1. 性能优化的将整数转换成字符串的时候用 num+””,如：

Var num = 123 ; 转成字符串：num+”” ,然后可以用typeof()来简单的看看是不是字符串。

1. 在引入js的时候要把所有的<script src=””></script>要放在body的底部。
2. 在获取标签值的时候可以使用 document.getElementById(“”).innerHTML 和 document.getElementById(“”).innerText 来获取值，这两个的差别就是，如果标签里面还有子标签的话，使用 innerHTML返回的时候会带子标签和内容，而innerText 则不会。

**优化处**

1. 在如果要创造节点的时候呢，不要选择 document.createElement(“p”),去创建，这样的js性能不高，要选择 cloneNode()，来复制一个原有的，然后创建，然后 cloneNode()里面的 true 或者 false 的差别在于，如果true 的话，复制了内容回来，而 false则没有滴。
2. 在使用数组、对象、正则时候：

var arr = new Array(); //替换为 var arr=[];

var obj= new Object();//替换为 var obj={};

var reg=new RegExp();//替换为 var reg=’//’;

1. 在使用setTimeout()的时候，不要给setTimeout()传入字符串，要使用如下:

function test(){ num++; }

setTimeout(test,1000);

1. 运算符的运行速度比任何布尔值的还要快，所以判断的时候尽量利用好， || 和 && ，如下:

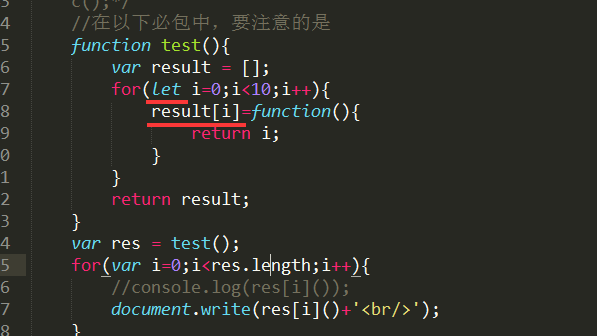
If(num){ doFunc(num); } 替换为： num&&doFunc(num);

If(!e){ var res=evet; } 替换为： var res = e||even;

1. 有个要注意的点来啦，什么时候该使用单引号，什么时候双引号：

在HTML中使用双引号，在JS中使用单引号，然后再JSON的话，用双引号保险，一面解析错误吗。

1. 闭包：就是一个函数可以访问另外一个函数的东西，但我们不知道里面的东西，然后再下图中执行后会看到，输出 10 个 10 的结果，因为var 是就近函数执行的，应改为 let ,因为let 是就近闭包函数执行的，并且在下面中，一定要 result[i] = function(){} 因为只有result的时候代表指针，也就是不会往里面存放东西：



* 1. 定义JS的函数方法有三个：

1. function myFunc(){}
2. myFunc = function(){}
3. var func = new Function(“[虚参]”,”函数体”)(这个是动态匿名函数)
   1. 自调用函数：

(function(){….})();

11、var x={name1:”Jackey”,name2:”Eting”}; 获取值可以用：x.name或者 x[“name”].

12、类型转换的时候，String(x)直接转成字符串，Number(x)转成数字，parseInt(x)返回整数，如果不能转则返回NaN ;返回浮点数是parseFloat(x).

13、在进行try{}catch(err){}的时候，如果没有自定义throw 的时候，错误信息是 err.message ，如果有throw的话，错误信息是 err 就可以了；

14、JS在进行调试的时候，使用浏览器的F12来进行console.log()来进行调试，也可以设置debugger 来设置断点bug调试。

15、命名规范：

**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)。

JS、PHP、MySQL 的基本都是下划线(res\_data)