为完成“搜查令（一个简单轻便的端口扫描脚本）”应用开发的代码编写工作，我们小组选择Python作为编程语言，由于对Python的不了解，作为团队的PM，除了做了PM的工作之外工作量还是不够，所以写了这个学习报告，本学习报告包括Python语言本身、Django的应用两部分的学习报告。（因为学习时间有限，对各知识点的都没有较深的细致认识，只是大致上了解的初学者阶段。）

# 一、 Python语言

Python是面向对象、解释执行的高级语言：Python是完全面向对象的，其中的函数、模块、数字、字符串都是对像，并且完全支持继承、重载、派生、多继承，由于之前c++的语言基础，Python的这一特性可以很快的上手使用；Python在解释执行时，首先会将.py文件中的源代码编译成Python的字节码，然后再由Python虚拟机来执行这些编译好的字节码。

Python的设计哲学是“优雅”、“明确”、“简单”：设计者开发时总的指导思想是，对于一个特定的问题，只要有一种最好的方法来解决就好了；Python的作者有意的设计限制性很强的语法，使得不好的编程习惯都不能通过编译。其中很重要的一项就是Python的缩进规则；Python的设计目标之一是让代码具备高度的可阅读性。它设计时尽量使用其它语言经常使用的标点符号和英文单字，让代码看起来整洁美观。

**基本语法：**

## （1）缩进：

Python语言利用缩进表示语句块的开始和退出（Off-side规则），而非使用花括号或者某种关键字。增加缩进表示语句块的开始，而减少缩进则表示语句块的退出。（根据PEP的规定，必须使用4个空格来表示每级缩进，在实际编写中可以自定义空格数，但是要满足每级缩进间空格数相等。使用Tab字符和其它数目的空格虽然都可以编译通过，但不符合编码规范。）

## （2）控制语句：

if语句，当条件成立时运行语句块。经常与else, elif(相当于else if) 配合使用。

for语句，遍历列表、字符串、字典、集合等迭代器，依次处理迭代器中的每个元素。

while语句，当条件为真时，循环运行语句块。

try语句，与except,finally配合使用处理在程序运行中出现的异常情况。

class语句，用于定义类型。

def语句，用于定义函数和类型的方法。

pass语句，表示此行为空，不运行任何操作。

assert语句，用于程序调适阶段时测试运行条件是否满足。

with语句，Python2.6以后定义的语法，在一个场景中运行语句块。比如，运行语句块前加密，然后在语句块运行退出后解密。

yield语句，在迭代器函数内使用，用于返回一个元素。自从Python 2.5版本以后。这个语句变成一个运算符。

raise语句，制造一个错误。

import语句，导入一个模块或包。

fromimport语句，从包导入模块或从模块导入某个对象。

import as语句，将导入的对象赋值给一个变量。

in语句。判断一个对象是否在一个字符串/列表/元组里。

## （3）表达式：

Python的表达式与c、c++、haskell、js等都有相似之处，但是又有很多不同，看起来像是融合了很多语言的特点，这里简要列举几个典型的表达式：

单引号和双引号：Python使用'(单引号)和"(双引号)来表示字符串，两种符号作用相同。一般地，如果字符串中出现了双引号，就使用单引号来表示字符串;反之则使用双引号。如果都没有出现，就依个人喜好选择。出现在字符串中的\(反斜杠)被解释为特殊字符，比如\n表示换行符。表达式前加r指示Python不解释字符串中出现的\。这种写法通常用于编写正则表达式或者Windows文件路径。

Lambda表达式：Python使用lambda表示匿名函数。匿名函数体只能是表达式。比如：add=lambda x, y : x + y

选择表达式：Python使用y if condelse x表示条件表达式。意思是当cond为真时，表达式的值为y，否则表达式的值为x。相当于C++和Java里的cond?y:x

列表推导式：Python支持“列表推导式”（listcomprehension），比如计算0-9的平方和:sum(x \* x for x in range(10))

4)        **函数：**Python的函数支持递归、默认参数值、可变参数，但不支持函数重载。为了增强代码的可读性，可以在函数后书写“文档字符串”(Documentation Strings，或者简称docstrings)，用于解释函数的作用、参数的类型与意义、返回值类型与取值范围等。可以使用内置函数help()打印出函数的使用帮助。

5)        **对象**：Python的对象定义和用法可以很大程度上参考C++，但要注意语法上的一些差异，比如Python在定义对象方法时要增加一个参数self用于访问对象内部数据。

6)        **类型**：Python采用动态、强数据类型，内置丰富的数据类型，可以有效减少代码长度，同时也支持自定义数据类型，值得注意的是，Python语言还用类型来表示函数、模块、类型本身、对象的方法、编译后的Python代码、运行时信息等等，因此，Python具备很强的动态性。包含的类型有：str、list、tuple、set、dict、int、float、complex、bool

7)        **数学运算**：支持整数与浮点数的数学运算，同时还支持复数运算与无穷位数（实际受限于计算机的能力）的整数运算。除了求绝对值函数abs()外，大多数数学函数处于math和cmath模块内，前者用于实数运算，而后者用于复数运算，使用时需要先导入它们。值得注意的是，Python允许像数学的常用写法那样连着写两个比较运行符。比如a < b < c与a < b and b < c等价。

# 二、  Django的应用

Django是一个开放源代码的Web应用框架，由Python写成。采用了MVC的软件设计模式，即模型M，视图V和控制器C，然而Django真实的软件设计模式为MTV，即模型M，模版T，视图V，至于原先控制器的功能就相当于转到了视图V和URLconf来实现，而模板T就相当于MVC中的视图，负责页面表现。

Django有几个特点对于此次的作业有很大作用，其一是对象关系映射，对象关系映射以Python类形式定义你的数据模型，ORM将模型与关系数据库连接起来，你将得到一个非常容易使用的数据库API，同时你也可以在Django中使用原始的SQL语句；其二是URL分派，URL 分派使用正则表达式匹配URL，你可以设计任意的URL，没有框架的特定限定，URL将一个httprequest分派给views的一个函数处理；其三是自动化管理界面，自动化的管理界面，不需要你花大量的工作来创建人员管理和更新内容，Django自带一个ADMIN site,类似于内容管理系统。

Django工作机制**：**

    前提：Django服务器已经正确设置好，setting.py中包含了所有设置信息，其中最重要的是ROOT\_URLCONF，这是用来设置本站的URLconf。

1、  当用户url时，Django服务器根据ROOT\_URLCONF来装载URLconf。

2、  匹配URLconf里的URLpatterns，匹配成功就会调用相关联的views中的函数，并把Httprequest对象作为第一个参数。

3、  Views函数处理后返回一个Httpresponse对象

**Django创建工程和应用：**

1.      建立 Django 站点工程：建好一个工程后会在工程根目录下有\_int\_.py/manage.py/setting.py/urls.py四个文件夹，他们用来设置管理工程，一个工程可以包含一个或多个APP

2.      建立App：建e好App后会在app根目录下有\_int\_.py/models.py/tests.py/views.py四个文件夹。之后的工作就是按照Django的软件设计模式开始一些模块的编程，并完成一些设置，如SQL、url等。