1. **爬取主题说明**
2. 背景  
    在本次phthon程序课程设计中，需要完成的是网络爬虫，在网站上爬取自己想要的信息。我选择的主题是：游戏。在我们日常生活中，游戏已经随处可见。电脑游戏、手机游戏、switch……无论是在地铁、奶茶店，还是电竞店，都有游戏的“身影”。  
    在每个人的成长过程中，父母都会教导我们少玩游戏多读书。沉迷游戏自然不好，但是游戏是所有哺乳动物，特别是灵长类动物学习生存的第一步。过家家、捉迷藏这些都是我们的童年游戏，现在我们所说的“游戏”，普遍指的是电子游戏。 游戏是一种基于物质需求满足之上的，在一些特定时间、空间范围内遵循某种特定规则的，追求精神世界需求满足的社会行为方式，但同时这种行为方式也是哺乳类动物或者灵长类动物所需的一种降压减排的方式，不管是在出生幼年期，或者发育期，成熟期都会需要的一种行为方式。所以这就是为什么电子游戏会覆盖大多数年龄段的原因。只要我们不沉迷于游戏，合理适度的游戏允许人类在模拟环境下挑战和克服障碍，可以帮助人类开发智力、锻炼思维和反应能力、训练技能、培养规则意识等。  
    大数据时代数据获取的方式，其中热门的就是爬虫。爬虫简单来说就是用来爬取网页数据的程序。虽然java等语言也可以制作爬虫，但是python爬虫相对简单，相同的功能用更少的代码就可以实现。这就是python爬虫火遍全球的原因。随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。搜索引擎作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性，如：不同领域、不同背景的用户往往具有不同的检索目的和需求，通过搜索引擎所返回的结果包含大量用户不关心的网页；通用搜索引擎的目标是尽可能大的网络覆盖率，有限的搜索引擎服务器资源与无限的网络数据资源之间的矛盾将进一步加深；万维网数据形式的丰富和网络技术的不断发展，图片、数据库、音频、视频多媒体等不同数据大量出现，通用搜索引擎往往对这些信息含量密集且具有一定结构的数据无能为力，不能很好地发现和获取；用搜索引擎大多提供基于关键字的检索，难以支持根据语义信息提出的查询。使用网络爬虫可以很好地解决以上问题。
3. 数据来源  
   我选择的游戏主题来源为英雄联盟，url是：<https://lol.qq.com/main.shtml>  
   我选择英雄联盟的原因是英雄联盟是比较热门的端游，在LPL夺冠以后引起大众的欢呼和兴奋，可见英雄联盟饱受喜欢。
4. 数据项  
   爬取的数据项有：英雄名称、英雄皮肤名称、英雄皮肤图片
5. **技术可行性分析**
6. 技术方案  
   requests库：用来发出http请求，如get获取信息；  
   response：解析返回的数据；  
   os库：操作系统接口模块，用作数据的保存，文件夹的创建；

pprint库：格式化文档，可以让打印出的数据更直观；

re库：正则表达式，可以把爬取下来的数据带有不可保存的符号转换为指定符号

1. 相关依赖包说明  
   ①PyMySql:用作连接mysql数据库，把在网页上爬下的图片信息存到数据库中  
   ②cursor:数据库的交互对象类
2. **详细设计及实现**  
   1、确定需要爬取的网页地址，在网页地址中找到想要爬取的英雄信息，在浏览器中按下F12，找到图片的来源 ，即request URL,这个就是信息来源。  
   2、添加一个请求头，用作伪装浏览器发送请求。  
   3、向网站发送信息获取请求（get）  
    response = requests.get(url=url, headers=headers)

此时去网页上查看网页的请求方式是get，所以使用get获取信息。

4、在获取到的信息中，解析出英雄名称、英雄皮肤图片

hero\_list = response.json()['hero']

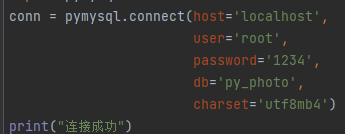
在爬取到的数据中，解析出hero内的内容，使用for循环，找出所有hero的英雄姓名。经分析发现，每一个英雄有独立的url，url的唯一区别就是url最后的数字，如第一个英雄的数字是1，第二个英雄的数字是2。

5、使用循环遍历出所有的英雄皮肤信息，然后解析数据中的skins（皮肤）信息

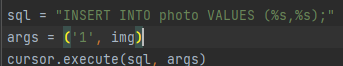
skins = response\_1.json()['skins']

再使用for循环，找到skins中的皮肤图片地址、英雄名字

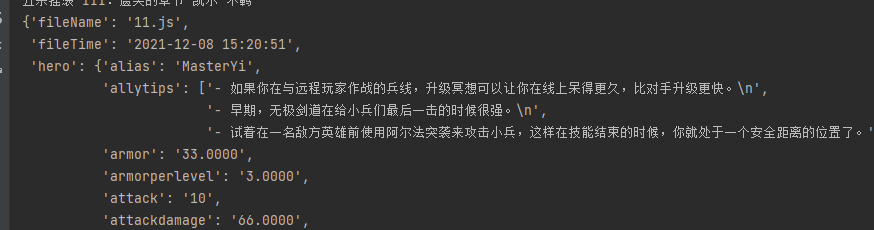
6、把爬取到的数据保存在本地。编写save函数，首先判断保存的文件地址是否存在英雄名称的文件夹，没有则创建文件夹。然后就使用request请求，获取图片的内容，即二进制数据，以jpg的文件格式保存在本地。

7、将爬取的图片文件保存到数据库。首先导入pymysq库，使用connect连接到已经创建好的数据库，如图（1）所示

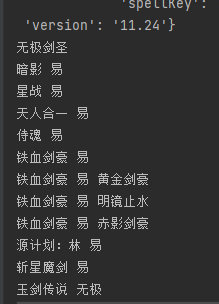
图表 1

接着找到需要上传数据库的文件地址，使用sql语句插入到数据库的表中。如图（2）

图表 2

1. **运行测试**代码编写好后，运行时，首先打印出当前爬取英雄的所有数据（如图3）  
   

图表 3

接着爬取下来后，print打印出所爬取的英雄图片名称（如图4）：  


图表 4

在所有英雄爬取完成后， 便可以在本地查看爬取到的英雄文件夹，文件夹里保存了该英雄的皮肤图片，如图5、图6：  


图表 5



图表 6

最后运行存入数据库代码，把图片存到数据库中，如图7：



图表 7

1. **总结**

在本次python爬虫课程设计中，在这一个月内通过在网上的学习，完成了python网络爬虫实训。

要做一只爬虫，首先就得知道他会干些什么，是怎样工作的。所以得有一些关于HTML的前置知识，这一点做过网页的应该最清楚了。我选的课程有前端工程，所以读懂html不是什么难题。了解了这个然后还得了解网页和服务器之间是怎么通信的，这就得稍微了解点HTTP协议，基于TCP/IP的应用层协议，规定了浏览器和服务器之间的通信规则，简单粗暴的介绍几点和爬虫相关的就是：浏览器和服务器之间有如下几种通信方式，GET，向服务器请求资源，请求以明文的方式传输，一般就在URL上能看到请求的参数；POST，从网页上提交表单，以报文的形式传输，请求资源。

了解基本过程后，就可以照着网上的方法来进行我们所需要爬的数据了。由于是python小白，要求自己独立写一个爬虫还是有些困难。还好可以参考网上的教程，进一步完成python设计。首先是进行内容选取，我选择了英雄联盟为爬取对象，英雄联盟是一款电脑游戏，可以爬取游戏英雄的名称和英雄皮肤，比较直观。去到英雄联盟官网，查询到英雄的图片信息来源后，再查看网页源代码，由于学过前端，所以看懂网页源代码不是大问题。先从总体入手，即找到所有英雄的总介绍页面，再到具体英雄的介绍，最后获取单个英雄的信息。从这里学到了从粗到细来查询信息。找到信息来源后，就可以开始信息获取了。

在获取英雄皮肤名称的过程中，程序运行的途中发生了错误。经查询错误原因，结果是保存的文件名称不可包含特殊符号。所以在网上查询资料，可以使用正则表达式，把特殊符号统一转换为“\_”,错误便修复成功。接下来又遇到了出错，查询原因原来是皮肤有两种，一种是普通皮肤，一种是炫彩皮肤，炫彩皮肤需要用另外的名称获取，所以修改了代码，使用if-else语句判断，如果是普通皮肤，则下载普通皮肤，否则下载炫彩皮肤，这样就再次解决了出现的报错。

在完成了英雄皮肤的爬取后，在文件夹中看到了所有的英雄皮肤，还是挺开心的。用了一种新方式保存图片，还是挺意外的。接着我又查找教程，试着把图片存进MySQL数据库。了解了具体过程后，便开始编写代码。首先用代码连接数据库，再获取图片的文件地址，最后输入SQL语句，insert进数据库中，就可以把图片存进数据库中了。又完成多了一个功能，程序的完善性又近了一步。

最后完成了python网路爬虫程序设计，让我感受到了python的强大之处，简单的几十行代码就可以完成相对复杂的功能。怪不得有人“调侃”：python永远滴神！

附加：程序代码  
  
主程序：  
import requests #请求

import pprint #打印

import os #导入操作系统接口模块（内置）

import re #正则表达式

url = 'https://game.gtimg.cn/images/lol/act/img/js/heroList/hero\_list.js' #需要获取图片信息的地址

headers = {

'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36 Edg/96.0.1054.43' #加入伪装

}

def change\_title(title): #修改爬取的内容中出现的不能保存的符号

mode = re.compile(r'[\\\/\:\\*\?\"\<\>\|]')

new\_title = re.sub(mode, '\_', title) #把上面出现的符号转换为“\_”

return new\_title

def save(title,name,img\_url): #图片保存函数

filename = f'{title}\\' #保存到独立的英雄文件夹

if not os.path.exists(filename): #if判断是否存在文件夹

os.mkdir(filename) #创建文件夹

img\_content = requests.get(url=img\_url, headers=headers).content #获取图片内容，即二进制数据

with open(filename + name + '.jpg', mode='wb') as f:

f.write(img\_content) #写入图片数据

print(name)

response = requests.get(url=url, headers=headers) #解析数据，获取英雄数据信息

# pprint.pprint(response.json())

hero\_list = response.json()['hero'] #返回解析数据中hero的信息

for index in hero\_list: #使用for循环，遍历出所有的英雄名称（heroId）

# pprint.pprint(index)

hero\_id = index['heroId']

hero\_url = 'https://game.gtimg.cn/images/lol/act/img/js/hero/{}.js'.format(hero\_id) #format格式化函数，把hero\_id传入

response\_1 = requests.get(url=hero\_url, headers=headers) #获取英雄图片信息

pprint.pprint(response\_1.json())

skins = response\_1.json()['skins'] #返回解析数据中的skins信息（皮肤信息）

for lol in skins:

img\_url = lol['mainImg'] #皮肤图片地址

title = lol['heroTitle'] #英雄名字

name = lol['name'] #皮肤名字

new\_name = change\_title(name)

new\_title = change\_title(title)

if img\_url:

save(new\_title, new\_name, img\_url)

else:

chroma\_img = lol['chromaImg'] #获取炫彩的皮肤

save(new\_title, new\_name, chroma\_img)

存入数据库代码：  
import pymysql

conn = pymysql.connect(host='localhost',

user='root',

password='1234',

db='py\_photo',

charset='utf8mb4')

print("连接成功")

cursor = conn.cursor()

fin = open(r"C:\Users\13287\PycharmProjects\pythonProject\1\阿卡丽\离群之刺.jpg", 'rb')

img = fin.read()

fin.close()

sql = "INSERT INTO photo VALUES (%s,%s);" #将数据插入到mysql数据库中，指令

args = ('1', img)

cursor.execute(sql, args) #执行相关操作

conn.commit() #更新数据库

#print(2)

cursor.close()

conn.close()