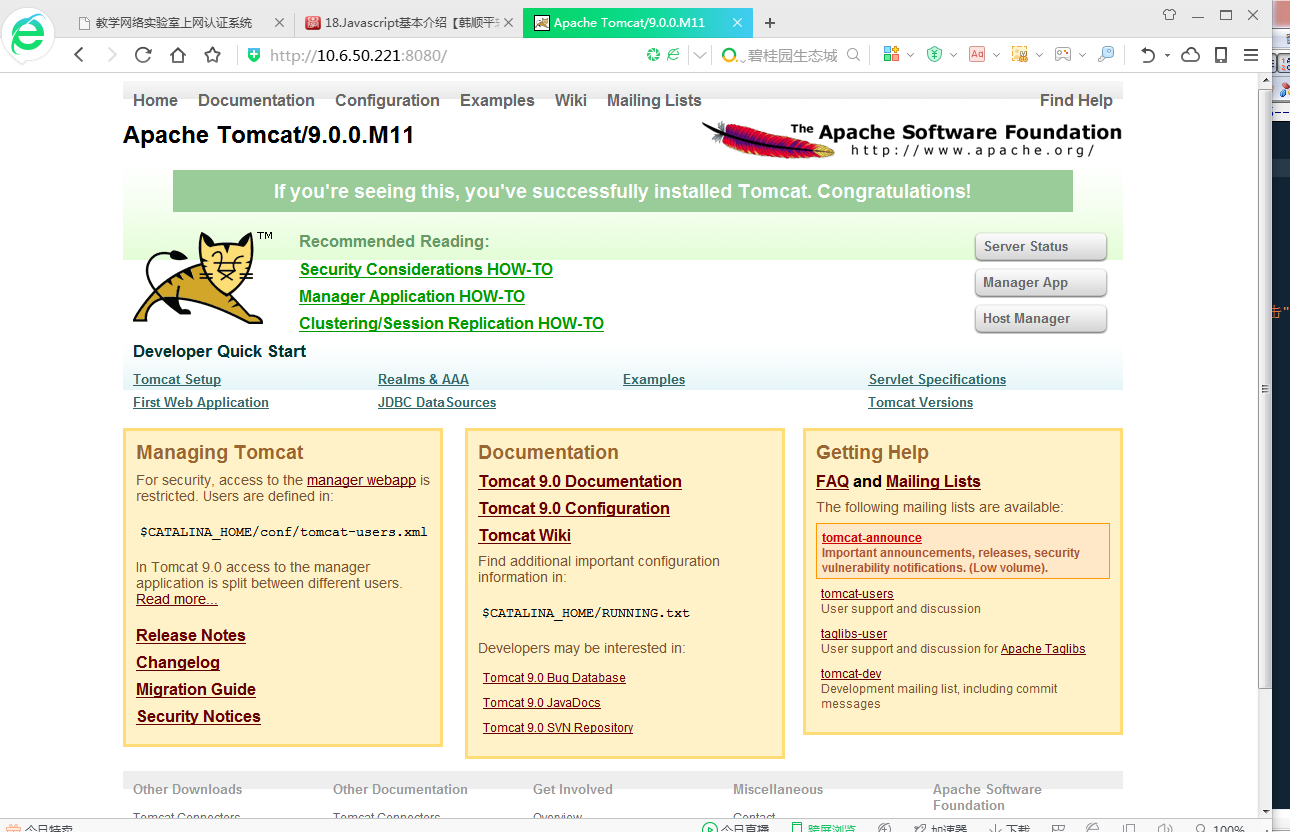
Js基础语法

Js基本介绍：

1：js是用于web开发的脚本语言。

脚本语言是什么？

1. 脚本语言往往不能独立使用，它和HTML，PHP，asp，asp.net配合使用。
2. 脚本语言有自己的变量，函数，控制语句（顺序，分支，循环）
3. 脚本语言实际上是解释性语言/编译语言（即在执行的时候，直接对源码进行执行）。
4. Java程序：.java->.class->jvm js->浏览器（js引擎来解释执行）
5. Js在客户端执行。



**Js代码：**

<html >

<head>

<script language="javascript">

function test(){

window.alert("hello");

}

</script>

</head>

<body>

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

</body>

</html>

-----------------------------------------------

1. 若没有使用<script>包起来，则浏览器会将其视为普通文本。
2. Js代码位置可以随意。（一般是放在head标签间）

<html >

<head>

<script language="javascript">

// 脚本语言，语法并不严谨。

//（js中变量的定义，js中变量用var表示，不管实际类型。）

var num1=456;

var num2=223;

var result=num1+num2;

window.alert('结果是'+result);//alert函数；

</script>

</head>

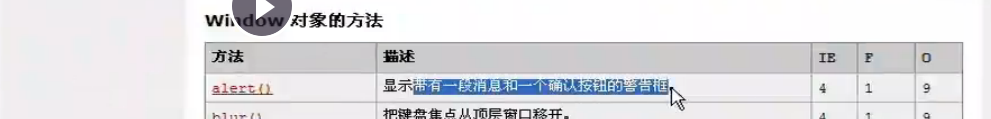
<body>

</body>

</html>

2.在一个html文件中，（jsp/php/asp）可以出现多对<scipt>，浏览器会按先后顺序执行。

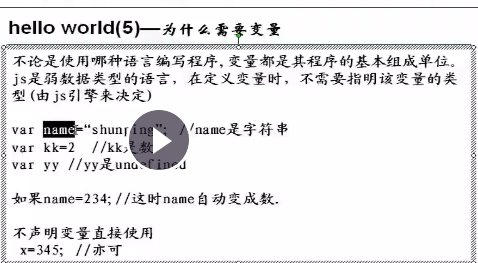
对前面的程序改为简单的加法运算程序：

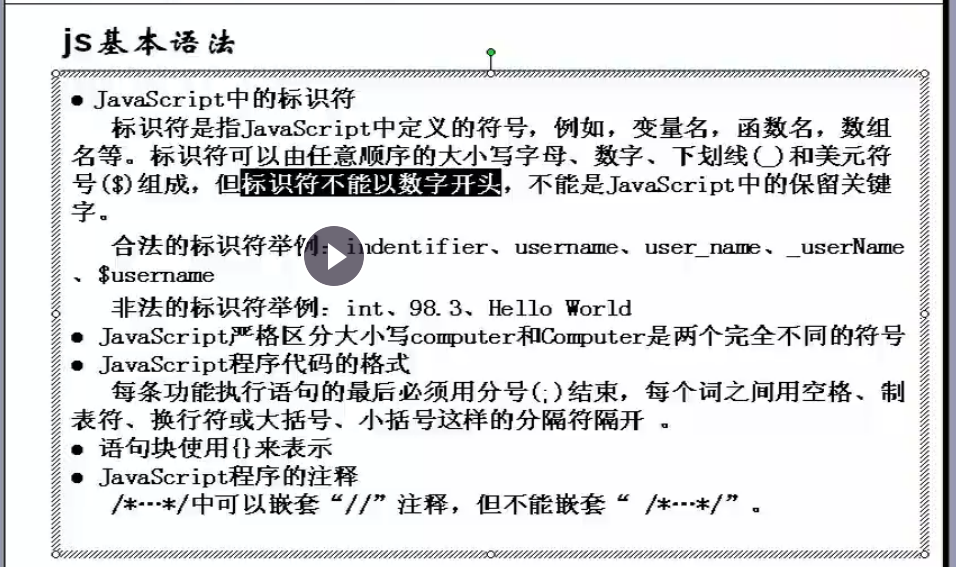


1）js是弱数据类型语言。

即：在定义变量的时候可以统一使用var表示，甚至可以去掉var这个关键字。

2）js中的变量的数据类型是由js引擎来决定的。

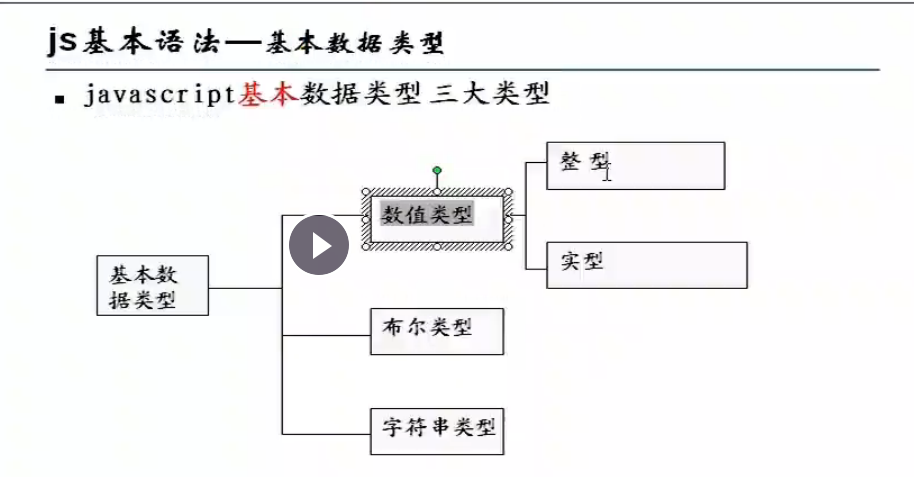




单行注释//

多行注释/\* \*/

4.



1. **js的数据类型：**

1：基本数据类型：

1）数值

2）字符串

3）布尔型

通过typeof关键字，可以看到变量的具体数据类型是什么。

举例：

<html >

<head>

<script language="javascript">

var n2=456;

var n1="ssss";

window.alert("n1是"+typeof n1);

window.alert("n2是"+typeof n2);

n1=565;//js的动态语言

window.alert("n1是"+typeof n1);

</script>

</head>

<body>

</body>

</html>

2.js的复合数据类型

复合数据类型： 1）数组

2）对象

1. js的特殊数据类型

1）null

var a=null;

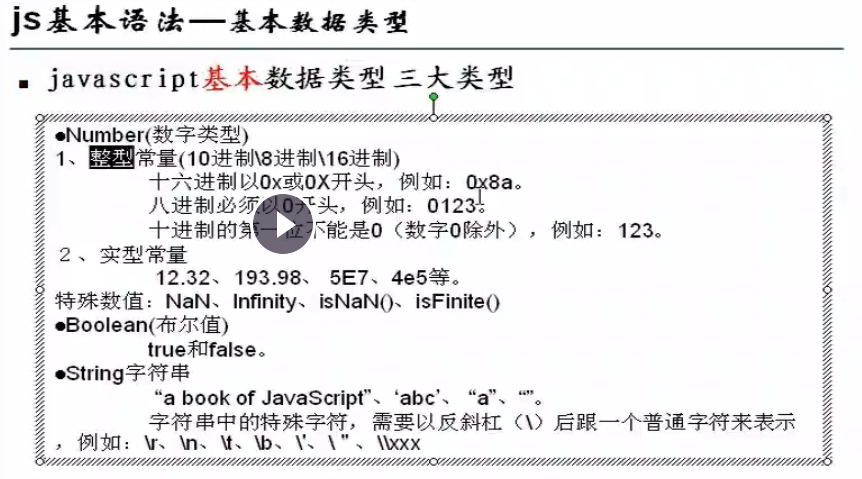
2)undefine

window.alert(t);

//报错：未定义。

var aa;//没有给值

Window.alert(aa);//如果未初始化，则显示的是：undefine.



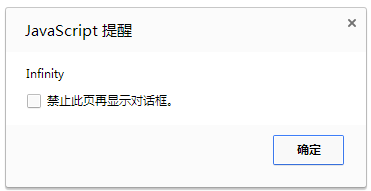
1.实型：小数类型

特殊类型：

NaN：不是一个数



Infenity:无穷大



有两个函数可以判断NaN，Infenity

<script language="javascript">

var a="abc";

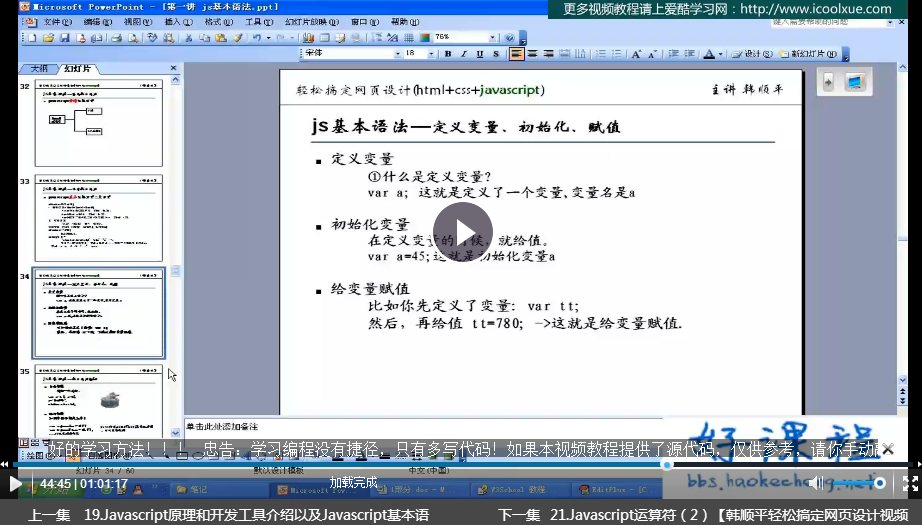
window.alert(isNaN(a));

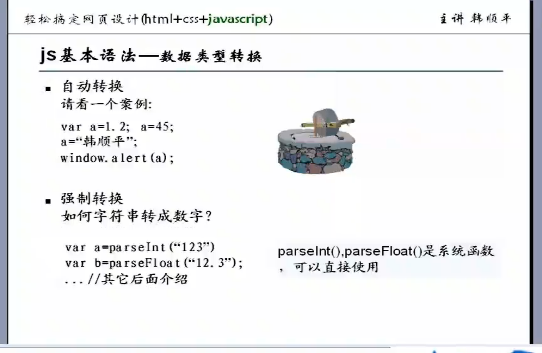
</script>

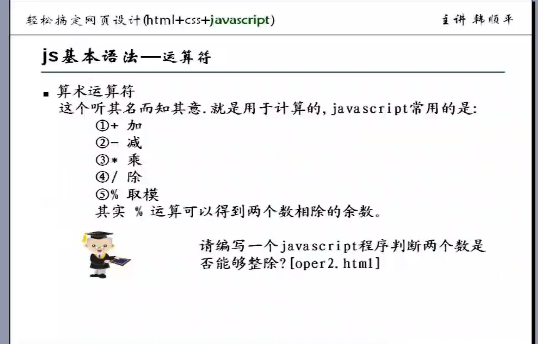
是数返回false，不是数返回true;



4.定义变量，初始化变量，给变量赋值。







Js的运算符：

+、-、\*、\、%

<script language="javascript">

var a=90;

var b=9;

window.alert(a%b);

if(a%b==0)

{

window.alert("能整除！");

}

else

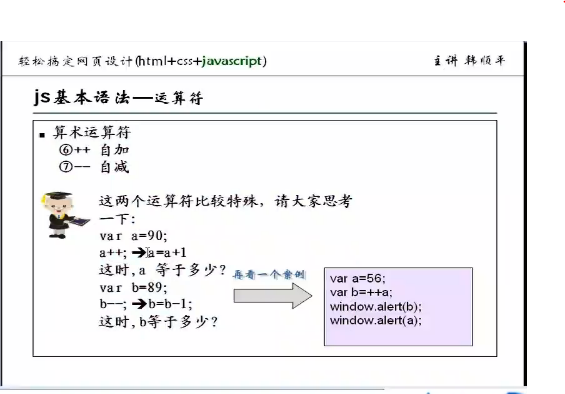
{

window.alert("不能整除！");

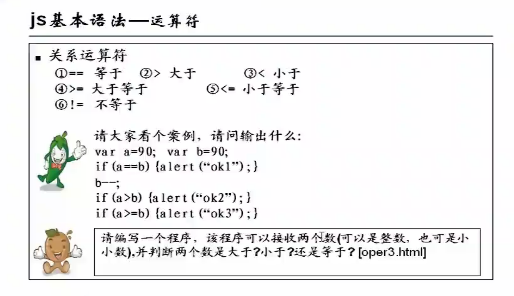
}

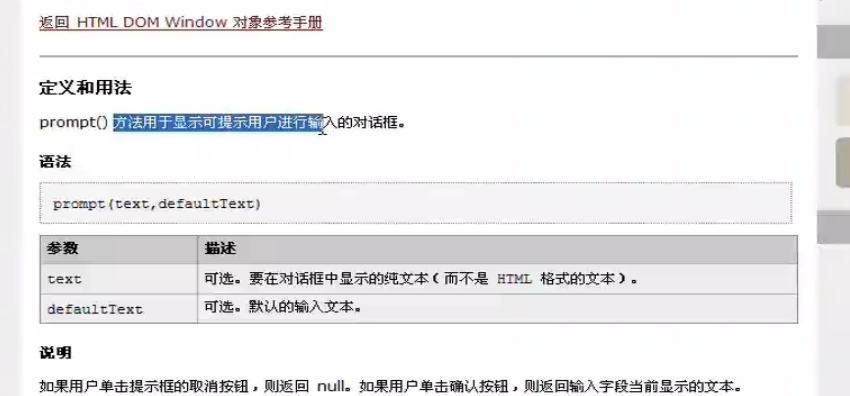
</script>

//取模主要用于整数型。



<http://www.icoolxue.com/play/1306>





介绍window.prompt,和document.writeln();

**(1)**

**<script language="javascript">**

**var va1=window.prompt("请输入");**

**var va2=window.prompt("请输入");**

**document.writeln("你的输入是："+(parseFloat(va1)+parseFloat(va2)));**

**</script>**

**(2)**

**<head>**

**<script language="javascript">**

**var va1=window.prompt("请输入");**

**var va2=window.prompt("请输入");**

**va1=parseFloat(va1);**

**va2=parseFloat(va2);**

**if(va1>va2)**

**window.alert("va1>va2");**

**else if(va1<va2)**

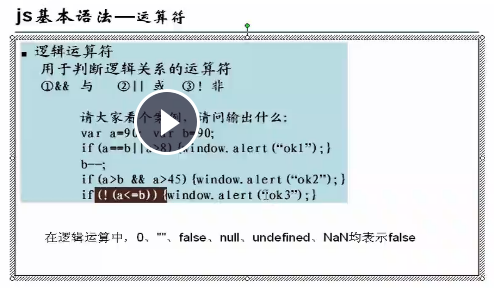
**window.alert("va1<va2");**

**else**

**window.alert("va1=va2");**

**</script>**

**</head>**



逻辑运算符：

**在逻辑运算中，0，“”，false,null,undefined,NaN均为false;**

**//Var a=0;**

**//Var a=””;**

**Var a=null;**

**If(!a)**

**{**

**Window.alert(“ok”);**

**}**

**在js中，||究竟返回什么值。这是一个非常重要的知识点。**

（1）

**<script language="javascript">**

**var a=0;**

**var b="";**

**var c=false;**

**var d=a||b||c;**

**window.alert(d);**

**</script>**

**//返回第一个不为false的值。当全部为false时，返回最后一个值。||返回的结果不一定是BooLean,大多数是布尔型。**

（2）

**<head>**

**<script language="javascript">**

**var a=0;**

**var b="";**

**var c=false;**

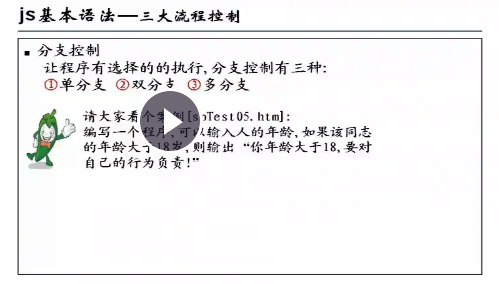
**var obj=new Object();**

**var d=a||b||c||obj;**

**window.alert(d);//返回第一个不为false的值。当全部为false时，返回最后一个值。**

**</script>**

**</head>**



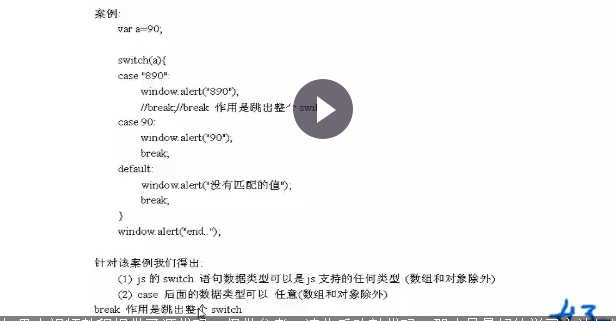


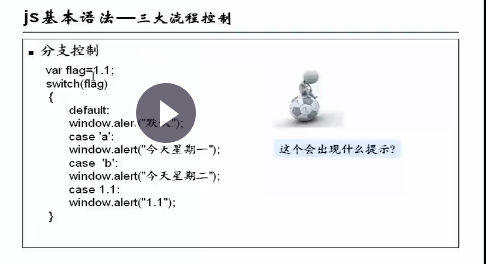
小结：（1）js的switch语句数据类型可以是js支持的任何类型（数组和对象除外）。

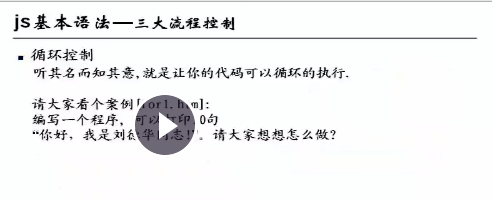
（2）case后面的数据类型可以任意。

（3）break作用跳出整个switch;

（4）default如果没有匹配，则执行。







---------------------------------------------------------------------------------------------------------------------------------js调试：



F11：逐句执行，遇到函数则跳入到函数中，再一句一句执行。

F10：按过程执行，当遇到一个函数的时候，F10就会把这个函数当成一个语句执行，而不跳入执行。

Shift+F11：从一个函数中跳出。

局部变量：可以看到变量的变化情况。

断点：可以看断点情况及取消断点。

==============================================================================================================================================================

**js函数**

1. 为什么需要函数
2. 函数的基本概念

基本语法：

Function 函数名 （参数列表）

{

//代码

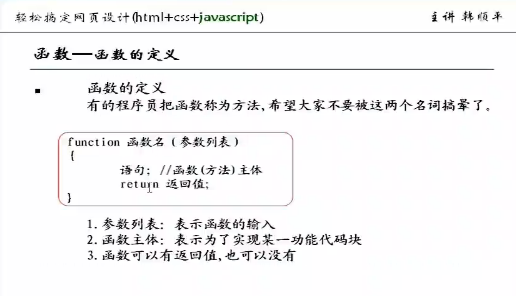
Return 值；//可以选择。

}

**说明：**

1. **参数列表**
2. **函数主体**

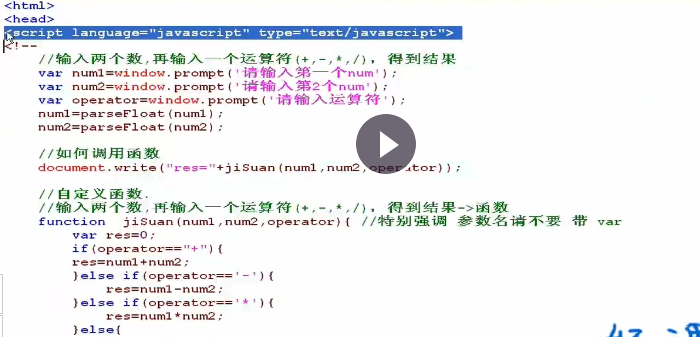
**3.函数可以有返回值，也可以没有。**

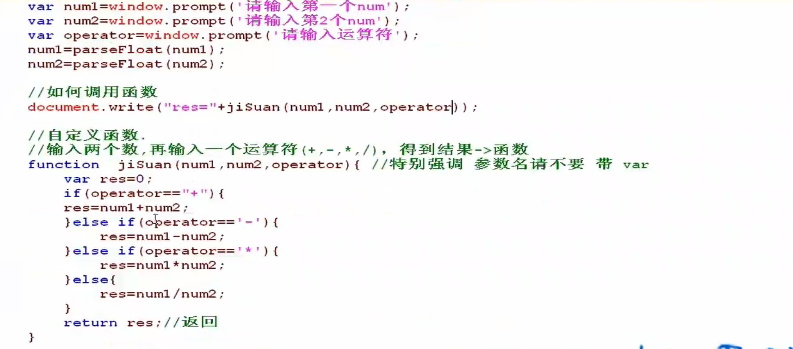


1. 自定义函数：

输入两个数，再输入一个运算符(+,-,\*,/),得到结果。

入门案例：





问题二：

如果其他的html，php要使用该函数要怎么办？

把上面的函数单独提出，写到js文件中，在需要的地方引入即可。

**Html文件：（该实例中，html文件与js文件放在相同目录下）**

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<script language="javascript" src="myfunction.js"></script>

<script language="javascript" >

var a=2;

var b=3;

document.write(jisuan(a,b));//函数的调用1

Test（‘kello’）;(1}

Window.alert(test);//将函数显示出来

Var myvar=test;

Window.alert(myvar)

Myvar(‘中国北京’);

</script>

</head>

<body>

</body>

</html>

**Js文件：**

function jisuan(a,b)//特别强调，参数名请不要带 var.

{

return a+b;

}

Function test(a)

{

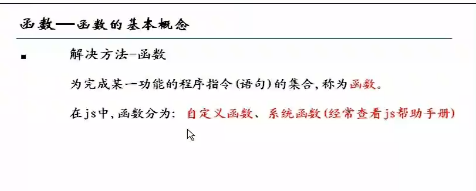
Windo.alert(“你输入的是”+a)；

}

在需要的文件中引入js：

<script language="javascript" src="jd路径"></script>

**系统函数：**





**(1)**

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<script language="javascript" src="myfunction.js"></script>

<script language="javascript" >

var a=2;

var b=3;

document.write(jisuan(a,b));

//<!--

var str="window.alert('ok!')";

//需求：是把str当做一个脚本执行。

eval(str);//系统函数，全局函数。

//-->

</script>

</head>

<body>

</body>

</html>

**（2）**

<script language="javascript" >

//escape和unescape往往配合使用，可以防止乱码出现。

var str1="顺平";

var str2=escape(str1); //编码

window.alert(str2);

var str3=unescape(str2);//反码

window.alert(str3);

</script>





**（3）**

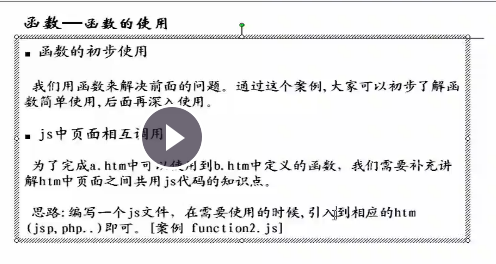
<script language="javascript" >

window.alert("ok");

window.moveTo(200,200);

window.resizeTo(200,300);

</script>



**函数的调用的方式**

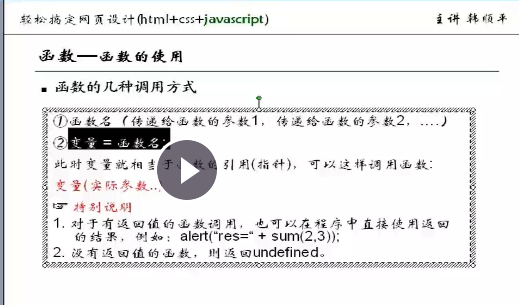
1. 普通调用

函数名（实际参数....）

1. 通过指向函数的变量去调用

Var myvar=函数名；

Myvar（实际参数）



**3.**

**关于接收函数返回值的问题讨论：**

Var myvar=test(‘abc’);//如果test函数没有返回值，但是你又接受了，则返回的就是undefined

如果有返回值，则是什么就是什么。

Window.alert(myvar);

**eg:**

**Html文件：**

<script language="javascript" >

/\* test('hello,world');

var myvar=test;

myvar('中国北京');\*/

var myvar=test('中国北京');

/\*带参数的话，相当于将函数返回的值交给它；

如果没有返回值的话，程序不会报错，但会返回undefined。

不带参数，相当于将函数的引用或指针交给它。\*/

</script>

**Js文件**

function jisuan(a,b)//特别强调，参数名请不要带 var.

{

return a+b;

}

function test(z)

{

window.alert("你输入的是："+z);

return 90;

}