**Daily Report**

1st February, 2018

1. **定义类并创建实例**

以 Person 为例，定义一个Person类如下：

class Person(object):

Pass

按照 Python 的编程习惯，类名以大写字母开头，紧接着是(object)，表示该类是从哪个类继承下来的

1. **创建实例属性**

例如：

xiaoming = Person()

xiaoming.name = 'Xiao Ming'

xiaoming.gender = 'Male'

xiaoming.birth = '1990-1-1'

1. **初始化实例属性**

在定义 Person 类时，可以为Person类添加一个特殊的\_\_init\_\_()方法，当创建实例时，\_\_init\_\_()方法被自动调用，我们就能在此为每个实例都统一加上以下属性。例如：

class Person(object):

def \_\_init\_\_(self, name, gender, birth):

self.name = name

self.gender = gender

self.birth = birth

\_\_init\_\_() 方法的第一个参数必须是 self（也可以用别的名字，但建议使用习惯用法），后续参数则可以自由指定，和定义函数没有任何区别

相应地，创建实例时，就必须要提供除 self 以外的参数：

xiaoming = Person('Xiao Ming', 'Male', '1991-1-1')

xiaohong = Person('Xiao Hong', 'Female', '1992-2-2')

1. **访问限制**

Python对属性权限的控制是通过属性名来实现的，如果一个属性由双下划线开头(\_\_)，该属性就无法被外部访问。看例子：

class Person(object):

def \_\_init\_\_(self, name):

self.name = name

self.\_title = 'Mr'

self.\_\_job = 'Student'

p = Person('Bob')

print p.name

# => Bob

print p.\_title

# => Mr

print p.\_\_job

# => Error

Traceback (most recent call last):

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

AttributeError: 'Person' object has no attribute '\_\_job'

1. **类属性**

实例属性每个实例各自拥有，互相独立，而类属性有且只有一份

class Person(object):

address = 'Earth'

def \_\_init\_\_(self, name):

self.name = name

print Person.address

# => Earth

p1 = Person('Bob')

p2 = Person('Alice')

print p1.address

# => Earth

print p2.address

# => Earth

1. **私有属性**

一个实例的私有属性就是以\_\_开头的属性，无法被外部访问；虽然私有属性无法从外部访问，但是，从类的内部是可以访问的

1. **定义实例方法**

实例的方法就是在类中定义的函数，它的第一个参数永远是 self，指向调用该方法的实例本身，其他参数和一个普通函数是完全一样的：

class Person(object)

def \_\_init\_\_(self, name):

self.\_\_name = name

def get\_name(self):

return self.\_\_name

get\_name(self) 就是一个实例方法，它的第一个参数是self。\_\_init\_\_(self, name)其实也可看做是一个特殊的实例方法

1. **定义类方法**

要在class中定义类方法，需要这么写：

class Person(object):

count = 0

@classmethod

def how\_many(cls):

return cls.count

def \_\_init\_\_(self, name):

self.name = name

Person.count = Person.count + 1

print Person.how\_many()

p1 = Person('Bob')

print Person.how\_many()

通过标记一个 @classmethod，该方法将绑定到 Person 类上，而非类的实例。类方法的第一个参数将传入类本身，通常将参数名命名为 cls，上面的 cls.count 实际上相当于 Person.count