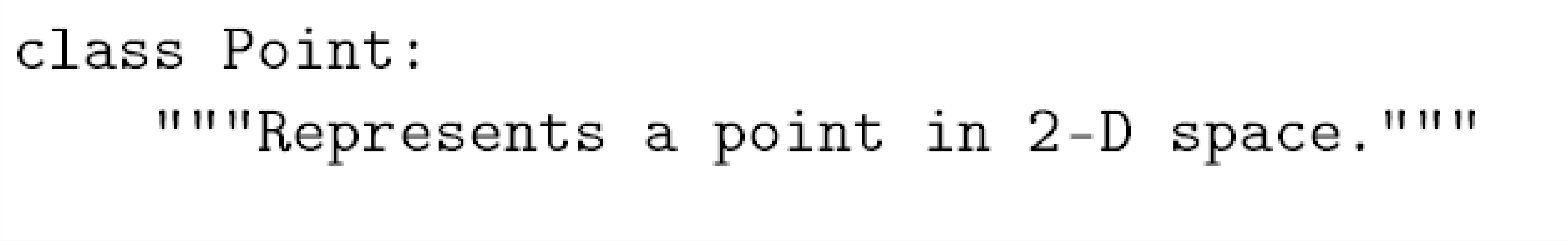
LECTURE 15: CLASSES AND OBJECTS

Hung-Yu Wei

Object Oriented Programming

- Class (類別; 一類東西)
- Object (物件; 一個東西)
  - *Instance*
- Define a new type



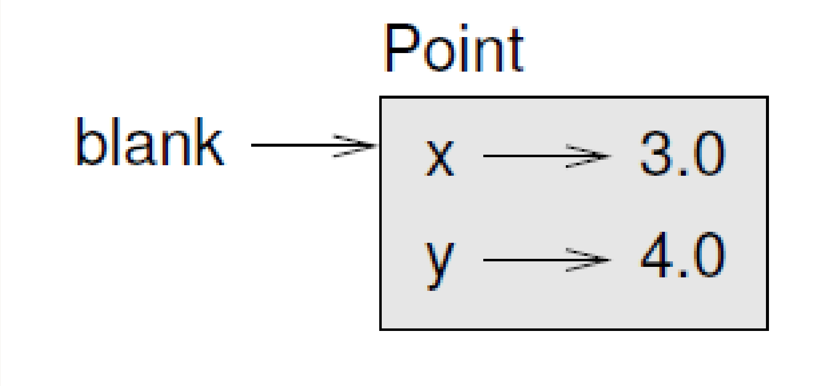
```
class Point:
    """Represents a point in 2-D space."""
```

- Syntax

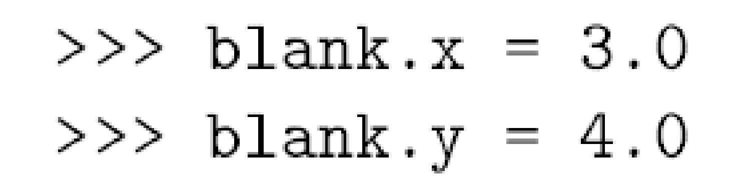
  class *ClassName:*

- Example
  - *Point (x,y)*

Define a class



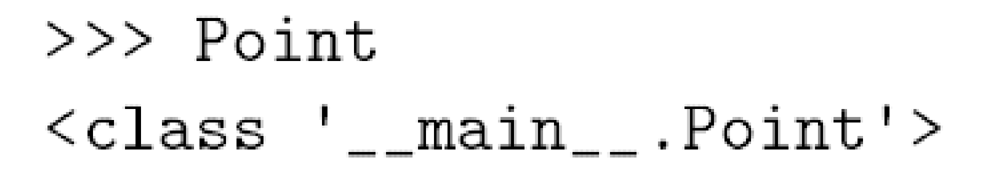
- Attributes



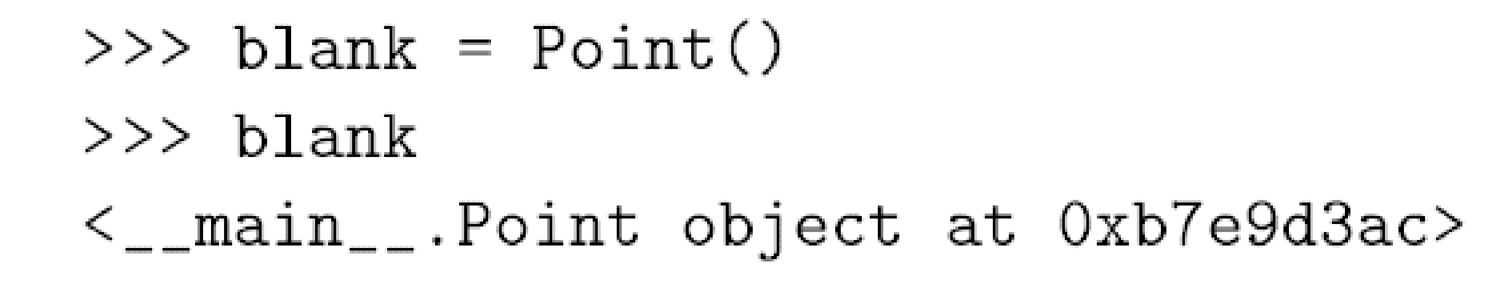
```
>>> blank.x = 3.0
>>> blank.y = 4.0
```

- **Dot** notation
  - *Syntax*

    objectName.attributeName

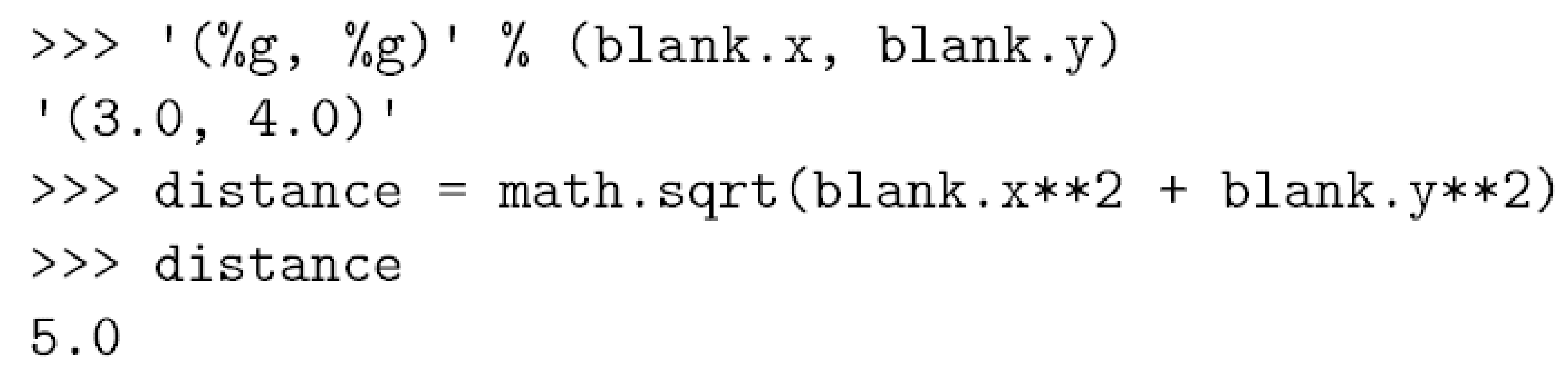


```
>>> Point
<class '__main__.Point'>
```

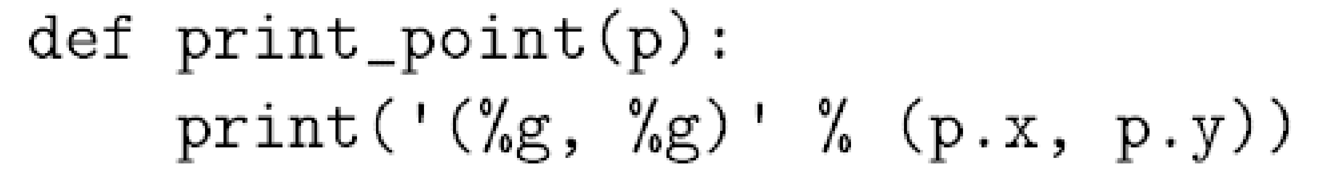


```
>>> blank = Point()
>>> blank
<__main__.Point object at 0xb7e9d3ac>
```

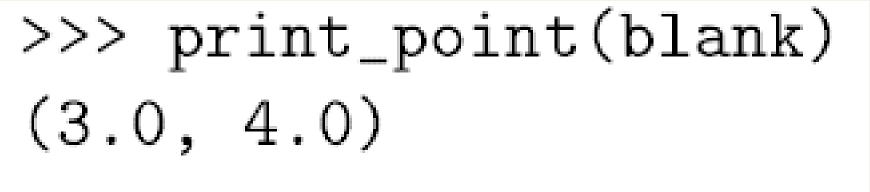
Example (Continued)



```
>>> '(%g, %g)' % (blank.x, blank.y)
'(3.0, 4.0)'
>>> distance = math.sqrt(blank.x**2 + blank.y**2)
>>> distance
5.0
```

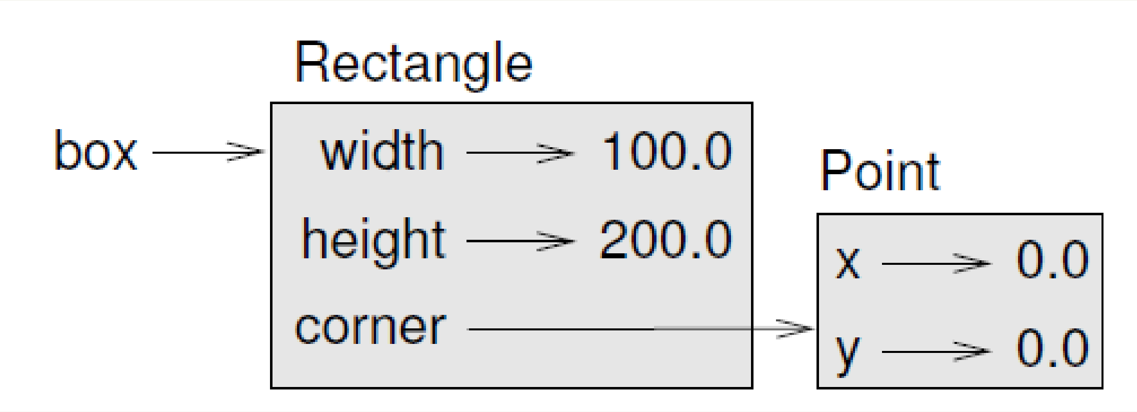


```
def print_point(p):
    print('(%g, %g)' % (p.x, p.y))
```

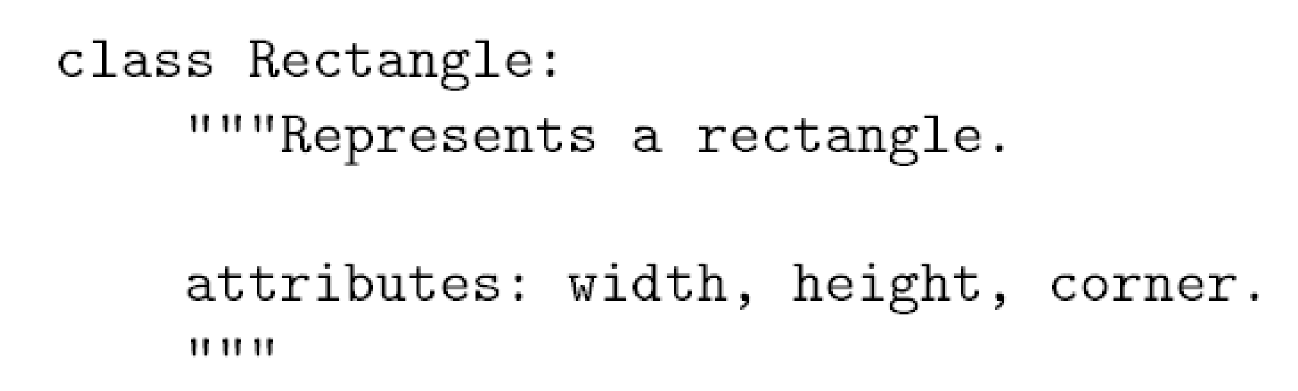


```
>>> print_point(blank)
(3.0, 4.0)
```

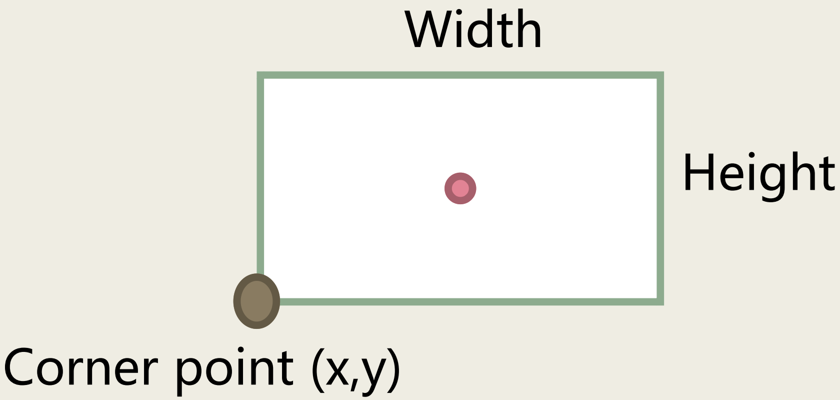
Example: Rectangle

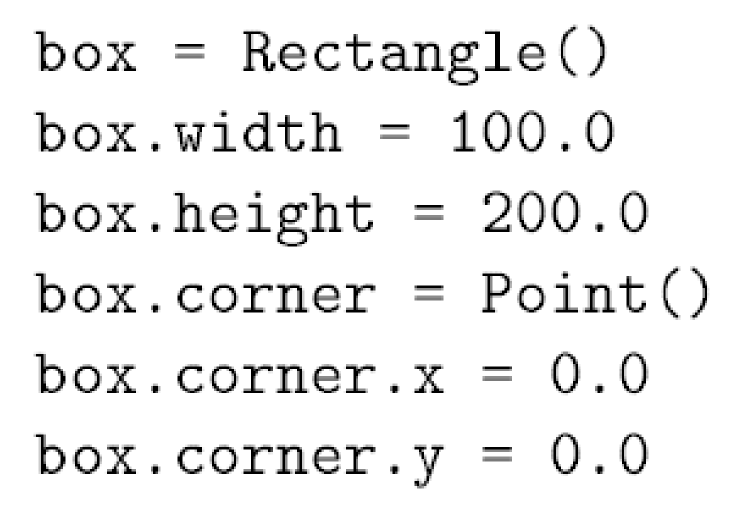


- Define a rectangle class
  - *Width*
  - *Height*
  - *Corner point (x,y)*
- Embedded object
  - *As an attribute of another object*
    - E.g. point in rectangle



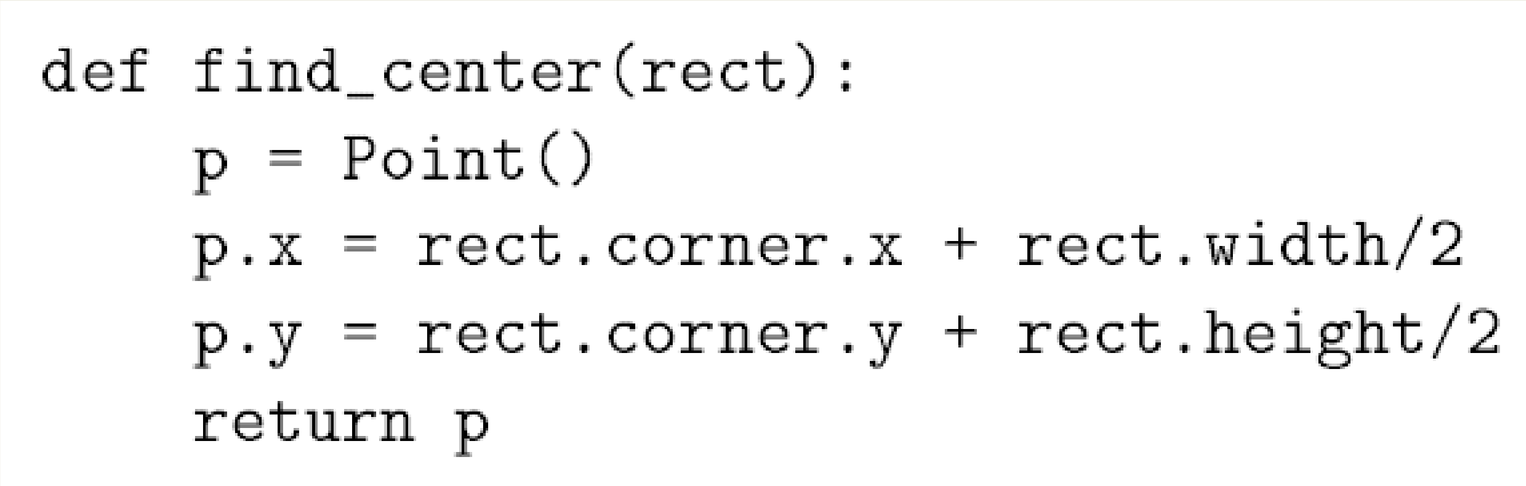
```
class Rectangle:
    """Represents a rectangle.

    attributes: width, height, corner.
    """
```



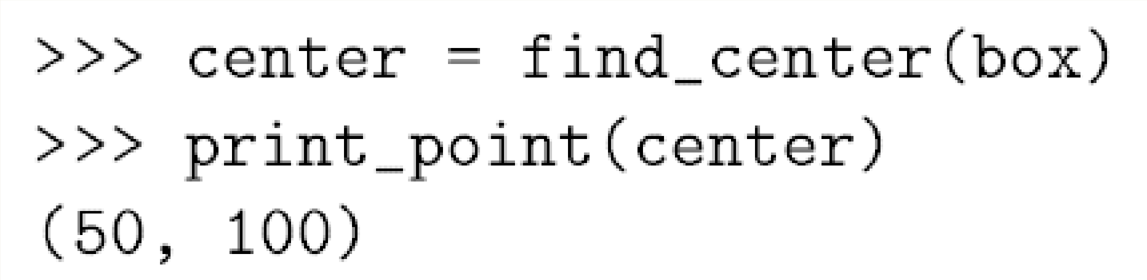


```
box = Rectangle()
box.width = 100.0
box.height = 200.0
box.corner = Point()
box.corner.x = 0.0
box.corner.y = 0.0
```

Instance as return value

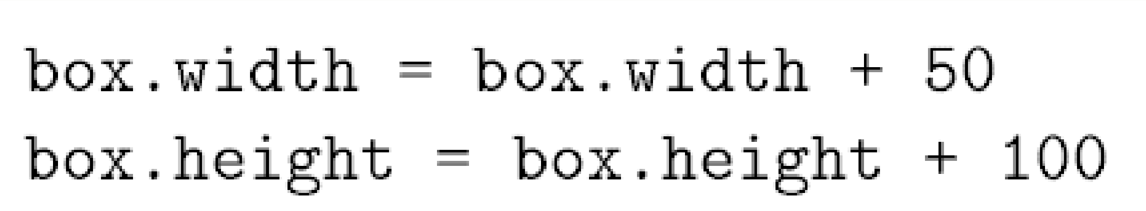


```
def find_center(rect):
    p = Point()
    p.x = rect.corner.x + rect.width/2
    p.y = rect.corner.y + rect.height/2
    return p
```

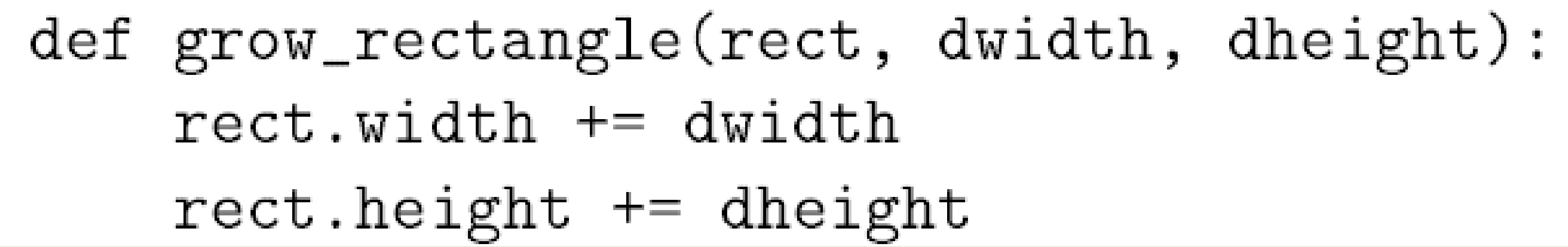


```
>>> center = find_center(box)
>>> print_point(center)
(50, 100)
```

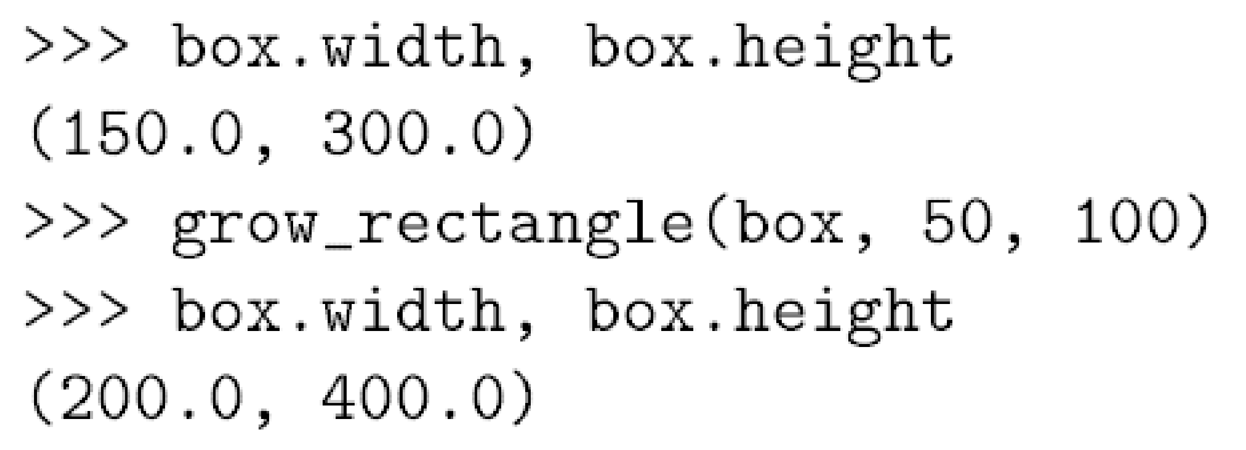
Objects are mutable



```
box.width = box.width + 50
box.height = box.height + 100
```



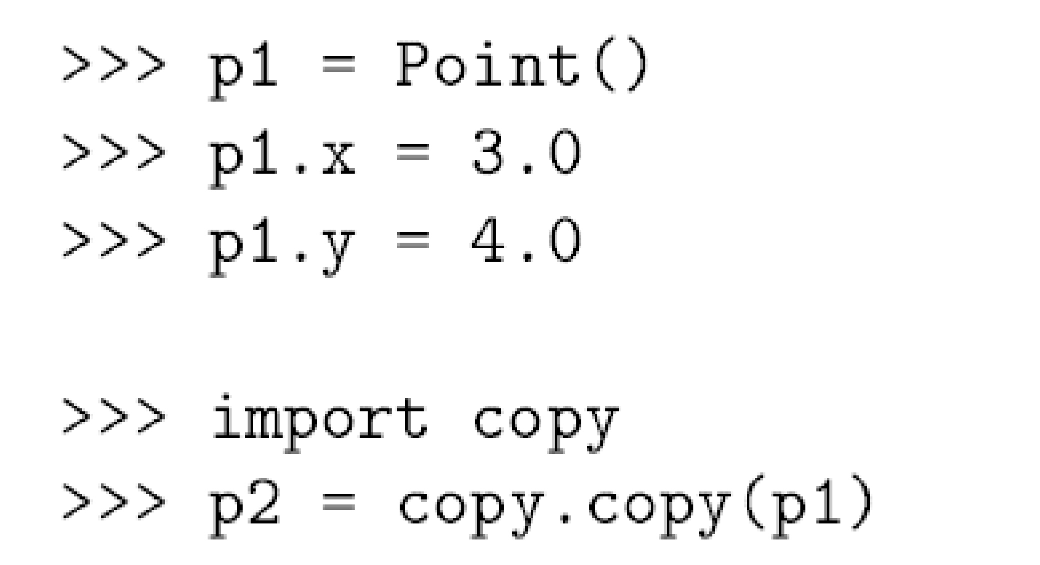
```
def grow_rectangle(rect, dwidth, dheight):
    rect.width += dwidth
    rect.height += dheight
```



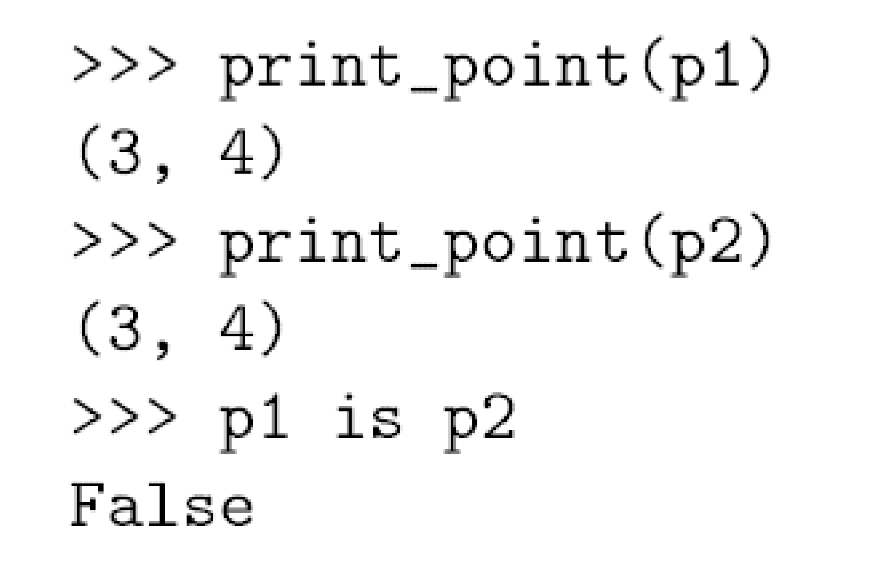
```
>>> box.width, box.height
(150.0, 300.0)
>>> grow_rectangle(box, 50, 100)
>>> box.width, box.height
(200.0, 400.0)
```

Copy an object

- Copy module
  - *Copy method in copy module*

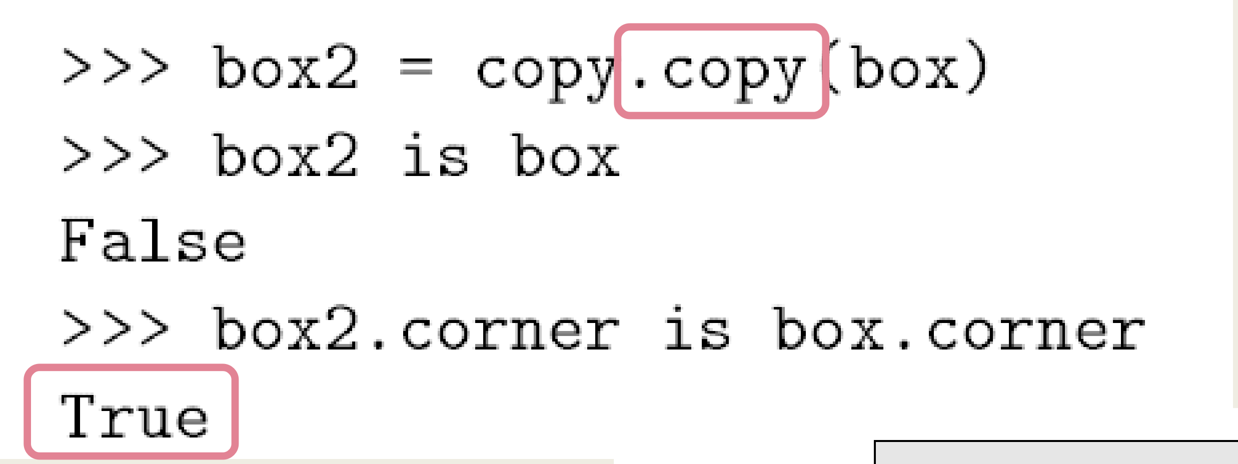


```
>>> p1 = Point()
>>> p1.x = 3.0
>>> p1.y = 4.0

>>> import copy
>>> p2 = copy.copy(p1)
```

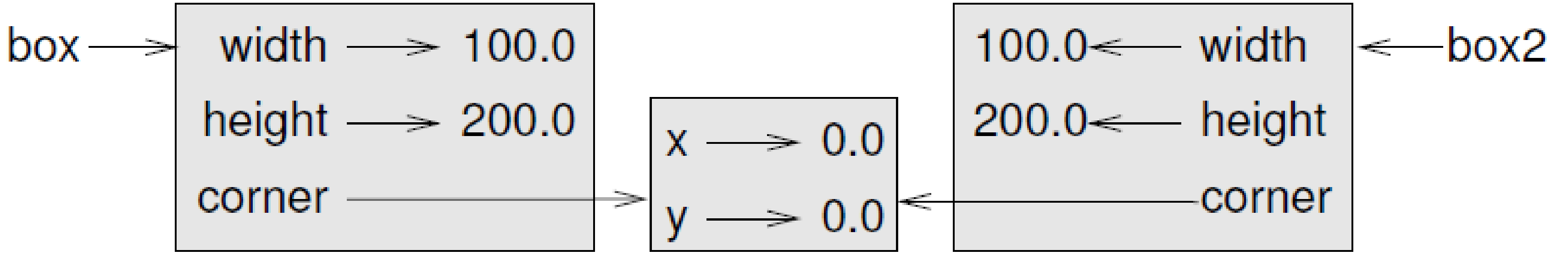


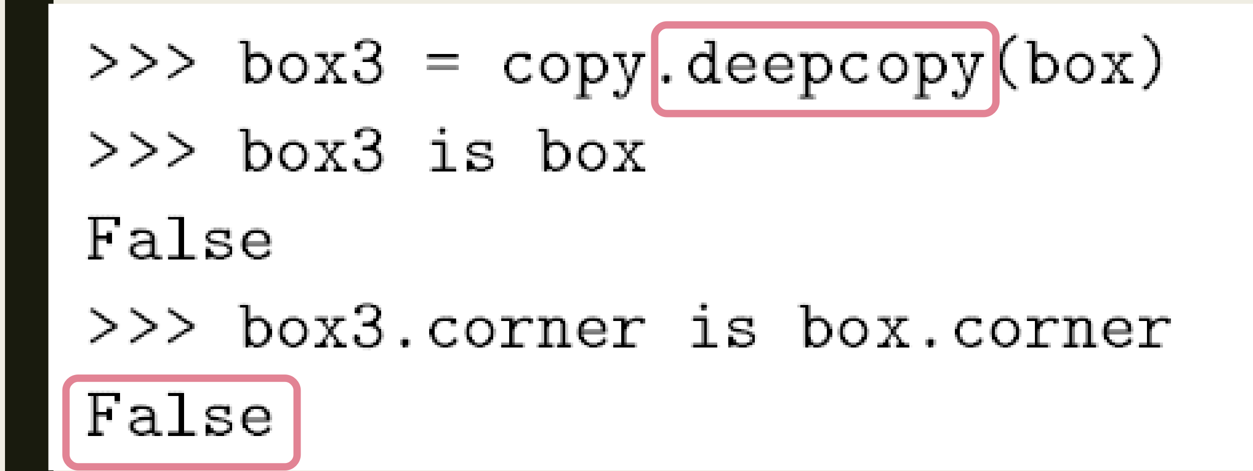
```
>>> print_point(p1)
(3, 4)
>>> print_point(p2)
(3, 4)
>>> p1 is p2
False
```

Shallow Copy v.s. Deep Copy



```
>>> box2 = copy.copy(box)
>>> box2 is box
False
>>> box2.corner is box.corner
True
```

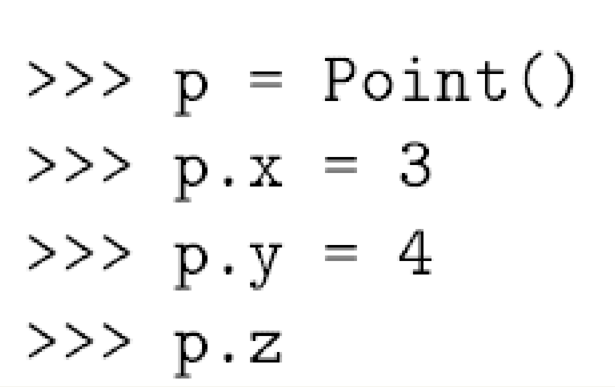




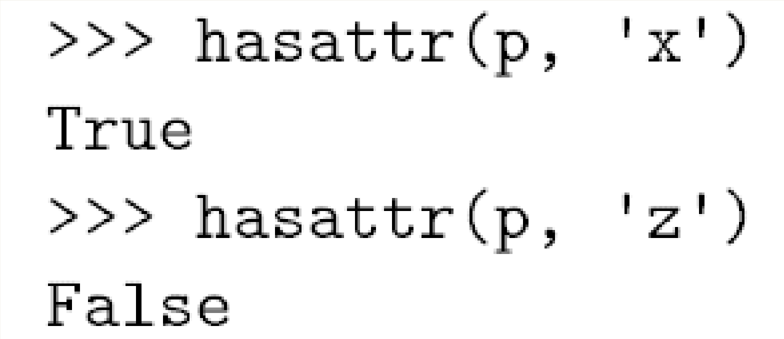
```
>>> box3 = copy.deepcopy(box)
>>> box3 is box
False
>>> box3.corner is box.corner
False
```

Do I have this attribute?

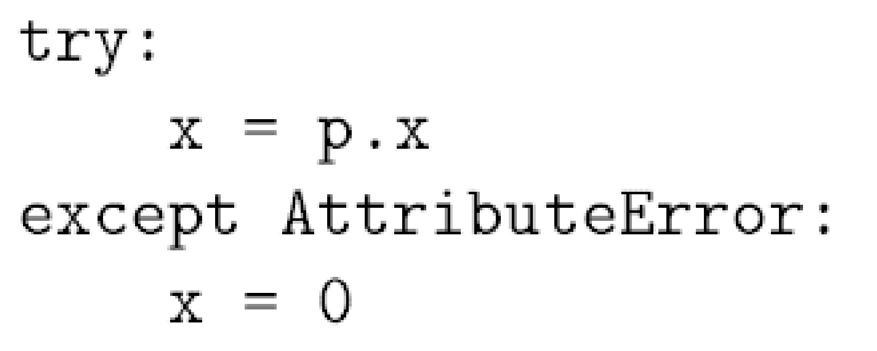
- Attribute Error
  - *Cannot find this attribute*
- Whether I have this attribute
  - *hasattr*
    - Has attribute?
  - ***try***



```
>>> p = Point()
>>> p.x = 3
>>> p.y = 4
>>> p.z
```



```
>>> hasattr(p, 'x')
True
>>> hasattr(p, 'z')
False
```



```
try:
    x = p.x
except AttributeError:
    x = 0
```

Reading

- Chapter 15 in textbook “Think Python”

- Summary
  - *Object and Class in Python*
  - *Shallow copy and deep copy*