## **JavaScript 代码规范**

所有的 JavaScript 项目适用同一种规范。

## **JavaScript 代码规范**

代码规范通常包括以下几个方面:

* 变量和函数的命名规则
* 空格，缩进，注释的使用规则。
* 其他常用规范……

规范的代码可以更易于阅读与维护。

代码规范一般在开发前规定，可以跟你的团队成员来协商设置。

## **变量名**

变量名推荐使用驼峰法来命名(****camelCase****):

firstName = "John";

lastName = "Doe";

price = 19.90;

tax = 0.20;

fullPrice = price + (price \* tax);

## **空格与运算符**

通常运算符 ( = + - \* / ) 前后需要添加空格:

### **实例:**

**var** x = y + z;**var** values = ["Volvo", "Saab", "Fiat"];

## **代码缩进**

通常使用 4 个空格符号来缩进代码块：

### **函数:**

**function** **toCelsius**(fahrenheit) {

**return** (5 / 9) \* (fahrenheit - 32);

}

注意:不推荐使用 TAB 键来缩进，因为不同编辑器 TAB 键的解析不一样。

## **语句规则**

简单语句的通用规则:

* 一条语句通常以分号作为结束符。

### **实例:**

**var** values = ["Volvo", "Saab", "Fiat"];

**var** person = {

firstName: "John",

lastName: "Doe",

age: 50,

eyeColor: "blue"

};

复杂语句的通用规则:

* 将左花括号放在第一行的结尾。
* 左花括号前添加一空格。
* 将右花括号独立放在一行。
* 不要以分号结束一个复杂的声明。

### **函数:**

**function** **toCelsius**(fahrenheit) {

**return** (5 / 9) \* (fahrenheit - 32);

}

### **循环:**

**for** (i = 0; i < 5; i++) {

x += i;

}

### **条件语句:**

**if** (time < 20) {

greeting = "Good day";

} **else** {

greeting = "Good evening";

}

## **对象规则**

对象定义的规则:

* 将左花括号与类名放在同一行。
* 冒号与属性值间有个空格。
* 字符串使用双引号，数字不需要。
* 最后一个属性-值对后面不要添加逗号。
* 将右花括号独立放在一行，并以分号作为结束符号。

### **实例:**

**var** person = {

firstName: "John",

lastName: "Doe",

age: 50,

eyeColor: "blue"

};

短的对象代码可以直接写成一行:

### **实例:**

**var** person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## **每行代码字符小于 80**

为了便于阅读每行字符建议小于数 80 个。

如果一个 JavaScript 语句超过了 80 个字符，建议在 运算符或者逗号后换行。

### **实例:**

### **实例:**

document.getElementById("demo").innerHTML =  
    "Hello W3Cschool.";

[尝试一下 »](https://www.w3cschool.cn/tryrun/showhtml/tryjs_line_break" \t "https://www.w3cschool.cn/javascript/_blank)

## **命名规则**

一般很多代码语言的命名规则都是类似的，例如:

* 变量和函数为驼峰法（ ****camelCase****）
* 全局变量为大写 (****UPPERCASE****)
* 常量 (如 PI) 为大写 (****UPPERCASE****)

变量命名你是否使用这几种规则： ****hyp-hens****, ****camelCase****, 或****under\_scores**** ?

****HTML 和 CSS 的横杠(-)字符:****

HTML5 属性可以以 data- (如：data-quantity, data-price) 作为前缀。

CSS 使用 - 来连接属性名 (font-size)。

注意:- 通常在 JavaScript 中被认为是减法，所以不允许使用。

****下划线:****

很多程序员比较喜欢使用下划线(如：date\_of\_birth), 特别是在 SQL 数据库中。

PHP 语言通常都使用下划线。

****帕斯卡拼写法(PascalCase):****

帕斯卡拼写法(PascalCase) 在 C 语言中语言较多。

驼峰法：

JavaScript 中通常推荐使用驼峰法，jQuery 及其他 JavaScript 库都使用驼峰法。

注意:变量名不要以 $ 作为开始标记，会与很多 JavaScript 库冲突。

## **HTML 载入外部 JavaScript 文件**

使用简洁的格式载入 JavaScript 文件 ( type 属性不是必须的):

<script src="myscript.js">

## **使用 JavaScript 访问 HTML 元素**

一个糟糕的 HTML 格式可能会导致 JavaScript 执行错误。

以下两个 JavaScript 语句会输出不同结果:

### **实例**

var obj =getElementById("Demo")  
var obj = getElementById("demo")

[尝试一下 »](https://www.w3cschool.cn/tryrun/showhtml/tryhtml_syntax_javascript" \t "https://www.w3cschool.cn/javascript/_blank)

HTML 与 JavaScript 尽量使用相同的命名规则。

[访问 HTML(5) 代码规范](https://www.w3cschool.cn/html5/html5-syntax.html" \t "https://www.w3cschool.cn/javascript/_blank)。

## **文件扩展名**

HTML 文件后缀可以是 ****.html**** (或r ****.htm****)。

CSS 文件后缀是 ****.css**** 。

JavaScript 文件后缀是 ****.js**** 。

## **使用小写文件名**

大多 Web 服务器 (Apache, Unix) 对大小写敏感： london.jpg 不能通过 London.jpg 访问。

其他 Web 服务器 (Microsoft, IIS) 对大小写不敏感： london.jpg 可以通过 London.jpg 或 london.jpg 访问。

你必须保持统一的风格，我们建议统一使用小写的文件名。