# 关于HTML

一，关于HTML代码

1.

1. 指定字符编码的meta必须是head的第一个子元素。 <meta charset=”UTF-8”>
2. 在link引入css时，可以不用type
3. title标签必须作为head的直接子元素，并跟在charset后面

二，关于缩进

1. 使用两个空格为一个缩进，不要用一个tab

三，关于命名

1. class属性的变量字母全小写，单词间以-为分割。
2. class必须以模块的功能命名，不能以样式信息命名。
3. 不要使用id命名搜索

四，关于语义化

1. 根据HTML的标签意义去使用它。如：跳转页面可以用a标签，而不是使用js操作

五，图片相关

1. 有下载需求的用img，在本地，无下载需求的使用CSS背景图片实现
2. 图像，资源src资源不要跟协议。如：

<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>

# 关于JavaScript

一，基本声明

1. 统一只使用let和const。

可变的引用用let。不可变的引用用const

二，关于对象函数和变量的简写

1.

// 推荐，方法的简写

const atom = {

value: 1,

addValue(value) {

return atom.value + value;

},

};

const lukeSkywalker = 'Luke Skywalker';

// 推荐，属性值简写

const obj = {

lukeSkywalker,

};

三，运算的操作

1. 数组复制。

// 推荐const itemsCopy = [...items];

1. 类数组转数组

// ES6

const foo = document.querySelectorAll('.foo');

const nodes = Array.from(foo);

3. 字符串拼接

// 推荐function sayHi(name) {

return `How are you, ${name}?`;

}

1. 不要使用argument，可以使用rest语法......替代

// 推荐function concatenateAll(...args) {

return args.join('');

}

1. 类直接使用Class，继承用extends

// 推荐class Queue {

constructor(contents = []) {

this.\_queue = [...contents];

}

pop() {

const value = this.\_queue[0];

this.\_queue.splice(0, 1);

return value;

}

}

// 推荐class PeekableQueue extends Queue {

peek() {

return this.\_queue[0];

}

}

1. 计算使用。尽量使用forEach()，map(),reduce()这些es6的计算。
2. If的条件判断使用简写。

// 推荐if (name) {

// ...stuff...

}

// 推荐if (collection.length) {

// ...stuff...

}

8， 类型转换

// 推荐const val = parseInt(inputValue, 10);

// 推荐const hasAge = Boolean(age);

9，命名规则

使用小驼峰去明名函数和实例

1. 使用\_表示私有属性。使用下划线表示私有变量
2. 当你导出的是单例，或者单类。则要与当前的文件名相同。

# Vue编码规范

1. 都用 v-on和v-bind
2. v-if和v-else结合key使用
3. 优先通过Prop和事件进行父子间通信。

一个理想的Vue是prop向下传递，事件向上传递。

1. 优先使用Vuex而不是一个全局总线或this.$root
2. 可以把计算属性切割为尽可能多的更简单的属性。
3. 组件命名适合第一个就大写，且单词写完整
4. 组件名应该以高级别的 (通常是一般化描述的) 单词开头，以描述性的修饰词结尾。

components/

|- SearchButtonClear.vue

|- SearchButtonRun.vue

|- SearchInputQuery.vue

|- SearchInputExcludeGlob.vue

|- SettingsCheckboxTerms.vue

|- SettingsCheckboxLaunchOnStartup.vue

# CSS编码

1. 选择器

当一个 rule 包含多个 selector 时，每个选择器声明必须独占一行。

/\* good \*/

.post,

.page,

.comment {

line-height: 1.5;

}

1. 使用两个空格作为一个缩进
2. 选择器的嵌套层级应不大于 3 级，位置靠后的限定条件应尽可能精确。

/\* good \*/

#username input {}

.comment .avatar {}

/\* good \*/

.page-container {

.content {

.profile {

// STOP!

}

}

}

1. 同一 rule set 下的属性在书写时，应按功能进行分组，并以 Formatting Model（布局方式、位置） > Box Model（尺寸） > Typographic（文本相关） > Visual（视觉效果） 的顺序书写，以提高代码的可读性。

解释：

Formatting Model 相关属性包括：position / top / right / bottom / left / float / display / overflow 等

Box Model 相关属性包括：border / margin / padding / width / height 等

Typographic 相关属性包括：font / line-height / text-align / word-wrap 等

Visual 相关属性包括：background / color / transition / list-style 等

.sidebar {

/\* formatting model: positioning schemes / offsets / z-indexes / display / ... \*/

position: absolute;

top: 50px;

left: 0;

overflow-x: hidden;

/\* box model: sizes / margins / paddings / borders / ... \*/

width: 200px;

padding: 5px;

border: 1px solid #ddd;

/\* typographic: font / aligns / text styles / ... \*/

font-size: 14px;

line-height: 20px;

/\* visual: colors / shadows / gradients / ... \*/

background: #f5f5f5;

color: #333;

-webkit-transition: color 1s;

-moz-transition: color 1s;

transition: color 1s;

}

1. 清楚浮动

通过对伪类设置clear，来清楚浮动

1. 尽量不要使用！Import

当需要强制指定样式不允许任何场景覆盖，通过内联和！Important定义样式

1. url()函数的路径不加引号

body {

background: url(bg.png);

}

Url()函数中的绝对路径可以省去协议名

1. 长度为0时，强制省略单位
2. 颜色必须使用rgba(),且rgba每个逗号后面必须保留一个空格。、

颜色值可以用缩写时，必须先使用缩写形式

1. Background的2D位置

必须同时给出水平和垂直的方向

/\* good \*/body {

background-position: center top; /\* 50% 0% \*/

}

1. font-weight必须使用数值的方式描述
2. line-height在定义文本段落时，应使用数值
3. 关于变换和动画

尽可能在浏览器能高效实现的属性上添加过渡和动画

* transform: translate(npx, npx);
* transform: scale(n);
* transform: rotate(ndeg);
* opacity: 0..1;

/\* good \*/.box {

transition: transform 1s;

}.box:hover {

transform: translate(20px); /\* move right for 20px \*/

}