# 1.前端编码规范文档设计

## 1.1项目，文件的命名

针对项目以及文件的命名提出统一的规范标准（小驼峰命名），来更好的进行统一的开发加强开发的效率。

例如针对项目或者文件命名的写法，组合，表达语意以及禁止用语等方式对文件以及项目的命名做出一个统一的规范。

## 1.2 HTML的规范

### 语法

规定统一的编写方式来进行约束避免不必要的麻烦同时保证代码在各种环境下显示一致性。

### 字符编码

统一使用utf-8的字符编码格式。

### Css以及Javascript的引用

对css以及javascript的引用进行统一的约束规范。

### 标签属性的使用

根据HTML标签的使用以及属性的写法排列进行规定以及约束，尽量使用语义化的标签增强HTML文档的可读性。

## 1.3 CSS的规范

### 语法

规定统一的编写方式来进行约束避免不必要的麻烦同时保证代码在各种环境下显示一致性。

### 代码组织以及选择器

对代码组织进行制定一些统一的约束使得对以后的维护以及修改有一个更好的帮助。

对选择器的使用以及进行一个合理的规范，提高浏览器对页面的渲染性能以及优化。

### 类名和属性

针对类名的命名进行一个有效的约束规定，使得这些类名在样式表中变的更有意义已经通过。

对类名的查看能够大概理解它所表达的意思。

对样式表中的属性名进行一个相对规范的约束，避免属性的滥用对页面进行不必要的影响。

## 1.4Javascript的规范

### 语法

规定统一的编写方式来进行约束避免不必要的麻烦同时保证代码在各种环境下显示一致性。

### 变量

对变量的命名、使用、进行一个统一标准的制定，并且按照规定进行编写约束。

### 函数

对函数的命名、使用、进行一个统一标准的制定，并且按照规定进行编写约束。

统一：小驼峰命名

### 注释

对注释进行一个使用的约束，表名在不同是地方不同的时候使用不同的注释帮助阅读代码的人更好的进行对代码的理解。

Html文件命名：英文命名

首页：index.html； 收藏：collection.html 市集: fair.html 邮箱: maibox.html

个人中心：myCenter.html 菜谱分类：foodclassify.html 分类列表：classifylist.html

添加厨友：addChefs.html 主题：theme.html 他人主页：Othershomepage.html

登录：signin.html 注册 register.html 上传：upload.html 创建菜谱1:setupfood1.html

创建菜谱2:setupfood2.html 加入菜单 addMenu.html 创建菜单 createMenu.html

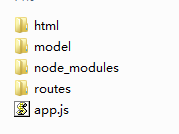
分类详情classificationDetails.html 菜谱详情 menuDetails.html

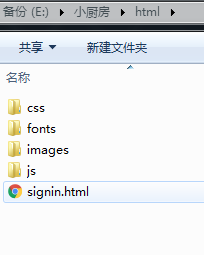
Node.js 文件 ： 命名

App.js

使用控制台可以打开。

文件夹格式类似这样





Model文件夹存放连接数据库文件 比如说增删改查操作，要操作的表名可以当参数传到公共组件中，需要的时候直接调用就行

Routes文件夹存放api.js文件 传输数据

Node文件夹是加载的组件

app.js 文件是需要用控制台打开的文件，这是搭建好的服务器

# 2．应用技术

## jQuery库

### 1）使用单引号

### 2）缓存变量：DOM 遍历是昂贵的，所以尽量将会重用的元素缓存。

### 3）避免全局变量：jquery 与 javascript 一样，一般来说，最好确保你的变量在函数作用域内。

### 4）使用驼峰式命名：使用[驼峰式](http://www.cnblogs.com/yjzhu/p/4883059.html)命名，在前面添加 $ 作为前缀，以便于标示为 jquery 对象.

### 5) 使用单 let 模式: 将多条 var 语句合并为一条语句，建议将未赋值的变量放到后面。

### 6) 使用 on 来处理事件: 在新版 jquery 中，更短的 on('click') 用来取代类似 click() 这样的函数。在之前的版本中 on() 就是 bind()。自从 jquery 1.7 版本后，on() 是附加事件处理程序的首选方法。出于一致性考虑，你可以简单的全部使用 on() 方法.

### 7) 链式操作:jquery 能够很轻易的实现链式操作.

### 8) 维持代码的可读性:伴随着精简代码和使用链式的同时，可能带来代码的难以阅读。添加缩进和换行能起到很好的效果。

### 9) 避免多个 id 选择符:在此强调，id 选择符应该是唯一的，不需要添加额外的选择符，更不需要多个后代 id 选择符。

### 10) 坚持最新版本新版本通常更好：更轻量级，更高效。当然你需要考虑你要支持的代码的兼容性。例如，2.0 版本不支持 ie 6/7/8

## 2、使用Bootstrap响应试框架。

### 1）所有命名全部统一Bootstrap按照本身的样式进行布局与页面的重构。

### 2）IE 兼容模式：IE 支持通过特定的 <meta> 标签来确定绘制当前页面所应该采用的 IE 版本。除非有强烈的特殊需求，否则最好是设置为 ****edge mode****，从而通知 IE 采用其所支持的最新的模式。

## 3、Vue框架开发

### vue 方法放置顺序:

* 1. components
  2. props
  3. data
  4. created
  5. **mounted** //生命周期，也叫钩子行数
  6. activited
  7. update
  8. **beforeRouteUpdate** //生命周期，也叫钩子行数
  9. metods
  10. filter
  11. computed
  12. watch

### method 自定义方法命名:

* 1. 动宾短语（good：jumpPage、openCarInfoDialog）（bad：go、nextPage、show、open、login）
  2. ajax 方法以 get、post 开头，以 data 结尾（good：getListData、postFormData）（bad：takeData、confirmData、getList、postForm）
  3. 事件方法以 on 开头（onTypeChange、onUsernameInput）
  4. init、refresh 单词除外
  5. 尽量使用常用单词开头（set、get、open、close、jump）
  6. 驼峰命名（good: getListData）（bad: get\_list\_data、getlistData）

### data props 方法注意点:

* 1. 使用 data 里的变量时请先在 data 里面初始化
  2. props 指定类型，也就是 type
  3. props 改变父组件数据 基础类型用 $emit ，复杂类型直接改
  4. ajax 请求数据用上 isLoading、isError 变量
  5. 不命名多余数据，现在是详情页、你的数据是 ajax 请求的，那就直接声明一个对象叫 d，而不是每个字段都声明
  6. 表单数据请包裹一层 form

### 生命周期方法注意点

* 1. 不在 mounted、created 之类的方法写逻辑，取 ajax 数据，
  2. 在 created 里面监听 Bus 事件

## 4、Node.js

//加载path,路径处理模块

//设定port变量，意为访问端口

//设定views变量，意为视图存放的目录

//加载模板处理模块EJS

//加载md5的加密模块

//加载自定义数据库模块

// 把API封装成文件进行处理

//设定使用的模板的后缀名

app.set("view engine","html");

//运行模板的方法

app.engine("html",ejs.\_\_express);

//渲染模板

使用：app.get('/xxx, function(*req*, *res*)