CodingStandards-frontend

caoshuangna

Abstract

Front-end coding specifications, including basic, JavaScript, css, HTML. The purpose of this specification is to call software engineers to focus on code quality, adhering to industry-recognized standards and specifications, and ultimately improving software quality.

Contents

Chapter 1

概述

> All code in any code-base should look like a single person typed it, no matter how many people contributed. — idiomatic.js

此文档的主要意义在于两方面： 1) 代码一致性 2) 最佳实践。

通过保持代码风格和传统的一致性，我们可以减少遗留系统维护的负担，并降低未来系统崩溃的风险。而通过遵照最佳实践，我们能确保优化的页面加载、性能以及可维护的代码。

使用过程中如碰到问题或疑问，请到 Github 进行提问。

总体原则

重要 编码应遵循的基本原则

缩进

对于所有的编程语言，所有的缩进应用空格字符。在你的代码编辑器中1个tab键应当等于2个空格。

可读性

在所有的代码编辑中，我们认为可读性是最重要的，艰涩难懂的语句和为减少文件大小而进行的压缩是不必要的。我们会在build时进行最小化并压缩相关文件。

名称

项目名、目录名、文件名不进行严格统一，但请遵循基本原则：

- 保持项目及项目内命名格式统一，如：小写字母、中划线分隔。
- 复数结构时，采用复数命名法，如：scripts、styles。

- 命名中字母应全部使用小写，因不同操作系统对字母大小写敏感度不同，且这种问题不易被察觉。

注释

注释是别人了解代码的重要途径，不要只写代码干了什么，要写代码为什么这么写，背后的考量的是什么，可以加入所思考的问题或解决方案的链接地址。

Chapter 2

JavaScript

代码风格

- switch 下的 case 和 default 必须增加一个缩进层级。
- 二元运算符两侧必须有一个空格，一元运算符与操作对象之间不允许有空格。
- 用作代码块起始的左花括号 { 前必须有一个空格。
- if / else / for / while / function / switch / do / try / catch / finally 关键字后， 必须有一个空格。
- 函数声明、具名函数表达式、函数调用中，函数名和 (之间不允许有空格。
- , 和 ; 前不允许有空格。如果不位于行尾，, 和 ; 后必须跟一个空格。
- 在函数调用、函数声明、括号表达式、属性访问、if / for / while / switch / catch 等语句中， () 和 [] 内紧贴括号部分不允许有空格。
- 单行声明的数组与对象，如果包含元素，{} 和 [] 内紧贴括号部分不允许包含空格。
- 行尾不得有多余的空格。

换行

- 每个独立语句结束后必须换行。
- 运算符处换行时，运算符必须在新行的行首。
- 在函数声明、函数表达式、函数调用、对象创建、数组创建、for 语句等场景中，不允许在 , 或 ; 前换行。
- 在语句的行长度超过 120 时，根据逻辑条件合理缩进。

语句

- 不得省略语句结束的分号。
- 在 if / else / for / do / while 语句中，即使只有一行，也不得省略块 {…}。

命名

- 变量 使用 Camel命名法。例如：myData、userName。
- 常量 使用 全部字母大写，单词间下划线分隔 的命名方式。
- 函数 使用 Camel命名法。
- 函数的 参数 使用 Camel命名法。
- 类 使用 Pascal命名法。例如：FirstName、LastName。
- 类的 方法 / 属性 使用 Camel命名法。
- 枚举变量 使用 Pascal命名法，枚举的属性 使用 全部字母大写，单词间下划线分隔的命名方式。
- 命名空间 使用 Camel命名法。
- 由多个单词组成的缩写词，在命名中，根据当前命名法和出现的位置，所有字母的大小写与首字母的大小写保持一致。
- 类名 使用 名词。
- 函数名 使用 动宾短语。

注释

单行注释

- 必须独占一行。// 后跟一个空格，缩进与下一行被注释说明的代码一致。

文件注释

- 文件顶部必须包含文件注释，用@description标识文件说明、用@author标识开发者信息等。

解释：

开发者信息能够体现开发人员对文件的贡献，并且能够让遇到问题或希望了解相关信息的人找到维护人。通常情况文件在被创建时标识的是创建者。随着项目的进展，越来越多的人加入，参与这个文件的开发，新的作者应该被加入 @author 标识。

@author 标识多人时，原则是按照 责任 进行排序。意思是如果有问题，就是找第一个人应该比找第二个人有效。比如文件的创建者由于各种原因，模块移交给了其他人或其他团队，后来因为新增需求，其他人在新增代码时，添加 @author 标识应该把自己的名字添加在创建人的前面。

@author 中的名字不允许被删除。任何劳动成果都应该被尊重。

业务项目中，一个文件可能被多人频繁修改，并且每个人的维护时间都可能不会很长，不建议为文件增加@author标识。 通过版本控制系统追踪变更，按业务逻辑单元确定模块的维护责任人，通过文档跟踪和查询，是更好的责任管理方式。

对于业务逻辑无关的技术型基础项目，特别是开源的公共项目，应使用 @author 标识。

函数/方法注释

- 函数/方法注释必须包含函数说明，有参数和返回值时必须使用注释标识。
- 参数和返回值注释必须包含类型信息，且不允许省略参数的说明。

细节注释

- 有时我们会使用一些特殊标记进行说明。特殊标记必须使用单行注释的形式。

常用标记：

1. TODO: 有功能待实现。此时需要对将要实现的功能进行简单说明。
2. FIXME: 该处代码运行没问题，但可能由于时间赶或者其他原因，需要修正。此时需要对如何修正进行简单说明。
3. HACK: 为修正某些问题而写的不太好或者使用了某些诡异手段的代码。此时需要对思路或诡异手段进行描述。
4. XXX: 该处存在陷阱。此时需要对陷阱进行描述。

语言特性

- 变量、函数在使用前必须先定义。
- 每个 var 只能声明一个变量。
- 变量必须 即用即声明，不得在函数或其它形式的代码块起始位置统一声明所有变量。
- 在 Equality Expression 中使用类型严格的 ===。 仅当判断 null 或 undefined 时，允许使用 == null。
- 使用对象字面量 {} 创建新 Object。
- 不允许修改和扩展任何原生对象和宿主对象的原型。
- 使用数组字面量 [] 创建新数组，除非想要创建的是指定长度的数组。
- 字符串全部使用单引号。

Chapter 3

CSS

语法

- 不允许有空的规则。
- 所有声明应该以分号结尾，每条声明应该占用一行。
- 不要在颜色值 rgb(), rgba(), hsl(), hsla(), 和 rect() 中增加空格。
- 不要在属性取值或者颜色参数前面添加不必要的 0（比如，使用 .5 替代 0.5 和 -.5px 替代 0.5px)。
- 所有的十六进制值都应该使用小写字母，例如 #fff。在浏览文档时，他们能够更轻松的被区分开来。
- 尽可能使用短的十六进制数值，例如使用 #fff 替代 #ffffff。
- 不要为 0 指明单位，比如使用 margin: 0; 而不是 margin: 0px;。
- url() 函数中的路径不加引号，函数中的绝对路径可省去协议名。

编码原则

- 从外部文件加载CSS，尽可能减少文件数。加载标签必须放在文件的 HEAD 部分。
- 用 LINK 标签加载，永远不要用@import。

Class 命名

- 保持 Class 命名为全小写，可以使用短划线（不要使用下划线和 camel-Case 命名）。短划线应该作为相关类的自然间断。(例如，.btn 和 .btn-danger)。
- 避免过度使用简写。.btn 可以很好地描述 button，但是 .s 不能代表任何元素。
- Class 的命名应该尽量短，也要尽量明确。
- 使用有意义的名称；使用结构化或者作用目标相关，而不是抽象的名称。
- 命名时使用最近的父节点或者父 class 作为前缀。

选择器

- 使用组合选择器时，保持每个独立的选择器占用一行，>、+、~ 选择器的两边各保留一个空格。
- 选择器 与 { 之间必须包含空格，} 应另起一行。
- 为选择器中得属性取值添加引号，属性选择器中的值必须用双引号包围，例如 input[type=“text”]。 他们只在某些情况下可有可无，所以都使用引号可以增加一致性。
- 减少选择器的长度，每个组合选择器选择器的条目应该尽量控制在 3 个以内。

Chapter 4

HTML

语法

- 嵌套的节点缩进（两个空格）。
- 在属性上，使用双引号，不要使用单引号。
- HTML标签名、类名、标签属性和大部分属性值统一用小写
- 对于无需自闭合的标签，不允许自闭合 （例如，<input> 和 <img>）。
- 对 HTML5 中规定允许省略的闭合标签，不允许省略闭合标签（例如， 和 </body>）。

HTML5 doctype

HTML文件必须加上 DOCTYPE 声明，并统一使用 HTML5 的文档声明， 虽然doctype-不区分大小写，但是按照惯例，doctype大写。

```
<!DOCTYPE html>
```

lang属性

推荐使用属性值 cmn-Hans-CN（简体，中国大陆），但是考虑浏览器和操作系统的兼容性，目前仍然使用 zh-CN 属性值

```
<html lang="zh-CN">
```

字符编码

统一使用 “UTF-8” 编码，页面必须使用精简形式，明确指定字符编码。
指定字符编码的 meta 必须是 head 的第一个直接子元素

```
<meta charset="UTF-8">
```

类型属性

根据HTML5规范，　通常在引入CSS和JS时不需要指明　type，因为　text/css和text/javascript 分别是他们的默认值。 引入 CSS 时必须指明 rel=“stylesheet”。

```
<link rel="stylesheet" href="" >
<script src=""></script>
```

标签

- 在编写 HTML 代码时，需要尽量避免多余的父节点。
- 在 JavaScript 文件中生成标签让内容变得更难查找，更难编辑，性能更差。应该尽量避免。