datejs
A JavaScript Date Library

Project Home   Downloads   **Wiki**   Issues   Source

Search   for

FormatSpecifiers
Format Specifiers are used to specify date formats for display and input.
Featured

Updated Feb 4, 2010 by geoff%co...@gtempaccount.com

toString (available in `core.js`)

◈ .toString(`String format`) : `String`

The .toString() function is available in `core.js` or any compiled `/build/` file.

Parameters

- {`String`} A format string consisting of one or more format spcifiers. **optional**.

Return Type

{`String`} A string representation of the current Date object.

The following Custom and Standard format specifier strings/characters are used with the `.toString` `(format)` function.

CUSTOM DATE AND TIME FORMAT SPECIFIERS

```
Format  Description
   Example
------
  --------------------------------------------------------------------  ----------------------
  s       The seconds of the minute between 0-59.
  "0" to "59"
  ss      The seconds of the minute with leading zero if required.
```

```
                       "00" to "59"

     m       The minute of the hour between 0-59.
"0"   or "59"
     mm      The minute of the hour with leading zero if required.
 "00" or "59"

     h       The hour of the day between 1-12.
 "1"  to "12"
     hh      The hour of the day with leading zero if required.
"01" to "12"

     H       The hour of the day between 0-23.
 "0"  to "23"
     HH      The hour of the day with leading zero if required.
"00" to "23"

     d       The day of the month between 1 and 31.
"1"  to "31"
     dd      The day of the month with leading zero if required.
 "01" to "31"
     ddd     Abbreviated day name. Date.CultureInfo.abbreviatedDayNames.
 "Mon" to "Sun"
     dddd    The full day name. Date.CultureInfo.dayNames.
   "Monday" to "Sunday"

     M       The month of the year between 1-12.
 "1" to "12"
     MM      The month of the year with leading zero if required.
"01" to "12"
     MMM     Abbreviated month name. Date.CultureInfo.abbreviatedMonthNames.
 "Jan" to "Dec"
     MMMM    The full month name. Date.CultureInfo.monthNames.
   "January" to "December"

     yy      The year as a two-digit number.
 "99" or "08"
     yyyy    The full four digit year.
   "1999" or "2008"

     t       Displays the first character of the A.M./P.M. designator.
 "A" or "P"
            $C.amDesignator or Date.CultureInfo.pmDesignator
     tt      Displays the A.M./P.M. designator.
"AM" or "PM"
            $C.amDesignator or Date.CultureInfo.pmDesignator

     S       The ordinal suffix ("st, "nd", "rd" or "th") of the current day.
 "st, "nd", "rd" or "th"
```

.

STANDARD DATE AND TIME FORMAT SPECIFIERS

A Standard format specifier for `.toString(format)` consists of a single character. If multiple characters are supplied, the string is treated as a Custom format specifier.

```
Format   Description                                                     Example ("en-US")
------   -------------------------------------------------------------   ---------------------
d        The CultureInfo shortDate Format Pattern                        "M/d/yyyy"
D        The CultureInfo longDate Format Pattern                         "dddd, MMMM dd, yyyy"
F        The CultureInfo fullDateTime Format Pattern                     "dddd, MMMM dd, yyyy h:mm:ss tt"
m        The CultureInfo monthDay Format Pattern                         "MMMM dd"
r        The CultureInfo rfc1123 Format Pattern                          "ddd, dd MMM yyyy HH:mm:ss GMT"
s        The CultureInfo sortableDateTime Format Pattern                 "yyyy-MM-ddTHH:mm:ss"
t        The CultureInfo shortTime Format Pattern                        "h:mm tt"
T        The CultureInfo longTime Format Pattern                         "h:mm:ss tt"
u        The CultureInfo universalSortableDateTime Format Pattern        "yyyy-MM-dd HH:mm:ssZ"
y        The CultureInfo yearMonth Format Pattern                        "MMMM, yyyy"
```

. .

format (available in `extras.js`)

⬥ .format(String format) : String

The .format() function is available in the optional `extras.js` module. The `extras.js` module is NOT included with the `/build/` file and must be added separately to your page.

The `core.js` (or any `/build/` file) is required and must be included before for `extras.js`.

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

Parameters

- {String} A format string consisting of one or more format spcifiers. **optional**

Return Type

{String} A string representation of the current Date object.

The following format specifiers are used with the `.format(format)` or `.$format(format)` function available by including the `extras.js` module.

The following descriptions are from http://www.php.net/strftime and http://www.php.net/manual/en/function. date.php. Copyright © 2001-2008 The PHP Group

PHP/UNIX FORMAT SPECIFIERS

```
Format  Description
   Example
------
   -------------------------------------------------------------------  ----------------------
 %a     abbreviated weekday name according to the current localed
 "Mon" through "Sun"
 %A     full weekday name according to the current locale
   "Sunday" through "Saturday"
 %b     abbreviated month name according to the current locale
"Jan" through "Dec"
 %B     full month name according to the current locale
   "January" through "December"
 %c     preferred date and time representation for the current locale
   "4/13/2008 12:33 PM"
 %C     century number (the year divided by 100 and truncated to an integer)
"00" to "99"
 %d     day of the month as a decimal number
"01" to "31"
 %D     same as %m/%d/%y
   "04/13/08"
 %e     day of the month as a decimal number, a single digit is preceded by a space
 "1" to "31"
 %g     like %G, but without the century                                              "08"
```

```
%G      The 4-digit year corresponding to the ISO week number (see %V).
  "2008"
        This has the same format and value as %Y, except that if the ISO week number
        belongs to the previous or next year, that year is used instead.
 %h      same as %b
"Jan" through "Dec"
 %H      hour as a decimal number using a 24-hour clock
"00" to "23"
 %I      hour as a decimal number using a 12-hour clock
"01" to "12"
 %j      day of the year as a decimal number
 "001" to "366"
 %m      month as a decimal number
 "01" to "12"
 %M      minute as a decimal number
"00" to "59"
 %n      newline character                                                  "\n"
 %p      either "am" or "pm" according to the given time value, or the
 "am" or "pm"
        corresponding strings for the current locale
 %r      time in a.m. and p.m. notation
"8:44 PM"
 %R      time in 24 hour notation
  "20:44"
 %S      second as a decimal number
"00" to "59"
 %t      tab character                                                      "\t"
 %T      current time, equal to %H:%M:%S
   "12:49:11"
 %u      weekday as a decimal number ["1", "7"], with "1" representing Monday
"1" to "7"
 %U      week number of the current year as a decimal number, starting with the
"0" to ("52" or "53")
        first Sunday as the first day of the first week
 %V      The ISO 8601:1988 week number of the current year as a decimal number,
"00" to ("52" or "53")
        range 01 to 53, where week 1 is the first week that has at least 4 days
        in the current year, and with Monday as the first day of the week.
        (Use %G or %g for the year component that corresponds to the week number
        for the specified timestamp.)
 %W      week number of the current year as a decimal number, starting with the
"00" to ("52" or "53")
        first Monday as the first day of the first week
 %w      day of the week as a decimal, Sunday being "0"
"0" to "6"
 %x      preferred date representation for the current locale without the time
   "4/13/2008"
 %X      preferred time representation for the current locale without the date
   "12:53:05"
```

```
%y      year as a decimal number without a century
"00" "99"
%Y      year as a decimal number including the century
 "2008"
%Z      time zone or name or abbreviation
   "UTC", "EST", "PST"
%z      same as %Z
%%      a literal "%" character                                              "%"

d      Day of the month, 2 digits with leading zeros
"01" to "31"
D      A textual representation of a day, three letters
"Mon" through "Sun"
j      Day of the month without leading zeros
"1" to "31"
l      A full textual representation of the day of the week (lowercase "L")
 "Sunday" through "Saturday"
N      ISO-8601 numeric representation of the day of the week (added in PHP
5.1.0) "1" (for Monday) through "7" (for Sunday)
S      English ordinal suffix for the day of the month, 2 characters
"st", "nd", "rd" or "th". Works well with j
w      Numeric representation of the day of the week
  "0" (for Sunday) through "6" (for Saturday)
z      The day of the year (starting from "0")
"0" through "365"
W      ISO-8601 week number of year, weeks starting on Monday
"00" to ("52" or "53")
F      A full textual representation of a month, such as January or March
 "January" through "December"
m      Numeric representation of a month, with leading zeros
"01" through "12"
M      A short textual representation of a month, three letters
"Jan" through "Dec"
n      Numeric representation of a month, without leading zeros
"1" through "12"
t      Number of days in the given month
"28" through "31"
L      Whether it's a leap year
"1" if it is a leap year, "0" otherwise
o      ISO-8601 year number. This has the same value as Y, except that if the
 "2008"
       ISO week number (W) belongs to the previous or next year, that year
       is used instead.
Y      A full numeric representation of a year, 4 digits
  "2008"
y      A two digit representation of a year                                  "08"
a      Lowercase Ante meridiem and Post meridiem
"am" or "pm"
A      Uppercase Ante meridiem and Post meridiem
```

```
   "AM" or "PM"
   B     Swatch Internet time
"000" through "999"
   g     12-hour format of an hour without leading zeros
 "1" through "12"
   G     24-hour format of an hour without leading zeros
 "0" through "23"
   h     12-hour format of an hour with leading zeros
"01" through "12"
   H     24-hour format of an hour with leading zeros
"00" through "23"
   i     Minutes with leading zeros
"00" to "59"
   s     Seconds, with leading zeros
 "00" through "59"
   u     Milliseconds
  "54321"
   e     Timezone identifier
   "UTC", "EST", "PST"
   I     Whether or not the date is in daylight saving time (uppercase i)
"1" if Daylight Saving Time, "0" otherwise
   O     Difference to Greenwich time (GMT) in hours
   "+0200", "-0600"
   P     Difference to Greenwich time (GMT) with colon between hours and minutes
   "+02:00", "-06:00"
   T     Timezone abbreviation
   "UTC", "EST", "PST"
   Z     Timezone offset in seconds. The offset for timezones west of UTC is
"-43200" through "50400"
         always negative, and for those east of UTC is always positive.
   c     ISO 8601 date
"2004-02-12T15:19:21+00:00"
   r     RFC 2822 formatted date
"Thu, 21 Dec 2000 16:01:07 +0200"
   U     Seconds since the Unix Epoch (January 1 1970 00:00:00 GMT)          "0"
```

Comment by [phillip....@gmail.com](), Feb 11, 2008

Any possibility of getting the ordinal suffix where appropriate?

Comment by [geoff%co...@gtempaccount.com](), Feb 11, 2008

The Ordinal suffix ('st', 'nd', 'rd', 'th') is available by calling the .getOrdinal() function on a date.

Example

Date.today().getOrdinal(); new Date().add(1).week().getOrdinal();

Hope this helps answer your question.

Comment by baxton...@gmail.com, Feb 13, 2008

Hi My display string is like this dateResult = curDate.toString('dddd, dd MMMM, yyyy hh:mm tt');

If i enter "1:30p" is shows 01:30 PM

If i enter "1:30p tomorrow" it shows 01:30 AM

Why? What am I doing wrong?

Thanks

Comment by [geoff%co...@gtempaccount.com](), Feb 13, 2008

@baxton015 - Can you confirm you are using the latest code from SVN ([http://www.datejs.com/svn)?]() When I try the strings in the "Mad Skillz" widget on the Datejs home page, they appear to parse correctly. FYI, I think this help request should be posted in the forums.

Comment by [balup...@gmail.com](), Mar 19, 2008

Could there please be added support for: [http://php.net/strftime]() [http://www.smarty.net/manual/en/language.modifier.date.format.php]()

Comment by [geoff%co...@gtempaccount.com](), Mar 19, 2008

@balupton - Early on we made the decision to go with the much cleaner Java/.NET style of date and time format specifiers. The PHP/ Unix format specifier syntax will not be supported in the library. If you have any questions regarding how to format a date into a string, please feel free to ask in the Datejs forums ([http://www.datejs.com/forums/]()).

Comment by depink...@gmail.com, Mar 24, 2008

Any change of getting a timezone format specifier? I know I can invoke getTimezone(), but it would be great to get it within the toString () format! Otherwise, this is great stuff!

Comment by [geoff%co...@gtempaccount.com](), Mar 25, 2008

@balupton - I've reconsidered adding functionality to convert the PHP/Unix date format to a Java/.NET format. I think the conversion function will be added to an optional "extras.js" module. See the follow Issue for more information. http://code.google.com/p/datejs/issues/detail?id=34

Comment by [geoff%co...@gtempaccount.com](), Mar 25, 2008

@depinkham - The timezone format specifier is absolutely a missing piece. There's some known loose ends regarding the timezone support in the library, so I'm going to spend some energy hammering out those details.

Comment by [geoff%co...@gtempaccount.com](), Apr 12, 2008

Format Specifier support for the Ordinal Suffix has been added to .toString(). Please use "S" to get the ordinal suffix.

Example

Date.today().toString("S"); // "st", "nd", "rd" or "th"

The "S" format specifier can be combined with other format specifiers.

Example

Date.today().toString("MMMM dS, yyyy"); // "April 12th, 2008"

The .getOrdinal() function has been removed from the library. THE FOLLOWING WILL NO LONGER WORK.

Example

Date.today().getOrdinal(); // ERROR: getOrdinal is not a function

Comment by [redeemer...@gmail.com](), May 24, 2008

I'm trying to parseExact a date string like "yyyy-MM-ddTHH:mm:ss.000-05:00" but the latter part (-05:00) breaks if it's a different time zone (-06:00) is present. I've tried using "P" (noted above) but no joy. I'm using trunk. I can't change the format... google calendar RSS. :)

I've also tried using just "c", but the presence of milliseconds in the string breaks it.

Anything I can try?

Comment by [gr4v3...@gmail.com](), Sep 30, 2008

where can i get extras.js ?

Comment by [geoff%co...@gtempaccount.com](), Sep 30, 2008

@gr4v3m4n - http://code.google.com/p/datejs/source/browse/#svn/trunk/src

---

Comment by [tonyhs...@gmail.com](), Apr 15, 2009

The custom format and standard format for "d" overlaps. How might I get just the custom format for just the date number?

Comment by [tonyhs...@gmail.com](), Apr 15, 2009

As a temporary hack, I deleted the following:

case"d":return x.t(c.shortDate);

That got rid of the shortDate. But the bad news is, that got ride of the short date. =(

Comment by [geoff%co...@gtempaccount.com](), Apr 15, 2009

@tonyhschu - just use the native .getDate() function.

Example

new Date().getDate()

Comment by [dcorle...@gmail.com](), May 13, 2009

So, I notice that the native function getTimezoneOffset returns the number of minutes from GMT. But this is completely useless because it doesn't give the sign - just the absolute value. Any plans to provide a format specifier or method to return the offset, in secs or msecs, of the current Date object? E.g. like the Java TimeZone[?].getOffset() method. Right now it looks like what I'll have to do is use getUTCOffset, extract just the sign char, prepend it to the getTimezoneOffset, and then multiply by 600000. Ouch.

Comment by [rosh...@gmail.com](), Jan 26, 2010

I keep getting d.format is not a function. What is wrong with the following code

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

```
javascript" src="extra.js"></script>

<script type="text/javascript">

    //alert(Date("1264426428434").toString("d "));

        var d = new Date("1264426428434");

// alert(d.toString("y"));

    alert(d.format("o"));

</script>
```

Comment by track...@gmail.com, Feb 22, 2010

@rosh.pr d is NaN, it's an invalid date, now if you remove the quotes from the number you will get a Date set to a ms offset from 1970-01-01T00:00:00.000Z ...

Comment by jnthnl...@gmail.com, Jul 27, 2011

Is there a way to escape certain characters? I'd like to produce e.g. "13hrs" and Date.toString('hhrs') is not going to do it, as 'hrs' is going to be interpreted with 'h' and 's' as special characters.

Comment by danor...@gmail.com, Aug 28, 2011

Ditto the previous. How do you escape a character?

Comment by m...@udargo.com, Jan 10, 2012

danor, etc...

I had the same question and I found a work-around. Use the HTML ASCII code for the characters you want to insert in the format string. For instance, I wanted to insert "at" into the string (e.g., "Sunday, April 4, 2022 at 4:44 PM"), but "t" is reserved, so I used "at" (""dddd, MMMM dd, yyyy at h:mm tt"). Here is a link with all the HTML ASCII codes: http://www.ascii.cl/htmlcodes.htm.

Comment by peng...@gmail.com, Mar 27, 2012

is it possible to format a date string for a specified timezone in stead of the locale timezone of the client browser?

e.g. new Date().toString("yyyy-MM-dd HH:mm:ss", "PDT");

Comment by c.philli...@gmail.com, Apr 29, 2012

Why aren't more than just 4 US time zones supported?

Comment by jglawre...@gmail.com, Jun 6, 2012

When I try Date().toString('r') for the rfc1123 format all I get back is "r"

Am I doing something wrong?

Comment by schmeng...@gmail.com, Aug 24, 2012

Use the trunk version, in the Alpha version the "r" format is not yet supported. But the rfc1123 format is broken anyways, the string contains "GMT" while the "M" is interpreted as month. I tried changing the constant, setting "O" instead of "GMT" but the identifier for timezones is not recognized in toString().

It //should// work with format() though.

Comment by peter.lu...@gmail.com, Oct 4, 2012

The S for ordinal suffix does not work, i.e. it outputs S instead of the appropriate st, nd, rd, th. Despite claims in these comments the toOrdinal() method is still in the source. I guess that means I'll have to do myDate.toString("dddd d{0} MMMM yyyy").replace("{0}", myDate.getOrdinal())

Comment by ASHWANI7...@gmail.com, Oct 10, 2012

following is code to remove ordinal of date IT is running successfully:

import java.text.DateFormat?; import java.text.ParseException?; import java.text.SimpleDateFormat?; import java.util.Date; import java.util.regex.Pattern; import java.util.regex.Matcher;

public class patrn {

    private static String deleteOrdinal(String dateString) {

        Pattern p1 = Pattern.compile("([0-9]+)(st|nd|rd|th)");

```
                    Matcher m = p1.matcher(dateString); while (m.find()) {

                            dateString = dateString.replaceAll(Matcher.quoteReplacement(m.group(0)), m.group(1));

                    } return dateString; }

            public static void main(String args) {

            String dateString="August 21st, 2012"; SimpleDateFormat? sdf = new SimpleDateFormat?("MMMM dd, yyyy"); Date emp1joinDate = null;
            try { emp1joinDate=sdf.parse(deleteOrdinal(dateString)); } catch (ParseException? e) {

                // TODO Auto-generated catch block e.printStackTrace();

            }

            }

    }
```

Comment by [d...@gmx.ch](#), Oct 11, 2012

Unfortunately, one of the supposedly most needed functions when dealing with locales doesn't work:

.format('%x')

It just returns the day's number, instead of the preferred date representation for the current locale without the time "4/13/2008",
as documented above.

In extras.js I found: [...] case"%x":return t("d"); [...] Obviously, this returns just the day. Any Idea how this can be fixed?

When the user types a date, e.g "13.4.2008" on a localized german page, I must parse it (which works fine) and then correct it to
display "13.04.2012", but of course I do not want to hard-code the format string as "dd.MM.yyyy", since with a different locale it
should perhaps be "4/13/2008".

Thanks, Don P

Comment by [d...@gmx.ch](#), Oct 11, 2012

Hi again,

Now it seems to work with .toString(Date.[CultureInfo](#).formatPatterns.shortDate); and without extras.js.

Think I can live with it :)

Don P

Finally: to make the %x format work as desired in extras.js, I replaced `case"%x": return t("d");` with `case"%x": return t(Date.CultureInfo.formatPatterns.shortDate);`

Don P

► [Sign in](#) to add a comment

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)