**JavaScript 基础语法**

# 授课内容

授课科目：Web开发二；

授课内容：JavaScript 基础语法；

授课类型：讲授；

授课时间：4学时；

主讲教师：。

# 教学目标要求

## 能力目标：

1. 掌握 JavaScript 基础语法；
2. 掌握在 JavaScript 中定义变量的方法；
3. 掌握 JavaScript 中运算符和常用流程控制结构的使用；
4. 了解 JavaScript 编码规范；
5. 掌握 JavaScript 调试工具 Firebug 的使用。

## 知识目标：

1. JavaScript 定义变量；
2. JavaScript 内置数据类型及类型转换的相关知识；
3. JavaScript 中运算符和常用流程控制结构的使用。

# 课件分析

概 述：本章是Web开发二的第二章，主要介绍JavaScript 基础语法、JavaScript 变量及内置数据类型、JavaScript 流程控制结构、基本的 JavaScript 编码规范。

教学重点：JavaScript 基本语法、变量、运算符、数据类型的转换、流程控制结构的使用。

教学难点：JavaScript 基本语法、变量、运算符、流程控制结构的使用。

# 教学方法

问题教学法、讲授法、类比法、讨论法等。

# 教学过程

## 课前补充

（0分钟）

## 上节回顾

（0分钟）

## 作业点评

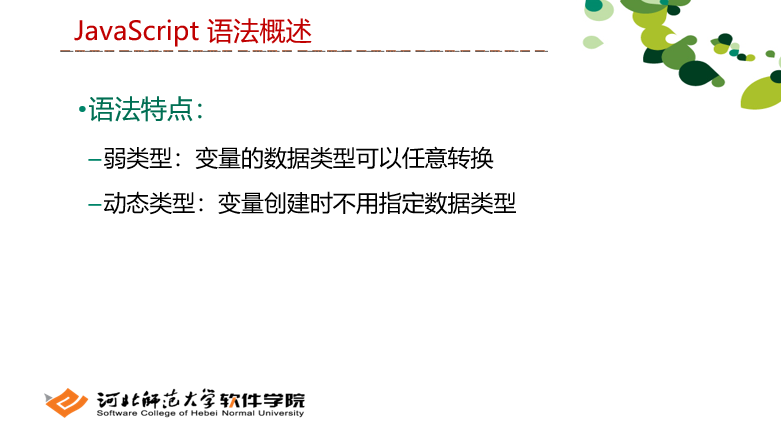
（0分钟）

## 导入新课

（2分钟）



在学习完JavaScript的概述以及基本用法之后，从第二章开始学习JavaScript 基础语法的相关知识。



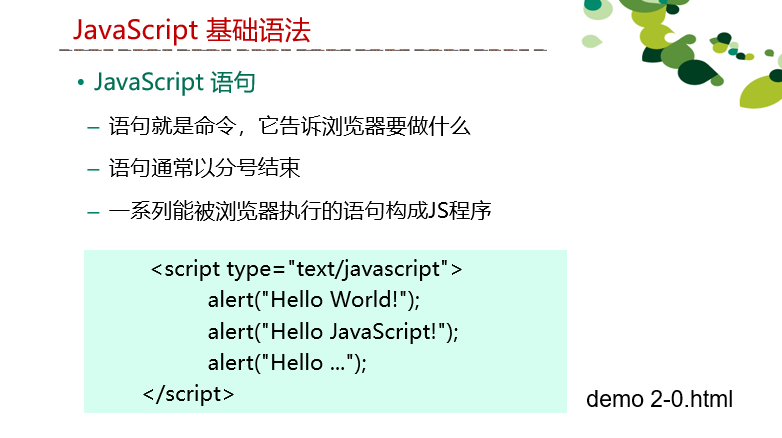
JavaScript是一门脚本语言。脚本语言是由传统编程语言简化而来的语言，因此有很多相似之处。构成相同、语法结构类似、编程工具不同、编程过程不同、运行方式不同。脚本语言有自己的语法特点，弱类型、动态类型。

## 讲授新课

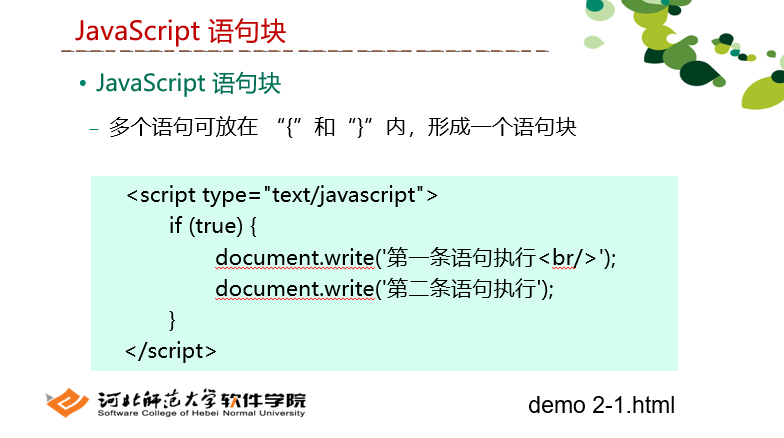
（170分钟）



第一小节学习JavaScript 基础语法知识。



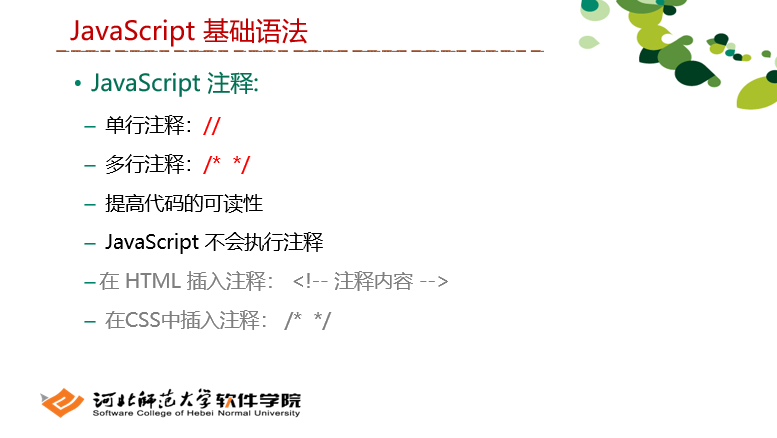
JavaScript 语句就是命令，它告诉浏览器要做什么。语句通常以分号结束，但结束分号不是必须的。如果语句的结尾没有分号，JavaScript会自动将该行代码的结尾作为语句的结尾。



一系列能被浏览器执行的语句构成JavaScript程序，多个语句可放在 “{”和“}”内，形成一个语句块。在2-1的demo中，document是JavaScript中的一个重要对象，该对象表示整个HTML文档，可以通过该对象来访问HTML文档中的所有元素。Write（）是document对象的一个方法，该方法的作用是在网页中输出一行字符串。



JavaScript 输出内容的 3 种方式:① document.write( ) 作用是在页面输出内容；②console.log( ) 作用是在控制台输出内容；③alert( ) 作用是以弹出框的形式输出内容。

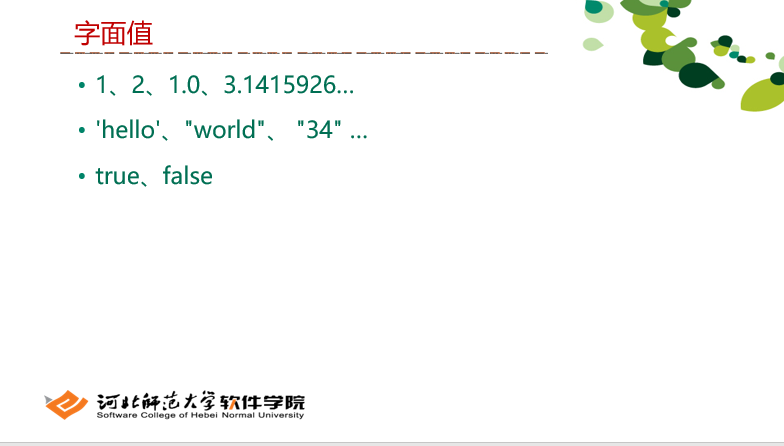


JavaScript 注释的两种写法：①单行注释：// ②多行注释：/\* \*/

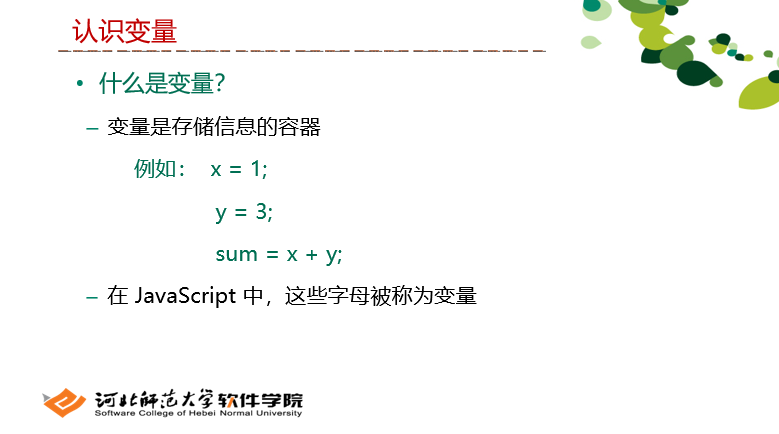
JavaScript 注释语句不会执行，可用来提高代码的可读性。应养成好的编码习惯，写注释。



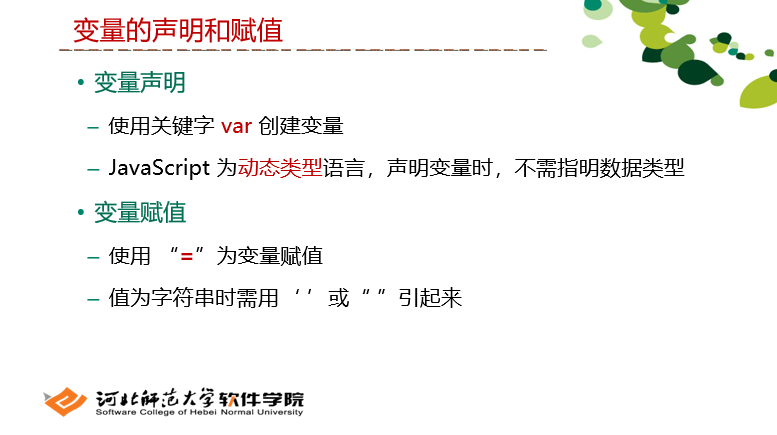
第二小节学习JavaScript 变量及原始数据类型。



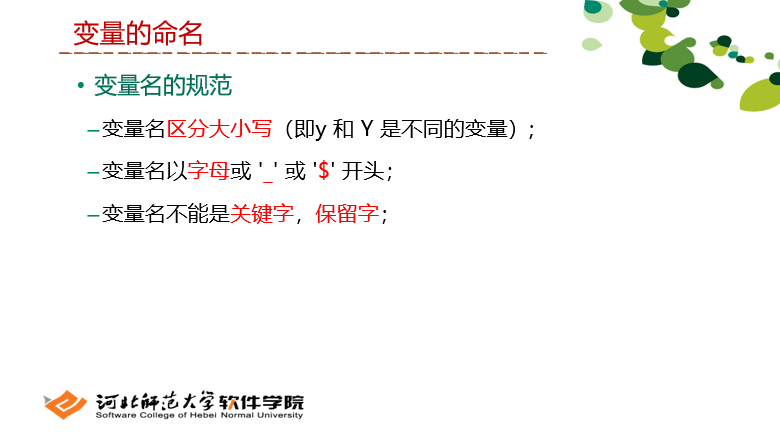
JavaScript字面值表示某种特定类型的一个值，是常量。比如 123是一个number的字面值常量， “abc”是一个字符串的字面值常量。



在JavaScript中，和常量相对的有变量。JavaScript 变量是存储数据值的容器。在编程中，类似代数，我们使用变量来存放值，在表达式中也可以使用变量。



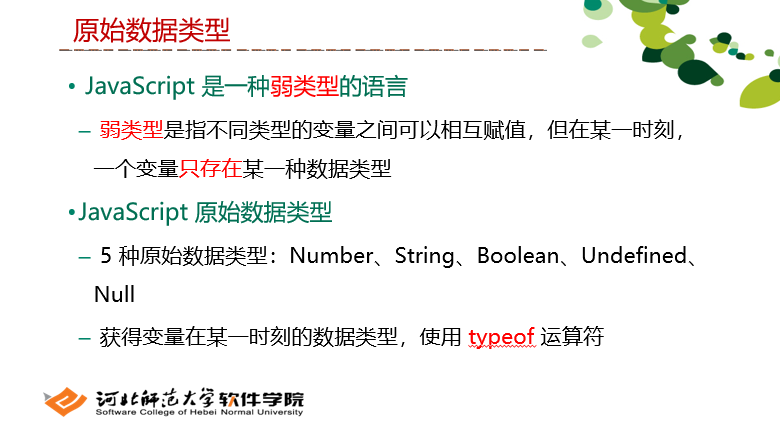
要使用变量，就必须首先声明变量。在JavaScript中，可以使用关键字var声明变量，而不必显示地声明变量的数据类型，JavaScript会根据需要自动的进行数据类型的转换。变量的赋值使用 “=”。



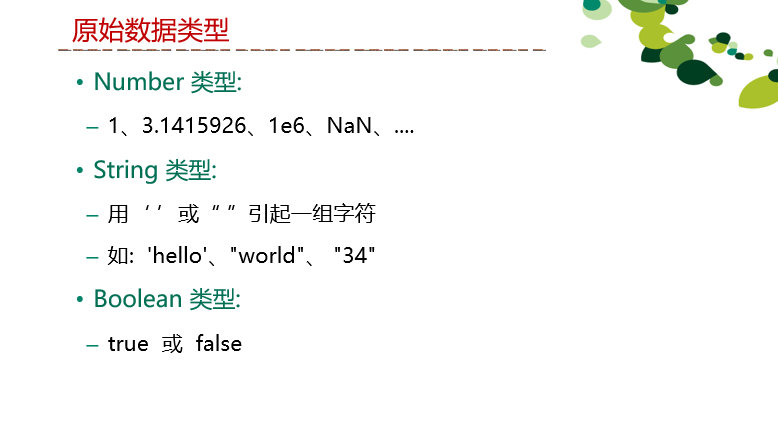
在声明变量时需要遵守变量命名规范，注意JavaScript变量是区分大小写的。还有一点需要注意：变量名不能是一个关键字、保留字或逻辑常量。



JavaScript中的关键字。



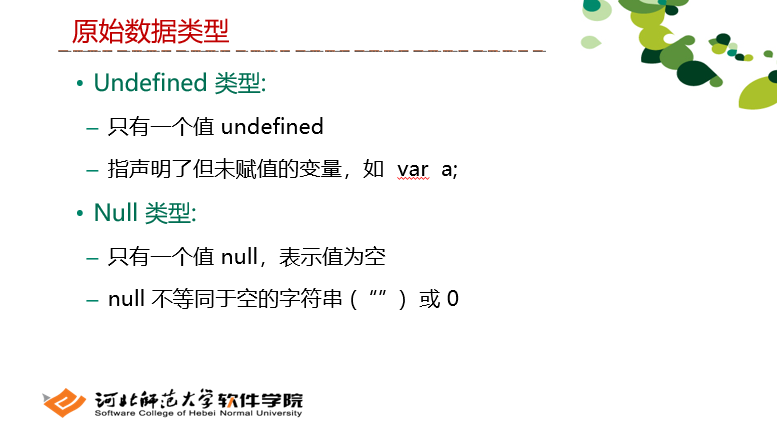
数据类型就是将各种数据加以分类，每一个数据或变量都属于其中确定的一类。JavaScript 是一种弱类型的语言，弱类型意味着用户不必显示地声明变量的数据类型，JavaScript将根据需要自动进行数据类型转换。JavaScript 的5种原始数据类型为：Number、String、Boolean、Undefined、Null。但在某一时刻，一个变量只存在某一种数据类型，可以使用 typeof 运算符来获取。



Number 类型: JavaScript 只有一种数值类型。写数值时用不用小数点均可。

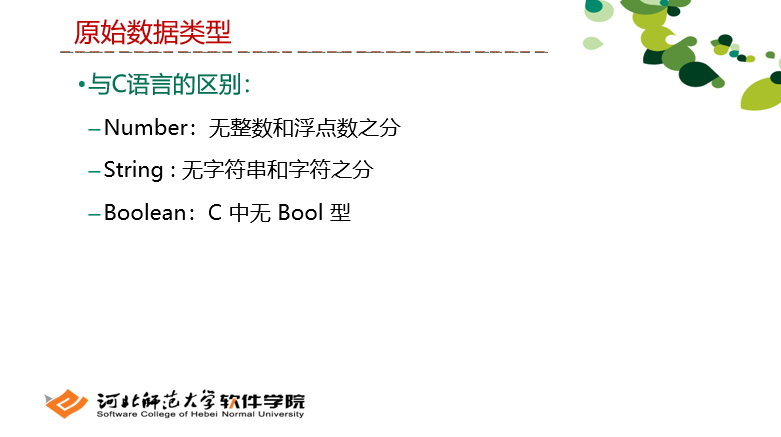
String 类型: 字符串是一串字符（比如 "Bill Gates"）。字符串被引号包围，使用单引号或双引号。

Boolean 类型: 布尔值只有两个值：true 或 false。

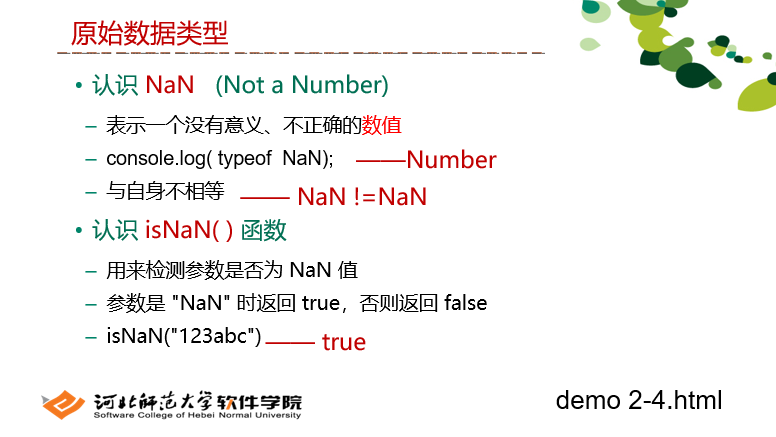


Undefined 类型: 在 JavaScript 中，没有值的变量，其值是 undefined。typeof 也返回 undefined。任何变量均可通过设置值为 undefined 进行清空。其类型也将是 undefined。

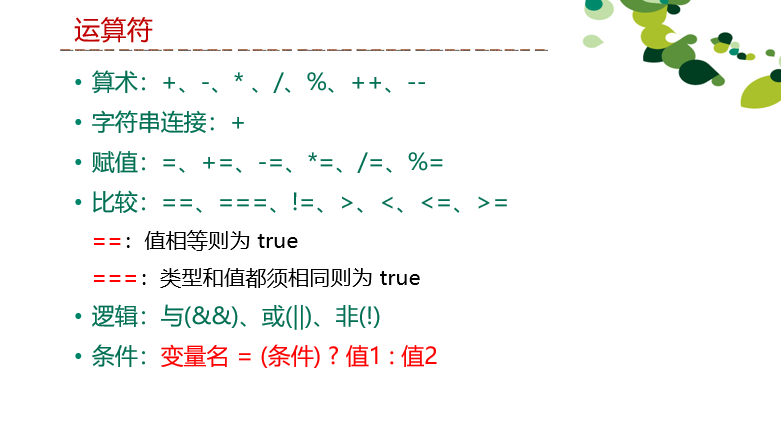
Null 类型: 在 JavaScript 中，null 是 "nothing"，被看做不存在的事物。但是在 JavaScript 中，null 的数据类型是对象，可以将此理解为一个 bug，它本应是 null。可以通过设置值为 null 清空对象



JavaScript原始数据类型与C语言的区别：① Number：无整数和浮点数之分；② String : 无字符串和字符之分；③ Boolean：C 中无布尔型。



NaN表示一个没有意义、不正确的数值，方法 parseInt() 和 parseFloat() 在不能解析指定的字符串时就返回这个值。使用 isNaN() 来判断一个值是否是数字，原因是 NaN 与所有值都不相等，包括它自己。当参数是 "NaN" 时返回 true，否则返回 false。



定义好变量后，需要对它们进行赋值、改变和执行计算，这些都由运算符来完成。

算数运算符：加、减、乘、除、取余、累加、递减。

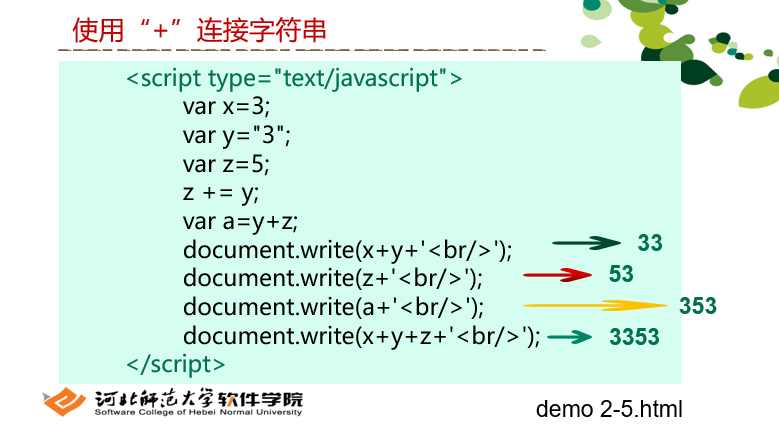
字符串运算符：将两个字符串连接成一个新字符串。该运算符要求两个操作数的数据类型都是字符串型。

赋值运算符：=将右边的值赋到左边；+=将左边的值加右边的值，并将结果赋给左边；\*=将左边的值乘以右边的值，并将结果值赋给左边。

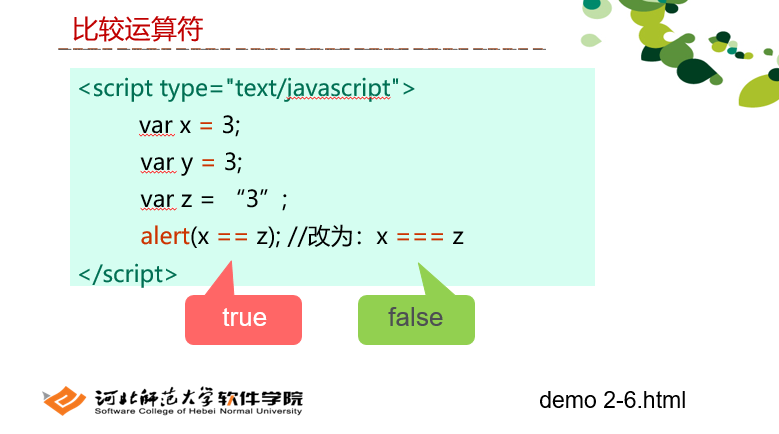
比较运算符：将两个操作数进行比较，返回一个布尔值来说明比较的结果。

逻辑运算符：逻辑与，两个条件都为真，结果才为真；逻辑或，两个条件中只要有一个为真，结果即为真；逻辑非，将结果取反，原条件为真返回假，原条件为假返回真。

条件运算符：JavaScript中唯一一个三元运算符。执行方式与if…else相似。

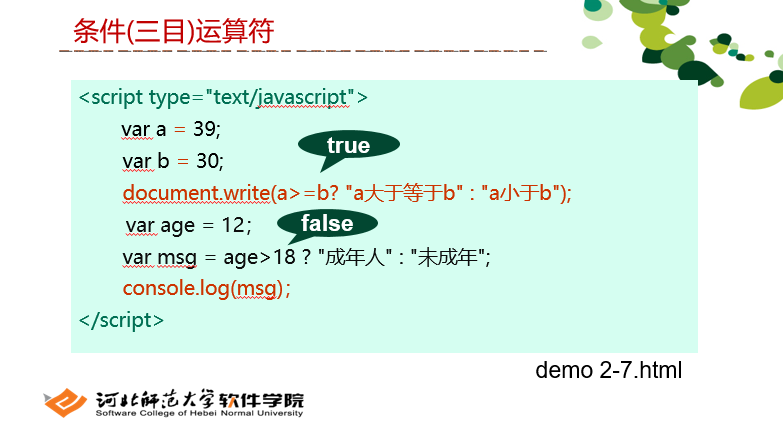


x+y=33 z=53 x+y+z=3353。均涉及到字符串拼接运算。

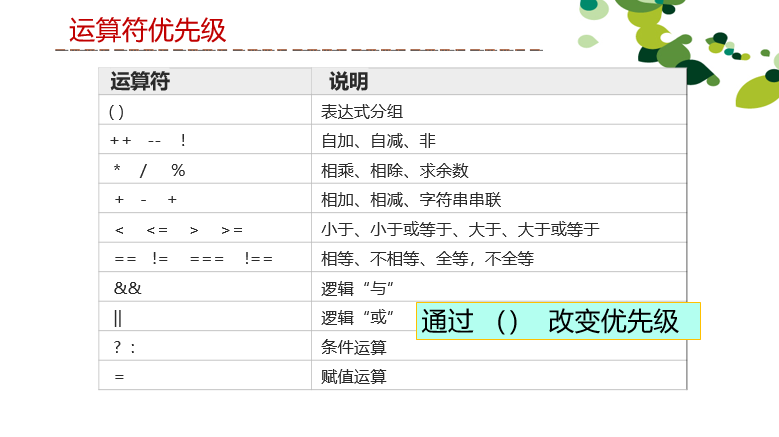


alert(x == z) 输出true。

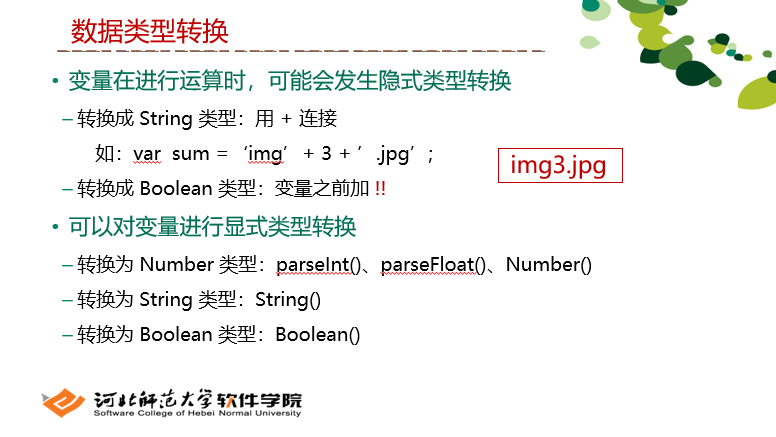
alert(x === z) 输出false。因为===要求类型和值都相等才为true。



条件（三元）运算符条件后面会跟一个问号（?），如果条件为 true，则问号后面的表达式A将会执行；表达式A后面跟着一个冒号（:），如果条件为 false，则冒号后面的表达式B将会执行。

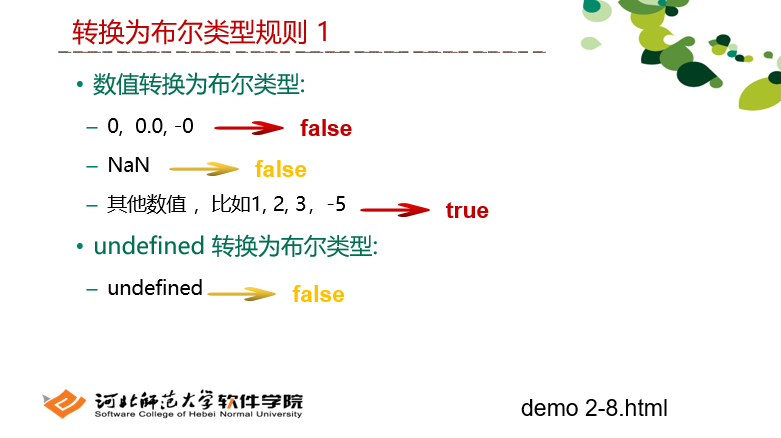


运算符按照优先级的不同从高到低排列。通过（）可改变优先级。



JavaScript 中变量可以赋予任何类型的值。但是运算符对数据类型是有要求的，如果运算符发现运算子的类型与预期不符，就会自动转换类型。数据类型除了可以自动转换以外，还可以手动强制转换。

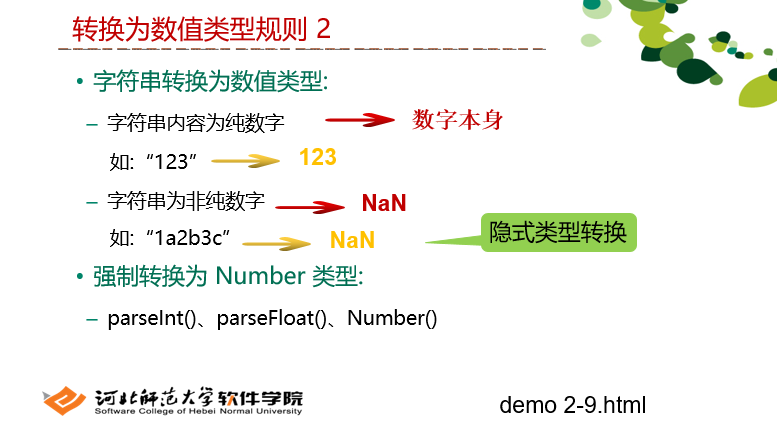
例如：num + ""，当 + 两边一个操作符是字符串类型，一个操作符是其它类型的时候，会先把其它类型转换成字符串再进行字符串拼接，返回字符串，即隐式类型转换。



使用Boolean(value)方法可以强制转换任意值为boolean类型，除了以下六个值，其他都是自动转为true：undefined、null、-0、+0、NaN、“”（空字符串）。



隐式转换为数值类型最简单的方法就是 \*1。除此之外，需要记住几个特殊的值转换为数值类型的规则。

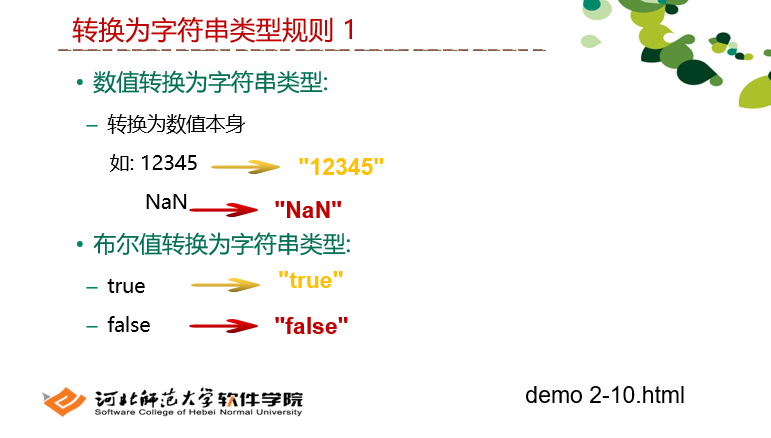


强制转换为 Number 类型涉及三个函数：parseInt()、parseFloat()、Number()。

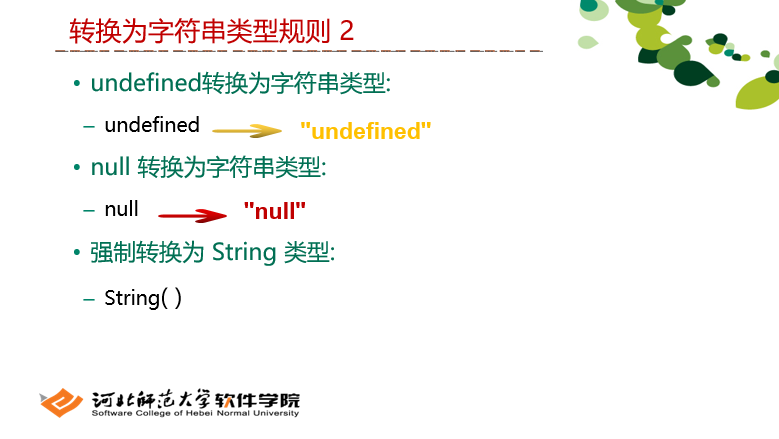
Number()：如果是Boolean，true转化为1,false转化为0；如果是null，返回0；如果是undefined，返回NaN；如果是字符串；若只包含数字（包括正号负号的情况），则转化为十进制数值；同理可以转化浮点数和十六进制数值；空字符串转化为0；其它则转化为NaN。

parseInt()：处理字符串时会忽略字符串前面的空格，若遇到第一个字符不是数字或符号，则返回NaN；如果是数字，则会解析直到所有字符串或遇到非数字字符；第二个参数表示转换的进制数。

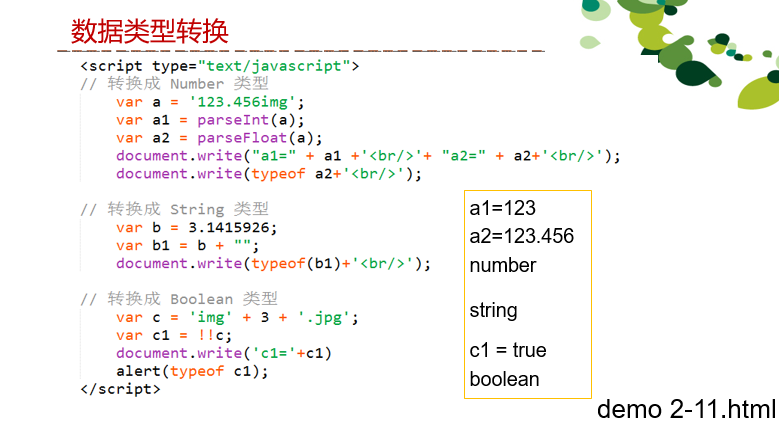
parseFloat()：同理，但是第一个小数点是有效的。两者区别是，parseFloat()会忽略前导的0，只能解析十进制浮点数。



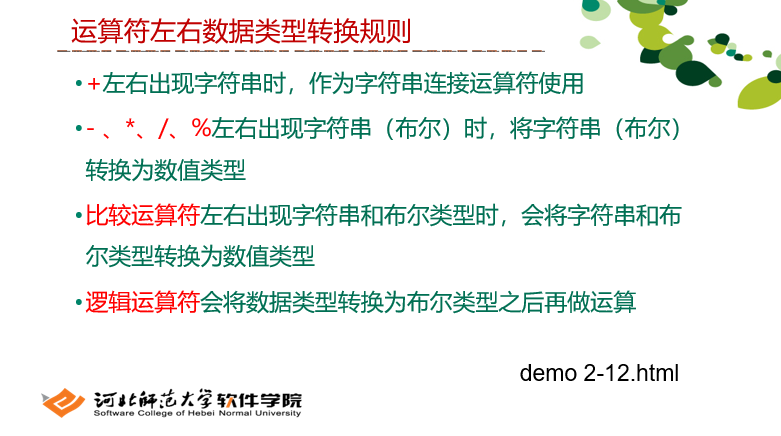
隐式转换为字符串类型的方法就是加引号“”。



强制转换为 String 类型使用String( )函数。在不知道变量是否为null或者undefined时使用String()函数来转换为字符串类型，其可以将任何类型的数值转换为字符串。如果转换值是null，则返回"null"，如果转换值是undefined，则返回"undefined"。



数据类型转换实例，包括隐式类型转换和强制类型转换。a1=123，a2=123.456，a2的数据类型为number。b1的数据类型为string。c1= true，其数据类型为boolean。

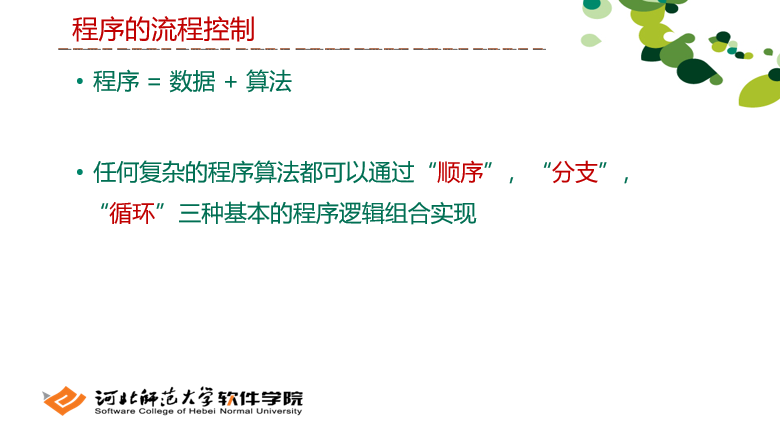


运算符左右数据类型转换规则总结。

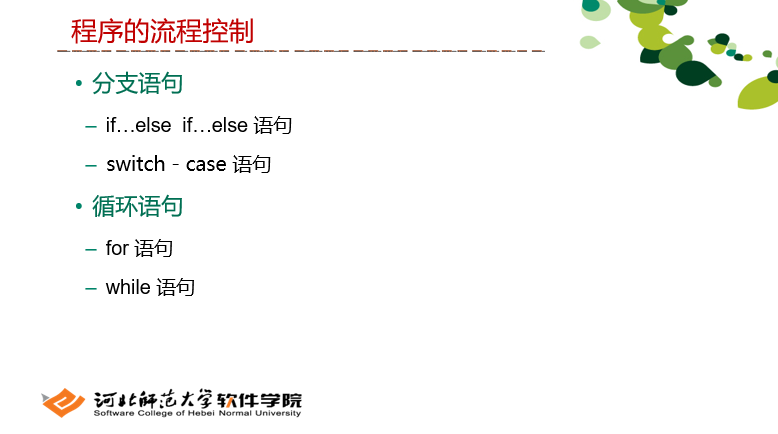
① +左右出现字符串时，作为字符串连接运算符使用；② - 、\*、/、%左右出现字符串或布尔类型时，将字符串、布尔类型转换为数值类型；③ 比较运算符左右出现字符串和布尔类型时，会将字符串和布尔类型转换为数值类型；④ 逻辑运算符会将数据类型转换为布尔类型之后再做运算。



最后，学习JavaScript 流程控制结构。



任何复杂的程序算法都可以通过“顺序”，“分支”，“循环”三种基本的程序逻辑组合实现，JavaScript也不例外。



在写代码时，经常会需要基于不同判断执行不同的动作。可以在代码中使用条件语句来实现这一点。

在 JavaScript 中，可使用如下条件语句：

if 语句，来规定假如条件为 true 时被执行的 JavaScript 代码块；

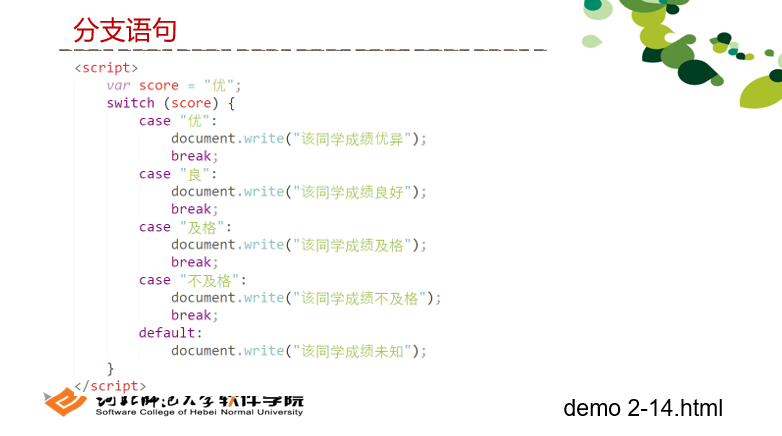
else 语句，来规定假如条件为 false 时的代码块；

else if 语句，来规定当首个条件为 false 时的新条件。

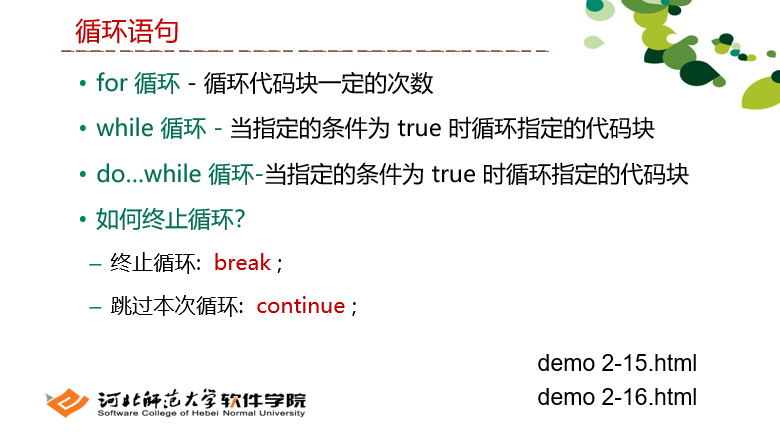
假如需要运行代码多次，且每次使用不同的值，那么循环（loop）相当方便使用。



如果满足条件，不会执行下面任何分支；如果以上都不满足，则输出不及格。



switch 语句用于基于不同条件执行不同动作。使用 switch 语句来选择多个需被执行的代码块之一。计算一次 switch 表达式，把表达式的值与每个 case 的值进行对比，如果存在匹配，则执行关联代码。

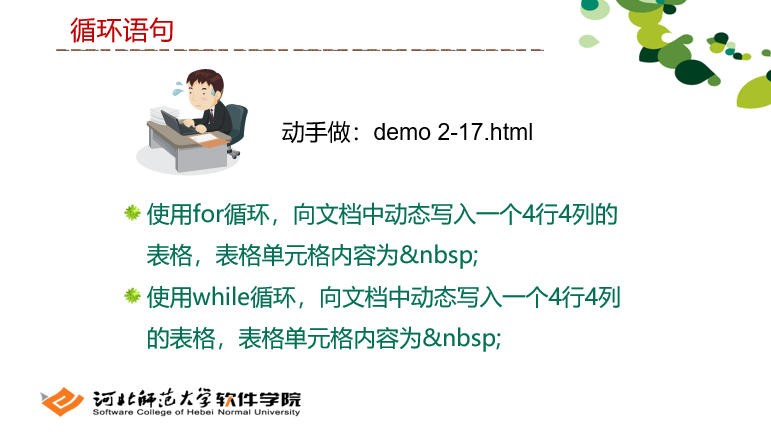


JavaScript 支持不同类型的循环：

for ，多次遍历代码块；

while ， 当指定条件为 true 时循环一段代码块，while 循环会一直循环代码块，只要指定的条件为 true；

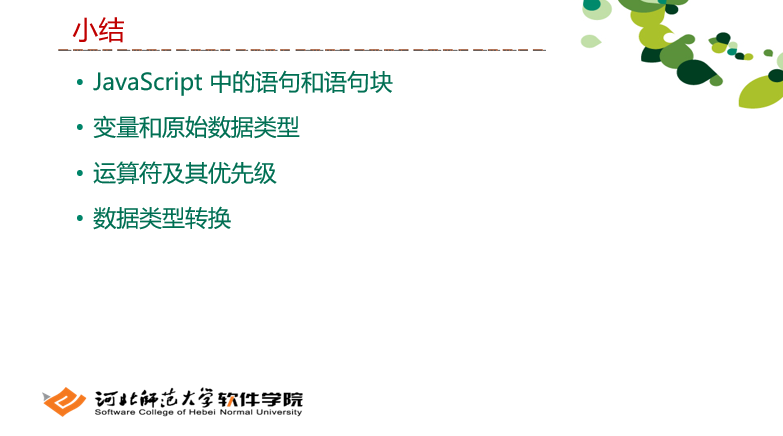
do/while ，当指定条件为 true 时循环一段代码块。do/while 循环是 while 循环的变体。在检查条件是否为真之前，这种循环会执行一次代码块，然后只要条件为真就会重复循环。



接下来，我们来做一个练习，分别使用for循环和while循环，向文档中动态写入一个4行4列的表格。

## 课时小结

（7分钟）



最后我们来进行一下总结。JavaScript 语句就是命令，它告诉浏览器要做什么。语句通常以分号结束，但结束分号不是必须的。一系列能被浏览器执行的语句构成JavaScript程序，多个语句可放在 “{”和“}”内，形成一个语句块。

在JavaScript中，可以使用关键字var声明变量，而不必显示地声明变量的数据类型。JavaScript 的5种原始数据类型为：Number、String、Boolean、Undefined、Null。但在某一时刻，一个变量只存在某一种数据类型，可以使用 typeof 运算符来获取。

定义好变量后，需要对它们进行赋值、改变和执行计算，这些都由运算符来完成。运算符有优先级的区分。通过（）可改变优先级。

JavaScript 中变量可以赋予任何类型的值，但是运算符对数据类型是有要求的。可以进行隐式类型转换和强制类型转换。

## 布置作业

（1分钟）

1. 完成本次课所对应的雪梨作业。