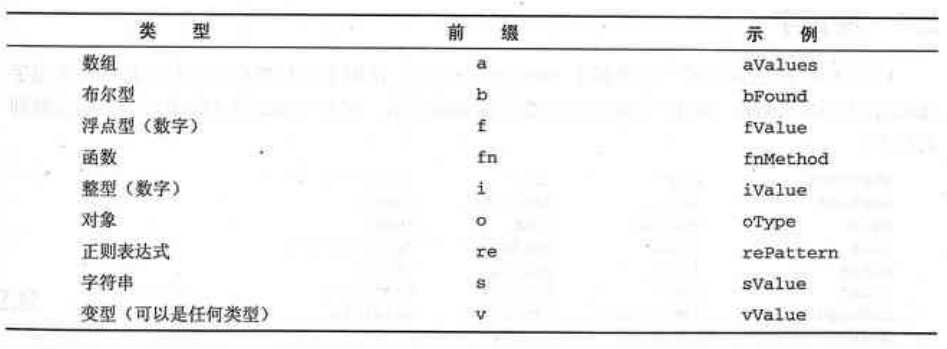
**VIEWSOFT JAVASCRIPT 代码规范**

**一、变量命名规范**

Javascript标示符是指变量的名字，具体是指实参、形参、函数名。必须以字母、下划线、美元符号开头，其他字符还可以使用数字。所以便有了jquery的$()；underscore的\_()。

变量名的重要意义不仅仅在于保护其语法正确，代码中采用恰当好处的命名规范会使得代码变得优雅而可读。大多数人的code水平，都可以从命名变量中窥见一二。经验越丰富，阅历优秀代码越多，写出的程序越赏心悦目！

基本的命名规则：大体有Gamel标记法(首字母大写，余下单词以大写字母开头。如：javascriptLib)、Pascal标记法(所有的单词都以大写字母开头。如：JavascriptLib)、匈牙利类型标记法-在以Pascal标记法命名的变量前附加一个标明该变量类型的小写字母。如：i表示整数，s表示字符串。下面列出了用匈牙利标记法定义ECMAScript变量使用的前缀。



ECMAScript弱类型语言，通俗的讲就是其变量可以保存任何类型的数据。但是弱类型语言不是没有数据类型！ECMAScript有5种基本数据类型：Undefined、Null、Boolean、Number和String。还有1种复杂数据类型Object，Object本质上一组无序的键值对。有点扯远，现在回归到变量的命名上。每个人都有自己的代码风格，笔者看过诸多Javascript框架，也均是风格迥异。以个人最佳实践的角度出发总结以下Javascript变量命名规范，该命名规范借鉴C、Java等语法规范，实战于O Javascript Framework（origin.js）。坚强的的运行在全球超过6000，000个web app之上（此处意淫一下）！

1、实参

\*实参命名，以具有实际意义的英文单词按照Gamel标记法来表示。例如：

function Origin(){

var versionDemo = '1.0';

var authorDemo = 'james yu';

}

\*属性命名

function Origin(){

this.versionDemo = '1.0';

this.authorDemo = 'james yu';

}

2、形参，因为多为提供函数调用。没有实参的上下文环境仅仅依靠变量名较难理解其本质，以匈牙利标记法来标注让弱类型语言也别太弱了！。Number(i,f)、String(s)、Boolean(b)、对象(函数：fn/数组：a/正则：re)

var fnDemo = function(oElement, sClass) {

}

3、函数

\*普通函数/类方法，以Gamel标记法来表示，以动词开头。

function getVersion () {

}

or

var getVersion = function(){

}

function Origin(){

this.getVersion = function(){

}

}

\*类

function Origin(){

}

var origin = new Origin();

\*私有函数，Javascript没有作用范围的概念,所以只能从非语法的角度进行标注。下划线+Gamel标记法来表示。如：\_addEvent()只在Origin函数内部使用！

function Origin(){

this.\_addEvent = function() {

}

}

4、常量，标示符单词字母全部大写，以下划线连接单词。

eg: USER\_ID。

**二、代码规范**

1、Json数据，value中的字符串一律使用""!

var com.viewsoft = {author:"James Yu"};

var author = com.viewsoft&& com.viewsoft.author;

alert(author);