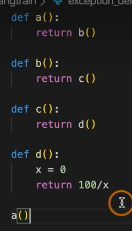
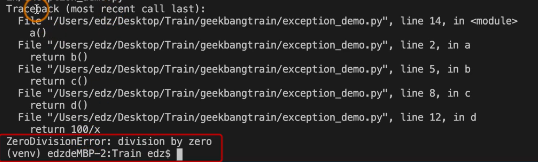
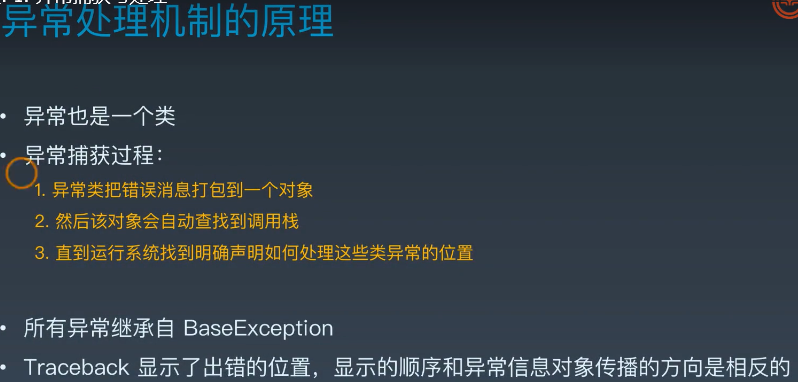
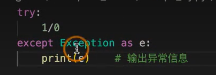
所有异常通过traceback函数追踪

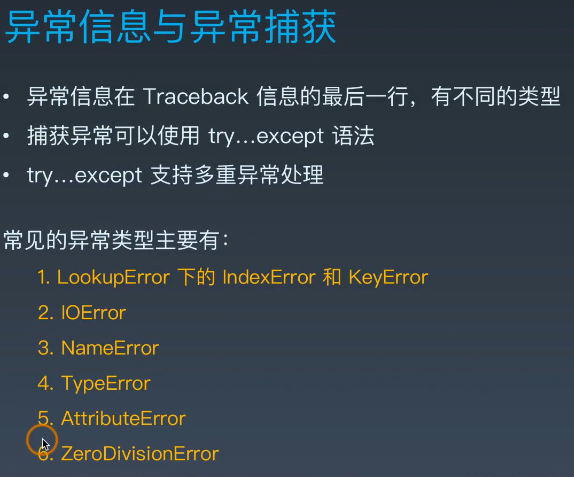




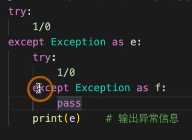
Abcd四处均可捕获



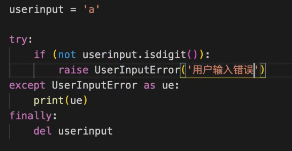
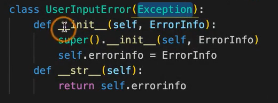
Exception能捕获所有异常，并print输出



捕获异常嵌套，捕获链上已经捕获的 异常不会再捕获了。



自定义异常：

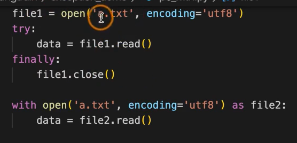


\_\_init\_\_相当于构造函数，定义了\_\_str\_\_后类就可以像字符串一样去使用。

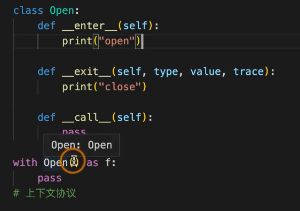
Pretty\_errres:第三方库 美化异常



文件异常：with写法更加优雅



自定义with类：有自带的enter和exit方法自动调用。



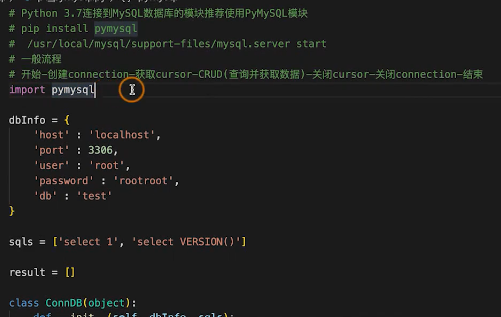
使用PyMySQL进行数据库操作

JAVA ORACLE,PYTHON MYSQL MANGO

先安装，成功后Mysql sever start 启动服务

Ps -ef |grep mysql查看是否有mysql进程，检查服务是否启动

Python连接mysql需要安装第三方库，建议用PyMySQL



数据库配置专门建一个独立文件，方便共享开源，以免泄露信息。

反爬虫：模拟浏览器的头部信息

发起请求时带浏览器看不到的头部信息：



第一种方式：F12抓包，直接把整个header复制过来

Referrer防跨站

如果请求的链接地址是加密的，可用模拟浏览器工具,WebDriver。

第三方库：fake\_useragent

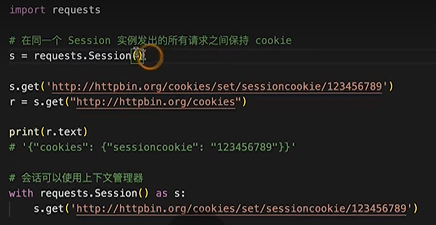
它会从网络中去请求目前常用订单一些浏览器，不去进行ssl验证，否则经常出现下载失败



可用写进scrapy的settings.py文件中

有的网站会验证useragent、referer（从哪个网址链接而来）、host（主机，大型网站集群中比较常见）

用户名和密码登录采用post方式请求，并会在本地的Cookies里面保存一份加密的用户名、密码及有效期。



用cookies模拟用户登录代码如下：







用webdriver模拟用户点击浏览器行为

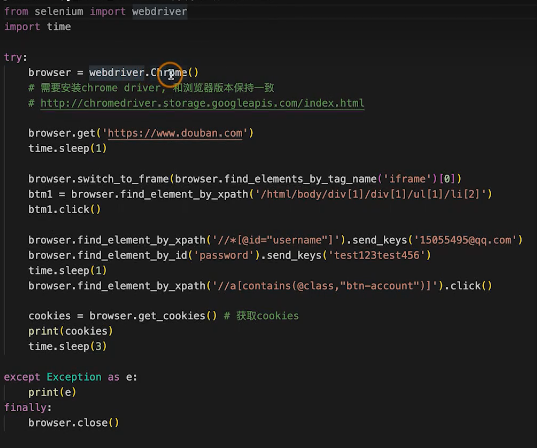
**WebDriver 文档：**

* <https://www.w3.org/TR/webdriver/>
* <https://www.selenium.dev/selenium/docs/api/py/>

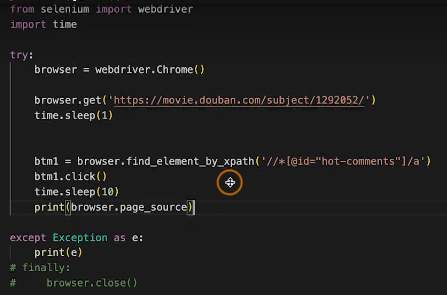
**ChromeDriver 下载地址：**  
<https://chromedriver.storage.googleapis.com/index.html>

Mirror index of <http://chromedriver.storage.googleapis.com/>

要安装包，并且要安装相应浏览器的驱动，下载自己对用的版本，并把Chromedriver放到和你当前解析的python相同的目录中，（在cmd用where python可以查看python解析器的位置）。通过他可以让python和浏览器联动。

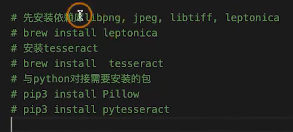


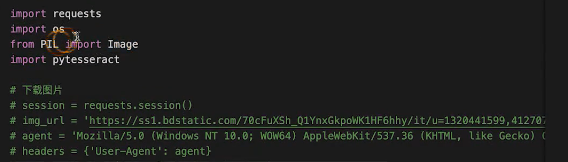
加密的javascript时，可以用这样方式，如下代码：

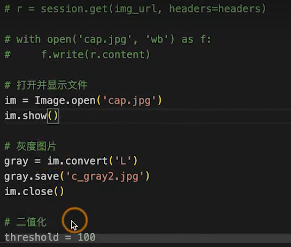


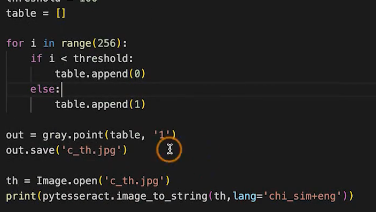
反爬虫：验证码识别

简单识别码：图形校验码









各种语言的识别库：

<https://github.com/tesseract-ocr/tessdata>



爬虫中间件和系统代理IP

执行scrapy加参数：--nolog，不打印日志

中间件有优先级，去从小到大，回来从大到小

