1，JavaScript的用途

JavaScript用来制作web页面交互效果，用于提升用户体验

web前端三层来说: w3c的规范 （行内样式 淘汰）

结构层 HTML 从语义的角度，描述页面结构

样式层 CSS 从审美的角度，美化页面

行为层 JavaScript 从交互角度，提升用户体验

ECMAScript是一个标准

Ajax异步交互技术

JavaScript是弱变量类型的语言，变量只需要用var来声明

html:掌握所有标签的使用，以及标签的语义。

css:掌握页面常见的布局，以及属性的使用。

拿到一个页面以后能有自己独立完成这个页面的思路。

JavaScript:变量的使用，数据类型，if判断，以及循环语句。

1，代码存放位置: 如果没有style代码放在title标签下，如果有style，那么代码放在style下面

alert（）； 在页面上弹出一个提示框

2，关于javascript代码书写格式: <script type="text/javascript"> <script>

不存在JavaScript标签，只有<script>

3，JavaScript与html，css，对空格，换行，以及锁紧是不敏感的

规范：将来在写代码的时候一定要所有的代码块都放在script标签中，

4，javascript的执行顺序：从上到下，从左到右

5，javascript注释 //注释一行代码 注释多行代码/\*\*/

6，javascript代码虽然对空格，换行，缩进不敏感，但是在写的时候一定要注意规范 ：

alert(); 弹框

console.log(""); 向浏览器的控制台输出一句话

作用:用来进行代码调试

prompt(""); 弹出一个输入框，给用户提供输入信息的位置。

直接量:如果页面上直接使用数据，那么这些内容叫做直接量

数据类型string:

1,字符串: 用引号引起来的文本。

注 意:引号既可以是引号也可以用双引号

如果字符串有双引，可以将字符串的引号变成单引号

转译符; \n;换行 \t;缩进 \b;空格 \r;回车 \\

数值类型number;表达的是一种数据的量

加号的作用;两个字符串用加号链接，那么加号起到的作用是连接这个两个字符串

如果是两个数值用加号连接，作用是运算符

typeof判断数据类型

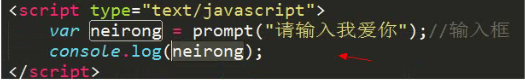
typeof;直接量/变量

## 变量;变量是用来存储数据的

prompt("");输入框

console.log("");

alert("");



1. **声明变量;var来声明**

var a;

**2 给变量赋值;**

a=”123”;

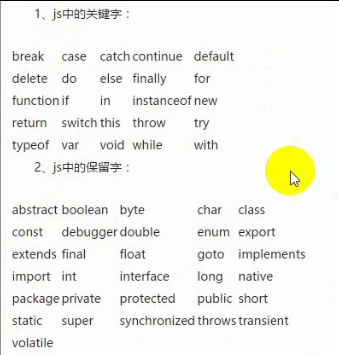
**输出**; console.log（）相当于system.out.println（）

**3，变量名的注意点; 变量名只能用英文字母，数字，下划线以及￥符号组成，并且数字不能放在名称开头。**

**4，变量的命名不能使用javascript中的关键字和保留字**

**关键字; 已经被javascript内部使用过的。**

**保留字; 还没有被JavaScript内部使用，但是可能有一天会被使用（备胎）**



**7，变量的数据类型;这个变量中存储的是什么样的数据，那么这个变量就是什么样类型。**

**8，变量之间的赋值和输出**

**9，boolean类型**

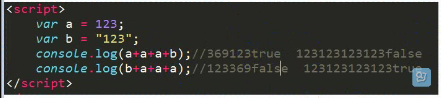
**取值; true;对**

**False;错**

**用于判断条件的结果**

**10，undefined; 如果一个变量声明了但没有赋值，他的结果就是undefined**

**11，运算符**



**12， NaN ; not a number🡺 非数字**

**作用 ; 用来表示数值的一种不正常的状态 一般会在计算错误的时候出现**

**判断当前的数据是否是NaN的关键字; isNaN; 是一个非数字**

**如果结果是非数字，则返回true**

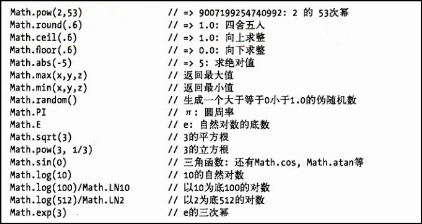
**如果结果是数字，则返回false**

**13，运算的先后顺序; 先乘除，再加减**

**14，% 取余 :**

**15,（ ）; 改变运算的优先级**

## Math对象



**逻辑运算符的优先级：**

**! > && > ||**

**== (是否相等) 判断数据内容，不判断数据类型**

**===（判断是否全等）既关心内容也关心类型**

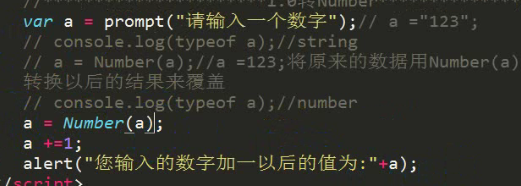
**注意：有一个比较特殊的数字NaN，它特殊到自己都不等于（==，===）自己**

**！== 只要内容或者类型不一样，就返回true，反之为false**

**通过prompt（）； 输入数据的类型：不管我们在输入框中输入什么样的内容，在浏览器上都显示string类型**

## 数据类型的转换

**1，Number类型转换**



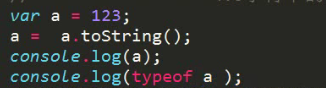
1. **如果转换的内容可以转成数字，那么就直接返回这个内容对应的数字**
2. **如果不可以转换，那么返回NaN**
3. **如果在内容中出现小数，那么小数会保留**
4. **如果内容为空，那么转换成0**
5. **parseInt（）将内容转换成Number**
6. **如果转换的内容可以转成数字，那么就直接返回这个内容对应的数字**
7. **如果不可以转换那么直接返回NaN**
8. **如果内容中出现小数，那么小数不会保留 不是四舍五入**
9. **如果第一个字符是数字，则继续解析至字符串解析完毕或者遇到一个非数字符号为止**
10. **parseFloat（）转Number**

**与parseint一样，唯一不同的是可以保留小数**

**Boolean在内存中存储的时候 true用1表示 false为0**

# 字符串的转换

1. **如果要将其他类型转成字符串类型，直接调用toString()方法可以将内容转成字符串**



1. **String() 直接将要转的内容放在String后的括号中，就可以将内容转换成字符串**

# Boolean转换

1. **Boolean(); 注意：除了false，0，NaN，空字符，undefined在转换的时候会转成false以外，其他的都会转成true**

隐式转换 **隐式转换不通过程序员自己去写固定的代码转换，浏览器可直接进行转换**

1. **直接在要转换的内容前加上 +**

**var a = “123”；**

**a = +a；**

1. **var a = “123”;**

**a = a\*1 ;**

**3， var a = “123”;**

**a = a-0;**

**4, var a = “123”;**

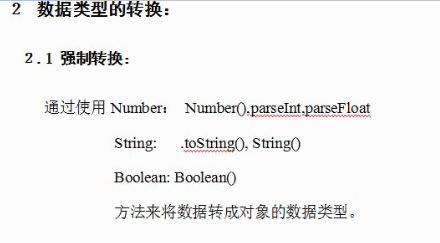
**a = a/1;**

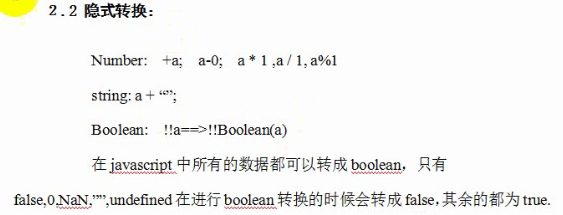
**5, var a = “123”;**

**a = a % 12;**

**隐式转换成字符串； var a = 123； a = a “ “ ;**

**隐式转换成布尔型； var a = 123； a = ! a ;**





# If语句

## 代码的调试：（debug）

1. **找到开发人员工具**
2. **找到sorce，并且找到查看代码的页面**
3. **开始调试： 断点**

# Switch case：

1. **作用与if 是一样的，用来判断多个条件中的哪一个条件满足要求**

**A, case的结构部分必须有break**

**B，case的变量值后面加冒号**

**C，default可写可不写，但是如果不写都满足条件代码直接结束。**

三元运算符（三目运算符）

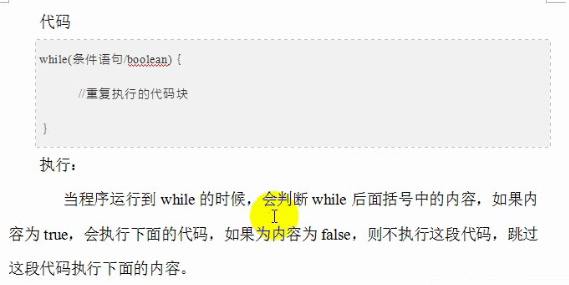
**三元运算符**:

如名字表示的三元运算符需要三个操作数。

语法是 **条件 ? 结果1 : 结果2;**. 这里你把条件写在问号(?)的前面后面跟着用冒号(:)分隔的结果1和结果2。满足条件时结果1否则结果2。

# While循环语句

1. **可以重复完成同样的事情**



**终止循环：break**

**结束本轮循环跳到下一轮：continue**

# Object对象：

**对象的创建：**

**Var a = new object();**

**对象属性的添加：**

1. **name = “zhangsan”;**
2. **age = “19”;**

**对象属性的取值：**

**Console.log(a.name);**

# 数组:Array

**创建数组： var a = new Array(); //创建一个数组**

**a [ ] = “19”; //在数组中保存了一个**

**数组的赋值：**



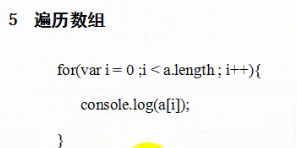
**数组的取值：**

**Console.log（a【 】）； //中括号内的数是数组下标**

**Javascript中的数组定义好了之后就是一个无穷大的容器 ，元素可以任意添加，不限制数量，数据类型**

**元素的下标是从0开始**

**数组的遍历：**



字符串也可以看成一个数组：

Var a = “abc”;

# 函数：function

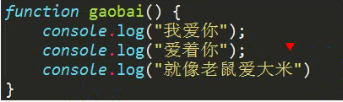
**作用：用来封装一些经常要用到的代码：**

**结构：function 函数名（）{**

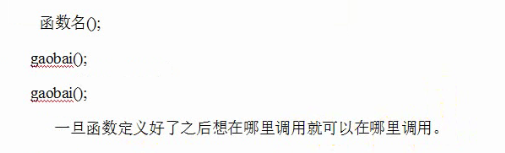
**//要封装的代码**

}

**函数的定义：**



**函数的调用：**



**函数的参数：实参 形参**

**就是放在函数名后面括号中的内容**

**函数的返回值：**

**就是跟在retrun后面的数据，叫做函数的返回值**

**一般情况下函数没有return，那么这个函数的返回值是默认返回：undefined**

**字符串的不可变 判断变量是否有值**

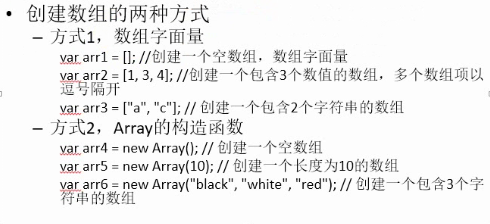
**空字符串，0，null，undefined**

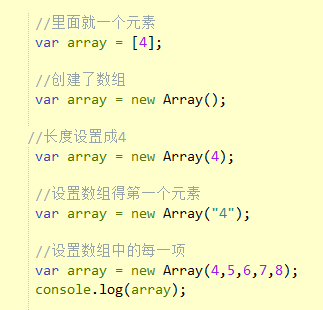
# 数组：

Var array = new Array();

**Array[] = “ ” ;**

**数组中的项，如果没有赋值 则是undefined**

1. **数组：数据中得有序列表，可以存放任意类型的数据，数组的大小可以动态调整。**
2. **数组的两种创建方式：**



**冒泡排序：**



# 函数：

**1,函数的基本语法**

**2,会使用函数解决一些问题**

**函数可以封装一段JavaScript代码。他只定义一次，但可以被执行或调用任意多次**

**编程就是把需求分球成一组函数与数据结构的能力**

**函数的定义：**

**function 函数名 （var1，var2，var3）{**

**//函数体**

**}**

**函数的调用：**

**函数名();**

**返回值计算的结果，函数返回，后续代码不会执行**

**function funName (形式参数) {**

**//函数体**

**}**

**//在调用函数的时候，传入的参数叫做实参**

**function(实参)**

**定义一个函数**

**function getSum () {**

**var sum = 0;**

**for(var i = 0;i<=100;i++) {**

**sum += i;**

**}**

**console.log(sum);**

**}**

**//调用函数**

**getSum();**

**函数的定义**

**fanction funName(参数1，参数2，参数3) {**

**//函数体**

**}**

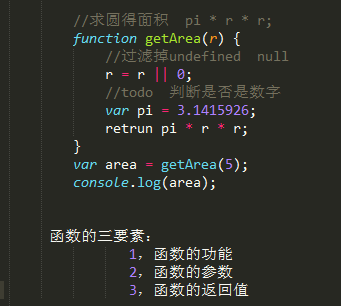
**//函数的调用**

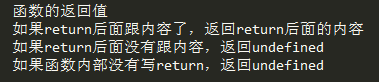
**funName(1,2,3);**

**//求n-m之间所有数的和**

**n,m 形式参数(形参)**

**求圆得面积**

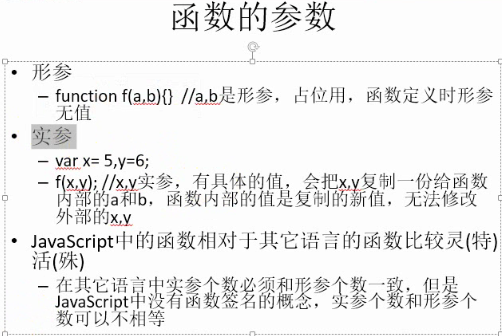






函数内部可以调用其他函数

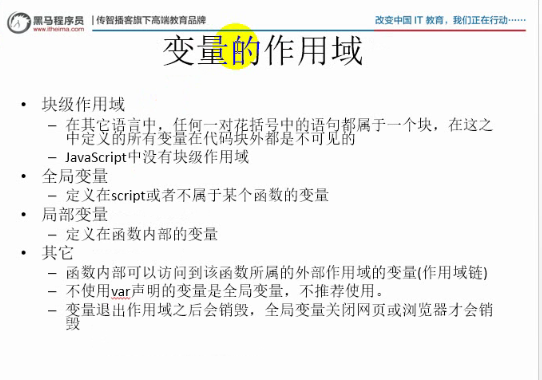
1. 执行的过程
2. 函数执行内部调用的函数的时候，会执行内部的函数，当内部的函数执行完毕，会继续执行当前的函数
3. 设置断点



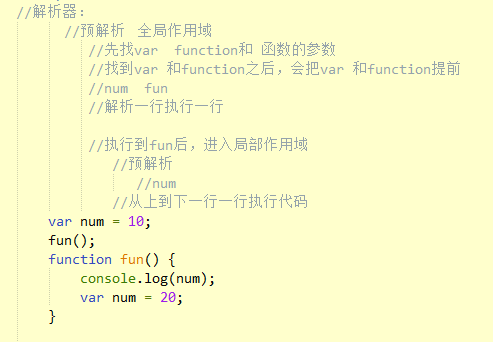
**Javascript中没有重载的概念**

下面的相同名称函数会把上面的相同名称函数覆盖

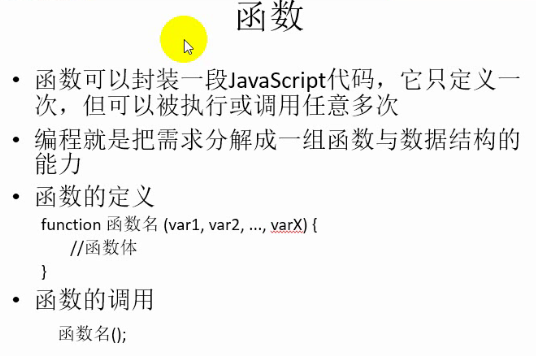












# 递归：

1. 自己调用自己

**函数是一种数据类型：**

1. **函数的类型是function**
2. **函数可以作为另一个函数的参数**

# 对象：

