大数据概论 #2.1

安装配置 & 一些规范和 tips & Git 简单入门

05.23

准备

安装：Anaconda: Jupyter Notebook / Pyecharts

一些 tips 和代码规范

Git 的基础用法

安装：Anaconda:
Jupyter Notebook

https://sspai.com/post/61799
其中包含了一些常用指令

安装：Pyecharts

```
pip install pyecharts
```

数据源文件

address	2022-03-22_fangcang	2022-03-23_fangcang	2022-03-24_fangcang	2022-03-25_fangcang	2022-03-26_fangcang	2022-03-27_fangcang	2022-03-27_yiyuan	2022-03-28_fangcang	2022-03-28_yiyuan
PudongNew	0	0	0	0	0	0	84	0	119
Huangpu	0	0	0	0	0	0	14	0	14
Xuhui	0	0	0	0	0	0	28	0	31
Changning	0	0	0	0	0	0	3	0	10
Jingan	0	0	0	0	0	0	21	0	30
Putuo	0	0	0	0	0	0	11	0	14
Hongkou	0	0	0	0	0	0	5	0	8
Yangpu	0	0	0	0	0	0	4	0	3
Minhang	0	0	0	0	0	0	56	0	63
Baoshan	0	0	0	0	0	0	17	0	25
Jiading	0	0	0	0	0	0	32	0	31
Jinshan	0	0	0	0	0	0	2	0	3
Songjiang	0	0	0	0	0	0	2	0	6
Qingpu	0	0	0	0	0	0	1	0	3
Fengxian	0	0	0	0	0	0	2	0	1
Chongming	0	0	0	0	0	0	2	0	2
Shanghai	4408	5170	6550	8674	11095	14376	284	18531	363

修改成 pandas.dataFrame()

shanghaiMedical =

	date	fangcang	hospital
0	2022-03-09	3	0
1	2022-03-10	4	0
2	2022-03-11	543	0
3	2022-03-12	65	0
4	2022-03-13	54	34
5	2022-03-14	43	39
6	2022-03-15	32	67
7	2022-03-16	213	112
…	…	…	…

将 dataFrame 导入 Pyecharts

Pyecharts 原生不支持 dataFrame，需要处理。

```
date = df['date'].tolist()
fangcang = df['fangcang'].tolist()
hospital = df['hospital'].tolist()
line = (Line()
        .add_xaxis(date)
        .add_yaxis('', fangcang)
        .add_yaxis('', hospital)
       )
```

关于地图 API

- 注册百度地图开发者，获取百度应用 AK 序列号
https://lbsyun.baidu.com/apiconsole/key#/home
如果觉得麻烦，我已经注册好了
AK：ivMnHuSCrEGDtGrEIXP5oOtbqg1S1KZ7

样例代码将传到 GitHub（可以参考，如果有更好的方案更好啦）

Git 入门

初始化步骤

如果你从未使用过 Git，请做以下步骤：(Windows)

1. 安装 Git https://blog.csdn.net/qq_40903378/article/details/121028609
2. 配置 SSH 密钥 https://blog.51cto.com/heboyme/3197921（Windows）
3. 将 SSH 密钥和你的 GitHub 连接

https://docs.github.com/cn/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account

本项目需要用到的 Git 命令

第一次下载：

`git clone <SSH链接>`

将 Git bash here 换成项目目录——然后

`git checkout dev`

每天开始写代码前：

`git pull origin dev` 从远程仓库下载最新代码（dev 分支）

每次提交代码时：

`git status` 查看在你上次提交之后是否有对文件进行再次修改

`git add .` （注意左边有个点）将所有文件添加到 git

`git commit -m “注释”` 在本地提交

`git push origin dev` 推送到云端代码库

当 push 无法成功

push 的前提条件是，你的 Git 版本和云端相同。

当你写完代码需要提交时，发现 push 不成功，一般原因是别人和你一同修改了文件并且上传到了云端。

所以，此时需要做的是：

git pull origin dev 先将云端的文件下载,此时客户端会提示你哪些文件改变了，你根据他的提示选择该文件的版本；然后修改完毕后重新上一页的“提交操作”。

其他常用 Git 命令

主要和分支、合并有关：

`git checkout <分支>`

`git branch`

`git merge <分支>`

我们的项目有 *dev* 和 *master* 两个分支，大家开发都选择 *dev* 分支，我们每次阶段性工作结束合完代码我会把它合并到 *master* 分支作为主版本。