**第9章 时间与日期**

学习要点：

1. Date类型
2. 通用的方法
3. 格式化方法
4. 组件方法

ECMAScript提供了Date类型来处理时间和日期。Date类型内置一系列获取和设置日期时间信息的方法。

1. **Date类型**

ECMAScript中的Date类型是在早期Java中java.util.Date类基础上构建的。为此，Date类型使用UTC (Coordinated Universal Time，国际协调时间[又称世界统一时间]) 1970年1月1日午夜(零时)开始经过的毫秒来保存日期。在使用这种数据存储格式的条件下，Date类型保存的日期能够精确到1970年1月1日之前或之后的285616年。

创建一个日期对象，使用new运算符和Date构造方法(构造函数)即可。

var box = new Date(); //创建一个日期对象

在调用Date构造方法而不传递参数的情况下，新建的对象自动获取当前的时间和日期。

alert(box); //不同浏览器显示不同

ECMAScript提供了两个方法，Date.parse()和Date.UTC()。Date.parse()方法接收一个表示日期的字符串参数，然后尝试根据这个字符串返回相应的毫秒数。ECMA-262没有定义Date.parse()应该支持哪种日期格式，因此方法的行为因实现而异，因地区而异。默认通常接收的日期格式如下：

1. '月/日/年'，如6/13/2011;
2. '英文月名 日, 年'，如 May 25, 2004;
3. '英文星期几 英文月名 日 年 时:分:秒 时区'，如 Tue May 25 2004 00:00:00 GMT-070

alert(Date.parse('6/13/2011')); //1307894400000

如果Date.parse()没有传入或者不是标准的日期格式，那么就会返回NaN。

alert(Date.parse()); //NaN

如果想输出指定的日期，那么把Date.parse()传入Date构造方法里。

var box = new Date(Date.parse('6/13/2011'));//Mon Jun 13 2011 00:00:00 GMT+0800

var box = new Date('6/13/2011'); //直接传入，Date.parse()后台被调用

PS：Date对象及其在不同浏览器中的实现有许多奇怪的行为。其中有一种倾向是将超出的范围的值替换成当前的值，以便生成输出。例如，在解析“January 32, 2007”时，有的浏览器会讲其解释为“February 1, 2007”。而Opera则倾向与插入当前月份的当前日期。

Date.UTC()方法同样也返回表示日期的毫秒数，但它与Date.parse()在构建值时使用不同的信息。(年份，基于0的月份[0表示1月，1表示2月]，月中的哪一天[1-31]，小时数[0-23]，分钟，秒以及毫秒)。只有前两个参数是必须的。如果没有提供月数，则天数为1；如果省略其他参数，则统统为0.

alert(Date.UTC(2011,11)); //1322697600000

如果Date.UTC()参数传递错误，那么就会出现负值或者NaN等非法信息。

alert(Date.UTC()); //负值或者NaN

如果要输出指定日期，那么直接把Date.UTC()传入Date构造方法里即可。

var box = new Date(Date.UTC(2011,11, 5, 15, 13, 16));

1. **通用的方法**

与其他类型一样，Date类型也重写了toLocaleString()、toString()和valueOf()方法；但这些方法返回值与其他类型中的方法不同。

var box = new Date(Date.UTC(2011,11, 5, 15, 13, 16));

alert('toString:' + box.toString());

alert('toLocaleString:' + box.toLocaleString()); //按本地格式输出

PS：这两个方法在不同浏览器显示的效果又不一样，但不用担心，这两个方法只是在调试比较有用，在显示时间和日期上，没什么价值。valueOf()方法显示毫秒数。

1. **日期格式化方法**

Date类型还有一些专门用于将日期格式化为字符串的方法。

var box = new Date();

alert(box.toDateString()); //以特定的格式显示星期几、月、日和年

alert(box.toTimeString()); //以特定的格式显示时、分、秒和时区

alert(box.toLocaleDateString()); //以特定地区格式显示星期几、月、日和年

alert(box.toLocaleTimeString()); //以特定地区格式显示时、分、秒和时区

alert(box.toUTCString()); //以特定的格式显示完整的UTC日期。

1. **组件方法**  
    组件方法，是为我们单独获取你想要的各种时间/日期而提供的方法。需要注意的时候，这些方法中，有带UTC的，有不带UTC的。UTC日期指的是在没有时区偏差的情况下的日期值。

alert(box.getTime()); //获取日期的毫秒数，和valueOf()返回一致

alert(box.setTime(100)); //以毫秒数设置日期，会改变整个日期

alert(box.getFullYear()); //获取四位年份

alert(box.setFullYear(2012)); //设置四位年份，返回的是毫秒数

alert(box.getMonth()); //获取月份，没指定月份，从0开始算起

alert(box.setMonth(11)); //设置月份

alert(box.getDate()); //获取日期

alert(box.setDate(8)); //设置日期，返回毫秒数

alert(box.getDay()); //返回星期几，0表示星期日，6表示星期六

alert(box.setDay(2)); //设置星期几

alert(box.getHours()); //返回时

alert(box.setHours(12)); //设置时

alert(box.getMinutes()); //返回分钟

alert(box.setMinutes(22)); //设置分钟

alert(box.getSeconds()); //返回秒数  
 alert(box.setSeconds(44)); //设置秒数

alert(box.getMilliseconds()); //返回毫秒数

alert(box.setMilliseconds()); //设置毫秒数

alert(box.getTimezoneOffset()); //返回本地时间和UTC时间相差的分钟数

PS：以上方法除了getTimezoneOffset()，其他都具有UTC功能，例如setDate()及getDate()获取星期几，那么就会有setUTCDate()及getUTCDate()。表示世界协调时间。