHONPATH`

docs.python.org/3/using/cmdline.html#envvar-PYTHONPATH

Summary

Importing nested packages

All modules in hierarchy are imported

- Only the first name is bound
- Use fully-qualified names for submodules

Package directory paths are stored in `__path__`

`sys.path` **controls module search**

- It is initialised from `PYTHONPATH`