

```
/Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5 -u
PyDev console: starting.

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/Users/joe/PycharmProjects/148WK1'])

Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
>>> # everything in Python is an object
>>> w1 = "words"
>>> type(w1)
<class 'str'>
>>> n = 4
>>> type(n)
<class 'int'>
>>> names = ['Joe', 'Danny', 'Amir']
>>> type(names)
<class 'list'>
>>> colleges = {'UC', 'Innis', 'Trinity'}
>>> type(colleges)
<class 'set'>
>>> # when we want to do something to an object, we usually call a
>>> w1.index('d')
3
>>> w1.islower()
True
>>> colleges.add('VIC')
>>> colleges
{'Trinity', 'Innis', 'UC', 'VIC'}
>>> w1.replace('d', 'dsmith')
'wordsmiths'
>>> w1
'words'
>>> # w1 is unchanged. Strings are immutable
>>> w2 = "swords"[1:]
>>> w1 == w2
True
>>> w1 is w2
False
>>> # why?
>>> w3 = w1
>>> w1 is w3
True
>>> # w1 and w2 are not the same object
>>> id(w1)
4326343656
>>> id(w2)
4349389264
>>> id(w3)
4326343656
>>> from turtle import Turtle
>>> t1 = Turtle()
>>> t1.pos()
(0.00,0.00)
>>> t1.forward(100)
>>> t1.pos()
(100.00,0.00)
>>> t1.neck
Traceback (most recent call last):
  File "<input>", line 1, in <module>
```

```
AttributeError: 'Turtle' object has no attribute 'neck'
>>> # neck is not a method in the class Turtle
>>> Turtle.neck = "very reptilian"
>>> # we just define a method called neck
>>> t1.neck
'very reptilian'
>>> # is this good?
>>> # even normal operators have methods we can call
>>> n < 3
False
>>> n.__lt__(3)
False
>>> n + 4
8
>>> n.__add__(4)
8
>>> class Rectangle:
...     def __init__(self, width, height):
...         print ("I'm initializing a new Rectangle instance")
...         self.width = width
...         self.height = height
...     def calculateArea(self):
...         return self.width * self.height
... type (Rectangle)
<class 'type'>
>>> issubclass(Rectangle, object)
True
>>> # no superclass - belongs to builtins.object
>>> rectangle1 = Rectangle(6,4)
I'm initializing a new Rectangle instance
>>> print(rectangle1.calculateArea())
24
```