**前端开发规范**

autherEmail: fuhuang@1218.com.cn

date: 2019.9.9

version: 0.0.1

统一缩进：一个tab（需要设置编译器一个tab为四个空格）或者四个空格

注释虽然不参与代码功能，但是后期维护中注释就显得格外重要了

一、HTML规范

1、注释：例如（<!-- HTML content -->）此注释可跨行

2、html属性定义：例如<div id="app" class="app" style="width: 100px; height: 100px; background-image: url('./theFilePath/fileName.filtType');"></div>

1）属性之间的间隔为一个空格；

2）属性与值之间用”=”连接（注意不要添加空格）；

3）属性的值采用英文双引号进行包裹，若内部需要再次出现引号时采用单引号进行包裹；

4）行内样式属性之间采用一个空格作为间隔，每个样式均为（name + 冒号 + 空格 + value）；

5）避免使用行内样式对页面进行样式控制，尽可能通过class控制。

3、class命名：例如<button class="submit-btn J-submit-btn"></button>

1）尽可能语义化的命名class名称；

2）class需要多个词组合时，每个词中间采用’-’(中划线)进行连接；

3）class应当样式与行为分离，设置样式的class直接命名就好，对于js将操作的class采用J- + ‘className’（此处className并不要求和前面的class名称一致）进行命名，这样有助于更快找到页面中哪些是需要js去操作的；

4）尽量减少页面上id的使用，除非一些特殊的需要，使用id时采用驼峰式命名Id。

4、html（body内）标签：标签之间不需要出现空行（根据需要可出现），标签内删除多余空行；

1）单标签通常采用一行展示，若需要换行，则可以将标签属性间隔处进行断行；

2）双标签内部没有内容（或者内容较少）的时候处理方式与单标签一致，内容较长时将双标签打开，中间独立一行作为内容展示区域，前面增加一个缩进，双标签夸行时需要注意，开始标签和结束标签的缩进距离是一样的，这样有助于区分页面内标签是否是成对的（开发中也请尽量写完标签再去填写内部的内容，避免出现标签未闭合的情况）；

3）尽量不添加一些无用的代码，如<div style="display: block;"></div>

此处的div原本就是block元素就需要再去重新申明其为block元素了。

5、link与script标签

1）link标签放至header中，注意将link标签统一放在一起，方便管理和查阅；

2）script标签引入js文件时，根据需要判断是否需要将script放置在header中，通常对页面初始加载过程中不需要的js应当放置在文档最末（body结束标签前，避免js阻塞文档加载），若需要在页面内写js，应该在引入的script之后（body结束标签前）添加双标签script并在标签内部编写js代码。

二、css规范

1、css的注释方式，如（/\* css content \*/）此注释可跨行

2、css的一般规范

1）css尽量采用link标签引入的方式，因为css样式文件通常情况下行数会比较的多，放在页面内可能导致页面行数过多，查找相应的代码效率低下；

2）若出现多个样式选择器，样式选择器之前采用’,’（英文逗号 + 换行）连接，例如

.my-rules-selector1,

.my-rules-selector2 {

    width: 100px;

}

对于较短的选择也可采用’, ’（英文逗号 + 空格）连接，例如

ul, li, a {

    list-style: none;

    text-decoration: none;

}

属性选择器与花括号之前留一个空格，并且左花括号与最后一个class选择器应该在一行内，

通常采用VSCode自动格式化时会自动将class选择器设置为第一种；

3）样式中，每一个样式属性独自占一行且缩进为当前选择器缩进增加一个缩进，同一个选择器内不要出现多个相同的属性控制（因为最终都是以最后一次设置为主）;

4）样式属性设置时以（name + 冒号 + 空格 + value）进行设置，每条属性设置完后请在最后加上分号；

5）样式设置中的一些细节问题，对于要出现小手一样的图标请用cursor: pointer，避免使用hand属性，因为目前的标准中已经没有了hand的css属性，只是部分浏览器仍保留了该值得解析，通常情况下设置边框的符合属性时尽量以 border-width, border-type, border-color顺序进行设置，例如，border: 1px solid black，css编写过程中，页面内同一地方周围的css放需要放在在一起，必要时请添加备注说明，这样利于后期对该样式的维护修改，避免重复去覆盖之前的样式。

6）对于样式中需要设置路径的情况如图片引用，采用单引号将地址进行包裹（虽然不包裹也能识别，但是包裹之后可以增加相应的安全性），示例格式如下

.background-test {

    background: url('./testFileName.fileType') center/cover no-repeat;

}

符合属性时采用一个空格进行间隔；

7）css伪元素选择器（如：hover, focus, active, visited等），css对大小写不敏感（大小写都能识别），因此，在css中尽可能采用小写字母进行编写，方便阅读；

8）css公共文件中，对全体标签的编写中不要添加特殊属性，仅修改全体标签中的共有属性，或者将部分标签的特有属性统一为全体标签样式，对于特殊样式需求请移步至具体的css文件中进行编写；

9）css文件编写时采用多层级去定位需要修改样式的位置，对于可以进行统一控制的样式才使用少层级的低优先级去初始化样式，样式设置中通常情况下不对id进行样式设置，样式设置中尽量不使用!important的方式去增加样式的优先级，若需要使用此方式来增加优先级，请按如下格式添加，例如

.important-test {

    width: 100px !important;

}

在该条规的值上添加 空格 + !imprtant（注意!和important之间不需要空格）；

10）每条css规则之间可以不用空行（目前编译器自动格式化时默认添加一行空行，可以用配置文件修改，但此处不要求）。

3、公共css样式

1）公共样式文件中体现的是各种标签的初始化，写在common.css中，例如\*{} 对全体元素进行初始化字体大小，字体样式进行设置，用ul, li, ol, a, button, input, i{}等对初始标签进行默认样式处理，例如去除列表点，下划线，outline， 字体样式等等，用body, html{}设置所有页面的最底层背景颜色和大小等，当在某个地方需要此特定样式时，在对应具体的css文件中去添加相应的样式；

2）在对应私有css中不应该存在对全局的样式改变的css（例如上述公共样式文件内的初始化设置就不应该出现在私有css文件中），在私有文件中对dom进行区块化处理（在对父组件下的子组件设置样式的时候，所有的子组件样式选择器都是‘.祖父组件 .父组件 .子组件’），如果在编译允许的情况下可采用less或者sass等文件格式进行编写css文件，可降低大量css工作量。

三、js规范

1、js注释方式：（// var test = 'test'）注释为单行生效，并且注释可叠加，当为公共方法写注释时请用以下注释方式（不可叠加注释）：

/\*\*

 \* @method 函数名

 \* @param {参数类型} a 参数1说明

 \* @param {number} b 参数2说明

 \* @param {string} c 参数3说明

 \* @param {\*} d 参数4说明

 \* @return {String} 返回说明

 \*/

更多参数参考：（@auhor 作者 @example 示例 @link 链接 @namespace 命名空间 @requires 依赖模块 @return 返回值 @version 版本号 等等）

就是中注释非常重要，请注意尽可能详细的写当前代码段的注释。

2、js基本规范

1）变量申明后再使用，否则变量会自动变成全局变量，在严格模式下会直接undefined的错误，现在主流浏览器都已经支持es6及以上的语法，因此在没有明确要求只能es5及以下的项目中，申明变量方式采用let申明，申明常量采用const方式申明，js缩进与前面html以及css一致；

2）在js中运算符左右两侧均保留一个空格，js中的字符串采用单引号进行包裹，若字符串内部需要再套引号则内部使用双引号进行包裹，避免使用转义符号；

3）{}的写法，对象内的键值对格式为（name + 冒号 + 空格 + value）若对象内的数据较少时可写为一行,键值对之间用‘逗号 + 一个空格’方式隔开，如{a: ‘test1’, b: ‘test2’};对象内数据较多时键值对之间采用‘逗号 + 换行符’方式隔开，如

let obj = {

    name: 'test',

    value: 10

}

注意缩进对齐；

4）[]的写法，数组内部的值全部由‘逗号 + 一个空格’方式隔开，例如：[1, 2, 3, 4];

5）函数写法，显式函数申明采用function fun() {}方式申明，（函数关键字 + 一个空格 + 函数名 + 一对圆括号 + 空格 + 左花括号），注意左花括号和函数名在同一行，函数赋值申明方式let fun = function () {}（函数关键字 + 一个空格 + 一对圆括号 + 空格 + 左花括号）；

5）条件判断，例如if (true) {}，格式为（if + ‘空格’ + ( + 条件 + ) + 空格 + 左花括号），switch case如下所示

switch (type) {

    case 1:

        // do some thing

        break;

    default:

        // do some thing

        break;

}