

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

主讲教师: 李炎恢

合作网站: <http://www.ibeifeng.com>

讲师博客: <http://hi.baidu.com/李炎恢>

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。

一. 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,有的浏览器会将其解释为“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));
```

## 二. 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 these 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。

## 三. 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.toString()); //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。
```

## 四. 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。

# 感谢收看本次教程!

本课程是由北风网([ibeifeng.com](http://ibeifeng.com))

瓢城 Web 俱乐部([yc60.com](http://yc60.com))联合提供:

本次主讲老师: 李炎恢

我的博客: [hi.baidu.com/李炎恢/](http://hi.baidu.com/李炎恢/)

我的邮件: [yc60.com@gmail.com](mailto:yc60.com@gmail.com)

