JavaScript

## 前言

### 编程语言和标记语言的区别

**编程语言**有很强的逻辑和行为能力在编程语言中有很多具有逻辑性和行为能力的指令，这是主动的

**标记语言（HTML）**不用于向计算机发出指令，常用于格式化和链接，标记语言的存在是用来被读取的，是被动的

### 计算机基础

#### 数据存储单位

位（bit）：1bit可以保存一个0或一个1（最小的存储单位）

字节（Byte）：1B=8b

1KB=1024B 1MB=1024KB 1GB=1024MB 1TB=1024GB

#### 浏览器执行js简介

浏览器分为两部分，渲染引擎和js引擎

渲染引擎：用来解析HTML和css，俗称内核，如Chrome的blink，老版本的webkit

Js引擎：也叫js解释器，用来读取网页中的js代码，对其处理后运行，如Chrome的v8

浏览器本身并不会执行js代码，而是通过内置的js引擎来执行，js引擎执行代码时逐行解释每一句源码（转换为机器语言）然后由计算机执行

### JavaScript

JavaScript是世界上最流行的语言之一，是**运行在客户端的脚本语言**

脚本语言：不需要编译，运行过程中由js解释器（js引擎）逐行解释并执行

注释：

单行注释：“//” 快捷键：Ctrl+’/’

多行注释：“/\*\*/” 默认快捷键：Ctrl+alt+A

## 变量

### 声明变量

Var 变量名;

Var 变量名 = 值;

Var 变量名 = 值, 变量名 = 值, 变量名 = 值;

特殊情况：

1. 只声明不赋值，console.log(变量名)结果是undefined
2. 不声明不赋值直接用会报错
3. **不声明直接赋值可以使用，会成为全局变量**

### 变量命名规范

1、由字母（A-Za-z）、数字（0-9）下划线（\_）美元符号（$）组成

2、严格区分大小写

3、不能以数字开头

4、不能是关键字，保留字

5、变量名必须有意义

6、遵守驼峰命名法，首字母小写，后面的单词的首字母大写

7、推荐翻译网站：有道，爱词霸

**注：尽量不要使用‘name’作为变量名**

## 数据类型

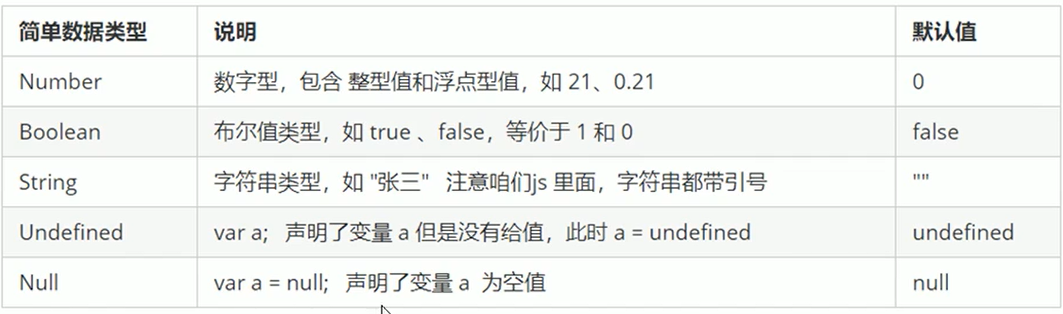
### 变量的数据类型

变量是用来存储值的所在处，他们有名字和数据类型，变量的数据类型决定了如何将代表这些值的位存储到计算机的内存中，JavaScript是一种弱类型或者说动态语言。这意味着不用提前声明变量的类型，在程序运行过程中，类型会被自动确定。

在代码运行时，变量的数据类型是由js引擎根据=右面变量值的数据类型确定的。

**Js是动态语言，var x =10后，可以再将x赋值为其他类型**

### 数据类型之简单数据类型



#### 数字型

进制：Js中八进制前加0，十六进制前加0x

数字型的最大值：Number.MAX\_VALUE，最小值同理

三个特殊值：

Infinity，代表无穷大，大于任何数值

-Infinity，代表无穷小，小于任何数值

NaN，not a number，代表一个非数值

isNaN()方法：用来判断非数字并且返回一个值，若是数字，返回的是false，如果不是数字，返回的是true。

#### 字符串型

字符串型可以使引号中的任意文本其语法为双引号””和单引号’’

因为HTML标签里属性使用的是双引号，js这里更推荐使用单引号

**字符串转义**



length属性：字符串长度。字符串名.length

字符串拼接“+”：字符串+任何类型=拼接后的新字符串

#### Undefined和Null

Undefined+字符串=字符串

Undefined+数字=“NaN”

Null+1=1

## 获取变量类型

typeof 变量名

### 字面量

字面量是在源代码中一个固定值的表示法，通俗来说就是字面量表示如何表达这个值。

数字字面量：7,8,2

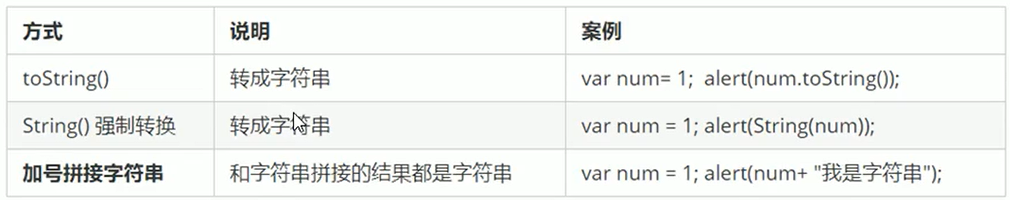
字符串字面量：“黑马”，‘前端’

布尔字面量：true，false

## 数据类型转换

一种数据类型的变量转换成另外一种

### 转换为字符串类型



toString用法：变量.toString

String用法：String(变量名)

### 转换为数字型（重点）



parseInt得到的是整数，只把整数部分提取出来，而且会把单位去掉如120px会变成120（数字必须在开头位置）

‘123’-‘120’结果是3

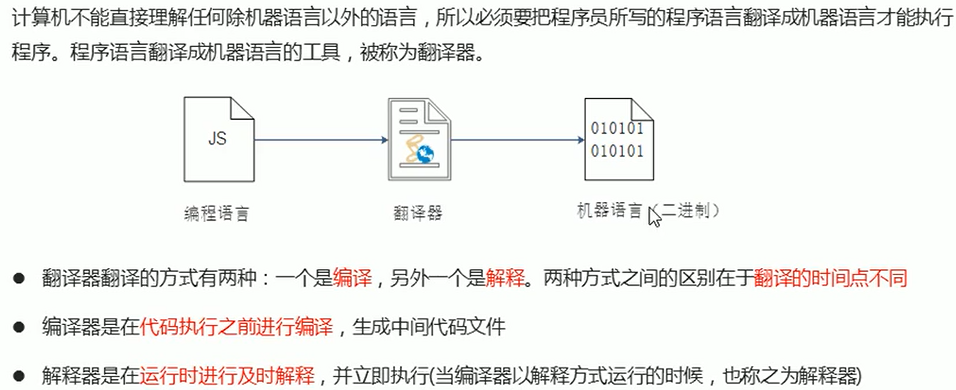
### 转换为布尔型

Boolean()函数

代表空、否定的值会被转换为false，如’’,0,NaN,null,undefined

其余值都会转换为true

## 解释型语言和编译型语言



## 运算符



==默认转换数据类型，会把字符串型转换为数字型，只要值相等就会为true，如37==’37’为true



### 逻辑运算符

#### 短路运算（逻辑中断）

短路运算原理：当有多个表达式（值）时，左边的表达式值可以确定结果时就不再继续运算右边的表达式值。

1. 逻辑与

语法：表达式1 && 表达式2

若第一个表达式值为真，则返回表达式2；若第一个表达式值为假，则返回表达式1

1. 逻辑或（重要，会影响程序运行结果）

若第一个表达式值为真，则返回表达式1；若第一个表达式值为假，则返回表达式2

## 运算符优先级



一元运算符里的逻辑非优先级很高

逻辑与比逻辑或优先级高

## 流程控制（顺序，分支，循环）

### 分支流程控制

#### If语句

If(条件表达式){

//执行语句1

}else{

//执行语句2}

#### 三元表达式

语法结构：条件表达式？表达式1：表达式2

若条件表达式为真，返回表达式1，若为假返回表达式2

#### Switch语句

Switch(表达式){

case value1:

执行语句1;

Break;

case value2:

执行语句2;

Break;

。。。

default:

执行最后的语句;

}

表达式值和case后的选项值匹配，如果匹配得上则执行该case里的语句。

**注：**

1. **开发时，表达式常写为变量**
2. **Switch括号里的变量和case里的值匹配时候是全等，即必须是值和数据类型一致才可以匹配。**
3. **Break如果当前的case里面没有break，则不会退出switch，是继续执行下一个case**

#### If语句和switch的区别

Switch用来比较固定值，if语句用来判断范围；

switch语句进行条件判断后直接执行到程序的条件语句，效率更高。而if…else有几种条件就要判断多少次；

分支比较少时，if语句的效率比switch高；分支多时，switch效率高，结构清晰

## 1

### 2

#### 3

##### 4