

软件开发架构规范

修订记录

版本号	修订人	修订日期	修订类型	修订说明	审核人	批准人	批准日期

修订类型包含：新增、修改、删除。

目录

1. 前言.....	1
1.1. 概述.....	1
1.2. 目的.....	1
1.3. 基本准则.....	1
2. 规范总结.....	1
2.1. HTML 规范.....	1
2.2. CSS/less 规范.....	4
2.3. JS 规范.....	6
2.4. Vue 规范.....	7
2.5. React 规范.....	10
2.6. Java 规范.....	11

1. 前

1.1. 概述

为提高团队协作效率，便于后台人员添加功能及前端后期优化维护，输出高质量的文档，特制订此文档。本规范一经确认，前端开发人员必须按本文档规范进行前台页面开发。本文档如有不对或者不合适的地方请及时提出，或者有更好的建议丰富此文档内容的，经讨论决定后可以更改此文档。

1.2. 目的

为了提高工作效率，便于后台人员添加功能及前端后期优化维护，输出高质量的文档，在项目及产品建设中，使结构更加清晰，代码简明有序，有一个更好的开发架构。

1.3. 基本准则

符合 web 标准，使用具有语义的标签，使结构、表现、行为分离，兼容性优良。页面性能优化，代码简洁、明了、有序，尽可能的减少服务器的负载，保证最快的解析速度。

2. 规范总结

2.1. HTML 规范

2.1.1、文件应以“<!DOCTYPE>”首行顶格开始，推荐使用“<!DOCTYPE html>”。

2.1.2、必须申明文档的编码 charset，且与文件本身编码保持一致，推荐

使用 UTF-8 编码<meta charset="utf-8"/>。

1.1.1、用四个空格来代替制表符（tab） -- 这是唯一能保证在所有环境下获得一致展现的方法。

1.1.2、嵌套元素应当缩进一次（即四个空格），保持良好的简洁的树形结构。

以上如图所示：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

1.1.3、class 命名必须单词全字母小写，单词间以 - 分隔，避免数字开头。

1.1.4、HTML 引号，必须使用双引号（"）而不是单引号（'）。

1.1.5、HTML 属性应当按照以下给出的顺序依次排列，确保代码的易读性。

class, id, name

data-*

src, for, type, href, value

title, alt

role, aria-*

```
<a class="..." id="..." data-toggle="modal" href="#">
  Example link
</a>

<input class="form-control" type="text">


```

2.1.3、建议提倡使用语义化标签。

不推荐：

```
<div class="head">头部</div>
<div class="body">body</div>
<div class="foot">尾部</div>
```



推荐：

```
<header>头部</header>
<article>正文</article>
<footer>尾部</footer>
```

2.1.4、不许使用 table 表格进行页面布局。table 仅可在表格列表信息数据渲染使用。

2.1.5 、 禁止在标签内嵌套样式，应给相应 class 名称去定义样式。

不推荐：

```
<div style="color:#000">标签</div>
```



推荐：

```
<div class="tab-color">标签</div>
```

2.1.6 、 禁止 a 标签的 href 取值为空或不写 href 属性，重构时默认尽量用脚本“javascript:;”或者“javascript:void(0);”代替，而不是“#”， 避免页面乱跳。

不推荐：

```
<a href="#">标签</a>
```



推荐：

```
<a href="javascript:;">标签</a>
```

2.1.7 、 html 标签统一小写，禁止大写，避免页面不统一。

不推荐

```
<DIV>
  <p>我是标签</p>
</DIV>
```



推荐

```
<div>
  <p>我是标签</p>
</div>
```

2.1.8 、 在页面底部引入 javascript 文件，让页面以最快的速度呈现在用户面前。

不推荐

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="jquery.js"></script>
  <script src="js.js"></script>
</head>
<body>
  <p>内容内容</p>
</body>
</html>
```



推荐

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <p>内容内容</p>
  <script src="jquery.js"></script>
  <script src="js.js"></script>
</body>
</html>
```

2.1.9 、建议为所有的图片加上 alt 属性，避免图片加载失败时对用户有所提示。并且有利于搜索引擎。firefox 不支持显示图像Alt 属性，可以加入title 属性。

2.2. CSS/less 规范

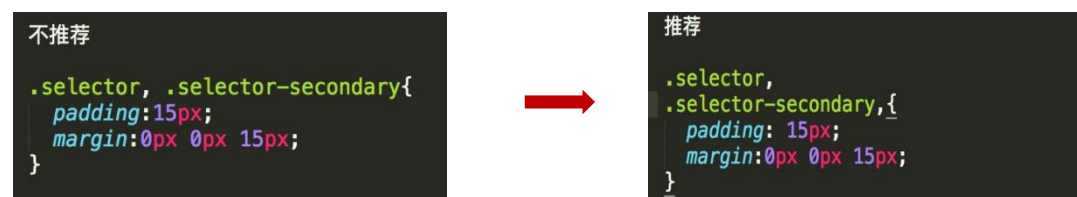
2.2.1、十六进制值应该全部小写，例如，#fff。在扫描文档时，小写字符易于分辨。

2.2.2 、避免为 0 值指定单位。例如：是 { margin:0 }，而不是 { margin:0px }。

2.2.3、不建议使用 “_” 下划线来命名 CSS 选择器，可以用 “-” 中划线，因为输入的时候少按一个 shift 键；浏览器兼容问题(比如使用_tips 的选择器命名，在 IE6 是无效的)能良好区分 JavaScript 变量命名（JS 变量命名是用 “_” ）。

2.2.4、避免选择器嵌套层级过多，尽量少于 3 级。

2.2.5、选择器分组时，保持独立的选择器占用一行。



2.2.6、避免使用 @import，与 <link> 相比，@import 要慢很多，不光增加额外的请求数，还会导致不可预料的问题。替代办法：

(1) 使用多个元素；

(2) 通过 Sass 或 Less 类似的 CSS 预处理器将多个 CSS 文件编译为一个文件；

(3) 其他 CSS 文件合并工具；

2.2.7、除了重置浏览器默认样式外，禁止直接为 html 、 a、img 等添加 css 样式设置。

2.2.8、合并 margin、padding、border 的-left/-top/-right/-bottom 的设置，尽量使用短名称。



2.2.9、如果没有边框时，不要写成 border:0，应该写成 border:none 。

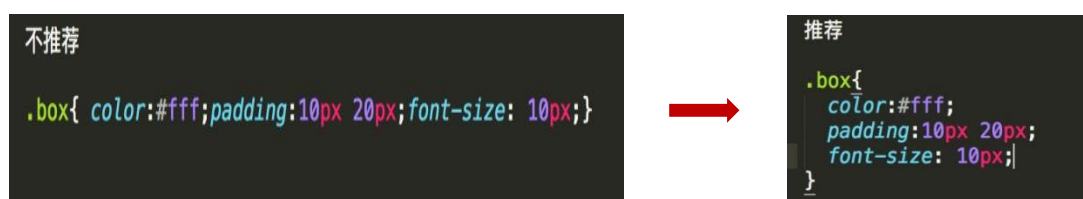
2.2.10、background、font 等可以缩写的属性，尽量使用缩写形式 。

2.2.11、层级(z-index)必须清晰明确，页面弹窗、气泡为最高级（最高级为 999），不同弹窗气泡之间可在三位数之间调整；普通区块为 10-90 内 10 的倍数；区块展开、弹出为当前父层级上个位增加，禁止层级间盲目攀比。

2.2.12 、不要在 rgb()、rgba()、hsl()、hsla() 或 rect() 值的内部的逗号后面插入空格。这样利于从多个属性值（既加逗号也加空格）中区分多个颜色值（只加逗号，不加空格）。

2.2.13 、建议 对于经常出现的组件，尽量避免使用属性选择器（例如，[class^="..."]）。浏览器的性能会受到这些因素的影响。

2.2.14 、为了获得更准确的错误报告，每条声明都应该独占一行。



2.2.15、建议 当使用特定厂商的带有前缀的属性时，通过缩进的方式，让每个属性的值在垂直方向对齐，这样便于多行编辑。

```
/* Prefixed properties */
.selector {
  -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.15);
    box-shadow: 0 1px 2px rgba(0,0,0,.15);
}
```

2.3. JS 规范

2.3.1、注释!!! ——为了方便其它同事，自己写代码的时候一定要写注释；若是定义的全局注释或者函数等注释采用标准等文档注释。

2.3.2、函数命名统一采用驼峰命名法。

2.3.3、建议 一次性修改样式属性；例如：

优化前：

```
1 ele.style.backgroundColor = "#e6e6e6";
2 ele.style.color = "#000";
3 ele.style.fontSize = "12rem";
```

优化后：

```
1 .addActive {
2     background: #eee;
3     color: #093;
4     height: 200px;
5 }
```

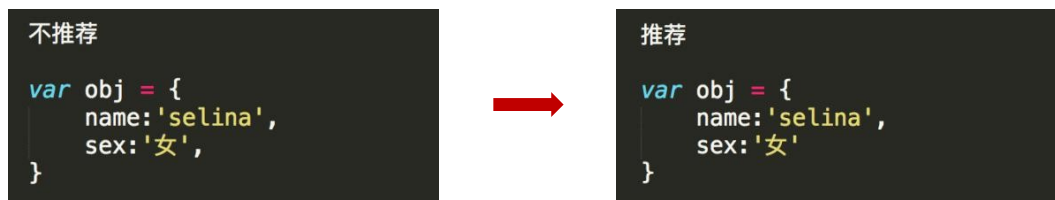
```
1 document.getElementById('myDiv').className = 'addActive';
```

2.3.4、建议 简短的判断，可用三元操作符替代 if 条件判断语句。

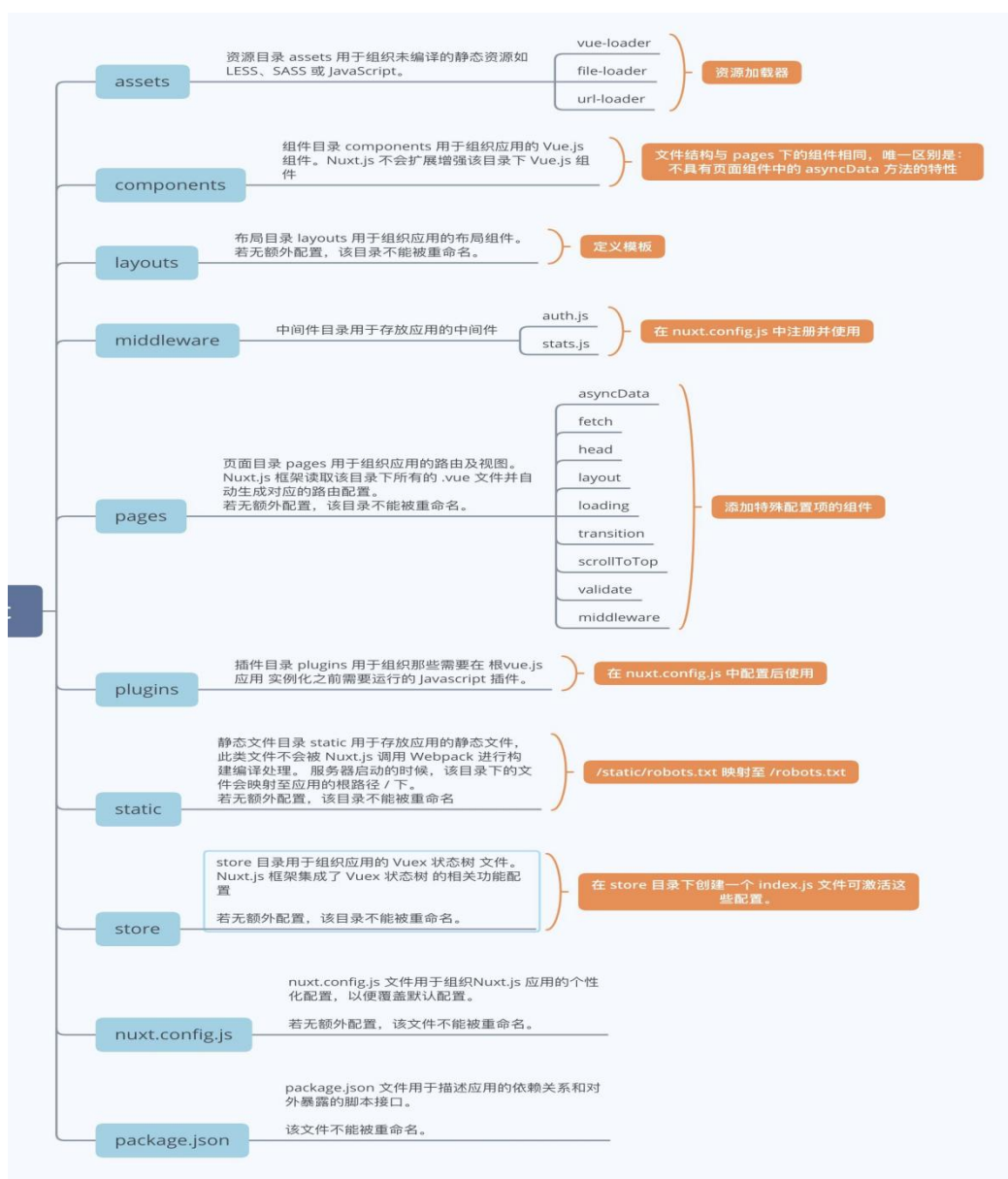
2.3.5、对字符串使用单引号 ' ' (因为大多数时候我们的字符串。特别 html 会出现")。

2.3.6、用 var 声明变量，不要使用 let。避免低版本浏览器报错。在 vue、react 项目中可使用 let。

2.3.7、不要加多余的逗号，这可能会在 IE 下引起错误，同时如果多一个逗号某些 ES3 的实现会计算多数组的长度。



2.4. Vue 规范



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。
如要下载或阅读全文，请访问：

<https://d.book118.com/435224224200011141>