## 01.JavaScript发展历程及语言特点

1、W3C（客户端）前端JavaScript构成：ECMAScript、DOM、BOM

ECMAScript（ES）是JavaScript的语法标准，是核心语言；DOM、BOM是宿主对象。

JavaScript 和 DOM 并不是不可分割的，它们的语言标准相互独立。

DOM 对 JavaScript 来说，是宿主对象，是语言中可更换的部分

ECMAScript 对 JavaScript 来说，是核心语言，是不可被替代的功能

服务端JavaScript ——Node.js

ECMAScript、fs、TCP、Stream、Buffer

2、ES5 —— 2009年12月发布

ES6——2015年6月份发布

以后每年更新一个版本，到目前为止，是ES10（2019年06月发布）

3、浏览器是客户端JavaScript运行的宿主环境。

Chrome：JavaScript实现方式（JS引擎）----v8

4、JavaScript语言特点

直译式脚本语言

在宿主（浏览器、Node）中解释执行

非编译语言，不是在执行前提前编译可执行文件或字节码

弱类型、动态类型语言

写程序时不用给变量指定特定的数据类型（动态类型）

可以动态的更改变量的类型（弱类型）

JavaScript语言特点

ES5没有块作用域

基于原型链的继承方式

5、JavaScript 只能够在浏览器中执行？

不是，JavaScript 除了在浏览器中运行，还可以在其他的运行环境中运行，如 node.js 环境。目前 JavaScript 的运行环境有浏览器和node.js 环境两种。

6、JavaScript 在浏览器中是如何运行的？

浏览器下载 JavaScript 脚本文件后，由浏览器 JavaScript 引擎解释执行。

## 02.数据类型以及存储和转换

1、ES5 数据类型（6种）及其划分（2类）

基本（原始）类型（Number、String、Boolean、Null、Undefined）

引用（对象）类型（Object（Array、Function、Date等））

2、判定数据类型可使用 typeof 操作符

返回一个字符串，表示未经计算的操作数的类型。

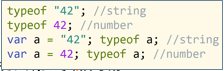
3、数据、变量和类型

JavaScript 中的变量没有类型

JavaScript 中的数据值有类型

变量可以随时持有任何类型的值

在对变量执行 typeof 操作时，得到的结果并不是该变量的类型，而是该变量持有的值的类型。



4、变量与内存

变量声明：使用方便的标识符，用于引用计算机内存地址；

变量声明指向一块内存空间，用于保存数据

变量赋值：向变量指向的内存空间中存放数据

一般来说，系统会划分出两种不同的内存空间：栈内存（stack）、堆内存（heap）

栈内存

存储的值大小固定

由系统自动分配内存空间

空间小，运行效率高

堆内存

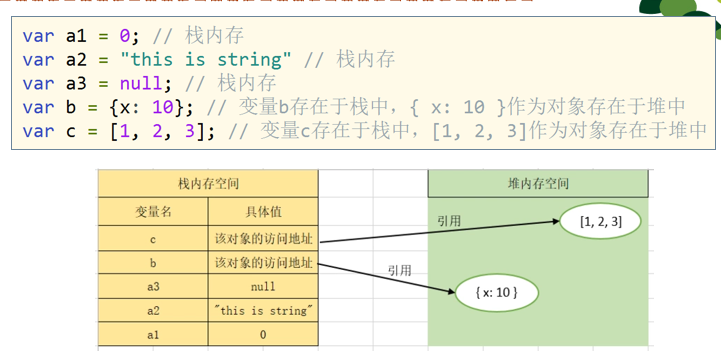
存储的值大小不定，可动态调整

由程序员通过代码进行分配

空间大，运行效率相对较低

基本类型的变量是存放在栈区的，基本类型的值是不改变的。

引用类型的值是同时保存在栈内存和堆内存中的对象，引用类型的值可以改变。



基本类型与引用类型的区别

访问机制不同

复制变量不同

比较变量不同

参数传递不同

访问机制

基本类型的值直接访问

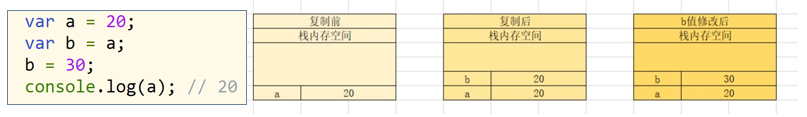
引用类型的值通过引用访问，不能直接访问

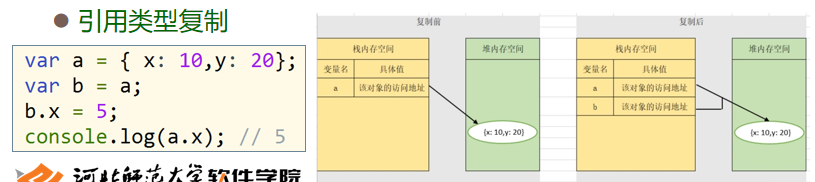
首先，从栈中获取该对象的地址引用

其次，再从堆内存中取得我们需要的数据

复制变量

基本类型复制 —— 相互独立互不影响





比较变量

基本数据类型与引用类型的比较

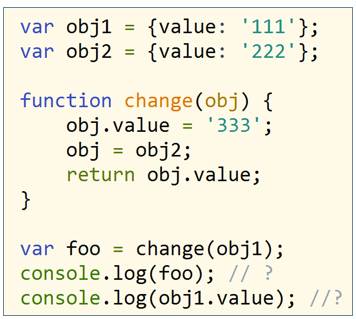
值类型是判断变量的值是否相等（值比较）

引用类型是判断所指向的内存空间（地址）是否相同（引用比较）

参数传递

ECMAScript 中所有函数的参数都是按值来传递的

基本类型值：把变量里的数据值传递给参数，之后参数和变量互不影响。

 引用类型值：把对象的引用（地址）值传递给参数，参数和对象都指向同一个对象，相互影响。

222 333

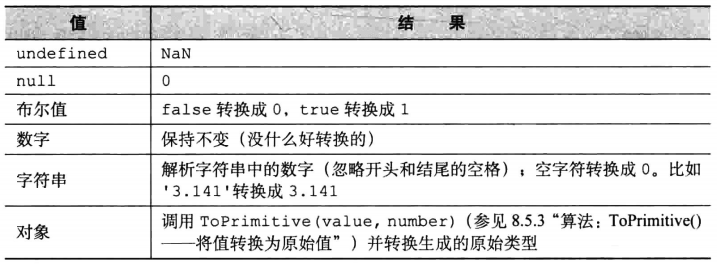
数据类型转换

类型转换 —— 将值从一种类型转换为另一种类型

隐式类型转换：通常是某些操作的副作用，不易看出

显示类型转换：可以在代码中明显看出

转换为 Number 类型规则



强制转换为 Number 类型

parseInt()、parseFloat()、Number()

认识 NaN (Not a Number)

表示一个没有意义、不正确的数值

console.log( typeof NaN);

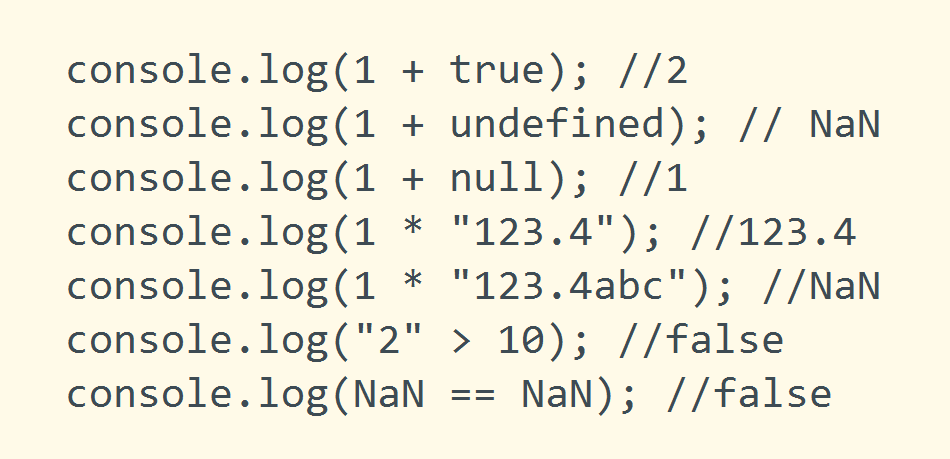
与自身不相等

认识 isNaN( ) 函数

用来检测参数是否为 NaN 值

参数是 "NaN" 时返回 true，否则返回 false

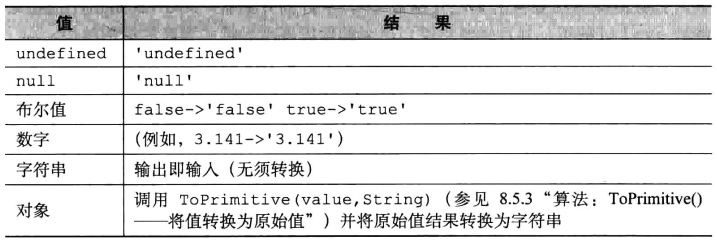
isNaN("123abc")——true





解析：a ：undefined，条件语句中undefined == undefined,所以运行 a\*1 语句，所以结果为NaN。

转换为 String 类型规则



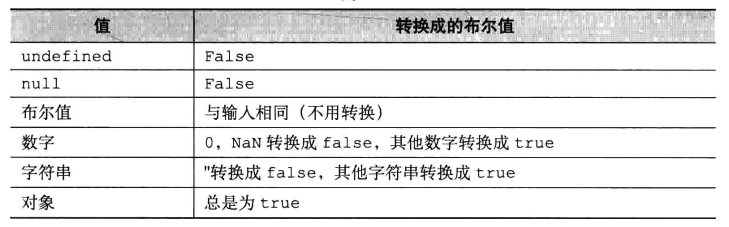
强制转换为 String 类型

String()

1. “+”运算符左右两侧有字符串时为拼接运算符。

2. 运算符等级相同时，从左往右计算。

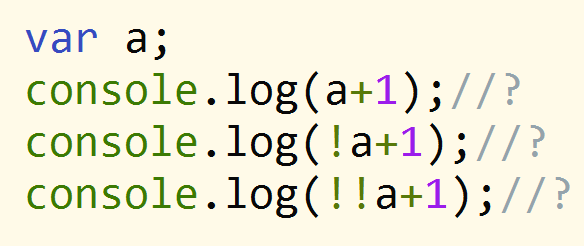
转换为 Boolean 类型规则



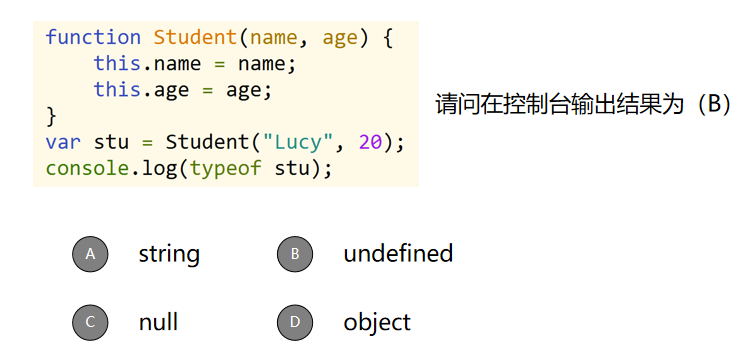
强制转换为 Boolean 类型

Boolean()

逻辑运算符会将数据类型转换为布尔类型之后再做运算



解析：a是个未被定义的变量，undefined，转换成布尔值为False，再转成数值为0,答案是1、2、1



函数没有return语句时，返回值为undefined，typeof undefined 为undefined（字符串类型）。