Core Python: Introspection

INTRODUCTING TYPES

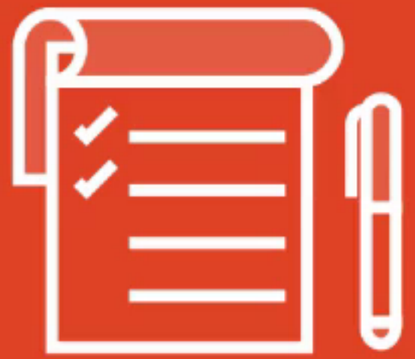
Austin Bingham
COFOUNDER - SIXTY NORTH

@austin_bingham

Robert Smallshire
COFOUNDER - SIXTY NORTH

@robsmallshire

Overview



Functions for introspecting the types of objects

Introspecting relationships between types

Introspecting relationships between objects and types

Explore fundamental aspects of the Python type system

Introspecting types

```
<class 'int'>
>>> int
<class 'int'>
>>> repr(int)
"<class 'int'>"
>>> type(i) is int
True
>>> type(i)(78)
78
>>> type(type(i))
<class 'type'>
>>> i.__class__
<class 'int'>
>>> i.__class__.__class__
<class 'type'>
>>> i.__class__.__class__.__class__
<class 'type'>
>>> issubclass(type, object)
True
>>> type(object)
<class 'type'>
>>> isinstance(i, int)
True
>>>
```

`issubclass()`

Determines if its first argument is a subclass of the second.

Second argument can be a single class.

Or it can be a tuple of classes.

`isinstance()`

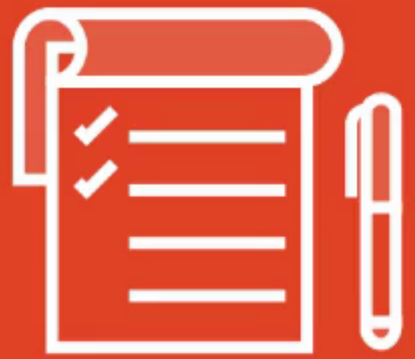
Determines if its first argument is an instance of a class.

The first argument can be an object of any type.

The second argument can be a single class or a tuple of classes.

When type checks are necessary, prefer `isinstance()` and `issubclass()` over direct comparison of type objects.

Summary



`type()` returns the type of its argument

Types are expressed in terms of class objects

The type of a class object is `type`

Every object in Python has a type

Objects store their type on their `__class__` attribute

`issubclass()` reports if one type is a subclass of another

`isinstance()` reports if an object is an instance of a type