

## OOP练习1

```
class Car:
    def __init__(self, c, b, p='无牌'):
        print("__init__ 被调用")
        self.color = c # 颜色
        self.brand = b # 品牌
        self.plate = p # 牌照

    # 定义方法
    def run(self, km):
        print(self.color, '的', self.brand, self.plate, '正在以', km,
              '公里/小时的速度行使')

# 创建对象
car1 = Car('红色', '比亚迪', '京A.88888')
car2 = Car('白色', '宝马', '京B.66666')

# 调用方法 (第1种方式)
car1.run(110)
car2.run(180)

# 调用方法 (第2种方式)
Car.run(car1, 120)
Car.run(car2, 160)
```

## 发送邮件

`email` 模块, 用于准备邮件内容

`smtplib` 模块, 用于与邮件服务器连接并发送邮件

使用本地邮件服务器发送邮件

一, 在本机启动邮件服务并添加邮箱账号

```
]# rpm -q postfix || yum -y install postfix
```

```
]# systemctl start postfix
```

```
]# netstat -utnlp | grep :25
```

```
]# vim /etc/hosts #把自定义的主机名与127地址绑定
127.0.0.1 teacher localhost
```

```
:wq
```

```
#添加邮箱账号
```

```
]#useradd bob
```

```
]#useradd tom
```

## 二、编写脚本

```
]# vim mail.py
```

```
#导入模块
```

```
from email.mime.text import MIMEText
```

```
from email.header import Header
```

```
import smtplib
```

```
msg = MIMEText('python test mail\n','plain' , 'utf8') #准备邮件正文
```

```
#为邮件添加头部信息
```

```
msg['From'] = Header("root",'utf8') # 发件人
```

```
msg['to'] = Header('bob','utf8') #收件人
```

```
msg['Subject'] = Header('py test', 'utf8') #邮件标题
```

```
#发送邮件
```

```
s = smtplib.SMTP('127.0.0.1') #指定邮件服务器
```

```
sender = "root" #指定发件人
```

```
receivers = ["bob","tom"] #指定收件人
```

```
s.sendmail(sender , receivers,msg.as_bytes()) #发送邮件
```

```
:wq
```

## 三、执行脚本

```
]# python3 mail.py
```

## 四、查看用户是否收到邮件

```
[root@teacher ~]# mail -u bob
```

```
Heirloom Mail version 12.5 7/5/10. Type ? for help.
```

```
"/var/mail/bob": 1 message 1 new
```

```
>N 1 =?utf8?q?root?=@teac Tue May 25 12:37 19/651 "py test"
```

```
& exit
```

```
[root@teacher ~]#
```

```
[root@teacher ~]# mail -u tom
```

```
Heirloom Mail version 12.5 7/5/10. Type ? for help.
```

```
"/var/mail/tom": 1 message 1 new
```

```
>N 1 =?utf8?q?root?=@teac Tue May 25 12:37 19/651 "py test"
```

```
& exit
```

```
[root@teacher ~]#
```

## JSON

### JSON(JavaScript Object Notation)

是一种轻量级的数据交换格式，可以对数据做转换，

JSON采用完全独立于语言的文本格式让不同语言之间可以互相识别对方的数据。

### 数据类型对比

Python	JSON
dict	object
list, tuple	array
str	string
int, float	number
True	true
False	false
None	null

### 导入JSON模块

```
import json
```

### 提供4种方法

函数	说明
json.dumps()	编码
json.loads()	解码

```
>>> import json
>>>
>>> dict1 = {'name':'tom' , 'age':20}
>>> type(dict1) #Python字典类型数据
<class 'dict'>
>>>
>>> json.dumps(dict1) #编码
'{"name": "tom", "age": 20}'
```

```

>>>
>>> data = json.dumps(dict1) #json编码后为 字符串
>>> type(data)
<class 'str'>
>>>
>>> json.loads(data) #解码
{'name': 'tom', 'age': 20}
>>>
>>> b=json.loads(data) #json解码后为 字典
>>> type(b)
<class 'dict'>
>>>

>>> json.dumps(101) #编码
'101'
>>> x=json.dumps(101)#解码
>>> json.loads(x)
101
>>>

```

**API : Application Programming Interface**，即应用程序编程接口

- 是一组定义、程序及协议的集合，
- 通过 **API**接口实现计算机软件之间的相互通信。
- **API** 的一个主要功能是提供通用功能集。
- **API**同时也是一种中间件，为各种不同平台提供数据共享。

开放平台：

众多站点将自身的资源开放给开发者来调用  
 站点提供开放统一的**API**接口，帮助使用者访问站点的功能和资源。  
 通过**API**接口,实现不同平台之间的数据共享

天气预报查询

搜索“中国天气网 城市代码”，查找城市代码  
 北京的代码101010100

## 城市天气情况接口

实况天气获取:

<http://www.weather.com.cn/data/sk/城市代码.html>

城市信息获取:

<http://www.weather.com.cn/data/cityinfo/城市代码.html>

详细指数获取:

<http://www.weather.com.cn/data/zs/城市代码.html>

## 城市代码表

<https://blog.csdn.net/wangqjpp/article/details/39957091>

例如:查北京的实况天气

<http://www.weather.com.cn/data/sk/101010100.html>



## requests模块（第3方模块）

模拟浏览器的行为，发送http协议请求，得到服务器响应的数据

## 安装模块

```
pip3 install requests
```

## http协议请求方式:

### 1. GET 请求

相当于`查看`，get请求可以获取网站的数据，请求参数通常跟在URL 的后面

### 2. POST请求

原意是`创建或者添加`，post请求通常用于提交表单或上传文件等

示例：使用requests获取网站数据

```
>>> import requests
>>> url3 = 'http://www.weather.com.cn/data/sk/101010100.html'
>>> r3 = requests.get(url3)
>>> r3.json()
{'weatherinfo': {'city': '\u8c\u97\u2013', 'cityid': '101010100',
'temp': '27.9', 'WD': '\u8d\u97\u2013\u8e', 'WS': '\u2013\u8f\u2013\u8e3\u2013',
'SD': '28%', 'AP': '1002hPa', 'njd':
'\u2013\u9a\u82\u97\u2013\u2013\u9e\u86\u2013', 'WSE': '<3', 'time': '17:55',
'sm': '2.1', 'isRadar': '1', 'Radar': 'JC_RADAR_AZ9010_JB'}}
>>>
>>> r3.encoding #查看字符编码
'ISO-8859-1'
>>>
>>> r3.encoding='utf8' #指定字符编码
>>> r3.json()
{'weatherinfo': {'city': '北京', 'cityid': '101010100', 'temp':
'27.9', 'WD': '南风', 'WS': '小于3级', 'SD': '28%', 'AP': '1002hPa',
'njd': '暂无实况', 'WSE': '<3', 'time': '17:55', 'sm': '2.1',
'isRadar': '1', 'Radar': 'JC_RADAR_AZ9010_JB'}}
>>>
>>> data = r3.json()
>>>
>>> type(data)
<class 'dict'>
>>>
>>> data
{'weatherinfo': {'city': '北京', 'cityid': '101010100', 'temp':
'27.9', 'WD': '南风', 'WS': '小于3级', 'SD': '28%', 'AP': '1002hPa',
'njd': '暂无实况', 'WSE': '<3', 'time': '17:55', 'sm': '2.1',
'isRadar': '1', 'Radar': 'JC_RADAR_AZ9010_JB'}}
>>>
```

```

>>> url2 = '图片文件的网址'
>>> r2 = requests.get(url2)
>>> r2.content #返回的是字节串

>>> with open("/root/a.gpg","wb") as fobj:
...     fobj.write(r2.content)
...
22343
>>> exit()

]# eog /root/a.gpg

```

## 钉钉机器人

### 1 访问网址 后扫码登录

<https://im.dingtalk.com/> 钉钉机器人网页版

### 2 添加机器人获取API

<https://oapi.dingtalk.com/robot/send?>

access\_token=4bad221c7a620f8cb47ddfc5abe7e29c5a0d144e238f351ea08b655a61a9c725

### 3 编写代码

```
]# vim dingtalk.py
```

```
import requests
```

```
import json
```

```
url = 'https://oapi.dingtalk.com/robot/send?
```

```
access_token=4bad221c7a620f8cb47ddfc5abe7e29c5a0d144e238f351ea08b655a61a9c725'
```

```
headers = {'Content-Type': 'application/json; charset=utf-8'}
```

```
data = {
```

```
    "at": {
```

```
        "atMobiles": [
```

```
            #"180xxxxxx"
```

```
        ],
```

```
        "atUserIds": [
```

```
            #"user123"
```

```
        ],
```

```
        "isAtAll": True
```

```
        },
        "text": {
            "content": "达内我就是我, @xxx 是不一样的烟火"
        },
        "msgtype": "text"
    }
}
```

```
r = requests.post(url, headers=headers, data=json.dumps(data))
print(r.json())
```

#### 4 执行脚本

```
[root@bogon ~]# python3 dingtalk.py
{'errcode': 0, 'errmsg': 'ok'}
[root@bogon ~]#
```

#### 5 查看群消息