

HTML 编码规范

对当前规范有更好的建议，请到 issue
(<http://gitlab.alibaba-inc.com/specs/style-guide/issues>)
页面反馈给我们，谢谢！

基本规范

HTML5 doctype

在每个 HTML 页面开头使用这个简单地 doctype 来启用标准模式，使其每个浏览器中尽可能一致的展现。

```
<!-- bad -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
  </head>
</html>

<!-- good -->
<!DOCTYPE html>
<html>
  <head>
  </head>
</html>
```

字符编码

由于历史原因，淘系不同产品都是采用了不同的字符编码。对于以后的新业务如无特殊要求，统一使用 UTF-8 字符编码，便于统一。

Language attribute

根据 HTML5 规范：

鼓励网站作者在 `html` 元素上指定 `lang` 属性，来指出页面的语言。这样做会有助于语言合成工具来确定怎样发音，以及翻译工具决定使用的规则，等等。

通过规范 (<http://www.w3.org/TR/html5/dom.html#attr-lang>) 中的 `lang` 属性了解更多相关内容。

ISO Language Codes

(http://www.w3schools.com/tags/ref_language_codes.asp)

IE 兼容模式

Internet Explorer 支持使用兼容性 `<meta>` 标签来指定使用什么版本的 IE 来渲染页面。如果不是特殊需要，通常通过 **edge mode** 来通知 IE 使用最新的兼容模式。

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

引入 CSS 和 JavaScript

- 根据 HTML5 规范，通常在引入 CSS 和 JavaScript 时不需要指明 `type`，因为 `text/css` 和 `text/javascript` 分别是他们的默认值。

```
<!-- bad -->
<!DOCTYPE html>
<head>
  <link type="text/css" rel="stylesheet" href="code-
  <script type="text/javascript" src="example.js"><
</head>

<!-- good -->
<!DOCTYPE html>
<head>
  <link rel="stylesheet" href="code-guide.css">
  <script src="example.js"></script>
</head>
```

- 此外，引入 CSS 必须在 `<head></head>` 标签里引入。对于引入 Javascript，除了基础库等比较基础性的脚本文件，其他都在靠近 `body` 结束标签前面引入。

```
<!-- bad -->
<!DOCTYPE html>
<html>
  <head>
    <script src="mod-a.js"></script>
    <script src="jquery.js"></script>
  </head>
  <body>
    <style>
      .mod-example {
        padding-left: 15px
      }
    </style>
  </body>
</html>

<!-- good -->
<!DOCTYPE html>
<html>
  <head>
    <style>
      .mod-example {
        padding-left: 15px
      }
    </style>
    <script src="jquery.js"></script>
  </head>
  <body>
  </body>
</html>
```

缩进使用两个空格

```
<!-- bad -->
<div>
  <p>just a example</p>
</div>

<!-- good -->
<div>
  <p>just a example</p>
</div>
```

标签名称和标签属性统一使用小写

```
<!-- bad -->
<Div Id="test">
</Div>

<!-- good -->
<div id="test">
</div>
```

建议的 html 脚手架

由上面规范可以得出下面建议的 html 脚手架：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta lang="zh">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="renderer" content="webkit">
    <meta name="description" content="">
    <meta name="keyword" content="">
    <title>淘宝网</title>
    <link rel="stylesheet" href="example.css">
  </head>
  <body>
    <script src="example.js"></script>
  </body>
</html>
```

属性规范

属性引号用双引号

```
<!-- bad -->
<link rel='stylesheet' href='example.css'>

<!-- good -->
<link rel="stylesheet" href="example.css">
```

Boolean 属性

Boolean 属性指不需要声明取值的属性。XHTML 需要每个属性声明取值，但是 HTML5 并不需要。

了解更多内容，参考 [WhatWG section on boolean attributes]:

一个元素中 Boolean 属性的存在表示取值 true，不存在则表示取值 false。

简而言之，尽量不要为 **Boolean** 属性添加取值。

```
<!-- bad -->
<input type="text" disabled="disabled">

<input type="checkbox" value="1" checked="checked">

<select>
  <option value="1" selected="selected">1</option>
</select>

<!-- good -->
<input type="text" disabled>

<input type="checkbox" value="1" checked>

<select>
  <option value="1" selected>1</option>
</select>
```

自定义属性

建议自定义属性必须以 `data-` 为前缀，便于阅读。

```
<!-- bad -->
<a modal="toggle" href="#">
  Example link
</a>

<!-- good -->
<a data-modal="toggle" href="#">
  Example link
</a>
```

属性顺序

HTML 属性应该按照特定的顺序出现以保证易读性，而且一致的属性顺序可能提升 1.5% 的 gzip 压缩率 (<http://gtmetrix.com/enable-gzip-compression.html>)

- `class`
- `id` , `name`
- `data-*`
- ...

Classes 是为高可复用组件设计的，所以他们处在第一位。**Ids** 更加具体而且应该尽量少使用（例如，页内书签），所以他们处在第二位。

```
<!-- bad -->
<a href="#" data-modal="toggle" id="..." class="...">
  Example link
</a>

<input type="text" class="form-control">

<!-- good -->
<a class="..." id="..." data-modal="toggle" href="#">
  Example link
</a>

<input class="form-control" type="text">
```

标签规范

不要在自动闭合标签的结尾处使用斜线，如

`
`、`<hr>`、`<input>`、`<meta>`、`<source>`

命名规范

- 对于需要自定义属性，属性名命名统一使用 `data-` 前缀，具体参考 `custom-data-attribute` (<http://www.w3.org/html/wg/drafts/html/master/dom.html#custom-data-attribute>)
- `class` 命名和 CSS 编码规范中的保持一致 ([./css-style-guide.html#class](http://css-style-guide.html#class))
- `id` 命名，当作为 **JS** 钩子使用时，以 `J_` 为前缀，后面接大驼峰命名，且这类选择器不能出现在 `css` 中，例如 `J_ExampleIdForJs`。其他使用用法，例如充当锚点，`label` 对应 `input` 的 `id`，都正常默认使用即可。

多媒体退化

- 为 `img` 标签添加 `alt` 属性以声明替代文本；
- 在多媒体标签内部提供指示浏览器不支持该标签的说明，如 `object`、`audio`、`video`

例子：

```
<!-- bad -->
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
</audio>

<!-- good -->
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

注释规范

由于 html 代码一般不会经过预处理，出于安全考虑，html 代码中不能出现任何关于业务相关敏感信息的注释。

参考资料

- Code Guide by @mdo (<http://mdo.github.io/code-guide/>)