

Web 前端技术实训 基础作业一

实验报告

卫国扬 软件 42 2014013436

个人主页链接：<https://foresummer.github.io>

一、 实验要求

- 基础要求

- 结合课上所学，设计实现个人主页，整体协调美观，内容丰富
- 具有个人介绍区，包括个人头像和部分个人信息
- 具有课程展示区，用来放置每次课程实验的发布链接
- 具有个人日志，与首页风格保持一致
- 完成结果通过 Github Pages 发布

- 进阶要求

- 使用前端自动化工具对 web 资源进行压缩
- 具有响应式设计以适配移动端等不同尺寸大小的浏览器

二、 完成实现

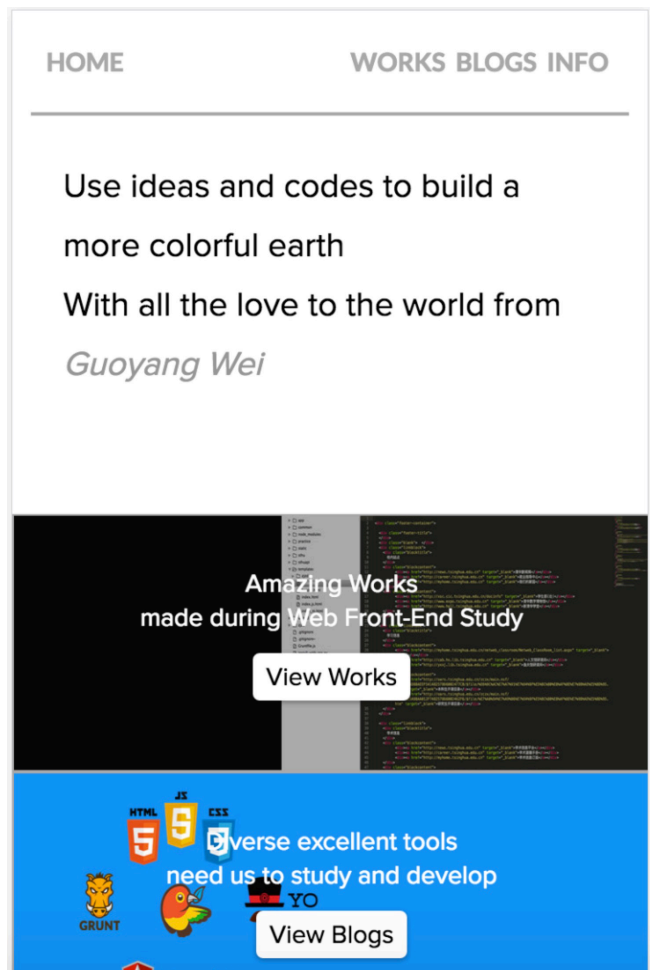
- 概述

结合课上所学并查阅相关资料, 完成稿可以实现本次实验的全部基础要求与进阶要求。

- 实现细节

- 布局控制

在布局设计上, 采用了经典的 NavBar – Header – Content – Footer 的设计, 方便用户使用。正文部分, 清晰展示了主页主要的三部分, 并附有按钮方便用户跳转。按钮和导航栏均通过 ‘:hover’ 属性增加了动画, 提高用户交互时的体验。Works 可以进入课程展示区, blogs 可以查看日志, info 是个人信息。



图为移动端主页效果图

布局的实现也没啥好说的，都是常见的 css 属性，跳转也是用的 a 标签的 href 跳转，不涉及 js。

■ 响应式设计

响应式设计采用了 css 的 @media 功能来实现。在设备宽度 500px 以下和 500px – 900px 之间增加额外的 css 属性覆盖掉默认的宽屏下的样式设计，就可以实现响应式设计。在 html 文件的 meta 标签中也要加上 'content="width=device-width, initial-scale=1.0, maximum-scale=1.0"' 保证 @media 正确应用。相关代码在 main.less 与 main.css 中都可以看得到。

■ 自动化工具 – Grunt

Grunt 有许多插件，一般常用的有如下几个插件，如下图

package.json 文件所示

```
{
  "name": "grunt_test",
  "version": "1.0.0",
  "devDependencies": {
    "grunt": "^1.0.1",
    "grunt-contrib-cssmin": "^1.0.1",
    "grunt-contrib-htmlmin": "^1.4.0",
    "grunt-contrib-less": "^1.3.0",
    "grunt-contrib-uglify": "^1.0.1",
    "grunt-contrib-watch": "^1.0.0",
    "load-grunt-tasks": "^3.5.0"
  }
}
```

cssmin 和 htmlmin 顾名思义是用来压缩的，less 是好用的 css 预处理语言工具，和 grunt 有较好的集成，写起来比原生 css 更舒服更方便，uglify 是用来压缩 js 的，在实验一中还用不到，watch 是监测工具，检测在工作中指定的文件的变化并进行相应的处理（例如在实验一中我们用 watch 检测 less 文件的变化并自动编译 css 文件，这样写 less 不需要手动执行命令编译，和直接写 css 一样可以实时看到展示的效果）。

想要执行 grunt 还需要配置文件 Gruntfile.js，在这个文件中详细定义了我们运行各个命令想要进行的工作。

在本次实验中，运行 grunt 命令可以将写在 dist 文件夹下的待压缩 html 压缩到根目录，同时将 less 进行编译、js 文件进行压缩（虽然实验一中并没有什么用），在压缩 css 方面，由于

less 编译好的就是压缩过的，所以直接采用了 less 编译后的版本作为压缩后的版本，同时 main.css 是一份和 main.less 功能完全相同的手写 css 文件，供老师和助教检查批阅（其实 less 文件看起来更舒服更清晰 0.0)

三、实验收获

通过这次实验，自身的 web 技术变得更加熟练。以前自己和别人合作写的时候 gruntfile 和依赖都是学长写好的自己直接用。当自己写过一次 gruntfile 后对整个 grunt 工具的运行和配置都有了更深的了解。感谢老师助教的讲授与指导。