

第 2 章

JavaScript 基础

主讲师：黄玲





- 在页面引入 JavaScript 代码的方式有哪些？



- 常用的输入输出语句有哪些？

学习目标



- ▲ • 掌握常用数据类型
- ▲ • 简单记忆基本语法
 - 变量的定义与赋值
 - 数据类型与转换 parseFloat 等
- 掌握逻辑控制语句
 - 条件语句
 - 循环语句
- ▲ • 掌握函数定义
- ▲

本章任务



- ▲ 任务 1 显示变量数据类型
- ▲ 任务 2 根据成绩给出学生考评
- ▲ 任务 3 实现猜数游戏
- ▲ 任务 4 制作简易计算器

简易计算器 页面



任务 1: 拍卖叫价



▲ 任务描述：

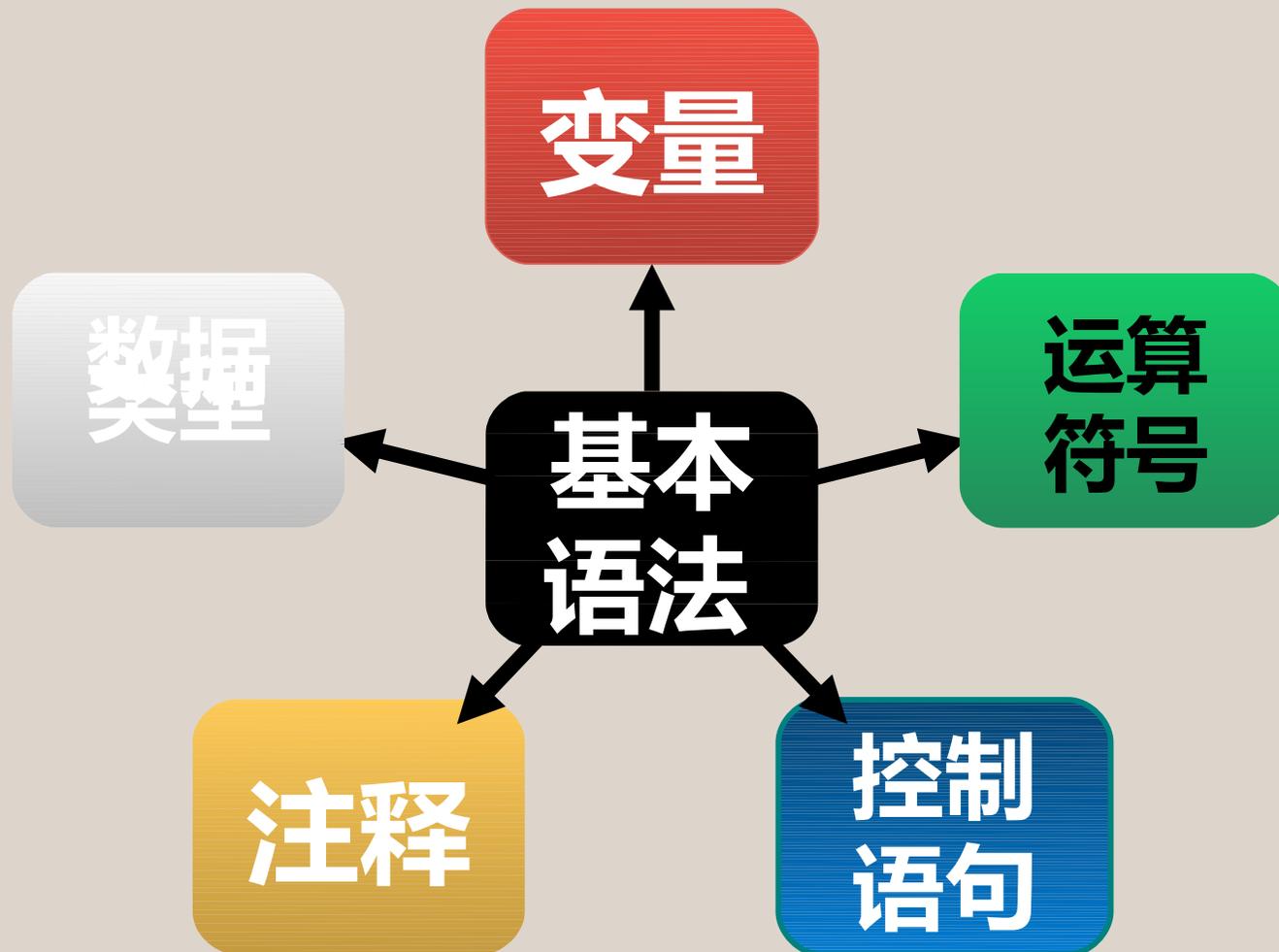
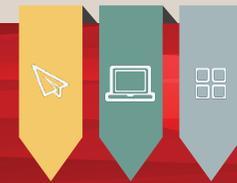
在页面显示最终拍卖价格。

▲ 任务分析：

在页面显示变量数据类型需要以下步骤：

1. 声明变量
2. 为变量赋值
3. 使用 `typeof` 方法显示数据类型名称

1.JavaScript 的基本语法



1.1 变量



- 变量名必须以字母或下划线（ "_" ）开头
- 变量可以包含数字、从 A 至 Z 的大小写字母
- JavaScript 区分大小写，即变量 `myVar`、`myVAR` 和 `myvar` 是不同的变量



变量的声明和赋值



定义变量

```
var count;
```

“var” - 用于声明变量的关键字
“count” - 变量名

赋值

```
count = 5;
```

同时声明和赋值变量

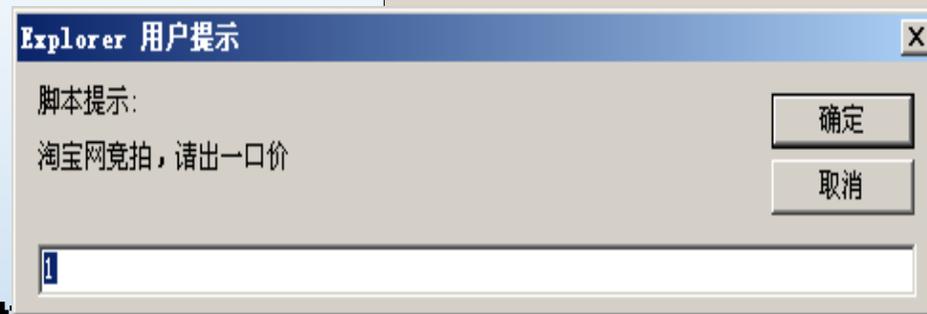
```
var count = 10;
```

```
var x, y, z = 10;
```

声明变量

```
<HTML>
<HEAD>
<TITLE> 使用变量 </TITLE>
<SCRIPT LANGUAGE = "Javascript">
var x;
x=prompt(" 淘宝网竞拍 , 请出一口价 ",1);
document.write(" 拍卖价格 "+x+"<br/>") // "+" 用来连接多个子字符串
document.write(" 恭喜您 , 您以最高价拍卖成功! ");
alert(" 欢迎下次光临! ");
```

`prompt("提示信息", "默认值")`
将弹出提示对话框, 接受用户的输入。点击确定返回输入的字符串, 点击取消反馈空字符串。

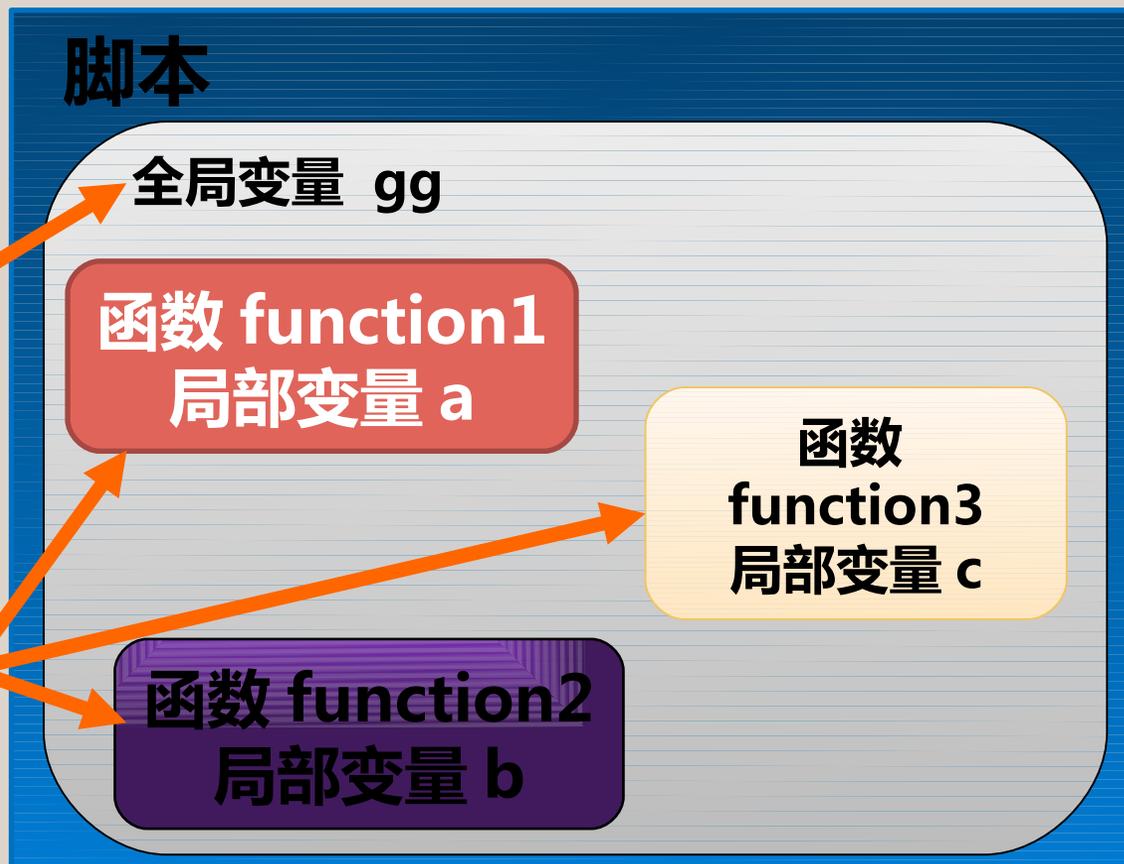


变量的作用域



可由函数 1、函数 2
和函数 3 访问

变量 a、b 和 c 只
能在其各自的函数中
被访问



根据编程目的决定将变量声明为全局变量还是局部变量。

常量



- 整型
- 浮点型
- 字符串型

字符	说明	示例
<code>\b</code>	退格符	<code>alert(" 这是第一句 \b 这是第二句")</code>
<code>\f</code>	换页符	<code>alert(" 这是第一局 \f 这是第二句")</code>
<code>\n</code>	换行符	<code>alert(" 这是第一局 \n 这是第二句")</code>
<code>\r</code>	回车符	<code>alert(" 这是第一局 \r 这是第二句")</code>
<code>\t</code>	制表符	<code>alert(" 这是第一局 \t 这是第二句")</code>

浮点型 - 浮点型字面量至少必须含有一个数字。此数字可包含小数点或采用科学记数法表示的数字。科学记数法中的整数可以是正整数 (+) 或负整数 (-) , 指数 (e) 表示 “十次幂”。

例如 10.24、1.20e+22、4E-8、.1815 等。

和 C 语言一样，js 也有转义字符，常用的就是：“`\n`”

1.2 数据类型



弱类型，区分大小写，句尾分号非强制要求。

- **数值型 (number)**
 整型、浮点型
- **字符串型 (string)**
 单引号或双引号引起来
- **布尔型 (boolean)**
 true, false
- **undefined 声明后没有初始值**
- **Null 未定义或不存在或定义后没赋值**
- **案例**



运算符



- **算术运算符**

+、-、*、/、%、++、--

- **比较运算符**

==、!=、>、>=、<、<=

- **逻辑运算符**

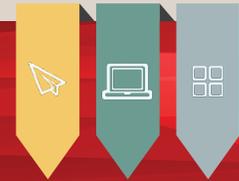
&&、||、!

- **条件运算符**

? :



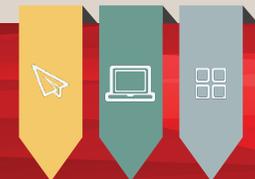
练一练：



▲完成拍卖加价小程序，实现以下功能



"+" 号的用法



```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "JavaScript">
var x=100;
var y;
var z;
document.write("<H2> 竞拍 SONY 数码相机 600 万像素 "
+x+"$ 起价 </H2>");
y=prompt(" 加多少银子? ", "1");
z=x+y;
alert(" 您最终的出价 \n"+z+"$"); // "\n" 用于换行显示
</SCRIPT>
</HEAD>
<body>
</body>
</HTML>
```



Prompt 函数返回输入的字符串
+ 字符串相连：
100+" 200"

1.3 类型转换



▲ **parseInt (String)**

将字符串转换为整型数字

如：`parseInt ("86")` 将字符串 "86" 转换为整型值 86

▲ **parseFloat (String)**

将字符串转换为浮点型数字

如：`parseFloat ("34.45")` 将字符串 "34.45" 转换为浮点值 34.45

"+" 号的用法



```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "JavaScript" >
var x=100;
var y;
var z;
document.write("<H2> 竞拍 SONY 数码相机 600 万像素 "
+x+"$ 起价 </H2>");
y=prompt(" 加多少银子? ", "1");
z=x+parseFloat( y );
alert(" 您最终的出价 \n"+z+"$"); //"\n" 用于换行显示
</SCRIPT>
</HEAD>
<body>
</body>
</HTML>
```

parseFloat() 函数将字符串转换为 float 数据
parseInt() 函数将字符串转换为 int 数据
如果转换失败, 返回 NaN 值 (not a number)

任务 2: 根据成绩给出学生考评



任务描述：



提示用户输入成绩，根据成绩给出学生的考评：

如果成绩在 90-100 分，考评为“优”，

如果成绩在 80-89 分之间，考评为“良”，

成绩如果在 70-79 分之间，考评为“中”，成绩如果在 60-

69 分之间，考评为“及格”，否则为“不及格”。

1. 条件语句



▲ · if 条件语句

▲ · switch 多分支语句

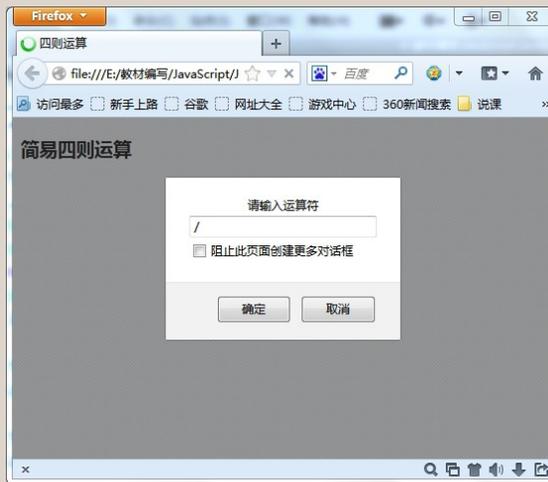
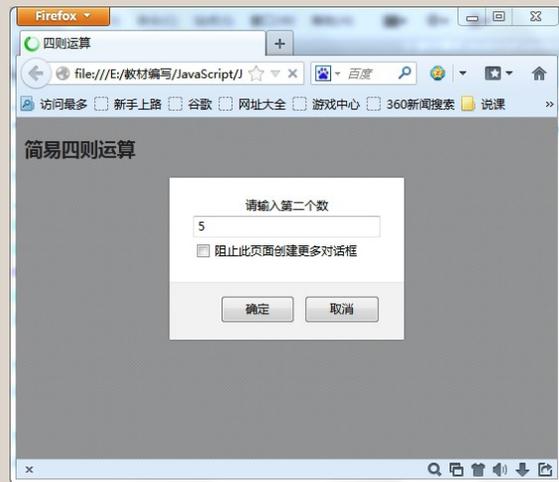
```
if( 条件 )
{
    //JavaScript 代码 ;
}
else
{
    //JavaScript 代码 ;
}
```

```
switch ( 表达式 )
{
    case 常量 1 :
        JavaScript 语句 1;
        break;
    case 常量 2 :
        JavaScript 语句 2;
        break;
    ...
    default :
        JavaScript 语句 3;
}
```

练一练：



▲ 使用 prompt 和控制语句设计四则运算



任务 3: 实现猜数游戏



任务描述：



系统随机生成一个 1-100 之间的数，然后让玩家猜该数。若玩家猜对该数，游戏则结束；若玩家猜得不对，则计算机告知玩家，数字猜大了还是小了；当玩家最终猜对数字，提示恭喜猜对了，并告知猜了几次，当玩家中途退出游戏则提示游戏没有完成。

下面进行猜数游戏
请输入1-100之间的整数：

^_^,你猜的数字大了

 阻止此页面创建更多对话框

恭喜你，猜对了，幸运数字是：81 您共猜了7次

 阻止此页面创建更多对话框

1. 循环语句



- ▲ · for 语句
- ▲ · while 语句
- ▲ · for...in 语句
- ▲ · 中断循环语句

```
for( 初始化; 条件;  
    增量 )
```

```
{  
    语句集;  
}
```

```
while( 条件 )
```

```
do{
```

```
    语句集
```

```
for( 变量 in 对象 )
```

```
{  
    语句集;  
}
```

break 语句

continue 语句

1. 循环语句



· for...in 语句遍历数组

```
<script type="text/javascript">
  var cars=["Audi","Volvo","BMW","Benz"];
  for(var i in cars)
  {
    document.write(cars[i]+"<br/>");
  }
</script>
```

· **结果** Audi Volvo BMW Benz



任务 4: 制作简易计算器



任务描述：

在页面实现简易计算器，用户在页面输入第一个数和第二个数，点击相应操作符将操作结果显示在计算结果文本框中。



什么是函数



- ▲ · 函数的含义：
 - 类似于 Java 中的方法，是执行特定任务的语句块。

如何希望点击某个按钮后才显示 “HelloWorld” ，并能输入显示的次数，怎么办？

演示示例：调用函数输出 “HelloWorld”



什么是函数

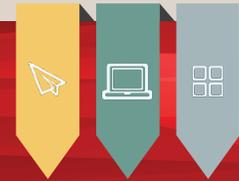


- 1、在页面中放置一个文本框和一个按钮，让用户输入次数后点击按钮才显示
 - 2、把循环显示的代码放置到函数中，当点击按钮时，才调用函数显示 Hello World
- 经过分析，该如何解决上面问题？

使用自定义函数



如何使用函数



创建函数

```
function 函数名 ( 参数 1, 参数 2 , ... )  
{  
    语句 ;  
}
```

调用函数

函数调用一般和表单元素的事件一起使用，调用格式为：

事件名 = “函数名” ；



如何使用函数



```
function showHello( )  
{  
  var count=document.myForm.txtCount.value ;  
  for(i=0; i<count; i++)  
    document.write("<H2>HelloWorld</H2>");  
}
```

表示单击此按钮时，调用函数
showHello() 执行

```
<INPUT type="submit" name="Submit" value=" 显示 HelloWorld"  
onClick="showHello( )">
```



简单了解获取文本框值的方法



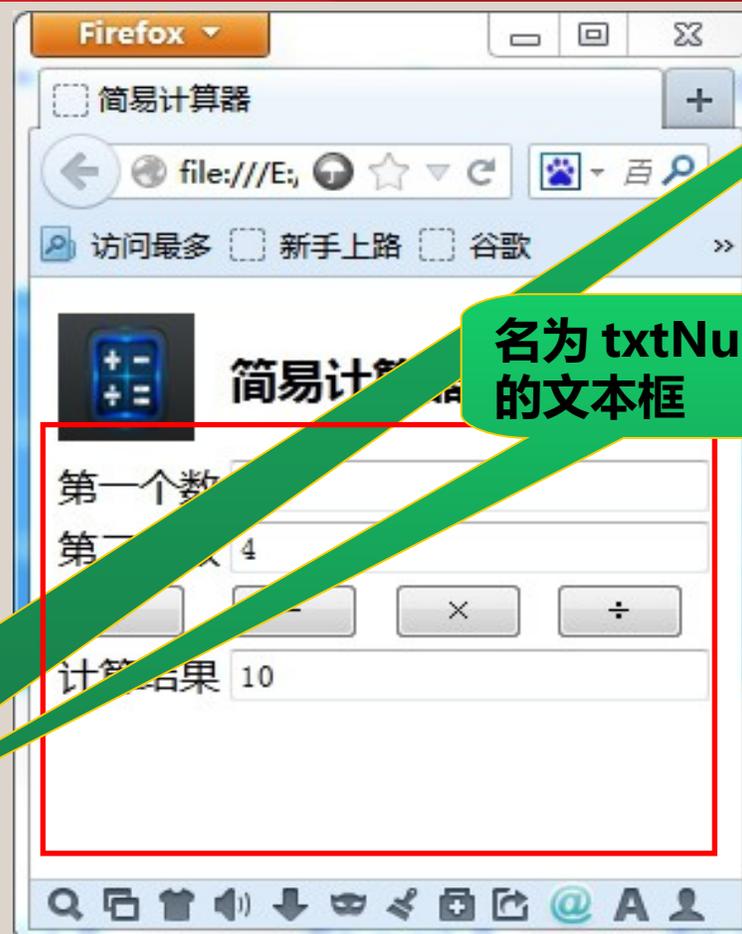
- `document. 表单名 . 文本框名 .value`
- 从文本框中获取的值都是字符串，
- 若进行加减，都要进行类型转换。



改进任务 4 的程序



▲ 编写如左图所示，具有能对两个操作数进行加、减、乘、除运算的简易计算器



名为 txtNum1 的文本框

名为 myform 的表单

提示：加、减、乘、除四个按钮分别调用四个函数实现两个数相加、相减、相乘和相除



如何使用函数



- 在简易计算器中 4 个按钮调用的函数的代码很类似，怎么优化代码？

比较 4 个函数，只有运算符号不一样，将运算符号作为函数的参数即可。
怎么用 一个函数来代替 4 个结构相似的函数？

使用有参函数



定义有参函数



```
<HEAD>
<SCRIPT language= JavaScript
function compute(op)
{
    var num1,num2;
    num1=parseFloat(document.myform.txtNum1.value);
    num2=parseFloat(document.myform.txtNum2.value);
    if (op=="+")
        document.myform.txtResult.value=num1+num2 ;
    if (op=="-")
        document.myform.txtResult.value=num1-num2 ;
    if (op=="*")
        document.myform.txtResult.value=num1*num2 ;
    if (op=="/" && num2!=0)
        document.myform.txtResult.value=num1/num2 ;
}
</SCRIPT> </HEAD>
```

定义有参函数 compute()，完成计算功能，op 参数代表运算符。



调用有参函数



调用有
参函数

```
<FORM action="" method="post" name="myform" id="myfo
.....
<TR>
  <TD><INPUT name="addButton2" type="button"
    id="addButton2" value=" + " onClick="compute('+')">
  </TD>
  <TD><INPUT name="subButton2" type="button"
    id="subButton2" value=" - " onClick="compute('-')">
  </TD>
  <TD><INPUT name="mulButton2" type="button"
    id="mulButton2" value=" × " onClick="compute('*')">
  </TD>
  <TD><INPUT name="divButton2" type="button"
    id="divButton2" value=" ÷ " onClick="compute('/')">
  </TD>
</TR>
.....
</FORM>
```



养成良好的编程习惯



命名风格

- 函数主要是用来实现功能，通常用“动词 + 名词是”形式
- 变量名通常是用来存储数据，通常用“名词“或”形容词 + 名词”形式
- 变量名以小写字母开头，对于多个词的情况，如果没有分隔符，则从第二个开始每个词语第一个字母大写。
- 类的命名通常是使用名词，类名应该以大写字母开头。



注释



单行注释以 // 开始，以行末结束

例如：

```
<SCRIPT language= "javascript" > // 表示 JavaScript 代码的开始
```

多行注释以 /* 开始，以 */ 结束，符号 /*..... */ 指示中间的语句是该程序中的注释。

例如：

```
/*  
    helloWorld.html  
    2007-9-29  
    第一个 JavaScript 程序  
*/
```



总结



- 掌握 JavaScript 中 “弱类型” 的含义
- 能熟练运用 JavaScript 中的各种逻辑控制语句：if 条件语句、switch 多分支语句、for/while/do while 循环语句
- 掌握函数的创建和调用方法
- 学会使用 JavaScript 的几个方法：document.write()、alert()、prompt()、parseInt() parseFloat()



谢谢观看

