前端开发规范

AJAX技术流行起来之后，前端技术突飞猛进，html5在标准出来之前就炒得火热，浏览器厂商不断地提高自己产品的性能和对新技术的支持，如此背景下，前端应用的规模越来越大，从而大大提升开发和管理的复杂性，因此，前端开发规范化势在必行。

YHUI经过九个月的开发，组件越来越多，代码量越来越大，和后端工程师的交流越来越频繁，如何科学地管理组件、编写代码，从而可以让使用者更容易使用组件以及阅读对应的代码，成为一个不得不考虑的问题。

因此，才有了前端开发规范。本规范多就代码格式和编码风格进行了规整，从而避免五花八门的格式和看上去个性十足但毫无阅读性可言的风格，从感性地层面上提高代码的可读性。

规范分为三个部分，分别讨论YHUI开发过程中采用的或者即将采用的 HTML、CSS、JAVASCRIPT 的代码格式和编码风格。

1. HTML

超文本标记语言，看上去最简单，实际上却是最复杂的。

* 1. 大小写

和内容无关的文本一律小写，比如标签名、属性等。

* 1. 文档声明

统一使用如下格式：

<!doctype html>

不能没有，也不能写成其他的声明。YHUI1.3 所有页面都会采用这种文档声明。

* 1. 缩进

一律缩进四个空格。有些编辑器的 tab 键是一个制表符，请手动改为四个空格。

* 1. 空格

不得存在不必要的空格。

不推荐：

<p>你好帅。 </p>

推荐：

<p>你好帅。</p>

* 1. 引号

属性名一律使用双引号。

不推荐：

<div class=’header’></div>

推荐：

<div class=”header”></div>

* 1. 自定义属性

必须以 “data-“ 开头。

不推荐：

<div role=”button”></div>

推荐：

<div data-role=” button”></div>

* 1. 页面编码

采用 utf-8 编码，如下：

<meta charset=”utf-8” />

* 1. 标签

必须要闭合，不管是成对的标签还是单个标签。

不推荐：

<div>

推荐：

<div></div>

不推荐：

<br>

推荐：

<br/>

1. CSS

层叠样式表，在前端开发中占有重要位置。CSS3的出现更是给页面设计师开辟了一个新的天地。

* 1. 命名

一律小写，必要时使用 “-“ 连接符。

不推荐：

.yhuiHeader {}

推荐：

.yhui-header {}

* 1. 选择器

尽量使用类选择器，尽量不要使用 id 、nodeName 和其他选择器。

尽量使用单一选择器。

不推荐：

#yhui-header {} 或 div {} 或 div:first-child {}

推荐：

.yhui-header {}

不推荐：

.yhui-header .yhui-header-role

推荐：

.yhui-header-role

* 1. 简写属性

尽量简写属性。

不推荐：

padding-top: 2px; padding-bottom: 4px; padding-left: 3px;

推荐：

padding: 2px 0 4px 3px;

* 1. 空格

冒号后面保留一个空格。

如果所有属性在一行，属性之间空格。

“{“ 之前有空格。

不推荐：

.yhui-header{height:20px;width:30px;}

推荐：

.yhui-header { height: 20px; width: 30px; }

* 1. hacks

尽量避免使用 “\*”、”\_” 等特殊符号。

尽量使用浏览器嗅探。

不推荐：

.yhui-header { \_zoom: 1; }

推荐：

.yhui-ie6 .yhui-header { zoom: 1; }

1. JAVASCRIPT

关于javascript的规范，实在是太多，比如专业校验工具 jsLint 、 jsHint，各大公司以及一些javascript框架都有自己的规范，比如 google 公司以及 kissy 框架。

YHUI 的核心是 jQuery，所以较多地采用 jQuery 的官方规范。以下是一些摘录，当然做了翻译。

* 1. 空格

使用 tab 键缩进代码，有些编辑器的 tab 键是一个制表符，请手动改为四个空格。

在编码过程中，尽可能多的使用空格。

不推荐：

if(blah===”foo”){

foo(“bar”,”baz”,{zoo:1});

}

推荐:

if ( blah === “foo” ) {

foo( “bar”, “baz”, {

zoo: 1

});

}

* 1. 等于

在任何可能的情况下，使用 “===” 而不是 “==” 。

* 1. 代码块

if/else/for/while/try 总是用 “{}“ 包住，而且一律换行。

条件语句和执行语句严禁放在同一行。

不推荐：

if(true) blah();

推荐：

if ( true ) {

blah();

}

不要使用三元运算符代替 if/else 语句。

除非是在条件语句中，否则不要使用 object && object.method() 代替 if/else 语句。

* 1. 函数调用

总是在参数列表的两头加上空格，如：

foo( true ) ;

特别说明：下面这种情况可以不用加空格。

foo( bar(true) );

特别说明：参数有且只有数组、对象、函数、字符串等字面量时，可以不用空格。

foo(function() { });

foo({ });

foo([ ]);

foo(“single string argument”);

特别说明：如果参数是函数、数组和对象字面量并且有多行时，请以友好的方式结尾。

不推荐：

foo( true, { blah: “baz” } );

推荐：

foo( true, {

blah: “baz”

});

冒号和逗号后面总是有一个空格。

* 1. 变量声明

变量声明必须用逗号隔开，并且逗号紧跟变量而不是另起一行。

声明但不初始化的变量应该总是放在较前位置，声明并初始化的变量应该单独占一行。

推荐：

var a, b, c,

test = true,

test2 = false;

* 1. 类型检测

字符串： typeof object === “string”

数字： typeof object === “number”

布尔值： typeof object === “boolean”

对象： typeof object === “object”

函数： $.isFunction()

数组： $.isArray()

元素： object.nodeType

null：object === null

null 或者 undefined：object == null

undefined：

全局变量： typeof variable === “undefined”

局部变量： typeof variable === undefined

属性： object.prop === undefined

* 1. 正则

总是使用 .test() 和 .exec() ， “string”.match() 不推荐使用。

* 1. 字符串

总是使用双引号，而不是单引号。

* 1. 节点

.nodeName 优于 .tagName 使用。

用 .nodeType 来判断一个节点是什么类型，而不是 .nodeName 。

不要给节点绑定布尔值。

总是使用 .data() 来给节点绑定数据。