



JS代码需编写到script标签中  
 <script type="text/javascript">  
 ...  
 </script>

\* 控制浏览器弹出一个警告框 \*

```
alert("...");
```

\* 在页面中输出一个内容 \*

```
document.write("...");
```

向body中输出一个内容

\* 向控制台输出内容 \*

```
console.log("...");
```

\* 可以将js代码编写到标签的onclick属性中，当我们点击按钮，js代码会执行

```
<button onclick="alert('...');">...</button>
```

也可以将js代码写在超链接的href属性中

```
<a href="javascript:alert('...');"></a>
```

虽然可以写在属性中，但属于结构与行为耦合，不方便维护





Mo Tu We Th Fr Sa Su

Memo No.

Date

可以将js代码编写到外部js文件中,然后通过script标签引入,编写到外部文件中可以在不同的页面中同时引用,也可以利用浏览器的缓存机制。

`<script type="text/javascript" src="js/script.js">`  
... `</script>`

指明路径

script 标签一旦用于引入外部文件,就不能在编写代码了,即使编写了浏览器也会忽略。

## 基本语法

### 1. 注释

/\*

...

多行注释

\*/

// 单行注释

### 2.

1) js中严格区分大小写

2) js中每一语句以分号结尾

3) js中会忽略多空格和换行,可对代码格式化



扫描全能王 创建





Memo No.

Date

## 字面量和变量

字面量，都是一些不可改变的值

字面量可直接使用，但一般不直接使用

变量用来保存字面量

变量更加方便使用

并都通常用变量去保存字面量

在JS中使用var关键字来声明一个变量

```
var a;      a = 123;  var var b = 123;  
console.log(a);
```

## 标识符

一 在JS中所拥有的可以由我们自主命名的都可以称为标识符

- 变量名，函数名，属性名

- 命名一个标识符时需要遵守如下规则

1. 标识符中可以含有字母，数字，-，\_，\$

2. 不能以数字开头

3. 标识符不能是JS中的关键字或保留字

4. 标识符一般都用驼峰命名法

(首字母小写，每个单词开头字母大写，其余小写)



扫描全能王 创建

- ES 底层 保存标识符时实际上是采用

的 Unicode 编码,

所以理论上讲, 所有 utf-8 中会有  
内容都可以作为标识符, 如中文.

## 数据类型.

六种

- ① string
- ② Number
- ③ Boolean
- ④ Null
- ⑤ Undefined 未定义
- ⑥ object 对象

①~⑤ 属于基本数据类型

⑥ 引用数据类型.

①: var str = "hello"; 3/3 单双可

str = "我说:\ "...!\", \ 转义

In 换行 化制表符.





typeof x; 检查类型

Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

Memo No. \_\_\_\_\_  
Date \_\_\_\_/\_\_\_\_/\_\_\_\_

- ② JS中所有数值都是 Number 类型  
Number.MAX-VALUE 最大值  
NaN → Not a Number

Var i = 123 + 456;

- ③ 布尔值只有两个 true false

- ④ Null 专门用来表示空的对象  
Var a = null

~~typeof~~ typeof null → object

- ⑤ 声明一个变量但不赋初值  
就是 un defined.

### 类型转换

1. → string

- ① toString 方法      a.toString();
- ② 调用String方法      String(a);

null, undefined 无 toString 方法





Mo Tu We Th Fr Sa Su

Memo No.

Date

⑤  $\rightarrow$  Number

Number() 函数

$a = \text{Number}('abc') \rightarrow \text{NaN}$

$a = \text{Number}('123') \rightarrow 123$

$a = \sim(\text{true}) \rightarrow 1$

$a = \sim(\text{false}) \rightarrow 0$

$a = \sim(\text{undefined}) \rightarrow \text{NaN}$

⑥ parseInt() 把字符串  $\rightarrow$  整数

$a = \text{parseInt}('123px') \rightarrow 123$

把有效整数内容提取出来

parseFloat 获取有效浮点数

其它进制

1. 16进制 以0x开头

$a = 0x10 \rightarrow 16$

$a = 0xff \rightarrow 255$

② 8进制 以0开头

$a = 070;$



扫描全能王 创建

③ 创建 以 06 开头

`a = parseInt(a, 10)` 指定创建

### 3. $\rightarrow$ Boolean

① Boolean 函数  
非 0 数字, 非  $\rightarrow$  true  
null 或 空字符串, 0  $\rightarrow$  false  
undefined.

运算符  
算术运算  $+ - * / \%$

$true + false \rightarrow 1 + 0 = 1$

$2 + null \rightarrow 2 + 0 = 2$

$2 + NaN \rightarrow NaN$

"你好" + "!"  $\rightarrow$  "你好!"

任何值与字符串作运算, 先转换为字符串再 +.





$$a = 100 - "1" = 99$$

一元运算符 { + 正号  
- 负号

$$a = true ; a = -a ; a = -1$$

$$a = "1" ; a = -a ; a = -1$$

$$result = 1 + "2" + 3 = 6$$

自增 自减

自增  $a++ ; (a++)$  都立即使变量自增1  
 $++a ; (++a)$

$a++$  后值为  $a$  原值

$++a$  是自增后的值

$$\log(a++) = a ;$$

$$\log(++a) = ++a ;$$

自减  $a-- ;$  同自增

$$a--a ;$$





逻辑运算

!非

&&与

||或

非布尔值情况

对于非 Boolean 时先转换为布尔, 再运算并返回  
原值

Var result = 2 & 3;  $\rightarrow$  result = 3

与运算两个值都为 true, 返回后一个

如如果有 false, 返回 0

都为 false, 返回前一个

Var result = 2 || 1;  $\rightarrow$  result = 2

或运算如果第一个为 true, 则直接返回第一个值  
第一个为 false, 返回第二个



# 赋值运算符

=

+=  $a = a + 5; \Leftrightarrow a + = 5;$

-=

\*=

/=

%=

# 关系运算符

$result = 5 > 7 \rightarrow result = false$

# 非数值情况

①  $1 > '0' \Rightarrow 1 > 0$

$10 > null \Rightarrow 10 > 0$

$10 > 'hello' \Rightarrow 10 > 1$

任何值和 NaN 比较都是 false.

② 如果两边都是字符串, 则都按字符串的 Unicode

$'a' < 'b' \rightarrow false$  两个字符串一定要转型

$9.7 < 98$





Unicode 编码

在字符串中使用转义字符输入 Unicode 编码。

14 四位编码

ex: \u0021

在网页中使用 Unicode 编码  
&# 编码

ex: <h1>&#2020;</h1>

相等运算符

1. ==

"1"==1,  $\Rightarrow$  1==1 类型不同则会先转型。

undefined 衍生自 null

所以这两个值做相等判断，会返回 true。

NaN 不和任何值相等，包括本身。

可以通过 is NaN() 函数来判断一个值是否相等。

d. !=

