寒假读书心得

寒假在家买了一本JavaScript语言精粹，就是那本蝴蝶书，闲着没事的时候翻着看了看，感觉书里讲的比较精简（可能都是精华吧）。尽管书里尽可能的以最浅显、最容易理解的方法来讲，我还是有些看不懂，应该是我对JavaScript的基本知识不太掌握，而这本书又是深入关注JavaScript最精华、最有用的地方。尽管如此，读了这本书之后，我还是有不少的收获。

首先，我更加了解JavaScript。知道了它能够一诞生，就在没有经过实验室试用和打磨的情况下，被直接集成到网景的Navigator 2 浏览器中、被全世界所接受、正是因为它有一些非常精华的部分。JavaScript是一门重要的语言，因为它是web浏览器的语言。它与浏览器的结合使它成为世界上最流行的编程语言之一。但同时，JavaScript 是最被轻视的语言，因为它不是所谓的主流语言（C、C++、Java、Perl、Python等）。JavaScript令人惊异的事情是，在对这门语言没有太多了解的的情况下，你也能用它来完成工作，它的表达力极强。更重要的是，运行JavaScript程序很简单，只需要一个web浏览器和任意一个文本编辑器即可。

因为书里用“铁路图”来表示语法，所以在具体学习之前，我了解了铁路图的理解规则：

\*从左边界开始，沿轨道到右边界。

\*圆框中是字面量，方块中是规则或描述。

\*沿轨道能走通的序列均合法，反之都是非法的。

\*末端一个竖条，表示允许任意一对符号中间插入空白。而末端两个竖条的则不允许。

关于注释：原来知道JavaScript提供两种注释形式，一种是/\* \*/包围的块注释，一种是以//开头的行注释。在使用时比较随意。用/\* \*/包围的块注释来自PL/I语言，它选择不常见的符号对作为注释的标志，因为它们不大可能出现在语言中。而在JavaScript中，那些字符对也可能出现在正则表达式字面量里，所以块注释有时是不安全的。

关于对象字面量：包围在一对花括号中的零或多个“名/值”对，它可以出现在任何允许表达式出现的地方。如果属性名是一个合法的JavaScript 标识符且不是保留字，并不强制要求用引号括住属性名。比如，“first-name”要括住，而first\_name可以不用括住。

上学期考核时，有关数组方法的题基本都不会，所以我特意看了一下这部分。

关于方法：

\*array.concat(item...)

产生一个数组，包含一份复制并把一个或多个参数item附加在后面。如果参数item是一个数组，那么它的每个元素会被分别添加。例子：

var a = ['a','b','c'];

var b = ['x','y','z'];

var c = a.concat(b,true); // c变成 ['a','b','c','x','y','z',true];

\*array.push(item...)

把一个或多个参数item附加到数组的尾部。和concat 方法不同的是，它会修改数组。如果参数item是一个数组，它会把参数数组作为单个元素添加到数组中并返回这个array的新长度值。例子：

var a = ['a','b','c'];

var b = ['x','y','z'];

var c = a.push(b,true); // a是 ['a','b','c',['x','y','z'],true]

// c 是5

\*array.join(separator)

把一个array构造成一个字符串。它先把array中的每一个元素构造成一个字符串，接着用一个separator分隔符吧它们连接在一起。默认的separator是逗号。例子：

var a = ['a','b','c'];

a.push(d);

var c = a.join(''); // c 是 'abcd'

\*array.pop()

pop方法移除array中的最后一个元素并返回该元素。如果该array是empty，会返回undenfined.例子：

var a = ['a','b','c'];

Var c = a.pop(); // a是['a','b']并且c 是'c'

pop可以像这样实现：

Array .method('pop',function(){

return this.splice(this.length -1,1)[0];

});

\*array.reverse（）

reverse方法反转array里的元素顺序，并返回array本身。例子：

var a = ['a','b','c'];

var b = a.reverse(); // a和b 都是 ['c','b','a']

\*array.shift()

shift方法移除数组中的第一个元素并返回该元素。如果数组是空的，会返回undefined。shift 通常比pop慢得多。例子：

var a = ['a','b','c'];

var b = a.shift(); // a 是 ['b','c'] 并且c 是'a'

Shift 可以这样实现：

Array .method('shift',function(){

return this.splice(0，1)[0];

});

\*array.slice(start,end)

slice方法是对数组中的一段进行浅复制。首先复制array[start]一直复制到array[end]为止。end参数是可选的。默认值是该数组的长度array.length。如果两个参数中任意一个是负数，array.length会和它们相加，试图使它们成为非负数。如果 start大于等于array.length，得到的结果将是一个新的空数组。例子：

var a = ['a','b','c'];

var b = a.slice(0,1); // b 是['a']

var c = a.slice(1); // c 是['b','c']

var d = a.slice(1,2); // d 是['b']