## JS引用

1. Script标签一旦用于引入外部文件，就不能再编写代码了。需再创建一个新的script标签用于编写。
2. 可放在a标签和事件中



## 变量

1. 在js中，需要创建一个变量，需要var 声明

var a;

1. 为变量赋值

a=123;

或 var b=456;

## 标识符

在js中所有由我们自主命名的都可以称为标识符

例如：变量名，函数名，属性名

命名规则：

1. 标识符中可以含有字母、数字、\_ 、$
2. 标识符不能以数字开头
3. 标识符不能使用关键字和保留字（如：if、var）
4. 标识符一般采用驼峰命名法

var a\_1\_$;

## string

1. 使用字符串时引号不能嵌套：一对单引号中不能出现单引号，双引号同理。若需使用，则要用\转义字符。如： str=”我说:\”今天天气真好\””;

\n表示换行

\t 制表符（tab）

\\ 表示 \

1. 可以使用typeof来检查一个变量的类型

语法：typeof 变量

1. JS中可以表示的数字最大值：Number.MAX\_VALUE。如果使用number表示的数字超过了最大值，则会返回Infinity,表示正无穷，Infinity也属number。
2. 若使用js进行浮点数运算，可能得到一个不精确的结果。如：c = 0.1+0.2
3. 强制类型转换：
4. a = string(a);
5. a = a.toString(); //在转换null和 undefined会报错

## number

1. 使用Number（）函数
2. 如果是纯数字的字符串，则直接转换为数字
3. 如果字符串中有非数字的内容，则转换为NaN
4. 如果字符串是一个空串或者是一个全是空格的字符串，则转换为0
5. 使用parseInt()、parsefloat()

 

## 其它进制的数字

1. 在js中，如果需要表示16进制的数字，则需要以0x开头(并不是所有浏览器都支持)

 

1. 如果需要表示8进制的数字，则需要以0开头



1. 如果需要表示2进制的数字，则需要以0b开头



## 转换为boolean

1. 数字：除了0和NaN，都是true



1. 字符串：除了空串，都是true
2. null 和undefined都是false

## 运算符

1. 任何值和NaN做运算都得NaN
2. 任何值（包括NaN）和字符串（包括空串）相加都会被转换为字符串



1. 任何值做- \* /运算时都会自动转换为Number
2. 快速转换为number



## 自增和自减

1. a++和++a都会立即使原变量的值加一。但a++的值等于原变量的值（即等于a），++a等于自增后的值（即a+1）
2. 同上，a--的值等于原变量的值（即等于a），--a等于自增后的值（即a-1）

## 逻辑运算符

1. && （与）

找false，从左至右运行，当存在false，后面不再运行



1. || （或）

找true，从左至右运行，当存在true，后面不再运行



1. true && true ，当两个值都为true，则返回后边的

false && false ，当两个值都为false，则返回前面的

（若第一个值为true，则必然返回第二个值；若第一个值为false，则必然返回第一个值）





1. true || true ，当两个值都为true，则直接返回前面的

false || false ，当两个值都为false，则直接返回后面的

（若第一个值为true，则直接返回第一个值；若第一个值为false，则必然返回第一个值）

## 关系运算符

1. 若两个字符串相比较，则只会取第一位进行字符编码比较





## Unicode编码

1. 在字符串中使用转义字符\u 可以输出Unicode编码



1. 在HTML中，使用&#可输出Unicode编码

（Unicode编码默认为16进制，在html中使用需先转换为10进制）



## 相等运算符

1. 当使用==来比较两个值时，如果两个值的类型不同，则会自动进行类型转换，将其转换为相同类型，然后再比较
2. NaN不和任何值相等
3. != 判断两个值是否不相等
4. === 判断两个值是否全等



1. !== 判断两个值是否不全等



## 条件运算符（三元运算符）

1. 语法：

条件表达式?语句1：语句2;

1. 执行流程：

首先对条件表达式进行求值

如果该值为true，则执行语句1，并返回执行结果

如果该值为flase，则执行语句2，并返回执行结果



1. 如果条件表达式为非布尔值，则会先将其转换为布尔值

## prompt（）

该函数可以弹出一个提示框，并带有一个文本框，可输入一段内容

## 条件表达式

1. 语法：

switch(条件表达式){

case 表达式1：

语句1 。。。

break;

case 表达式2：

语句2 。。。

break;

case 表达式3：

语句3 。。。

break;

default:

语句。。。

break;

}

