以下是web请求前的准备工作

-1.在[main.py](main.py)中,pd.DataFrame.from()函数，未调用api，生成一个含有所有省份和直辖市简要资料的tsv档[fsnd\_data.tsv](fsnd\_data.tsv)保存到本地的[data](data)文件夹中。pd.DataFrame.from()函数，读取[fsnd\_data.tsv](fsnd\_data.tsv)并返回省份或直辖市简要资料的字典。用pd.DataFrame.from函数()，调用字典dict\_champion，以集合推导的方式，建立以id号为键，省份英文名为值的小字典to\_dict(见代码 data\_list = [v for k,v in data.items()])。跑含有code号的省份或者直辖市 , 并把每个code号下的省份或者直辖市以其英文为文件名，输成json档 [province\_code\_name.json]( province\_code\_name.json ) ，最终结果为34个省份或者直辖市，保存到本地的[data](data)文件夹中。

+1.在[main.py](main.py)中,pd.DataFrame.from()函数，未调用api，生成一个含有所有省份和直辖市简要资料的tsv档[fsnd\_data.tsv](fsnd\_data.tsv)保存到本地的[data](data)文件夹中。pd.DataFrame.from()函数，读取[fsnd\_data.tsv](fsnd\_data.tsv)并返回省份或直辖市简要资料的字典。用pd.DataFrame.from函数()，调用大数据pandas(pd.DataFrame.from)，以集合推导的方式，建立以code号为键，省份中文名为值的小字典to\_dict(见代码 data\_list = [v for k,v in data.items()])。跑含有code号的省份或者直辖市 , 并把每个code号下的省份或者直辖市以其中文code为文件名，输成json档 [province\_code\_name.json]( province\_code\_name.json ) ，最终结果为34个省份或者直辖市，保存到本地的[data](data)文件夹中。

2.在[datatemp.py ](datatemp.py )中，def get\_data()函数，打开[province\_code\_name.json ](province\_code\_name.json )，返回一个以省份或者直辖市的代码为键，中文名和代码为值的字典(见代码temp = df[['reg','sj','data']].set\_index('reg').to\_dict()['data'])，用做后面的函数和网页的表单界面，目标是用户输入英省份或者直辖市的中文名能得到想要的结果。def get\_data()函数，将省份的中文名输入转换成互联网普及率，调用函数def get\_data()，运用for循环，如（int\_pr = df.query("zb=='A0G0H05'")[['reg','sj', 'data']]）,从中可以得到该省份或者直辖市的代码、时间、数据.如果输入的名称不存在,返回'error'字符串.