from pydispatch import Dispatcher, Property

class MyEmitter(Dispatcher):

# Property objects are defined and named at the class level.

# They will become instance attributes that will emit events when their values change

name = Property()

value = Property()

class MyListener(object):

def on\_name(self, instance, value, \*\*kwargs):

print('emitter name is {}'.format(value))

def on\_value(self, instance, value, \*\*kwargs):

print('emitter value is {}'.format(value))

emitter = MyEmitter()

listener = MyListener()

emitter.bind(name=listener.on\_name, value=listener.on\_value)

emitter.name = 'foo'

# >>> emitter name is foo

emitter.value = 42

# >>> emitter value is 42