# **HTML**

### **语法**

* 标签不要大写，即便是 doctype 标签。
* 用两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。
* 嵌套元素应当缩进一次（即两个空格）。
* 对于属性的定义，永远全部使用双引号，绝不要使用单引号。
* 不要在自闭合（self-closing）元素的尾部添加斜线 — [HTML5 规范](http://dev.w3.org/html5/spec-author-view/syntax.html#syntax-start-tag) 中明确说明斜线是可忽略的。
* 不要省略可选的结束标签（closing tag）（例如，</li> 或 </body>）。

<!doctype html>  
<html>  
 <head>  
 <title>Page title</title>  
 </head>  
 <body>  
 <img src="images/company-logo.png" alt="Company">  
 <h1 class="hello-world">Hello, world!</h1>  
 </body>  
</html>

### **HTML5 doctype**

为每个 HTML 页面的第一行添加 [standards mode（标准模式）](https://developer.mozilla.org/en-US/docs/Web/HTML/Quirks_Mode_and_Standards_Mode) 声明，这样能够确保在每个浏览器中拥有一致的展现。

<!doctype html>  
<html>  
 <head>  
 </head>  
</html>

### **语言属性**

根据 HTML5 规范：

强烈建议为 html 根元素指定 lang 属性，从而为文档设置正确的语言。这将有助于语音合成工具确定其所应该采用的发音，有助于翻译工具确定其翻译时所应遵守的规则等等。

更多关于 lang 属性的知识可以从 [此规范](http://www.w3.org/html/wg/drafts/html/master/semantics.html#the-html-element) 中了解。Sitepoint 站点上 [给出了一份语言代码表](https://www.sitepoint.com/iso-2-letter-language-codes/)。

<html lang="en">  
 <!-- ... -->  
</html>

### **IE 兼容模式**

IE 支持通过特定的 <meta> 标签来确定绘制当前页面所应该采用的 IE 版本。除非有强烈的特殊需求，否则最好是设置为 **edge mode**，从而通知 IE 采用其所支持的最新的绘制模式。  
了解更多信息请 [阅读这篇 Stack Overflow 上的文章](https://stackoverflow.com/questions/6771258/what-does-meta-http-equiv-x-ua-compatible-content-ie-edge-do)。

<meta http-equiv="x-ua-compatible" content="ie=edge">

### **字符编码**

通过明确声明字符编码，能够确保浏览器快速并容易的判断页面内容的渲染方式。这样做的好处是，可以避免在 HTML 中使用字符实体标记（character entity），从而全部与文档编码一致（一般采用 UTF-8 编码）。

<head>  
 <meta charset="UTF-8">  
</head>

### **引入 CSS 和 JavaScript 文件**

根据 HTML5 规范，在引入 CSS 和 JavaScript 文件时一般不需要指定 type 属性，因为 text/css 和 text/javascript 分别是它们的默认值。

#### **HTML5 spec links**

* [Using link](http://www.w3.org/TR/2011/WD-html5-20110525/semantics.html#the-link-element)
* [Using style](http://www.w3.org/TR/2011/WD-html5-20110525/semantics.html#the-style-element)
* [Using script](http://www.w3.org/TR/2011/WD-html5-20110525/scripting-1.html#the-script-element)

<!-- External CSS -->  
<link rel="stylesheet" href="code-guide.css">  
<!-- In-document CSS -->  
<style>  
 /\* ... \*/  
</style>  
<!-- JavaScript -->  
<script src="code-guide.js"></script>

### **实用为王**

尽量遵循 HTML 标准和语义，但是不要以牺牲实用性为代价。任何时候都要尽量使用最少的标签并保持最小的复杂度。

### **属性顺序**

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-\*

class 用于标识高度可复用组件，因此应该排在首位。id 用于标识具体组件，应当谨慎使用（例如，页面内的书签），因此排在第二位。

<a class="..." id="..." data-toggle="modal" href="#">  
 Example link  
</a>  
<input class="form-control" type="text">  
<img src="..." alt="...">

### **布尔（boolean）型属性**

布尔型属性可以在声明时不赋值。XHTML 规范要求为其赋值，但是 HTML5 规范不需要。  
更多信息请参考 [WhatWG section on boolean attributes](http://www.whatwg.org/specs/web-apps/current-work/multipage/common-microsyntaxes.html#boolean-attributes)：

元素的布尔型属性如果有值，就是 true，如果没有值，就是 false。

如果*一定*要为其赋值的话，请参考 WhatWG 规范：

如果属性存在，其值必须是空字符串或 [...] 属性的规范名称，并且不要在首尾添加空白符。

**简单来说，就是不用赋值。**

<input type="text" disabled>  
<input type="checkbox" value="1" checked>  
<select>  
 <option value="1" selected>1</option>  
</select>

### **减少标签的数量**

编写 HTML 代码时，尽量避免多余的父元素。很多时候，这需要迭代和重构来实现。请看下面的案例：

<!-- Not so great -->  
<span class="avatar">  
 <img src="...">  
</span>  
<!-- Better -->  
<img class="avatar" src="...">

HTML

1、多个自定义属性情况下，每个属性换行。

<tr  
 class="info"  
 \*ngFor="let item of draftList.data; let i = index"  
 [ngClass]="{'active': item.id === draftdata.details.entity['id'], 'hasdone': item['properties']['status']  
 === 'Calibrated'}"  
 (click)="itenClick(item, i)"  
>

2、行元素不要换行缩进，如<span>，小程序中<text>等。防止出现多余空格。

3、html各个元素尽量使用class来控制样式，理论情况下，所有主要元素都应该命名class。

4、id可以用在需要js进行控制的元素上，一般情况下，在vue、angular、小程序项目不使用。

5、尽量不要使用行内样式。

6、注意使用中文标点符号。

7、一个自定义directive作为元素属性，vue、angular写法。  
  
8、{{ A?'A':'B' }}可以使用三元判断符来进行转换。

小程序  
1、小程序中的<icon>标签主要供其自带icon使用，iconfont更适合使用<text>标签。且务必设置字号。

2、rich-text可以用来进行富文本解析

3、小程序中没有自闭和元素

4、遵循统一规范，属性使用双引号。

5、设置wx:key不需要双花括号，引用需要。

angular  
1、ng-container包裹层。

2、#模板引用变量 使用注意