**利用selenum及requests进行百度图片自动化获取**

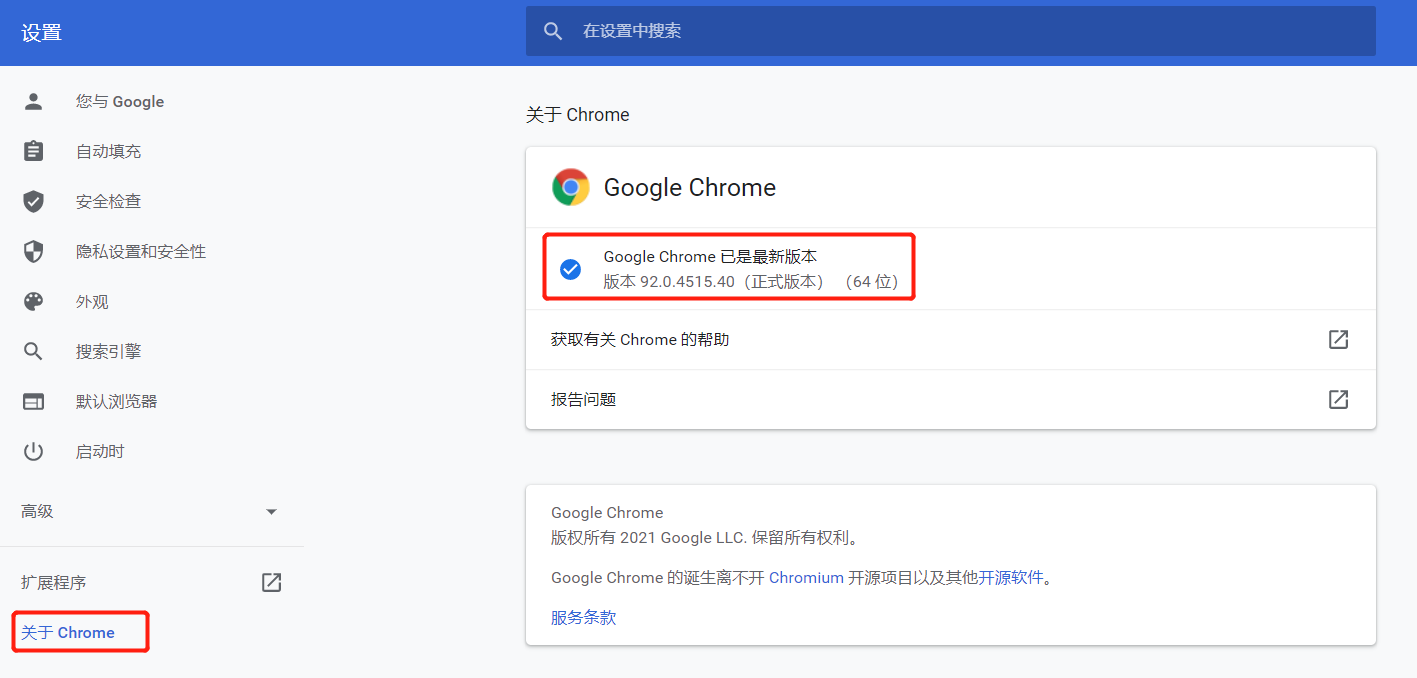
##### 环境准备

Python3.0版本及requests和selenium第三方包；

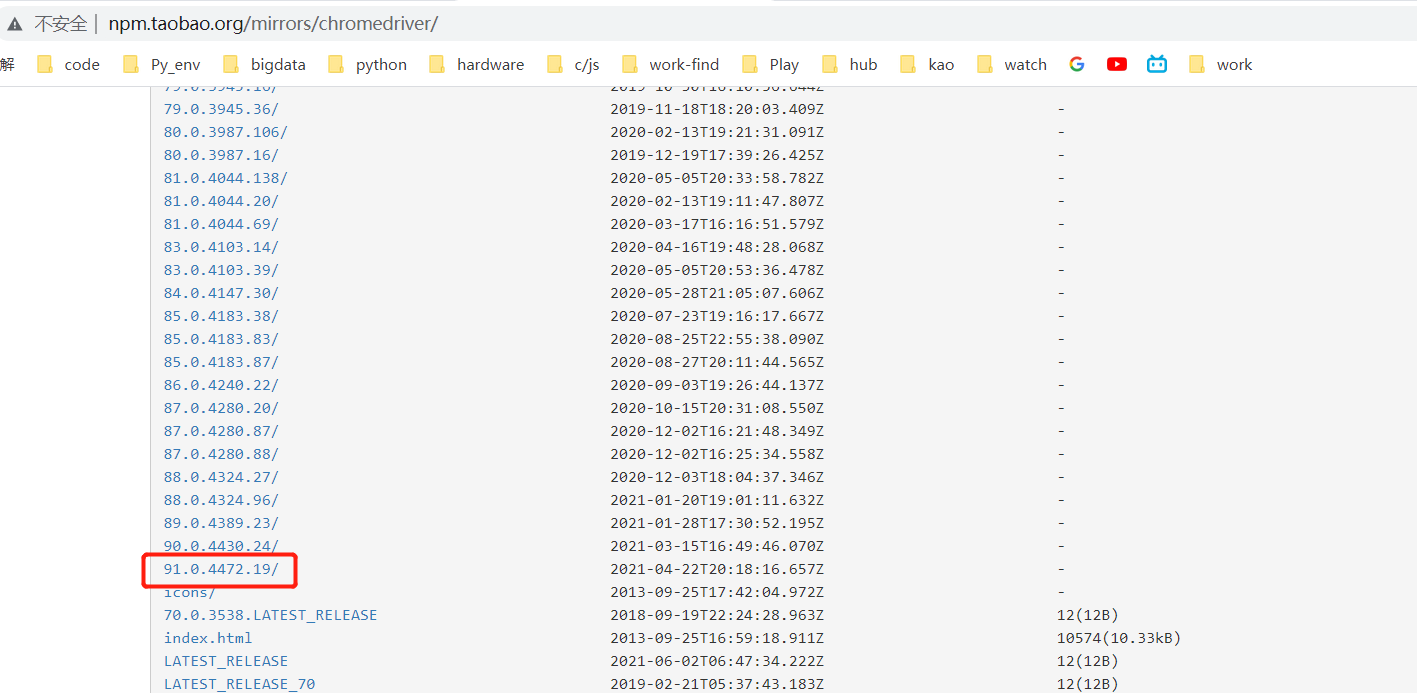
与chrome浏览器相匹配的chromedriver包

* 1. chromedriver获取

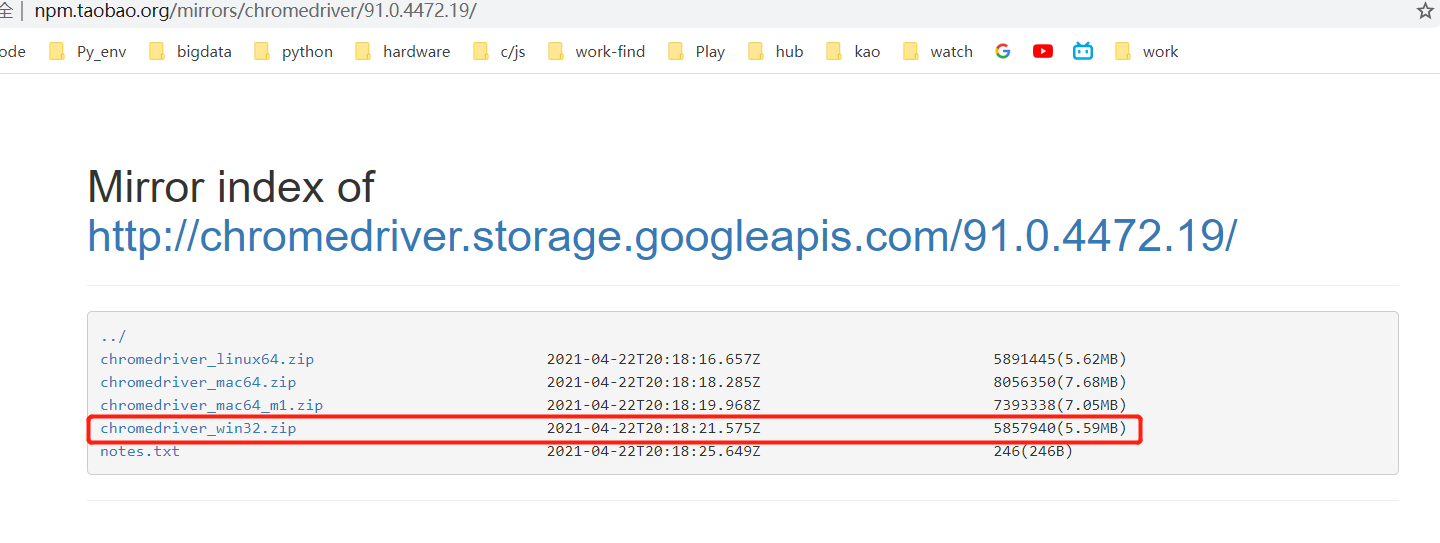
1. 在chrome浏览器设置中查看chrome版本



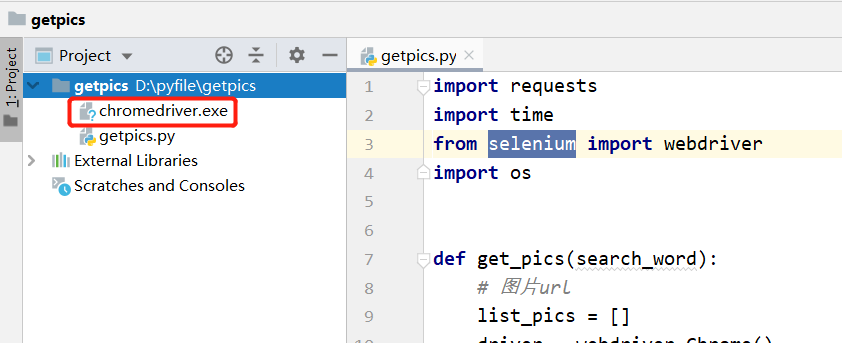
1. 在 <http://npm.taobao.org/mirrors/chromedriver/> 网站中下载与浏览器对应版本的chromedriver包（相差一个版本区别不是太大）



1. 选择与操作系统相匹配的chromedriver下载



1. 将下载后的chromedriver包拖拽至代书写脚本的同级目录下



至此，准备工作完成，可以开始书写脚本

##### **脚本书写**

2.1、导入需要用到的包，调用selenium启动chrome浏览器去访问百度图片地址，找到搜索框并键入搜索词，进行搜索；执行js代码让浏览器往下滑动页面，加载出更多的图片（原因：百度图片的xpath一页有21张图片，需要滑动多页来加载更多的图片，方便后续通过单张图片的xpath来获取图片url）



2.2、通过查看网页源码发现其xpath地址：{ //\*[@id="imgid"]/div[j]/ul/li[i]/div/a/img } ，有如下规律，div[j]可以控制其显示的页数，li[i]可以控制每页的第几张图片（从0到21），故而通过双层循环来获取单张图片的url地址，为后面下载图片做准备；

内层循环：从1到20来获取每页的图片，下图可获取每页20张图片（可自行更改）

外层循环：获取图片页数，下图是获取5页图片（可自行更改）

代码中加入了try...except...增加了代码的健壮性，原因是：百度图片某些搜索词，会出现一些广告图片，占用了目标图片的xpath地址，导致获取图片url失败，故而当遇到广告图片时，跳过图片xpath，以获取目标图片



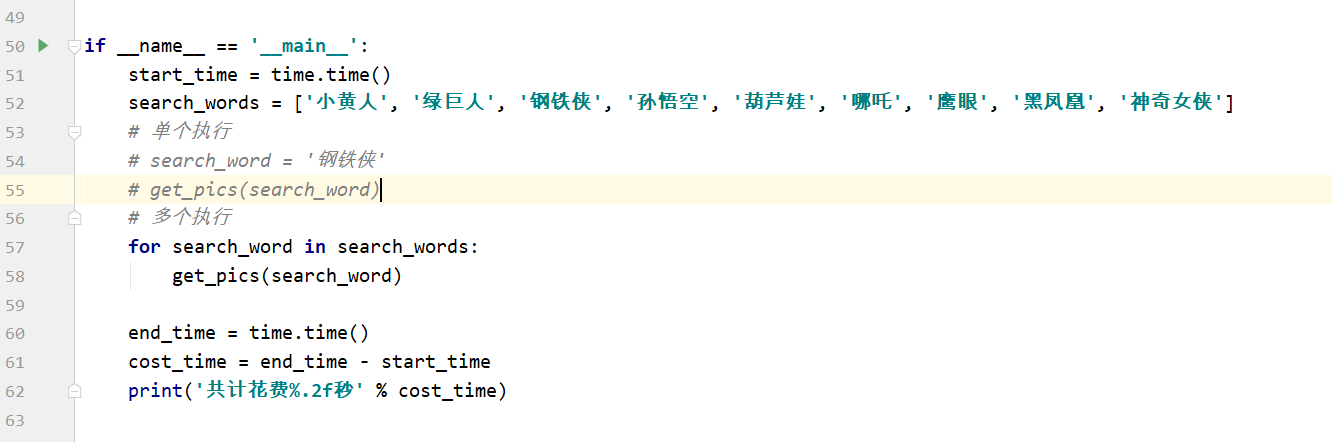
2.3、获取到单张图片url后，加到了列表里，然后退出浏览器

2.4、在当前文件夹下创建了一个以搜索词为命名的文件夹

从列表中取出单张图片url，调用requests发起get请求获取图片，并将图片以.jpg形式保存到搜索词文件夹下



2.5、运行主函数，可选择单个搜索词去获取图片，也可以将多个搜索词放入列表中，获取多个搜索词图片



2.6、运行效果



##### **脚本详情见插入附件**



注意事项：若再次搜索同名搜索词，需先将目录下已有搜索词文件夹删除！！！

完结，撒花^-^