[所有 2](#_Toc393447395)

[C# 2](#_Toc393447396)

[Css 2](#_Toc393447397)

[Javascript 3](#_Toc393447398)

[Webstorm 11](#_Toc393447399)

[Node.js 11](#_Toc393447400)

[Javascript 测试 11](#_Toc393447401)

[Asp.net MVC 12](#_Toc393447402)

# 所有

1. 命名原则

重点词放在最前面，组合词用“\_”分开。如：

“Index\_Admin”

“Login\_Admin”

# C#

1. Session的key小写开头，如：

Session["userName"]

1. ViewData的key大写开头，如：

ViewData["Data"]

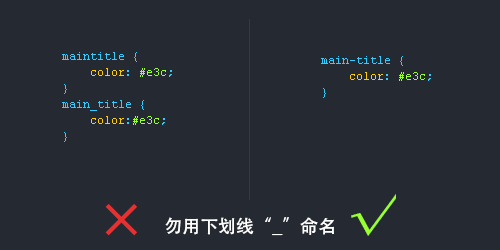
# Css

1. 使用破折号“-”命名css类名

不建议使用“\_”下划线来命名CSS选择器，为什么呢？

* 输入的时候少按一个shift键；
* 浏览器兼容问题 （比如使用\_tips的选择器命名，在IE6是无效的）
* 能良好区分JavaScript变量命名（JS变量命名是用“\_”）

这里有一篇破折号与下划线的详细讨论，英文：[点击查看](http://stackoverflow.com/questions/7560813/why-are-dashes-preferred-for-css-selectors-html-attributes) 中文篇：[点击查看](http://www.cnblogs.com/kaiye/archive/2011/06/13/3039046.html)



# Javascript

1. 统一使用Id命名，不使用ID命名

如Layer有setCanvasId方法，该方法不命名为setCanvasID

1. 注释

（1）“//\*”

表示一级小标题

"//\*\*"

表示二级小标题

。。。以此类推

（2）“//?”

表示未确定的，需要修改或补充的内容

（3）方法注释中，一行代码后出现“//”表示输出结果。如：

/\* 对每个数组元素调用iterator，返回调用iterator返回false的元素索引的数组

如 source\_arr = ["", "a", "b", ""];

result = operate.Judge(arr, function (value, index) {

if (value === "") {

return false;

}

else {

return true;

}

});

//result = [0, 3];

\*/

表述结果为“result = [0, 3];”

1. 序号

序号都是从0开始，0为第一个。

1. 类的方法、属性

方法：

构造函数首字母大写，如“Init”

非构造函数，首字母小写，如“myName”

私有方法加前缀“\_”，保护方法加前缀“P\_”。

使用动词或动词+名词命名。

属性：

首字母小写，如“myName”

私有属性加前缀“\_”，保护属性加前缀“P\_”。

1. 类名

以function定义的类，首字母大写。如：

function Asyn(){}

对象直接量首字母小写。如：

asyn = {

set: function(){}

}

1. 闭包（以类的概念）中的私有成员，加上“\_”前缀

如

var t = (function(){

function \_a(){}

return {

…

}

}());

function A(){

function \_get(){}

this.b = function(){}

}

测试describe中的function不加前缀

如：

Describe(“”,function(){

function a(){}

…

});

1. 参数

（1）函数中定义与形参同名的变量，前面加“\_”。如：

function A(args){

var \_args = null;

...

}

1. 文件

js文件名：

如果该.js文件中定义了一个类，则以大写开头。如：

Index.js

Operate\_Boss.js

如果该.js文件中不是定义了一个类（如var t = {}；或者定义了一个实例等），则以小写开头。如：

index.js

operate\_boss.js

js文件包名：

以小写开头。如：

index

operate\_boss

1. 接口以I开头。
2. 接口、抽象类首字母大写。如：

IIterator（接口）

Layer（抽象类）

1. 命名空间首字母大写。如：

YYC.Tool

1. 全局变量全部大写。如：

SIZE

1. 常量全部大写。如：

FPS

1. 事件句柄小写。如：

onload

onclick

1. 引号

首先使用双引号""，然后嵌套时再使用单引号''

1. 变量申明与初始化放在函数开头。如：

\_loadImg: function () {

var img = null,

i = 0,

len = this.\_images.length,

self = this,

image = null;

函数体...

}

1. for循环，在括号里定义i的初始值（即时已经在变量i申明时已经初始化了！）（因为我不习惯看到：for(;i < len;i ++)）的形式！）。如：

var i = 0,

len = this.\_images.length;

for (i = 0;i < len; i++) {

...

}

1. 枚举变量首字母大写，key全部大写，value从0开始。如：

var State = {

OPEN: 0,

CLOSE: 1

}

1. 配置信息属性最后一部分大写。如：

WIDTH: 50,

speed: {

SLOW: 1,

NORMAL: 2,

FAST: 5

}

1. 一行代码太长时，需要换成多行显示，此时应将操作符换行。

如下所示：

var str = “a”

+ “b”

+ ”c”;

var result = true

&& true;

## Webstorm

文件包名小写

## Node.js

1. 文件包名小写
2. 文件名与js一样的规则
3. 在文件最开始的地方require文件

## Javascript 测试

1. 用function定义函数，用var定义变量，且变量在函数之前定义。如：

var t = 1;

function a(){

xxx

}

1. 产品代码中的测试方法需要加上前缀“forTest\_”。如：

namespace("YE").Director = YYC.Class({

...

Static: {

...

//\*供测试使用

forTest\_clearInstance: function () {

\_instance = null;

},

forTest\_getGameStatus: function () {

return GameStatue;

}

}

});

# Asp.net MVC

1. 文件包名大写，Scripts中的文件包、js文件包的名称小写。如：

Content -> Css

Scripts -> bomber

jquery-1.7.js

1. Content中的资源名小写。如：

Content -> Css -> index.css

Content -> Image ->　index.jpg

1. 项目名大写。如：

Help

JavascriptTest