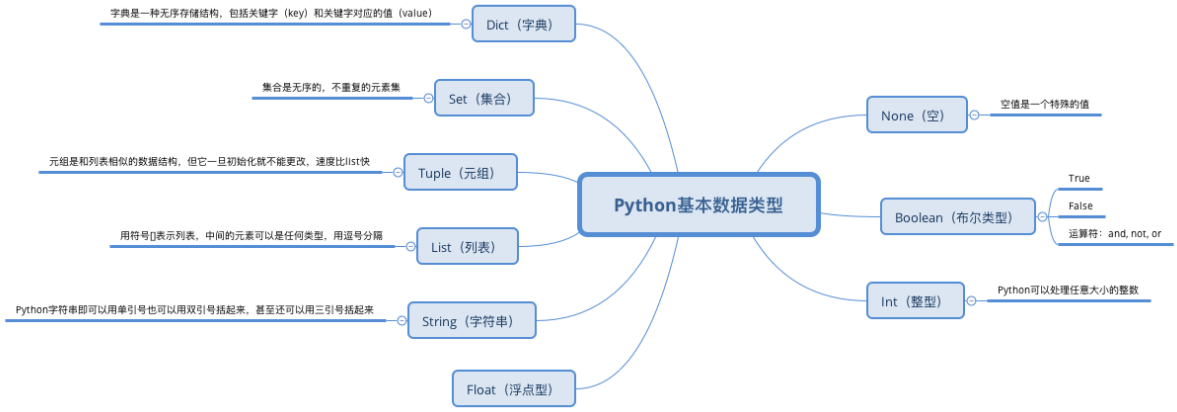
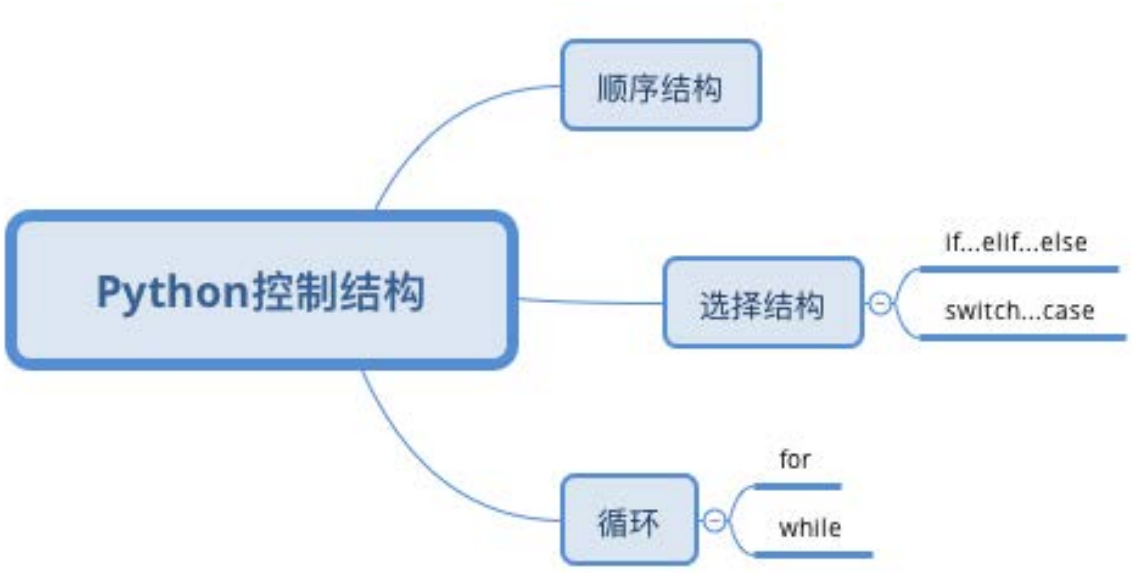
1、Python 基本成分和语言特性总结

1.1 Python 语言基本成分：





1.2 Python 语言特性：

在 python 的交互式命令窗口输入 import.this 我们就会得到一首 python 之禅：

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.

Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

这首禅的主要内容即是对 python 语言特性的描述，其主要内容是强调了“优美胜于丑陋”，“明了胜于晦涩”，“简洁胜于复杂”，“复杂剩余凌乱”，“扁平剩余嵌套”，“间隔胜于紧凑”，“可读性很重要”，这些内容恰恰都是 python 语言的特点。

python 语言的一些基本成分其实与 C 等其他编程语言很相似，具有自带的数据类型、数据结构等，python 原本是作为脚本语言的，其面向对象的开发能力远远不如 java、c++等语言，但随着 python 的使用越来越广，该语言也在不断发展，现在不仅支持面向对象，而且效果也不差。另外，python 的语法成分相比其他语言，显得尤为少，python 的嵌套语法是首行末尾使用冒号，代码块省去括号，按照缩进的方式书写。并且 python 每行之间完全不需要使用分号，一行的结束就是终止了该语句。缩进结束就意味着这段代码块的结束。所以缩进语法是 python 的一个核心语法。正是因为这一点，python 程序员可以整齐地写出根据程序逻辑结构以垂直方式来完成的代码。这使得 python 的程序代码更加具有可读性。

python 在很多领域都极为适用，如系统编程、用户图形接口、网络脚本、组件集成、数据库编程、快速原型、数值计算和科学计算编程、游戏、图像、人工智能等，我认为 python 比 c++在一般的开发情况下更具有优势，的确，python 是解释型语言，执行速度比 c 慢，但在实际开发中我们往往要考虑高昂的人力成本，熟悉掌握 c++所需的时间比 python 要长得多，且 c++开发项目的周期也要比 python 长的多。

2、Python 个人项目 ProxyipPool 分析

2.1 项目介绍

该项目的功能是从几个有免费代理 ip 的网址爬取免费 ip 到本地 mongodb 数据库，定时验证 ip 可用性。通过将程序写成 windows 服务的方式，让其在后台定时的维护代理 ip 池。

2.2 注释风格分析

程序名	来源	代码行	程序语言	序言性注释	功能性注释
CtrlFunc.py	Github/ProxyipPool	70	Python	1、处理爬虫获得的 ip 地址并存到未验证 ip 的 collection 中 :param collection: Mongodb 的 Collection 对象 :return: none 2、拉取 UnverifiedIP 中的所有 ip 进行验证，然后把可用的 ip 增加到 verifiedIP 中去 Param： uv_collection: UnverifiedIP 的 collection param： v_collection: verifiedIP 的 collection :return: none 3、随意获得一可用 IP	# 删除可用 # 更新所有不可用的检测次数 # 删除超过检测次数限制的 IP # 删除不可用 # 给不可用重新赋予为 0 的检测次数

				:param v_collection: 可用 ipcolleciton return: 可用 ip:port 4、获取 collection 的数据数量 :param collection: :return: 数量	
ProxyipService.py	Github/ProxyipPool	64	Python	1、爬取代理 ip 网页 ip10 分钟爬取一次 UnverfiedIP 中的 ip 没有之后强制 crawl 重新爬取一次 verfiedIP 中的 ip 数量少于 10 之后强制验证 UnverfiedIP 一次验证 UnverfiedIP60 秒一次 验证 verfiedIP 每 20 秒一次	
apschedulerjob.py	Github/ProxyipPool	95	Python	1、读取配置文件链接数据库 :return: 数据库对象	# 爬取代理 ip 网页 JOB # 验证 UnverfiedIP JOB # 验证 verfiedIP JOB # 强制 crawl JOB # 强制验证 UnverfiedIP JOB
getproxyip.py	Github/ProxyipPool	226	Python	1、爬取代理 ip 网页的免费代理 ip 并验证 ip 的可用性 2、uv_ip_list: 未验证可用性 ip 列表 3、通过爬取 ip 代理网页获得 ip :return: 没有经过验证可用性的一 组 IP 列表 4、lock: 互斥锁 useable_ip_list: 可用 ip 列表 unusable_ip_list: 不可用 ip 列表 uv_ip_list: 未验证可用性 ip 列表 5、多线程验证 ip 可用性 :param get: 'useable' = 获取可用的 ip 列表 'unusable' = 获取不可用的 ip 列表	# 西刺代理 (国内高匿和国内普通)42/200 左右 #瑶瑶代理 28/150 左右 # 讯代理 5/10 的可用性 # ip181 25/100 的可用性 # 西刺代理 (国内高匿和国内普通)42/200 左右 # 瑶瑶代理 28/150 左右 √(网站暂时无法访问) # 讯代理 5/10 的可用性 √ # ip181 25/100 的可用性

				:param ip_list: 尚未验证可用性的 ip 列表 不传入参数则为默认的 ip 列表 :return: 返回 get 字符需要的 ip 列表	# 验证 IP 的可用性 # 共享变量是未验证的 IP 列表，通过锁来保护数据的互斥访问
Mongopy.py	Github/ProxyipPool/Mongodb	114	Python	1、连接远程 MongoDB 数据库，加上验证用户权限 :param host: 数据库的 ip-address :param port: 开放连接的端口 :param user: dict 用户名和密码 :return: 数据库对象 2、:param database: 数据库表名 3、封装 pymong 的增加数据函数，增加一条数据和增加多条数据都使用此函数，增加一条数据只需要传入一个 dict，增加多条数据只需要传入一个由 dict 组成的 list :param data: 要增加的数据 :return: 成功返回 true 失败返回 false 4、封装 pymongo 删除数据函数 :param keyword: 要删除的数据的关键字 :return: 成功返回 true 失败返回 false 5、查询所有数据拼接成列表 :return: 返回所有数据组成的列表 6、通过随机数返回数据库内随机的一个元素 :return: 返回一个数据库内元素 7、更新 UnverfiedIP 中的验证次数 :return: 成功返回 true 失败返回 false 8、删除 UnverifiedIP 中	

				的验证次数大于 10 次的 ip :return: 成功返回 true 失败返回 false	

2.2 项目注释风格总结

1、该项目的描述文档十分详细，配备了详细流程图，直观易懂，能让项目开发者以外的人能迅速了解整个项目的流程和功能；

2、代码空格空行习惯良好，每个类中都有序言性注释，部分函数中也有序言性注释，解释了方法的接口参数和返回值，其余关键处也加上了适当的说明性注释，读代码时较为轻松；

3、变量名规范，结合注释易理解。

3、个人代码修改

我的项目代码划分较细，每个模块的功能较为简单，因此在每个方法函数前加上了少量注释，以 login.py 为例：

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'mainwindow.ui'
#
# Created by: PyQt5 UI code generator 5.10.1
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QApplication, QMainWindow
import selectCourse
import sys
import requests
import random
import re
import os

class Ui_MainWindow(object):

    def setSumCookie(self, sum_cookie):
        self.sumCookie = sum_cookie

    def setHeader(self, header):
        self.Header = header

    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(548, 363)
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap("nju/南京大学视觉形象规范化标准(jpeg 格式文件)/01 校标.jpg"), QtGui.QIcon.Normal, QtGui.QIcon.Off)
        MainWindow.setWindowIcon(icon)
        MainWindow.setStyleSheet("background-color: rgb(106, 0, 95);")
        self.centralwidget = QtWidgets.QWidget(MainWindow)
```

```
self.centralwidget.setObjectName("centralwidget")
self.label = QtWidgets.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(160, 40, 261, 41))
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(self.centralwidget)
self.label_2.setGeometry(QtCore.QRect(120, 110, 72, 15))
self.label_2.setObjectName("label_2")
self.label_3 = QtWidgets.QLabel(self.centralwidget)
self.label_3.setGeometry(QtCore.QRect(120, 150, 72, 15))
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(self.centralwidget)
self.label_4.setGeometry(QtCore.QRect(110, 190, 72, 15))
self.label_4.setObjectName("label_4")
self.pushButton = QtWidgets.QPushButton(self.centralwidget)
self.pushButton.setGeometry(QtCore.QRect(330, 230, 131, 28))
self.pushButton.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton.setObjectName("pushButton")
self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
self.pushButton_2.setGeometry(QtCore.QRect(150, 280, 93, 28))
self.pushButton_2.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton_2.setObjectName("pushButton_2")
self.pushButton_3 = QtWidgets.QPushButton(self.centralwidget)
self.pushButton_3.setGeometry(QtCore.QRect(280, 280, 93, 28))
self.pushButton_3.setStyleSheet("background-color: rgb(255, 255, 255);")
self.pushButton_3.setObjectName("pushButton_3")
self.toolButton = QtWidgets.QToolButton(self.centralwidget)
self.toolButton.setGeometry(QtCore.QRect(490, 10, 51, 21))
self.toolButton.setStyleSheet("background-color: rgb(255, 255, 255);")
self.toolButton.setObjectName("toolButton")
self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(180, 220, 141, 41))
self.label_5.setObjectName("label_5")
self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)
self.lineEdit.setGeometry(QtCore.QRect(170, 100, 151, 31))
self.lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);")
self.lineEdit.setObjectName("lineEdit")

# 加载历史输入记录，方便用户使用
f = open('HistoryRecord.txt')
content = f.readlines()
for i in range(0, len(content) - 1):
    content[i] = content[i][:-1]
self.lineEdit.setCompleter(QtWidgets.QCompleter(content))

self.lineEdit_2 = QtWidgets.QLineEdit(self.centralwidget)
self.lineEdit_2.setGeometry(QtCore.QRect(170, 140, 151, 31))
self.lineEdit_2.setStyleSheet("background-color: rgb(255, 255, 255);")
self.lineEdit_2.setObjectName("lineEdit_2")
#将密码输入设置为不可见状态
```

```
        self.lineEdit_2.setEchoMode(2)
        self.lineEdit_3 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_3.setGeometry(QtCore.QRect(170, 180, 151, 31))
        self.lineEdit_3.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_3.setObjectName("lineEdit_3")
        MainWindow.setCentralWidget(self.centralwidget)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        self.pushButton.clicked.connect(self.getValidateCode)
        self.pushButton_2.clicked.connect(self.getAccount)
        self.pushButton_3.clicked.connect(self.clearText)
        self.toolButton.clicked.connect(self.openHelpFile)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "南京大学抢课脚本"))
        self.label.setText(_translate("MainWindow", "<html><head/><body><p><span style=\" font-
size:18pt; font-weight:600; color:#ffffff;\">南京大学抢课脚本</span></p></body></html>"))
        self.label_2.setText(_translate("MainWindow", "<html><head/><body><p><span style=\"
color:#ffffff;\">学号：</span></p></body></html>"))
        self.label_3.setText(_translate("MainWindow", "<html><head/><body><p><span style=\"
color:#ffffff;\">密码：</span></p></body></html>"))
        self.label_4.setText(_translate("MainWindow", "<html><head/><body><p><span style=\"
color:#ffffff;\">验证码：</span></p></body></html>"))
        self.pushButton.setText(_translate("MainWindow", "点击刷新验证码"))
        self.pushButton_2.setText(_translate("MainWindow", "登录"))
        self.pushButton_3.setText(_translate("MainWindow", "取消"))
        self.toolButton.setText(_translate("MainWindow", "帮助"))
        self.label_5.setText(_translate("MainWindow", "TextLabel"))
        #打开登录界面时自动获取最新的验证码
        self.getValidateCode()

    # 获取验证码并获得 cookie，构造 header 供后续功能界面使用
    def getValidateCode(self):
        validatecode = requests.get('http://elite.nju.edu.cn/jiaowu/ValidateCode.jsp')
        picture = open('ValiDateCode.jpg', 'wb')
        picture.write(validatecode.content)
        picture.close()
        png_source = QtGui.QPixmap(r"D:\Python_Project\ui\ValidateCode.jpg")
        self.label_5.setPixmap(png_source)

        #获取 cookie
        cookies = validatecode.cookies
        cookie = str(cookies)
        combine_cookie_1 = re.findall(r'Cookie (.*) for .elite.nju.edu.cn/>, <Cookie yunsuo', cookie)
```

```
        combine_cookie_2 = re.findall(r', <Cookie (.*) for elite.nju.edu.cn/>, <Cookie JSESSIONID',
cookie)
        combine_cookie_3 = re.findall(r'<Cookie yunsuo_session_verify=(\w*) for elite.nju.edu.cn/>,
<Cookie (.*) for',cookie)
        self.setSumCookie(combine_cookie_3[0][1] + '; ' + combine_cookie_1[0] + '; ' +
combine_cookie_2[0] + '; ')

    # 获取账号密码及验证码文本并进行输入是否为空的简单的逻辑判断，并从服务器获取数据来
判断账号的信息的对错
    def getAccount(self):

        #获取账号密码及验证码信息
        userName = self.lineEdit.text()
        #print(userName)
        passWord = self.lineEdit_2.text()
        valiDateCode = self.lineEdit_3.text()

        #检测输入是否为空
        if len(userName) == 0:
            messageBox_1 = QtWidgets.QMessageBox()
            messageBox_1.setWindowTitle("提示")
            messageBox_1.setText("账号输入不能为空！")
            messageBox_1.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_1.exec_()
            return
        elif len(passWord) == 0:
            messageBox_2 = QtWidgets.QMessageBox()
            messageBox_2.setWindowTitle("提示")
            messageBox_2.setText("密码输入不能为空！")
            messageBox_2.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_2.exec_()
            return
        elif len(valiDateCode) == 0:
            messageBox_3 = QtWidgets.QMessageBox()
            messageBox_3.setWindowTitle("提示")
            messageBox_3.setText("验证码输入不能为空！")
            messageBox_3.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_3.exec_()
            return

        user_id = userName + ' 152'
        for i in range(10):
          user_id += str(random.randint(0, 9))

        #构造登录信息
        self.setSumCookie(self.sumCookie + user_id)
        self.setHeader({'Referer': 'http://elite.nju.edu.cn/jiaowu/login.do',
                'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/65.0.3325.181 Safari/537.36',
```

```
            'Cookie': self.sumCookie})
        data = {'userName': userName, 'password': passWord, 'returnUrl': 'null', 'ValidateCode':
valiDateCode}
        r = requests.post('http://elite.nju.edu.cn/jiaowu/login.do', data, headers = self.Header)
        #print(r.text)

        if "验证码错误" in r.text:
            messageBox_4 = QtWidgets.QMessageBox()
            messageBox_4.setWindowTitle("提示")
            messageBox_4.setText("验证码错误！")
            messageBox_4.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_4.exec_()
            self.lineEdit_3.clear()
            return
        elif "验证码已过期" in r.text:
            messageBox_5 = QtWidgets.QMessageBox()
            messageBox_5.setWindowTitle("提示")
            messageBox_5.setText("验证码已过期")
            messageBox_5.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_5.exec_()
            self.lineEdit_3.clear()
            return

        #验证账号密码正确性
        if "用户名错误" in r.text:
            messageBox_6 = QtWidgets.QMessageBox()
            messageBox_6.setWindowTitle("提示")
            messageBox_6.setText("用户名错误")
            messageBox_6.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_6.exec_()
            self.lineEdit.clear()
            return
        elif "密码错误" in r.text:
            messageBox_7 = QtWidgets.QMessageBox()
            messageBox_7.setWindowTitle("提示")
            messageBox_7.setText("密码错误")
            messageBox_7.setIcon(QtWidgets.QMessageBox.Information)
            messageBox_7.exec_()
            self.lineEdit_2.clear()
            return

        # 获取新用户的账号，存入历史记录中
        f = open('HistoryRecord.txt')
        content = f.readlines()
        for i in range(0, len(content) - 1):
            content[i] = content[i][:-1]
        if userName not in content:
            content.append(userName)
```

```
        self.clearText()
        #print(self.Header)
        self.openSelectWindow()

    # 清空文本框内容
    def clearText(self):

        #清空文本输入框信息
        self.lineEdit.clear()
        self.lineEdit_2.clear()
        self.lineEdit_3.clear()

    # 打开帮助文档
    def openHelpFile(self):

        #打开“帮助”文档
        os.popen(r'"%windir%\system32\notepad.exe" Help.txt')

    # 打开下一个界面
    def openSelectWindow(self):
        self.SelectWindow = QMainWindow()
        self.ui_2 = selectCourse.Ui_selectWindow()
        self.ui_2.setupUi(self.SelectWindow)
        self.ui_2.setHeader(self.Header)
        self.SelectWindow.show()
```