>>> class P(object):

... def do(self, val):

... print 'p', val

...

>>> class Q(object):

... def do(self, val):

... print 'q', val

...

>>> v = Observer()

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: \_\_init\_\_() takes exactly 2 arguments (1 given)

>>> v = Observer(0)

>>> v.val

0

>>> p = P()

>>> v.sign(p)

>>> v.set\_val(2)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "<stdin>", line 11, in set\_val

NameError: global name 'clients' is not defined

>>> class Observer(object):

... def \_\_init\_\_(self, val):

... self.val = val

... self.clients = []

... def sign(self, obj):

... self.clients.append(obj)

... def get\_val(self):

... return self.val

... def set\_val(self, val):

... self.val = val

... for client in self.clients:

... client.do(val)

...

>>> v = Observer(0)

>>> p = P()

>>> v.sign(p)

>>> v.set\_val(2)

p 2

>>> v.sign(Q())

>>> v.set\_val(3)

p 3

q 3

>>> v.sign(P())

>>> v.set\_val(4)

p 4

q 4

p 4

>>>

https://github.com/faif/python-patterns/