# 爬取番茄小说源码函数接口文档

*#把数据写入csv文件，这个是适用于首页的（无需解密）*

save\_csv(book\_dict, save\_path='./fanqie/',name='data.csv')

book\_dict：json类型的数据

save\_path：保存路径默认为'./fanqie/'

name：保存的文件名称

无返回值

*#保存排行榜数据的csv，区分上面是因为存储的json格式不一样*save\_csv\_phb(book\_dict, save\_path='./fanqie/',name='data.csv')

book\_dict：json类型的数据

save\_path：保存路径默认为'./fanqie/'

name：保存的文件名称

无返回值

*#筛选数据，分析小说类型和数量*Type\_Summary(filename)

Filename是csv数据路径，用于绘图分析

返回 data\_Type\_Counts

*#绘图分析小说类型*Draw\_BarChrat(data\_Type\_Counts)

绘图函数，绘制完返回1

*#用于获取response后返回解密的response*get\_request(url,headers,dict\_data)

url：目标地址

headers：头文件

dict\_data:用于解密的字典（因为每个页面加密的文字woff2文件都不一样）

返回解密后的json数据

*#展示排行榜上的书籍*show\_books(j1,page\_count)

J1:json类型数据

Page\_count：展示排行榜前多少个的数据

无返回值

*#把小说封面爬取下来*download\_books\_images(j1, headers, save\_path='./fanqie/')

J1：解密后的json数据

Headers：响应头文件

Save\_path: 保存路径

无返回值

*#保存小说图片*def save\_book\_img(j1,j,save\_path='./fanqie/')

J1：解密后的json数据

J：用于规定下载哪一个的小说封面参数

Save\_path：保存的路径，默认为'./fanqie/'

无返回值

*#保存小说内容*save\_book\_text(j1,j,save\_path='./fanqie/')

J1：解密后的字典类型数据

Save\_path：保存路径，不填写默认为'./fanqie/'

执行完毕函数返回1

*#主函数路口*if \_\_name\_\_ == '\_\_main\_\_'