# Data Classes



Robert Smallshire
COFOUNDER - SIXTY NORTH
@robsmallshire



Austin Bingham
COFOUNDER - SIXTY NORTH
@austin\_bingham

### Overview

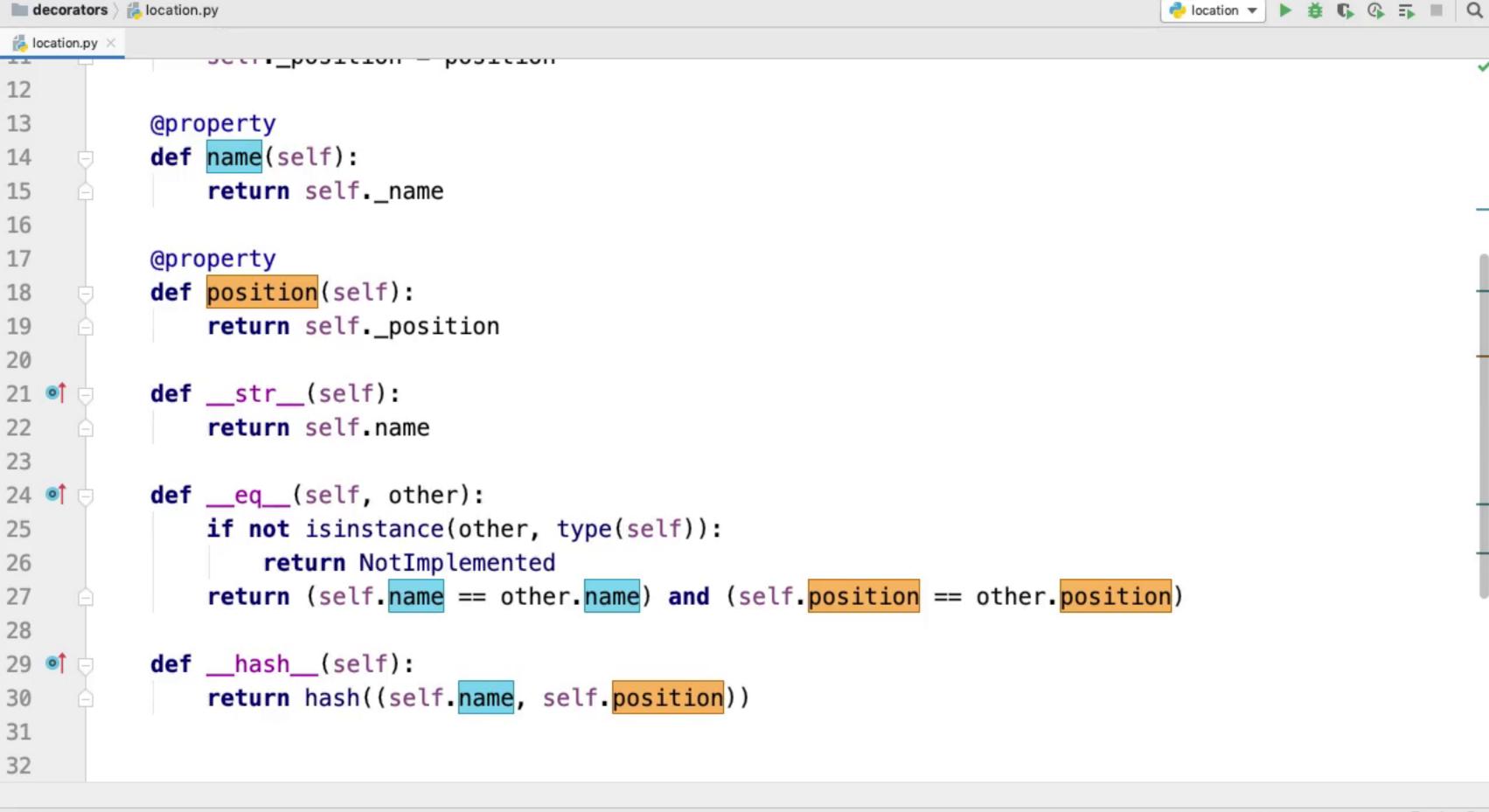


The data class concept

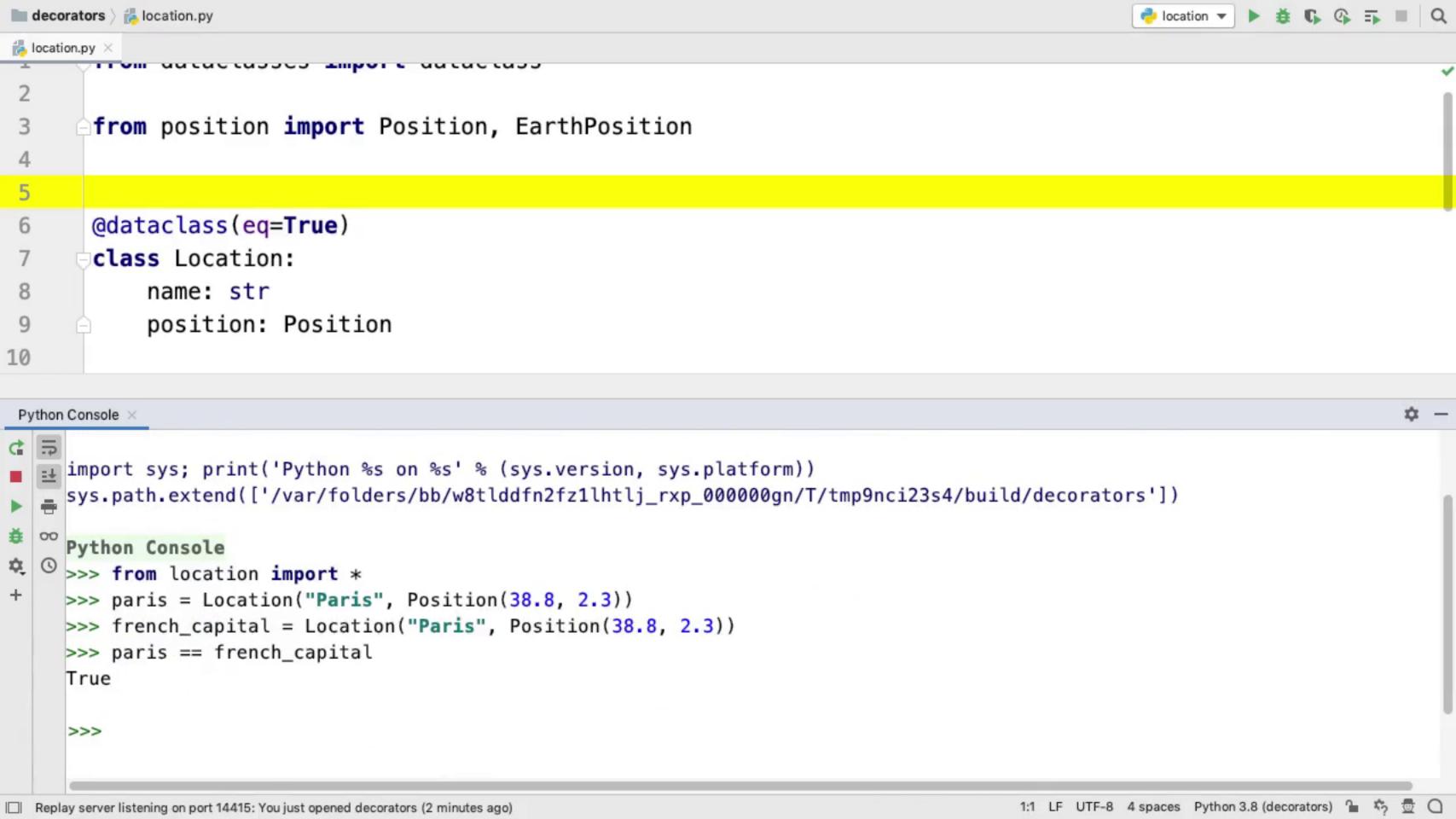
Define data classes

Applicable context for data classes

Avoid inappropriate data class use



# Defining Data Classes



```
@dataclass(
    init=True,
    repr=True,
    eq=True,
    order=False,
    unsafe_hash=False,
    frozen=False,
class MyDataClass:
    fred: int
    jim: int
    sheila: int
```

- **◆ enable** \_\_init\_\_
- **◄** enable \_\_repr\_\_
- **◆** enable \_\_eq\_\_
- **◄** enable \_\_1t\_\_, \_\_gt\_\_, etc.

Hash and Hashability

# Complicated Dataclass Hashability Rules



Immutability is difficult to express.



Hash-based collections require immutable elements.



Equality and hashing must be consistent.

```
@dataclass(
    init=True,
    repr=True,
    eq=True,
    order=False,
                               ◀ configure __hash__
    unsafe_hash=False,
    frozen=False,
class MyDataClass:
    fred: int
    jim: int
    sheila: int
```

```
decorators | location.py
                                                                                                            A C C E I Q
location.py X
       @dataclass(eq=True, frozen=True)
       class Location:
            name: str
            position: Position
10
       hong_kong = Location("Hong Kong", EarthPosition(22.29, 114.16))
                   Python Console >
     /Users/rjs/.virtualenvs/decorators/bin/python "/Users/rjs/Library/Application
      Support/JetBrains/Toolbox/apps/PyCharm-P/ch-0/192.6817.19/PyCharm.app/Contents/helpers/pydev/pydevconsole.py"
      --mode=client --port=50384
     import sys; print('Python %s on %s' % (sys.version, sys.platform))
     sys.path.extend(['/var/folders/bb/w8tlddfn2fz1lhtlj_rxp_000000gn/T/tmpiiokdr9a/build/decorators'])
     Python Console
     >>> from location import *
     >>> cities = {hong_kong, stockholm, cape_town, rotterdam, maracaibo}
     >>>
                                                                                       1:1 LF UTF-8 4 spaces Python 3.8 (decorators)
Replay server listening on port 14415: You just opened decorators (a minute ago)
```

## Prefer Immutable Dataclasses



Use immutable attribute types.



Declare the dataclass as frozen.

# Dataclass Invariants

# Tenets of Object-oriented Programming

#### **Encapsulation**

Managed access to hidden data.

#### Abstraction

Simple interfaces to complex objects.

#### Inheritance

Relating the general to the specific.

#### Polymorphism

A single interface to different types.

```
decorators | location.py
                                                                                    location.py ×
      from position import Position, EarthPosition
      @dataclass(eq=True, frozen=True)
      class Location:
          name: str
          position: Position
10
          def __post_init__(self):
11
              if self.name == "":
                  raise ValueError("Location name cannot be empty")
13
14
15
      hong_kong = Location("Hong Kong", EarthPosition(22.29, 114.16))
16
      stockholm = Location("Stockholm", EarthPosition(59.33, 18.06))
17
      cape_town = Location("Cape Town", EarthPosition(-33.93, 18.42))
18
      rotterdam = Location("Rotterdam", EarthPosition(51.96, 4.47))
19
      maracaibo = Location("Maracaibo", EarthPosition(10.65, -71.65))
20
```

#### @dataclass

#### class MyDataClass:

```
fred: int
```

jim: int

sheila: int

```
def __post_init__(self):
```

**◀** \_\_post\_init\_\_ accepts self

```
if self.fred < 0:</pre>
```

raise ValueError

**◀** Use to configure or validate instance

# Tenets of Object-oriented Programming

#### **Encapsulation**

Managed access to hidden data.

#### **Abstraction**

Simple interfaces to complex objects.

#### Inheritance

Relating the general to the specific.

#### **Polymorphism**

A single interface to different types.

# Tell! Don't ask.

Tell other objects what to do instead of asking them their state and responding to it.

# Keep your data-classes simple.

# Summary



Data-classes are simple compound data types

Apply the @dataclass class-decorator

Type-annotated class attributes specify the data-class members

Optional parameters to @dataclass control member generation

Establish class invariants in \_\_post\_init\_\_

Prefer immutable (frozen) data classes

Equality comparable and frozen dataclasses are hashable

# Well done!

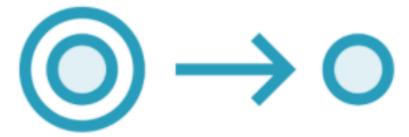
# Concepts to Classes



**Practice frequently** 



Learn from experience



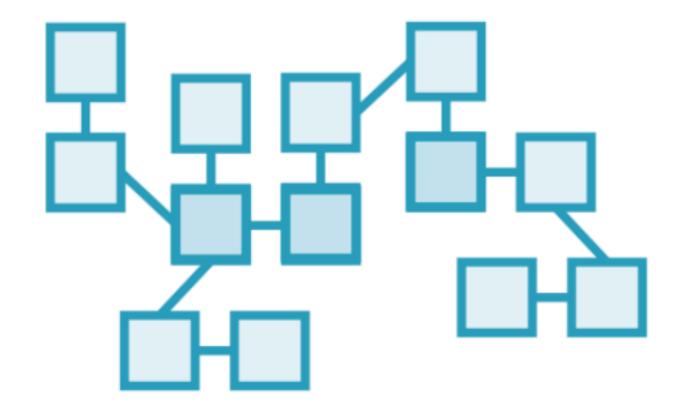
Refactor towards deeper insight

# Single Responsibility Principle

Few classes



Many small classes



One responsibility each

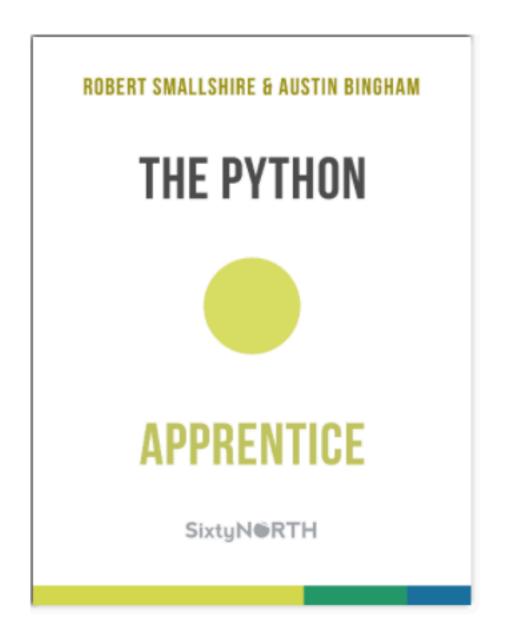
# Core Python

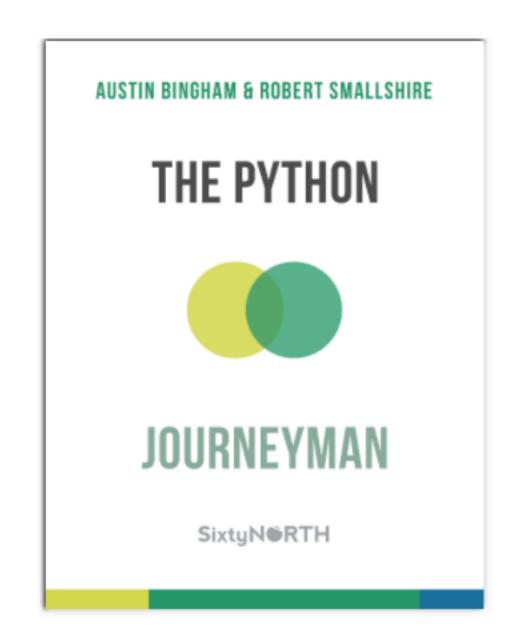


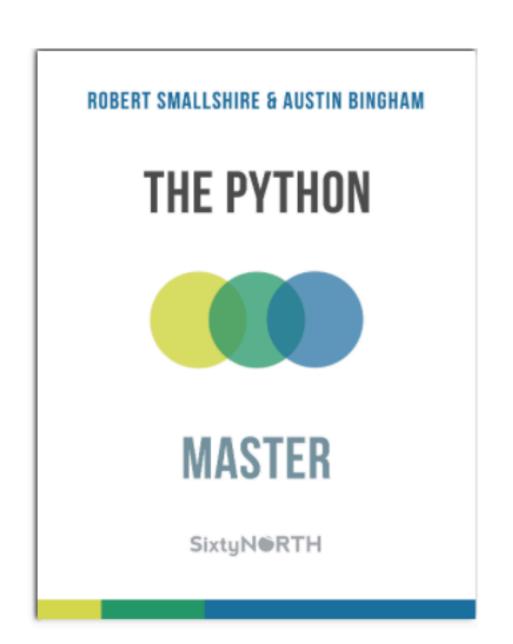
on

PLURALSIGHT

# The Python Craftsman

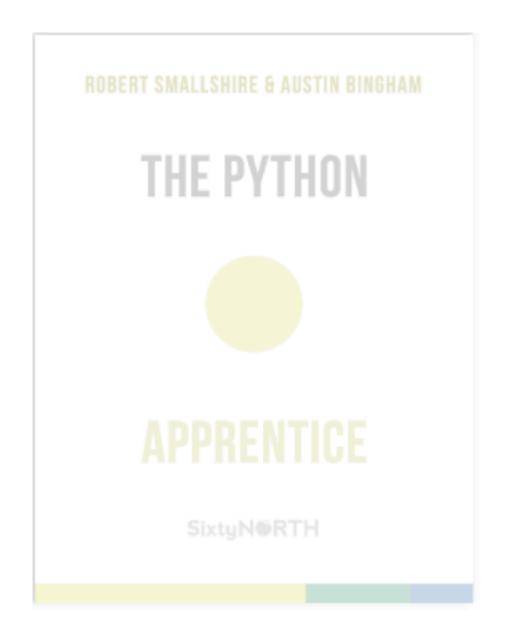


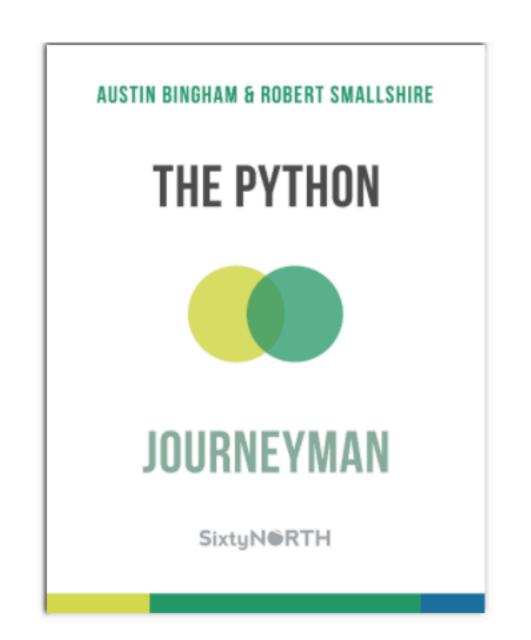


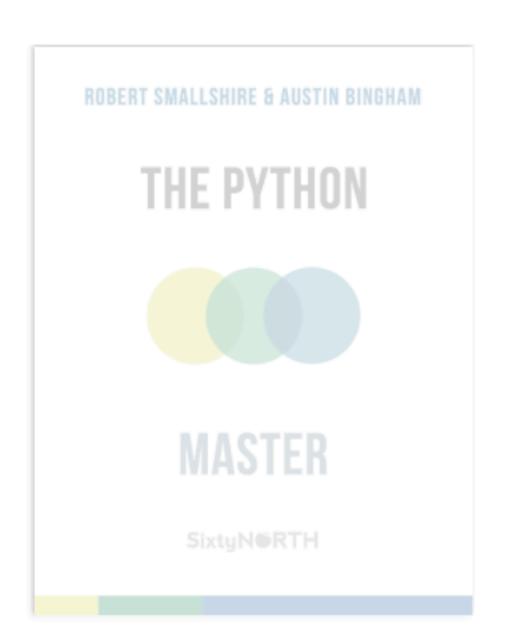


leanpub.com/b/python-craftsman

# The Python Journeyman







leanpub.com/python-journeyman

# Happy Programming!

