Js组成1.核心ecmascript2.dom3.bom

<script>console.log()</script>

<scriptsrc=”title.js”></script>

必须使用分号结束

Varage;Age=25;

Varage=25;

\u4e00--->\u9fa5chinese

typeof(num)获取数据类型

NaN:notanumbei

变量.toString（）

paeseInt(数据)碰到第一个非整数字符就停止，第一个就是非整数就为NaN

varresult=parseFloat("35.25");//35.25

varresult=parseFloat("35.2你好!");//35.2

varresult=parseFloat("你好35.2");//NaN

varresult=Number(数据)将字符串解析为number，如果包含非法字符，则返回NaN

==:判断等于不比较类型，只比较数值

===:全等除数值之外，连同类型也会一起比较

isNaN()会抛开数据类型来判断数据是否为数字

逻辑与:&&

如果第一个条件的结果已经为false的话，那么就不会再判断第二个条件,那么整个表达式的值，就是false如果第一个条件的结果是true的话，则继续判断(执行)第二个条件(表达式)，并且以第二个表达式的值，作为整个表达式的值

短路||

如果第一个条件的结果已经为true的话，那么就不会再判断第二个条件，那么整个表达式的值，就是true如果第一个条件的结果是false的话，则继续判断(执行)第二个条件(表达式),并且以第二个表达式的值，作为整个表达式的值

表达式1?表达式2:表达式3;

表达式1是一个条件，值为boolean类型

若表达式1的值为true，则执行表达式2的操作，并且以表达式2的结果作为整个表达式的结果

若表达式1的值为false，则执行表达式3的操作，并且以表达式3的结果作为整个表达式的结果

vara;//仅声明提前

console.log(a);//undefined

a=100;//赋值仍保留在原位置

console.log(a);//100

varn=100;//全局变量n

functionfun(n){//参数变量也是局部变量

n-=3;//修改的是局部变量n

console.log(n);//输出的是局部变量n

}

fun(n);//按值传递，方法内输出97;

console.log(n);//输出全局变量的值：100

1、if语句，条件位置处，如果条件不是boolean类型的话，JS会自动进行转换以下情况if(0/0.0/"“/null/undefined/NaN){}

除以上情况外，一律为真

if(1){console.log("真!");}

if("我帅吗"){console.log("真!!!!");}

2、if语句块的{}，如果省略，if只控制它下面的第一条语句

Switch（）{

Case1：

语句；

Break；

Case2：

语句；

Break；

Default：

语句；

}

两个case或多个case之间，没有任何的可执行代码，那么就以最后一块的case为主

do{

可执行语句;

}while(boolean表达式);

1、先执行循环操作

2、再判断循环条件如果条件为真，则继续执行循环操作如果条件为假，则终止循环即退出循环

特点：无论条件满足与否，至少要执行一次循环操作

for (表达式1; 表达式2; 表达式3) {

循环体语句;

}

for循环执行过程：

1. 计算表达式1的值；

2. 计算表达式2（boolean表达式）的值，如果为true则执行循环 体，否则退出循环；

3. 执行循环体；

4. 执行表达式3；

5. 计算表达式2，如果为true则执行循环体，否则退出循环；

6. 如此循环往复，直到表达式2的值为false。

用于实现固定次数循环：类似for I in range() : [python]

for ( var i = 0 ; i < 10 ; i + + ) {

console.log( i );

}

Var arr=[]; or var arr = new Array();

下标从0到length-1

Length的值永远是最大下标+1

var nums = [50, 90, 20, 10];

for( var i=0; i< nums.length; i++){

nums[ i ] += 10;

}遍历

var bookInfo = [ ];

bookInfo['bookName'] = '西游记' ;

bookInfo['price'] = 35.5 ;

关联数组 可自定义名字

遍历使用for in 循环

For (var key in hash){

Key 元素下标名

Hash[key]

}

String（arr）:转换为字符串，逗号分隔

固定套路: 对数组拍照: 用于鉴别是否数组被修改过

arr.join(‘连接符’) 自定义连接符分隔

判断数组是空数组: arr.join("")==""

concat() 拼接两个或更多的数组，并返回结果

var subArr=arr.slice(starti,endi+1)

选取arr中starti位置开始，到endi结束的所有元素组成新数组返回——原数组保持不变。

含头不含尾

复制数组arr.slice(0,arr.length); 可简写为:arr.slice();

arr.splice(starti,n); 删除arr中starti位置开始的n个元素不考虑含头不含尾

var deletes=arr.splice(starti,n); 返回值deletes保存了被删除的元素组成的临时数组

插入：

arr.splice(starti,0,值1,值2,...) 在arr中starti位置，插入新值1,值2,...原starti位置的值 及其之后的值被向后顺移

替换：

其实就是删除旧的，插入新的 arr.splice(starti,n,值1,值2,...) 先删除arr中starti位置的n个值，再在starti位置插入新值

强调: 删除的元素个数和插入的新元素个数不必一致。

arr.reverse() 颠倒数组中元素的顺序颠倒

arr.sort()将所有元素转为字符串再排列。

Bom

SetInterval（exp，time）

setInterval(function(){

console.log("Hello World");

},1000);

var timer = setInterval(function(){

console.log("Hello World");

},1000);

clearInterval(timer);停止

setTimeout（exp，time）：

setTimeout（function（）{

alert（）}；，

3000）；