2017.6.14

一:

<script type="text/javascript">

function test1(){

if(false){

alert("你真聪明!");

}

alert("测试结束!");

}

function test2(){

var sex="你猜";

if(sex=="女"){

alert("美女,你有男朋友吗?");

}else{

alert("帅哥,你有女朋友吗?");

}

alert("测试结束!");

}

function test3(){

var age=-1;

/\*

\* 30>age>18: 普通用户

\* 60>age>30: VIP用户

\* 100>age>60: 建议...

\*/

if(30>age && age>18){

alert("普通用户");

}else if(60>age && age>30){

alert("VIP用户");

}else if(100>age && age>60){

alert("建议回家");

}else{

alert("白白");

}

alert("测试结束!");

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-14

控制语句:

为什么使用控制语句?

在程序中经常需要 对 数据 或 信息 进行判断,然后进行选择性处理.比如: 判断用户的性别,如果是男 提供 抽烟场所. 否则,提供 空调休息场所.

若用户年龄>18,允许用户使用/访问/操作. 否则,提醒/警告/拒绝用户使用或操做.

JS中的控制语句:

if语句 switch语句.

特点: 选择性执行[多个代码块,最多执行1处]

if语句:

1.if中文意思: 如果 假如

2.格式:

a.

if(条件){

当条件为true时,执行此处代码

}

b.

if(条件){

当条件为true时,执行此处代码

}else{

当条件为false时,执行此处代码

}

c.

if(条件){

当条件为true时,执行此处代码 .并结束整个if代码块的执行

}else if(条件){

当条件为true时,执行此处代码 .并结束整个if代码块的执行

}else if(条件){

当条件为true时,执行此处代码 . 并结束整个if代码块的执行

}

......

[

else{

当条件为false时,执行此处代码

}

]

-->

JS程序 <br>

<!-- 第一种 -->

<input type="button" value="确定1" onclick="test1()">

<!-- 第二种 -->

<input type="button" value="确定2" onclick="test2()">

<!-- 第三种 -->

<input type="button" value="确定3" onclick="test3()">

</body>

</html>

二:

<script type="text/javascript">

function test1(){

var sex="男";

switch(sex){

case "男" : alert("帅哥,你好!"); break;

case "男" : alert("靓女,你好!"); break;

default: alert("滚蛋!");

}

alert("测试结束!");

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-14

控制语句:

为什么使用控制语句?

在程序中经常需要 对 数据 或 信息 进行判断,然后进行选择性处理.比如: 判断用户的性别,如果是男 提供 抽烟场所. 否则,提供 空调休息场所.

若用户年龄>18,允许用户使用/访问/操作. 否则,提醒/警告/拒绝用户使用或操做.

JS中的控制语句:

if语句 switch语句.

特点: 选择性执行[多个代码块,最多执行1处]

switch语句:

1.switch中文意思: 开关 交换 调换 控制

2.格式:

switch(表达式|值){

case 值1: 代码1; break;

case 值2: 代码2; break;

......

[

default: 默认代码;

]

}

说明:

1.在java及其他语言中要求,表达式的值必须与 case值 类型相同. 但是 JS不限制,因此JS是弱类型语言.

2.在java及其他语言中要求,表达式的值只能为 整数[long除外] 和字符. 但是JS不限制

3.在java及其他语言中要求,case值必须唯一. 但是JS不限制

原理:

JS执行switch代码块时,先执行表达式,然后拿着表达式的值 从上到下 开始 与 case 值匹配.

若表达式值与case值相同,将指定此处代码.执行结束后,将结束整个switch代码块.

若表达式值与case值不相同,将继续与下一个case执行进行比较.

若表达式值与case值都不相同,将执行default代码或结束整个switch代码块.

-->

switch语句测试 <br>

<!-- 第一种 -->

<input type="button" value="确定1" onclick="test1()">

</body>

</html>

三:

<script type="text/javascript">

function test1(){

//输出10次 你好,JS

/\*alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");

alert("你好,JS");\*/

/\*for( var i=0; i<10; i++){//0 1 2 3 4 5 6 7 8 9

alert("你好,JS "+i);

}\*/

/\*//输出数组中的所有元素

var allNum=[11,22,333,444,"AA",1,2,8,9];

//获得数组元素个数: 数组名.length

alert("元素个数:"+allNum.length);

for(var k=0; k<allNum.length; k++){

alert(allNum[k]);

}\*/

//输出10以内的偶数 / 奇数

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

if(i%2!=0){

alert("奇数:"+i);

}

}

}

</script>

</head>

<body>

for循环测试. <br>

<!--

为什么使用循环语句?

为了提高代码利用率及减轻开发,JS提供了循环语句用于重复执行某些代码.

JS含有哪些循环语句?

for循环, for in循环[forEach循环],while循环 ,do while循环,with循环[自学]

for循环:

格式:

for( 循环初始值; 循环条件; 循环控制条件 ){

循环体: 需要重复执行的代码

}

原理:

JS执行for循环时,先执行循环初始值,然后判断循环条件是否为true.

若循环条件为true,将执行循环体.循环体执行结束后,执行循环控制条件.

然后再次判断循环条件是否为true,若为true,将再次执行循环体.

依此类推,直到循环条件为false时,结束整个for循环的执行.

-->

<input type="button" value="确定" onclick="test1()">

</body>

</html>

四:

<script type="text/javascript">

function test1(){

var allNum=[11,22,33,44];

for(var i in allNum){

//alert("名字:"+i);

alert(allNum[i]);

}

}

</script>

</head>

<body>

for in循环测试. <br>

<!--

为什么使用循环语句?

为了提高代码利用率及减轻开发,JS提供了循环语句用于重复执行某些代码.

JS含有哪些循环语句?

for循环, for in循环[forEach循环],while循环 ,do while循环,with循环[自学]

for in循环: [实际就是forEach]

格式:

for(变量类型 变量名 in 数组名/集合 ){

循环体: 需要重复执行的代码

}

原理:

从集合或数组中取出一个元素,并将元素的名字[索引]赋给变量.

然后执行循环体.当循环体执行结束后,取出下一个元素并将名字赋给变量,再次执行循环体.

依此类推,直到取出所有元素为止.

用途:

专门用于输出集合或数组中的所有数据.

-->

<!-- 普通循环 -->

<input type="button" value="确定1" onclick="test1()">

</body>

</html>

五:

<script type="text/javascript">

function test1(){

//输出10次 你好JS

var i=0;//循环初始值

while(i<10){//循环条件

alert("你好,JS "+i);

i++;//循环控制条件

}

}

</script>

</head>

<body>

while循环测试. <br>

<!--

while循环:

1.while中文意思: 当 ..时候

2.格式:

while(条件){

循环体: 需要重复执行的代码

}

原理:

当JS执行while循环时,先判断循环条件是否为true.若条件为true,将执行循环体.

当循环体执行结束后,将再次判断循环条件是否为true.若条件为true,将再次执行循环体.

依此类推,直到循环条件为false时,结束整个while循环.

延伸:

for循环与while循环的区别:

1.for循环中通常需要知道循环次数,while循环中通常不需要知道循环次数.

-->

<input type="button" value="确定1" onclick="test1()">

</body>

</html>

六:

<script type="text/javascript">

function test1(){

var i=0;//循环初始值

do{

alert("你好,JS "+i);

i++;//循环控制条件

}while(i<10);//循环条件

}

function test2(){

var i=0;//循环初始值

do{

alert("do while: "+i);

i++;//循环控制条件

}while(i>10);//循环条件

var j=0;

while(j>10){

alert("while: "+j);

j++;

}

}

</script>

</head>

<body>

do while循环测试. <br>

<!--

do while循环:

1.do表示做 执行 while表示当..时候

2.格式:

do{

循环体:需要重复执行的代码

}while(条件);

原理:

当执行do while循环时,先执行do后面的循环体,然后再判断条件是否为true.

若条件为true,将再次执行循环体.循环体执行结束后,再次判断条件是否为true,若条件为true,将再次执行循环体.

依此类推,直到条件为false时结束整个 do while循环.

do while与while区别:

1.while循环中,每次都需要先判断循环条件,然后才会执行循环体

2.do while循环中,第一次执行循环体不需要判断条件,直接执行.

3.while循环的循环体可能一次都不执行,do while循环的循环体至少执行一次.

-->

<!-- do while原理 -->

<input type="button" value="确定1" onclick="test1()">

<!-- do while与 while区别 -->

<input type="button" value="确定2" onclick="test2()">

</body>

</html>

七:

<script type="text/javascript">

function test(){

//获得标签对象

var num1=document.getElementById("n1");

//获得标签的值

var num1Val=num1.value;

//简写:

//document.getElementById("n1").value;

var num2Val=document.getElementById("n2").value;

if(num1Val>num2Val){

alert("第一个数大: num1="+num1Val);

}else{

alert("第二个数大: num2="+num2Val);

}

}

</script>

</head>

<body>

输出最大值. <br>

num1:<input type="text" id="n1" ><br>

num2:<input type="text" id="n2"><br>

<input type="button" value="测试" onclick="test()">

</body>

</html>

八:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>100以内的和.html</title>

<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">

<meta http-equiv="description" content="this is my page">

<meta http-equiv="content-type" content="text/html; charset=UTF-8">

<!--<link rel="stylesheet" type="text/css" href="./styles.css">-->

<script type="text/javascript">

function test(){

/\*

\* 1+2+3+..+99+100 = 5050

\* 思路: 前N个数的和 + (N+1) \*\*\*\*> 前N+1个数的和

\* sum=1+2

\* sum新= sum+3 =1+2+3

\* sum新2= sum新+4=1+2+3+4

\*/

var num=document.getElementById("num").value;

var sum=0;//前N个数的和

var i=1;

while(i<=num){

//alert(i);

//console.log(i);//将i以日志信息的方式 打印到控制台

sum=sum+i;

i++;

}

alert("结果:"+sum);

}

</script>

</head>

<body>

计算100以内整数的和. <br>

0到:<input type="text" id="num">

<input type="button" value="计算" onclick="test()">

</body>

</html>

九:

<script type="text/javascript">

function test(){

/\*

\* 三行四列

\* \*\*\*\*

\* \*\*\*\*

\* \*\*\*\*

\*/

/\*for(var i=0; i<3; i++){

document.writeln("\*\*\*\*<br>");

}\*/

var row=10;

var col=40;

//外层循环控制 行数

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

//document.writeln("\*\*\*\*<br>");

//内层循环控制列数

for(var j=0;j<col;j++){

document.write("\*");

}

document.writeln("<br>");

}

}

</script>

</head>

<body>

打印矩形 <br>

<input type="button" value="打印" onclick="test()">

</body>

</html>

十:

<script type="text/javascript">

function test(){

/\* \*

\* \*\*

\* \*\*\*

\* \*\*\*\*

\* \*\*\*\*\*

\*

\*

\*/

//外层循环控制 行数

for(var i=0; i<9; i++){

//document.writeln("\*\*\*\*<br>");

//内层循环控制列数

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

document.write("\*");

}

document.writeln("<br>");

}

}

</script>

</head>

<body>

打印三角形 <br>

<input type="button" value="打印" onclick="test()">

</body>

</html>

十一:

<title>打印乘法表.html</title>

<script type="text/javascript">

function test(){

//外层循环控制行数

document.write("<pre>");

for(var i=1; i<=9; i++){

//document.write("\*\*\*\*\*\*\*\*\*\*<br>");

for(var j=1; j<=i; j++){

document.write(i+"\*"+j+"="+(i\*j)+"\t");

}

document.write("<br>");

}

document.write("</pre>");

}

</script>

</head>

<body>

打印乘法表 <br>

<input type="button" value="打印" onclick="test()">

</body>

</html>