**中文使用说明：**

## 目录

* [概述](http://javascript.ruanyifeng.com/library/datejs.html#toc0)
* [方法](http://javascript.ruanyifeng.com/library/datejs.html#toc1)
* [日期信息](http://javascript.ruanyifeng.com/library/datejs.html#toc2)
* [日期的变更](http://javascript.ruanyifeng.com/library/datejs.html#toc3)
* [日期的解析](http://javascript.ruanyifeng.com/library/datejs.html#toc4)
* [参考链接](http://javascript.ruanyifeng.com/library/datejs.html#toc5)

## 概述

Datejs是一个用来操作日期的库，官方网站为[datejs.com](http://www.datejs.com/)。

下载后插入网页，就可以使用。

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

官方还提供多种语言的版本，可以选择使用。

// 美国版

<script type="text/javascript" src="date-en-US.js"></script>

// 中国版

<script type="text/javascript" src="date-zh-CN.js"></script>

## 方法

Datejs在原生的Date对象上面，定义了许多语义化的方法，可以方便地链式使用。

### 日期信息

Date.today() // 返回当天日期，时间定在这一天开始的00:00

Date.today().getDayName() // 今天是星期几

Date.today().is().friday() // 今天是否为星期五，返回true或者false

Date.today().is().fri() // 等同于上一行

Date.today().is().november() // 今天是否为11月，返回true或者false

Date.today().is().nov() // 等同于上一行

Date.today().isWeekday() // 今天是否为工作日（周一到周五）

### 日期的变更

Date.today().next().friday() // 下一个星期五

Date.today().last().monday() // 上一个星期一

new Date().next().march() // 下个三月份的今天

new Date().last().week() // 上星期的今天

Date.today().add(5).days() // 五天后

Date.friday() // 本周的星期五

Date.march() // 今年的三月

Date.january().first().monday() // 今年一月的第一个星期一

Date.dec().final().fri() // 今年12月的最后一个星期五

// 先将日期定在本月15日的下午4点30分，然后向后推90天

Date.today().set({ day: 15, hour: 16, minute: 30 }).add({ days: 90 })

(3).days().fromNow() // 三天后

(6).months().ago() // 6个月前

(12).weeks().fromNow() // 12个星期后

(30).days().after(Date.today()) // 30天后

### 日期的解析

Date.parse('today')

Date.parse('tomorrow')

Date.parse('July 8')

Date.parse('July 8th, 2007')

Date.parse('July 8th, 2007, 10:30 PM')

Date.parse('07.15.2007')

**Example**

[html]  
<script type="text/javascript" src="date.js"></script>  
[/html]  
You can download the latest release from [datejs.com/download/](http://www.datejs.com/download/) or connect directly to the SVN source repository via [datejs.com/svn/](http://www.datejs.com/svn/).

The date.js file can be found within the /build/ folder.

Datejs has traveled to many lands and returns with knowledge of over 150 individual Cultures. Supporting your language of choice is simple, just swap out the date.js file for another culture-specific file.

**Example**

[html]  
en-US [English (United States)]  
<script type="text/javascript" src="date.js"></script>

en-US [English (United States)]  
<script type="text/javascript" src="date-en-US.js"></script>

de-DE [German (Germany)]  
<script type="text/javascript" src="date-de-DE.js"></script>

fr-FR [French (France)]  
<script type="text/javascript" src="date-fr-FR.js"></script>  
[/html]  
All 150+ CultureInfo files have been pre-compiled and are available within the same /build/ folder as date.js. Each culture file includes translations for many of the strings used in the Datejs library. Some strings have not been translated, although will be filled in over time as the community contributes.

**Step #2 - Start your training**

The Datejs library includes many helpful functions for easing the pain of developing with Dates and Times in JavaScript. Once the date.js file has been included into your page you can begin some serious training.

If you have not already done so, we highly recommend installing [FireBug](http://www.getfirebug.com/) for [FireFox](http://www.getfirefox.com/). Among many things, Firebug allows you to execute custom JavaScript code directly in the browser without having to edit your source page — kind of like a command-line for JavaScript. Get it now. The Ninja waits.

**Let's Get Started**

Before diving deep into the library, let's first limber up with some stretching. Exploring the natural language syntax is a good place to start.  
[js]  
// Get today's date  
Date.today();

// Add 5 days to today  
Date.today().add(5).days();

// Get Friday of this week  
Date.friday();

// Get March of this year  
Date.march();

// Is today Friday?  
Date.today().is().friday(); // true|false

// What day is it?  
Date.today().getDayName();

[/js]

Everything ok? A little out of breath? Soooo sorry.

Now, some Date *Assassin* exercises.  
[js]  
// Get the first Monday of the year  
Date.january().first().monday()

// Get the last Friday of the year  
Date.dec().final().fri()

// Set a date to the 15th of the current month at 4:30 PM,  
// then add 90 days and make sure that date is a weekday,  
// else move to the next weekday.  
var d1 = Date.today()  
.set({ day: 15, hour: 16, minute: 30 })  
.add({ days: 90 })  
if (!d1.isWeekday()) {  
d1.next().monday();  
}  
[/js]  
How about letting your users enter a few dates? Say into an <input> field or date picker? Included with the Datejs library is a powerful replacement for the native JavaScript Date parser.

The following examples all start with a String value that we convert into a Date object.  
[js]  
// Lets start simple. "Today"  
Date.parse('today');

// How about tomorrow?  
Date.parse('tomorrow');

// July 8?  
Date.parse('July 8');

// With a year?  
Date.parse('July 8th, 2007');

// And time?  
Date.parse('July 8th, 2007, 10:30 PM');

// Get the date, move to Monday (if not already Monday),  
// then alert the date to the user in a different format.  
var d1 = Date.parse('8-Jul-2007');  
if (!d1.is().monday()) {  
d1.last().monday();  
}  
alert(d1.toString('dddd, MMMM d, yyyy'));  
[/js]

The library also includes some Number fun. In order to execute functions directly on JavaScript Number objects, the number must be wrapped in parentheses. This is a requirement of JavaScript. If the number is declared first, the parentheses are not required.  
[js]  
// Get a date 3 days from now  
(3).days().fromNow();

// 6 month ago  
(6).months().ago();

// 12 weeks from now  
var n = 12;  
n.weeks().fromNow();

// Get a date 30 days after a user supplied date  
var d1 = Date.parse('07.15.2007');  
var d2 = (30).days().after(d1);  
[/js]

**Step #3 - Refine your skillz**

You are nearing a state of Date Ninja-ness, but more experience is required.

Please be sure to check out the following list of resources for further training.

* Datejs at [GoogleCode](http://www.datejs.com/googlecode/)
* [Documentation](http://www.datejs.com/documentation/)
* [Forums](http://www.datejs.com/forums/)
* [Downloads](http://www.datejs.com/download/)
* [SVN](http://www.datejs.com/svn/)

The Datejs Library also includes good size [Test Suite](http://www.datejs.com/test/), which is worth browsing to get a idea of what's possible.

Hope this helps.