types, base, instances, and subclasses

Notation

This is an object

Examples

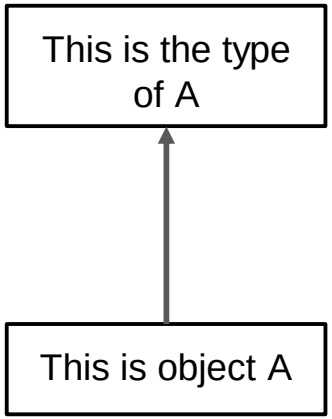
123

type

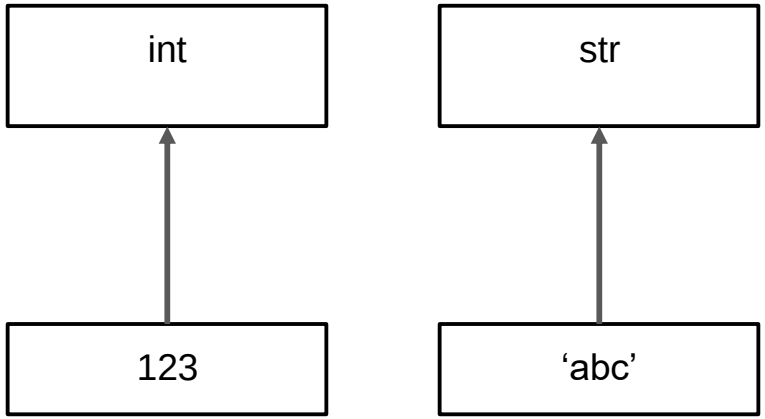
object

anything in python

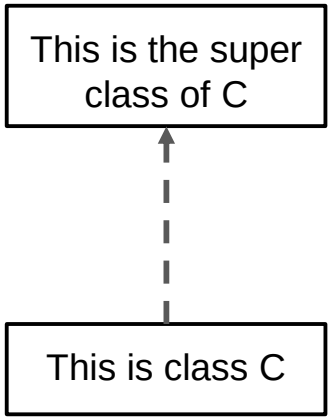
Notation



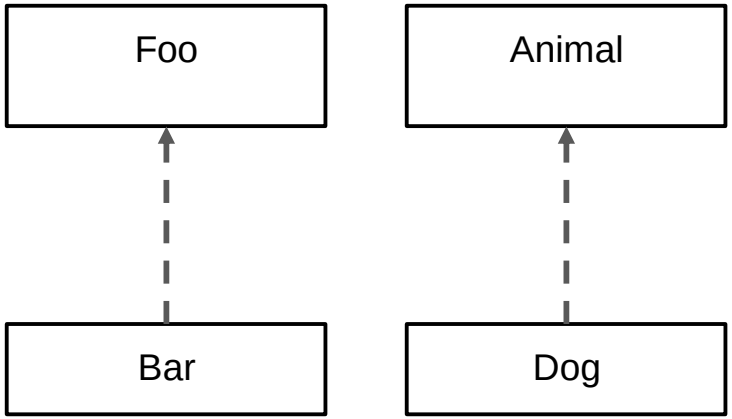
Examples



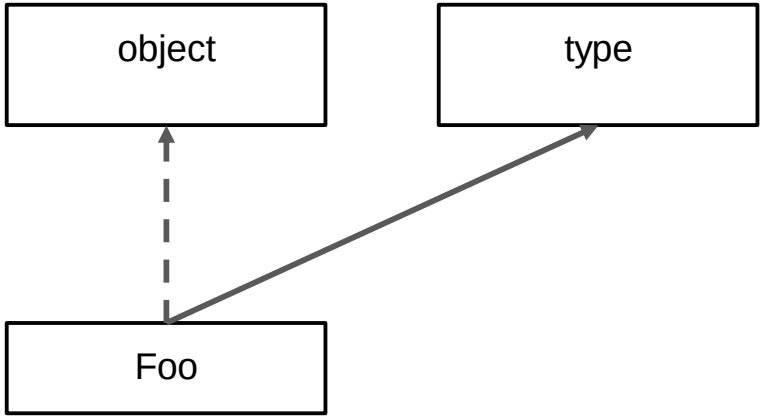
Notation



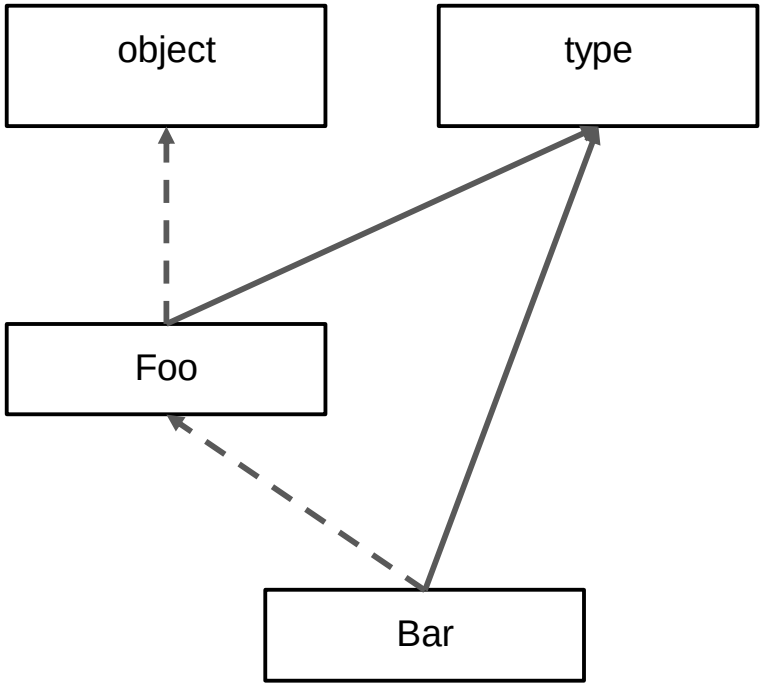
Examples



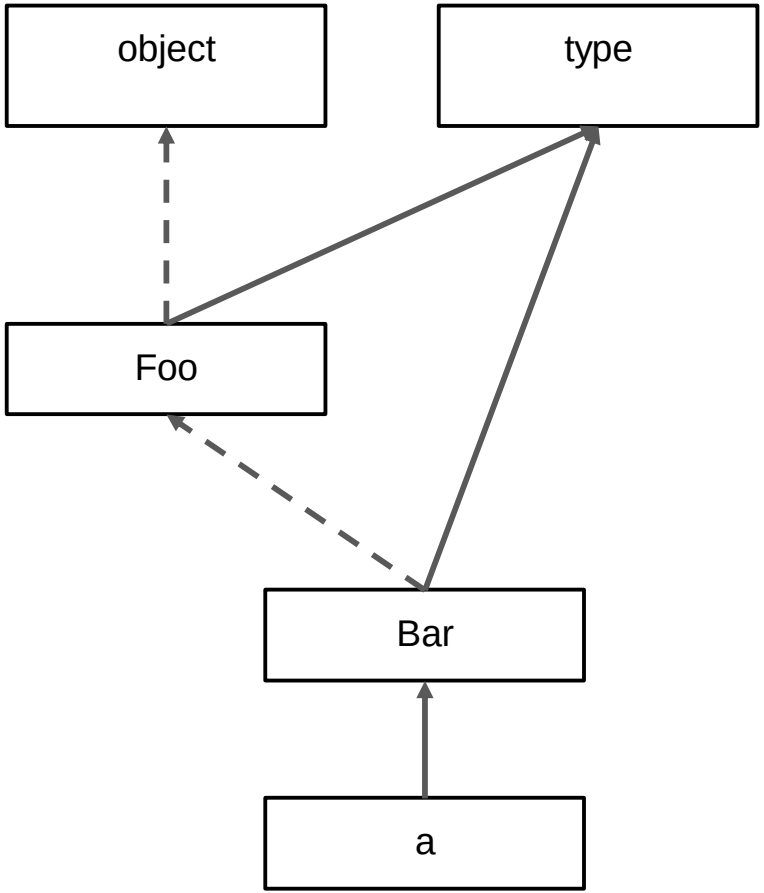
```
class Foo(object):
    pass
```

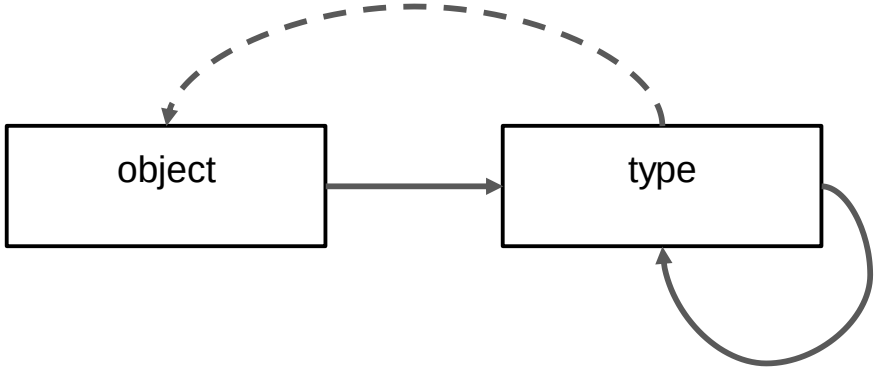


```
class Foo(object):
    pass

class Bar(Foo):
    pass
```



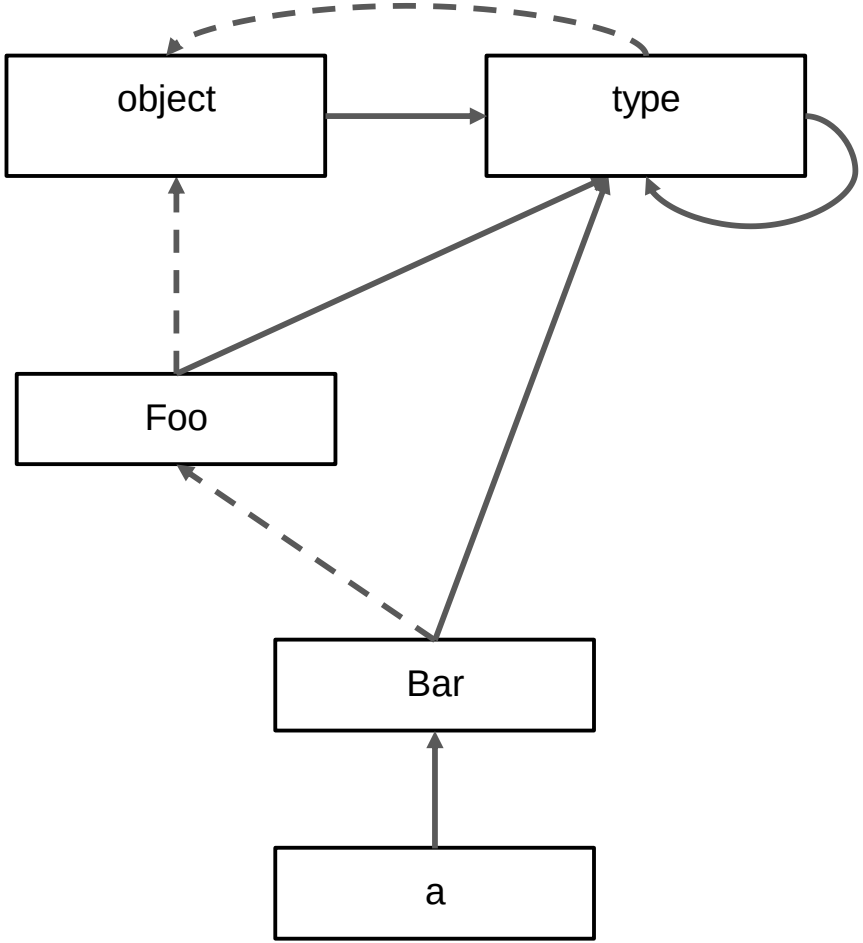
```
class Foo(object):
    pass

class Bar(Foo):
    pass

a = Bar()
```




object
type

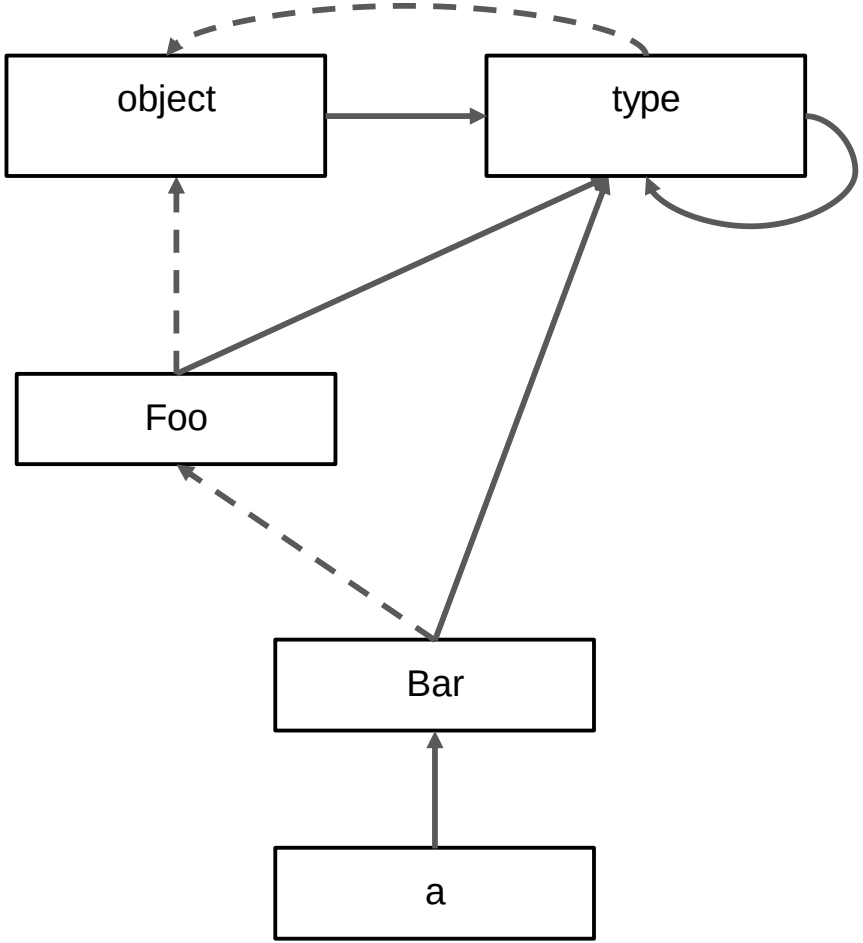
type(x): from x go up the solid arrow one step

type(Foo) == type
type(Bar) == type
type(a) == Bar
type(type) == type
type(object) == type



x.__base__: from x go up the dashed arrow one step

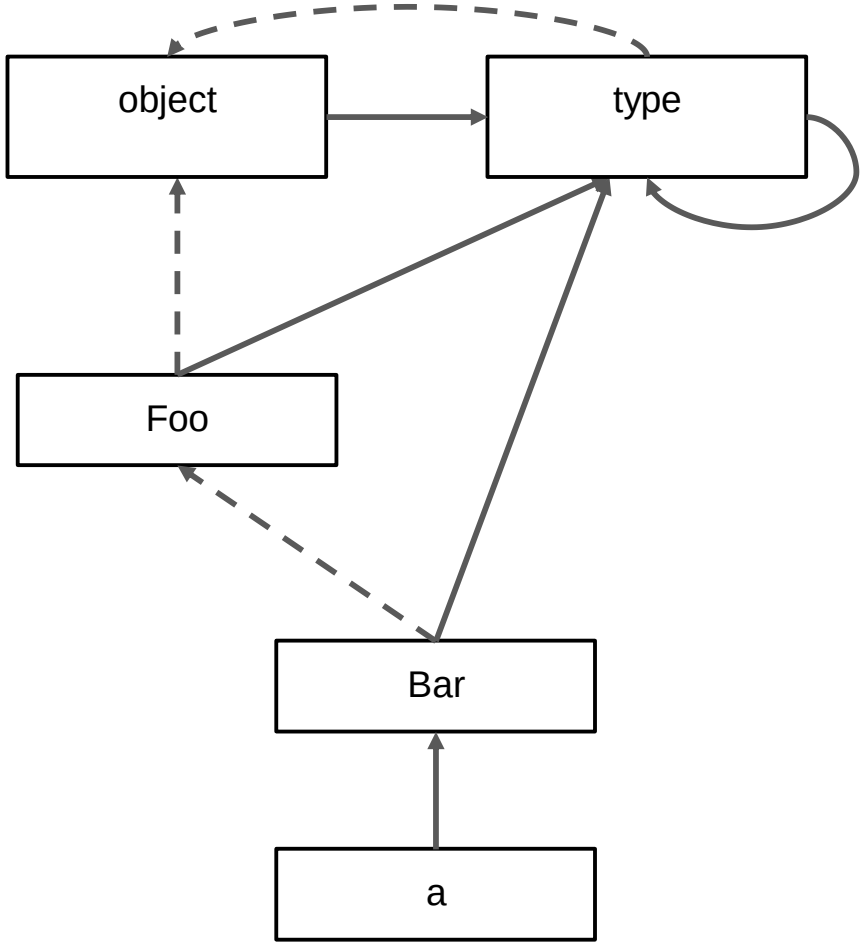
Foo.__base__ == object
Bar.__base__ == Foo
type.__base__ == object
object.__base__ == None



isinstance(x, y): from x go up the solid arrow one step then go up the dashed arrow, return True if see y

Why: first find the type of x then find all superclasses of that type.

```
isinstance(Foo, type) == True
isinstance(Bar, type) == True
isinstance(a, Bar) == True
isinstance(a, Foo) == True
isinstance(Bar, Foo) == False
isinstance(a, object) == True
isinstance(type, object) == True
isinstance(Bar, object) == True
isinstance(Foo, object) == True
isinstance(type, type) == True
isinstance(type, object) == True
```



issubclass(x, y): from x go up the dashed arrow, return True if see y

```
issubclass(Bar, Foo) == True
issubclass(Foo, object) == True
issubclass(Bar, object) == True
issubclass(type, object) == True
```