1，把值赋给尚未声明的变量，该变量将被自动作为全局变量声明。

carname="Volvo"; //将声明一个全局变量 carname，即使它在函数内执行。

2，javaScript 只有一种数字类型

var y=123e5;      // 12300000，科学计数法

var z=123e-5;     // 0.00123

1. 变量的值设置为 null/undefined来清空变量。

null==undefined，true；null===undefined，false

4，在 HTML 中, 全局变量是 window 对象: 所有数据变量都属于 window 对象。

可通过window.parent.var获取其他页面的变量。

自定义全局变量，或者函数，可以覆盖 window 对象的变量或者函数。

5，可以使用索引位置来访问字符串中的每个字符：

var character = 'HappyEveryday'[7];

不要创建 String 对象。它会拖慢执行速度，并可能产生其他副作用。

6，For/In，循环遍历对象的属性：

var person={fname:"John",lname:"Doe",age:25};

for (x in person) // x 为属性名

{

txt=txt + person[x];

}

7，JavaScript 标签，label:statements

continue 语句（带有或不带标签引用）只能用在循环中。

break 语句只能用在循环或 switch 中。通过标签引用，可跳出任何代码块

8，js类型转换，Number() 转换为数字， String() 转换为字符串， Boolean() 转化为布尔值。

9，可,使用 constructor 属性来查看对象是否为数组 (包含字符串 "Array")

function isArray(myArray) {

return myArray.constructor.toString().indexOf("Array") > -1;

}

10，typeof 用以获取一个变量或者表达式的类型，返回：number,boolean,string,function（函数

）,object（NULL,数组，对象）,undefined

instanceof 操作符来判断对象的具体类型，arr instanceof Array

11，JavaScript 中，函数及变量的声明都将被提升到函数的最顶部。

var size=0; //声明变量同时初始化的语句不会提升。

12，严格模式：在脚本或函数的头部添加 "use strict"声明。在函数内部声明只在函数内使用严格模式。

13，严格模式的限制：

不允许使用未声明的变量、

不允许删除变量或对象、

函数(delete x)，不允许变量重名，

不允许使用八进制，

不允许使用转义字符，

不允许对只读属性赋值，

不允许对一个使用getter方法读取的属性进行赋值，

不允许删除一个不允许删除的属性，

变量名不能使用 "eval"、"arguments" 字符串，

禁止this关键字指向全局对象。

1. js使用误区：

switch 语句会使用恒等计算符(===)进行比较，

浮点型数据运算时，可转换为对应的整数乘除法来代替；使用(1+2)/10代替(0.1+0.2)。字符串断行需要使用反斜杠(\)，

var x = "Hello \

World!";

定义数组元素/对象，最后不能添加逗号

15，JavaScript 表单

document.forms["myForm"]["fname"]，通过name获取表单组件

HTML表单自动验证，如果表单组件值为空, required属性会阻止表单提交。

约束验证HTML输入属性：disabled、max、min、pattern、required、type

约束验证CSS伪类选择器：:disabled、:invalid、:optional、:required、:valid

约束验证 DOM 方法：checkValidity()，setCustomValidity()，

约束验证 DOM 属性：validity、validationMessage、willValidate

Validity 属性：customError、patternMismatch、rangeOverflow、rangeUnderflow、stepMismatch、tooLong、typeMismatch、valueMissing、valid

16，JSON，对象{}，数组[]；值key:value。

JSON.parse();

JSON.stringify();

17，javascript:void(exp);执行exp，但不返回值；如：javascript:void(0);

18，自调用函数；函数表达式后面紧跟()，则会自动调用。不能自调用声明的函数。

自我调用函数只执行一次。

(function () {

var x = "Hello!!"; // 我将调用自己

})();

19，函数显式参数(Parameters)与隐式参数(Arguments)；

函数显式参数在函数定义时列出。

函数隐式参数在函数调用时传递给函数真正的值。

为参数设置一个默认值：y = y || 0;

arguments.length，返回函数接收到的参数个数。Arguments[i]获取第i个参数。

通过值传递参数时函数对值的修改不影响函数外的值。

通过对象传递参数时函数对对象的修改会修改函数外的对象。

20，闭包，私有化变量。

闭包是可访问上一层函数作用域里变量的函数，即便上一层函数已经关闭。

var add = (function () {

var counter = 0;

return function () {return counter += 1;}

})(); //add = function(){return counter +=1;}，自调用函数只执行一次。