**Date对象**

——内置对象，获取当前电脑的本地时间

——在js中使用Date对象来表示一个时间

**创建一个Date对象**

——如果直接使用构造函数创建一个Date对象，则会封装为当前代码执行的时间

**语法：**

var d=new Date();

**创建—个指定的时间对象**

——需要在构造函数中传递一个表示时间的字符串作为参数

——也可以直接传递从格林威治标准时间到指定时间的毫秒数

**日期的格式**

——"2020/12/12 时:分:秒"

——"2020-12-12 时:分:秒"

——2020,12,12

当传入的月份和日期大于11和月份对应的天数时 系统会自动换算成x年y月z日

**语法：**

var d2 = new Date( "12/13/2011 11:10:30");

**getFullYear()**

——从 Date 对象以四位数字返回年份

**语法：**

var year = d2.getFullYear()

**getMonth()**

——从 Date 对象返回月份 (0 ~ 11)

0表示1月

1表示2月

11表示12月

**语法：**

var month = d2.getMonth();

**getDate()**

——从 Date 对象返回—个月中的某—天（1 ~ 31)。

——获取当前的日期对象是几号

**语法：**

var date = d2.getDate();

**getDay ()**

——从 Date 对象返回—周中的某—天（0 ~ 6)

——获取当前日期对象是周几

0表示周天

1表示周一

**语法：**

var day= d2-getDay();

**getTime()**

——获取当前日期对象的时间戳

——时间戳，指从格林威治标准时建的1970年1月1日0时，0分0秒

为什么是1970年1月1日

因为1970年1月1日UNIX诞生

到当前日期所花费的毫秒数（1秒 = 1000毫秒）

——计算机底层在保存时间时使用的都是时间戳 为了统一进制方便保存

**语法：**

var d2 = new Date("2012/12/12")

var time = d2.getTime();

**Date.parse()**

——把字符串转换为Date对象 ，然后返回此Date对象与'1970/01/01 00:00:00'之间的毫秒值

**语法：**

Date.parse("2012-12-12 0:0:0")

**返回值：**

返回转换后的Date对象与起始时间之间的毫秒数。

**Date.now()**

——返回当前日期和时间的Date对象与'1970/01/01 00:00:00'之间的毫秒值(北京时间的时区为东8区，起点时间实际为：'1970/01/01 08:00:00')

**返回值：**

当前时间与起始时间之间的毫秒数。

**toDateString()**

把 Date 对象的日期部分转换为字符串

**toLocalDateString()**

根据本地时间格式，把 Date 对象的日期部分转换为字符串

**toTimeString()**

把 Date 对象的时间部分转换为字符串

**toLocalTimeString()**

根据本地时间格式，把 Date 对象的时间部分转换为字符串

**toString()**

把 Date 对象转换为字符串

**toUTCString()**

根据世界时，把 Date 对象转换为字符串

**Get**





**Set**



