**前端开发规范**

**制 定 者：王军法**

**制定时间：2019-4-30**

# 常规规范

1. tab缩进，统一4个空格，4个空格

2. 项目必有 README.md说明文件

2.1项目标题

2.2 项目描述

2.3 开发环境，开发框架，框架版本等

2.4开发命令，运行命令，运行配置等

2.5 项目规范，代码规范等

2.6 项目结构介绍

2.7复杂流程介绍，如登录方法，缓存技术等

# HTML规范

1. 全部使用小写
2. img标签添加alt属性
3. 属性使用双引号
4. class名使用中横线区分
5. class名理论不超过三个单词
6. 避免一行太长，多属性标签换行显示
7. 不使用空标签，使用before，after实现
8. 自定义属性前加 data-,自定义属性不建议太多
9. 在 HTML中指定编码 <meta charset="utf-8">

10.html模块化,标签语义化

效果如下：

|  |
| --- |
| *<!-- img标签添加alt属性 -->*  <img src="" alt="icon">  *<!-- 属性使用双引号 -->*  <div class="info"></div>  *<!-- class名使用中横线区分 -->*  <div class="user-name"></div>  *<!-- class名理论不超过三个单词 -->*  <div class="user-info-name"></div>  *<!-- 避免一行太长，多属性标签换行显示 -->*  <div  class=""  v-for=""  :key=""  data-id=""  data-index=""  >  </div>  *<!-- 自定义属性前加 data-,自定义属性不建议太多 -->*  <div  data-id=""  data-index=""  data-name=""  >  </div>  *<!-- html模块化,标签语义化 -->*  <div class="main">  <nav></nav>  <header></header>  <section></section>  <footer></footer>  </div> |

# CSS规范

1. 样式重置
2. 属性使用双引号
3. 选择器层级不要太高，尽量不超过两层
4. 合理使用伪类、伪对象选择器
5. 合理使用标签选择器，（尽量少用）
6. 属性值前，选择器'>', '+', '~'前后，'{'前 保留一个空格

代码如下：

|  |
| --- |
| */\* 使用双引号 \*/*  .user::after {  content: "";  }  */\* 选择器层级不要太高，尽量不超过两层 \*/*  */\* 合理使用标签选择器，（尽量少用） \*/*  .user img {}  */\* 属性值前，选择器'>', '+', '~'前后，'{'前 保留一个空格 \*/*  .user>.info{} */\* bad \*/*  .user > .info {} */\* good \*/*  .user {  color:red; */\* bad \*/*  class: red; */\* good \*/*  } |

# JS规范

## 4.1命名规范

|  |  |  |
| --- | --- | --- |
| **常量** | 全大写，使用下划线分割 | const PI = 3.14; |
| **变量** | 小驼峰 | let userName = ''; |
| **方法** | 小驼峰 | function getInfo () {}; |
| **构造方法** | 大驼峰 | function Student() {}; |
| **类** | 大驼峰 | class Student {}; |
| **类方法属性** | 小驼峰 |  |
| **类私有属性** | 下划线开头，小驼峰 |  |

代码如下：

|  |
| --- |
| *// 变量常量声明，总使用var,let,const，避免变量污染*  *// 常量--全大写*  const PI = 3.14;  *// 变量--小驼峰*  let userName = '';  *// 方法--小驼峰*  function getInfo () {};  *// 构造方法--大驼峰*  function Student (*name*) {  this.\_name = name; *// 私有成员--以下划线开头，小驼峰*  this.getName = function () {}; *// 公共方法*  };  *// 类方法--大驼峰*  class Student {}; |

## 4.2语法规范

1. 语句结尾使用分号;

2. ===,!==与==,!==合理使用（建议使用===,!==）

3. 逗号合理使用，(多余逗号不用)

|  |
| --- |
| *// 使用分号*  let name = '' *// bad*  let name = ''; *// good*  *// 使用===，！==, 不使用==,!=;*  if (name === '') {};  *// 逗号使用(末尾使用)*  let name,  sex;  *// 对象最后属性不使用逗号*  let obj = {  name: '',  sex: ''  } |

## 4.3语句规范

|  |
| --- |
| *// 单行if*  if (true) {return false}; *// bad*  if (true) return false; *// good*  *// 多行if*  if (true) {}; *// good*  *// 布尔值判断*  var arr = [];  var name = '';  if (arr.length) {};  if (name) {};  if (!name) {};  *// 类型转换*  var isTrue = !name;  var isTrue = !!name;  var num = 231232.123123;  var sum = Number(num);  var sum = parseInt(num, 10); *// 0-36之间*  var sum = parseFloat(num);  *// 逗号合理使用，（语句结尾使用）(不要放到语句开头)(多余逗号不加)*  var name,  sex; *// good*  var obj = {  name: '',  }; *// bad*  var obj = {  name: ''  }; *// good* |

## 4.4 空格规范

1. 运算符左右保留一个空格(+, -, \*, /, += , -=, \*=, /=)

2. 冒号右侧保留一个空格

3. 三目运算符冒号问好左右保留一个空格

4. 大括号前放一个空格

5. 在做链式操作使用缩进.

6. 箭头函数 左右保留一个空格

|  |
| --- |
| *// 运算符左右保留一个空格(+,-,\*,/,+=,-=,\*=,/=)*  var num=3-3; *// bad*  var num = 1 + 2; *// good*  var num = 1 - 2; // good  var num+=1; *// bad*  var num += 1; *// good*  var num \*= 1; *// good*  var num++; *// good*  var num--; *// good*  *// 冒号右侧保留一个空格*  var obj = {  name:'', *// bad*  age: 12 *// good*  }  *// 三目运算符冒号问号左右保留一个空格*  var sum = num?num + 1:num + 2; *// bad*  var sum = num ? num + 1 : num + 2; *// good*  *// 大括号前放一个空格*  function getInfo(){} *// bad*  function getInfo() {} *// good*  *// 在做链式操作使用缩进.*  axios().then((*res*) => {}).catch(*err* => {}); *// bad*  axios()  .then((*res*) => {})  .catch(*err* => {}); *// good*  *// 箭头函数 左右保留一个空格*  var log = ()=>{}; *// bad*  var log = () => {}; *// good* |

## 4.5 注释规范

|  |
| --- |
| 单行注释  let name; *// name*  *文件注释*  */\*\**  *\* @file LBS控制器*  *\* @author wjf*  *\* @copyright wjf版权*  *\* @createDate 2019-4-18 09:40:11*  *\*/*  *方法注释*  */\*\**  *\* @method name 方法说明*  *\* @param {String} params1 参数1说明*  *\* @returns {Number} param 返回值说明*  *\* @throws {String} 抛出异常*  *\*/*  @constructor 构造函数声明注释  @param 参数注释  @return 返回值注释  @example 示例注释  @overview 对当前代码文件的描述。  @copyright 代码的版权信息。  @author []代码的作者信息。  @version 当前代码的版本。 |