

# html编码风格

html编码风格，主要考虑到易读性，这里暂时只列一条。

对标签除第一个属性外的其他每个属性进行换行，并保持对齐，更加易读。

## 坏例子

```
<input class="input-unit" type="text" v-model="inputVal" @focus="setName" placeholder="请输入数字" />
```

## 好例子

```
<input class="input-unit"
      type="text"
      v-model="inputVal"
      @focus="setName"
      placeholder="请输入数字" />
```

# css编码风格

css编码风格主要考虑易读性、性能。

1、选择器 与 { 之间必须包含空格；属性名 与之后的 : 之间不允许包含空格，: 与 属性值 之间必须包含空格。

## 坏例子

```
.selector {
    color:red;
}
```

## 好例子

```
.selector {
    color: red;
}
```

2、复合型属性值 书写在单行时，, 后必须跟一个空格。

## 坏例子

```
font-family: Arial,sans-serif;
```

## 好例子

```
font-family: Arial, sans-serif;
```

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

## 坏例子

```
.post, .page, .comment {  
    line-height: 1.5;  
}
```

## 好例子

```
.post,  
.page,  
.comment {  
    line-height: 1.5;  
}
```

4、>、+、~ 选择器的两边各保留一个空格。

## 坏例子

```
.main>.nav {  
    padding: 10px;  
}
```

## 好例子

```
.main > .nav {  
    padding: 10px;  
}
```

5、属性定义必须另起一行。

## 坏例子

```
.selector { margin: 0; padding: 0; }
```

## 好例子

```
.selector {  
    margin: 0;  
    padding: 0;  
}
```

## 6、选择器的嵌套层级应不大于 3 级。

嵌套太深对性能有一定的影响。

## 坏例子

```
.page .header .login #username input {}
```

## 好例子

```
#username input {}
```

## 7、尽量不使用 !important 声明。

## 8、当数值为 0 - 1 之间的小数时，省略整数部分的 0。

## 坏例子

```
.panel {  
    opacity: 0.8;  
}
```

## 好例子

```
.panel {  
    opacity: .8;  
}
```

## 9、长度为 0 时须省略单位。(也只有长度单位可省)

## 坏例子

```
body {  
    padding: 0px 5px;  
}
```

好例子

```
body {  
    padding: 0 5px;  
}
```

10、url() 函数中的路径不加引号。

好例子

```
body {  
    background: url(bg.png);  
}
```