

# Flask 基础 (1)

Zhenghui Wang \*

*International School, Beijing University of Posts & Telecommunications*

2020 年 10 月 26 日

## 1 动因与概述

我在暑期使用 Axure 原型设计软件做出过一个个人主页的界面，当时的 html 文件是软件一键生成的，因此我只需要考虑排版问题即可。但随后我发现这种界面是不可靠的，因为每次我想增加一些新的东西进去或是修改一些内容时，我都要先对整个页面重新排版，然后再部署到服务器上才能显示，步骤非常繁琐、效率极低；且用这种方式做出的网站，几乎没有产生任何交互，简单来说就是网站没有后端。

为了解决这一问题，同时也为了积累一些 web 网站搭建的知识，我考虑利用 Flask 环境重新设计主页。在我开始以前，有很多系统的 web 框架供我挑选，其中 Java 的 Spring 以及 Python 的 Django 都是很好的选择。我近期 python 代码写的比较多一些，再加之 Java 的内容有些忘了…，且目前尚处于入门阶段，觉得 flask 框架应该是最合适的，因此暂时舍弃了另外两种主流框架，准备有基础后择日再探。

本周我成功更新了 my pc 端的个人主页（移动端目前排版会出问题，考虑要重新写一个样式表），同时利用之前爬虫获取的疫情数据制作了一个简易的疫情实时地图网页（暂未部署）。

pc 端主页地址：<http://www.zhenghui.tech>

## 2 实验过程

### 2.1 快速部署 html 文件到测试服务器（略）

### 2.2 创作 html

#### 2.2.1 确定需求，参考感兴趣的网站结构

着手制作以前，可以参考一些做的非常成熟的网站的结构，这种参考不是一种抄袭，而像是在与做这个网站的人员交流一般，可以从他们身上获取很多之前没有考虑过的点子。我在做主页时，就先参考了北邮人论坛的首页布局，获取了很多灵感，对整个页面的布局也更清晰了。

#### 2.2.2 结构化、层次化 html

渲染 html 时，除了要顾及页面当中的排版，更应该考虑整体 html 的结构。否则，在你的电脑中表现完美的页面，换到别人的电脑上可能全乱了。不过有些时候，即使你结构化了你的代码，换到分辨

---

\*Corresponding author: wangzhenghui@bupt.edu.cn

率差异很大的浏览器中（移动端）效果可能也会出乎意料的差，但不管怎样，结构化的代码好处多多，在写前端之前一定要考虑。

我所理解的结构化，就是将页面布局划分成几个重要的子部分，子部分又能继续向下划分，然后对子部分作单独的加工，这样能使不同部分的代码产生“隔离”，这种隔离一方面更加清晰方便维护，一方面将页面中代码相互的影响降到了最低



图 1: 北邮人论坛中的布局排版

### 2.2.3 页面的渲染与样式表（略）

### 2.2.4 数据的获取

数据的获取是后端需要考虑的内容。在 flask 中，我的理解就是写一个工具库 utils，在库内利用 python 获取数据库中的内容、获取时间戳等等。然后在 flask 路由函数中调用相应的函数获取数据，经过一些简单处理后再传给前端。

## 近期项目

- 1.任务1-利用自主构建的神经网络框架分类CIFAR [\[pdf\]](#)[\[src\\_code\]](#)
- 2.任务2-Tensorflow框架搭建VGG网络实现对CIFAR-10的训练 [\[pdf\]](#)[\[src\\_code\]](#)
- 3.项目3-基于Keras框架实现多种网络结构对agnews数据集进行分类 [\[pdf\]](#)[\[src\\_code\]](#)
- 4.项目4-李宏毅深度学习作业——回归（9.14-9.20） [\[pdf\]](#)[\[src\\_code\]](#)
- 5.LaTeX\_handbook [\[pdf\]](#)[\[src\\_code\]](#)
- 6.crawler\_handbook [\[pdf\]](#)[\[src\\_code\]](#)

图 2: 主页部分内容展示

上面的部分页面内容，就是利用 `utils` 库读取文件夹中的所有文件，然后将项目标题、项目报告以及项目源码的路径回传，经路由函数调用传到前端进行显示的。这种结构使得每次我更新项目时，只需要按照一定的结构要求将项目放入指定文件夹下即可，不需要通过手动的修改 `html` 文件来完成。

### 2.2.5 前端的调用函数

ajax: 异步的 JavaScript 和 XML, 允许在不重新加载页面的情况下与服务器交换数据并更新内容。之前提及的路由函数回传到前端就可以用 ajax 请求很轻松地获取。

jinja2: 基于 python 的模版引擎。模板在 Python 的 web 开发中广泛使用,它能够有效的将业务逻辑和页面逻辑分开,使代码可读性增强、并且更加容易理解和维护。利用 jinja2 可以将获取到的数据动态显示到页面中 (我理解就是 java 中的 jsp)

### 2.2.6 echarts 的使用

echarts 被我理解为基于 js 写出的一个第三方库。如果需要在网站中可视化数据，echarts 是最佳选择。echarts 的模版可以在官方网站中直接获取，然后根据需要作简单调整即可，显示出的效果十分强大。

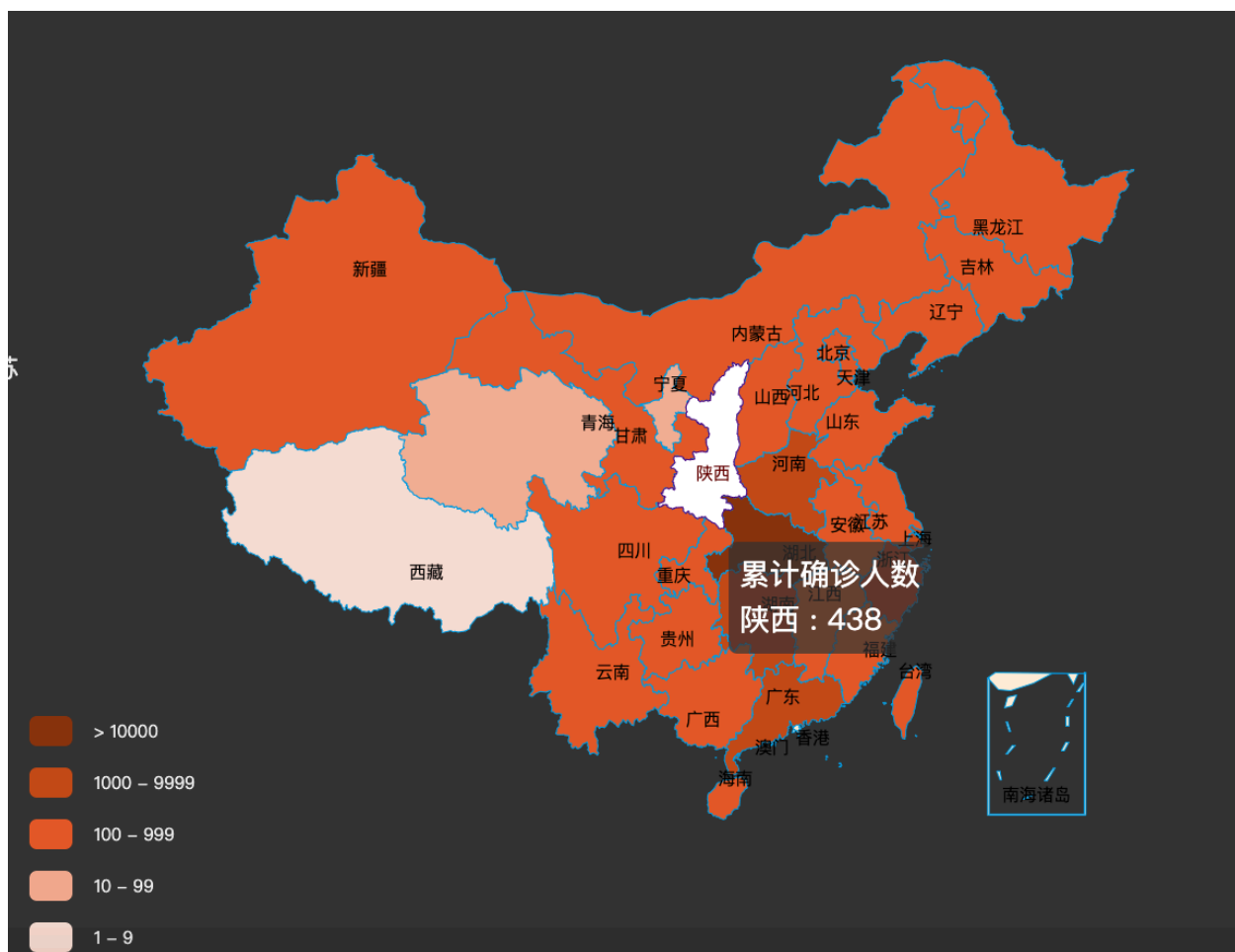


图 3: 用 echarts 制作的疫情地图

## 2.3 遭遇的问题

在整个项目实现过程中，花费时间最多的部分是：根据屏幕分辨率动态调整页面大小。我可能花了近 10 个小时尝试解决不同设备显示效果截然不同的问题，最终部分解决——目前在 PC 端（宽度 1000px 以上的设备）可以比较良好的显示界面，但在移动端问题还是很大。我考虑后面通过重写一个样式表来适配移动端。

一开始我都已经得意地将项目部署到服务器上了，然后在移动端打开后人都傻了哈哈哈哈哈哈…，简直是惨不忍睹。之后开始考虑这一问题，把网上教程翻了个遍硬没解决，后将样式表中所有涉及字体大小的部分改为 rem 格式、将所有图片适配宽度、所有子部分的宽度设为百分比显示，再在 js 中写入一个依据当前浏览器分辨率调整 html 整体 font 的函数，每次 resize 浏览器都调用，部分解决问题。目前，显示出的内容有的超出了显示范围，考虑分页面显示，这一部分尚未完成。

## 3 结果展示



图 4: 疫情页面