

## ~~继承~~ 继承

编写新类时要调用父类的 `__init__()`

```
class Derived(Base)
```

`super()` 可用于调用父类的方法

父类也称为超类，因此 `__` 名为 `super`

```
eg: def __init__(self):  
    super().__init__()
```

重写父类的方法：

在子类中定义一个与要重写的父类方法同名的方法

• 将实例作为属性：实例化对象作为类的属性

好处：在调整一些属性时不至于把整个类弄乱

## • 导入类

```
from module_name import class_name, ...
```

导入整个模块：

```
import module_name
```

使用时需 `module_name.class_name`

} 易于阅读

不建议 `from module_name import *`

不易于阅读，且名称容易冲突

亦可在一个模块中导入另一个模块

也可用 `as` 起别名

## Python标准库

import 即可

eg: import random / from random import ~~rand~~ randint  
~~rande~~ randint(a,b)  
[a,b] 内的随机数

另一个有用的函数是 choice ( )

将一个列表或元组作为参数，并随机返回其中的一个元素

注：安全相关的项目不用 random module (伪随机)

## • 类编码风格

类名应采用驼峰命名法 Class Name

实例名、模块名都采用小写格式且下划线分割 first\_up

每个类后跟一个文档字符串进行说明 ""

隔

类中可用一个空行来分割方法

模块中可用两个空行来分割 隔类

先 import 标准库 换一行 再 import 自写模块