Topic1：如何解决requests过程中因网络问题断链后重新开始爬虫程序导致的重复下载问题：

*if not* os.path.exists("deep\_scrapping\_html/" + str(file\_name) + ".html"):

*coding……*

*else*:  
 print('file exist')

判断要下载的文件是不是已经存在，若存在则不下载

Topic2：创建文件后因各种原因数据没有写入到文件中，下次运行程序时，虽能判断文件已经存在，但无法判断文件内是否有写入数据：

f = open("deep\_scrapping\_html/" + str(file\_name) + '.html.temp',"w",encoding="utf-8")

.

.

.

f.close()

os.rename("deep\_scrapping\_html/" + str(file\_name) + '.html.temp',"deep\_scrapping\_html/" + str(file\_name) + '.html')

第一次创建文件时，创建为临时文件，若程序中选，再运行时，可以判断到该文件没有被创建，则会重新下载或解析该文件

Topic3：创建稳定的函数方法，可以使用assert断言的方式规定传入函数的特性，如果不满足要求，系统会自动报错

*def* predict(self,*x\_test*):  
 '''断言中规范了x测试集必须是n行2列的矩阵'''  
  
 *assert x\_test*.shape[1] == 2, \  
 'x\_test must be a n by 2 matrix. Use ".reshape(len(x\_test),2)" to reshape your matrix'  
 *assert* self.\_x\_train *is not None and* self.\_y\_train *is not None*, \  
 'you should fit the machine before prediction!'  
 self.\_x\_test = *x\_test*

Topic4: 合并路径时，如果单纯的使用‘\’合并路径，会导致不同该操作系统中无法兼容。我们要使用os.path.join：

file\_path = os.path.join(self.path, hello.csv)

效果就是：D:\Yan\hello.csv