The `texdate` Package, v2.0

Donald P. Goodman III

December 8, 2018

Abstract

TEX and LATEX provide few facilities for dates by default, though many packages have filled this gap. `texdate` fills it, as well, with a pure TEX-primitive implementation. It can print dates, advance them by numbers of days, weeks, or months, determine the weekday automatically, and print them in (mostly) arbitrary format. It can also print calendars (monthly and yearly) automatically, and can be easily localized for non-English languages.

Contents

1 Introduction: The State of the TEX

TEX by default contains very little facilities for dealing with dates, and LATEX follows suit. As far as primitives go, TEX just offers the counters `\day`, `\month`, and `\year`, which give the current values of those units of time; e.g., `\the\year-\the\month-\the\day` will yield "2018-12-8" (which is the date on which this document was compiled). LATEX also has `\today`, which will produce the current date in the default American style: "December 8, 2018". But that's really about all there is.

Many packages have attempted to fill up this gap, some with excellent success; `datetime2` certain deserves special mention here, particularly as it goes beyond what `texdate` offers, given that `texdate` contains no facilities for *times* at all. `texdate` tries to fill the gap, as well; but it does it using only TEX-primitives, in the hope that the solution will be (a) pretty fast, (b) pretty portable, and (c) not requiring the loading of massive packages, only a fraction of the capabilities of which will actually be used.

For comparison, `datetime2` uses `pgfcalendar`, which of course requires `pgf`, which is a huge package. Our modern computers make loading such packages often a negligible overhead; but with large and complex documents, it's not always trivial. Also, it's an enjoyable challenge to write a usable package in TEX for something for which TEX was not designed; and some of us enjoy *just knowing* that we're using a lean package, even it makes little practical difference.

This document is numbered in *dozenal*, or base twelve; numbering proceeds 1, 2, 3, 4, 5, 6, 7, 8, 9, ↊, ↋, 10, 11, 12 . . . It uses the `dozenal` LATEX package to do this. For more information, see `http://www.dozenal.org`.

2 Dependencies

`texdate` requires the `padcount`, `modulus`, and `iflang` packages internally, so be sure that they are installed. They are all available on CTAN and in the TEXLive distribution.

3 Printing and Setting the Date

`texdate` works with an *internal date*, which is the current setting of all the date variables. When initiated, the internal date is 1 January of the current year. We
`\printdate` can print that date with `\printdate`:

```
\printdate
```

Sunday (Sun), January (Jan) 01, 2018

We can easily set the internal date to the current date by running the macro
`\initcurrdate` `\initcurrdate`:

```
\initcurrdate
\printdate
```

Saturday (Sat), December (Dec) 08, 2018

(This is the current date at the time this document was compiled.)

`\initdate` You can also easily *set* the internal date, by running the `\initdate` macro:

`\initdate {`⟨*year*⟩`} {`⟨*month*⟩`} {`⟨*day-of-month*⟩`}`

The elements of the date must be supplied to `\initdate` in that order, or `texdate` will become confused. It's obvious why; what should `texdate` do if the month is entered as `2019`?

```
\initdate{2019}{6}{24}
\printdate
```

Monday (Mon), June (Jun) 24, 2019

While internally dates are kept as zero-indexed, these dates are received by `\initdate` as one-indexed; that is, `24` will mean the twenty-fourth, not the twenty-fifth, because we count starting at 1 rather than 0.

4 Date Formats

The date format we've seen so far is the default, which is designed primarily to demonstrate several of the possible variables that can be in a date format. Naturally, you'll want to change it; and it can be changed, almost arbitrarily, simply by redefining a command, or by using one of several presets.

4.1 Preset Formats

`texdate` provides a number of preset formats that can be easily selected without having to design a format string (for which see Section 4.2, on page 4).

`\printfdate{ISO}` `\printfdate{ISO}` will print the current date in the default ISO 8601 format, which is *yyyymmdd*. In `texdate`'s formatting strings, this is `Ymd`; you'll learn more about these in Section 4.2. There is also the "ISO extended" form, `Y-m-d`.

```
\initdate{2019}{6}{24}
\printfdate{ISO}

\printfdate{ISOext}
```

```
20190624
2019-06-24
```

\printfdate{american}
\printfdate{shamerican}

For Americans fond of our curious customary format, you can use `\printfdate{american}`; in `texdate` format strings, this is `B\ d, Y`. There is also `\printfdate{shamerican}`, which is the abbreviated form, using slashes rather than hyphens.

```
\initdate{2019}{6}{24}
\printfdate{american}

\printfdate{shamerican}
```

```
June 24, 2019
06/24/2019
```

\printfdate{british}
\printfdate{shbritish}
\printfdate{shbritishdots}

The British also have their own ways of writing dates, which correspond largely to the way the American military writes them (which are consequently sometimes called "military dates," in the same way that twenty-four-hour time readings are sometimes called "military time"). These are `\printfdate{british}` and `\printfdate{shbritish}`, along with alternate form `\printfdate{shbritishdots}`,

```
\initdate{2019}{6}{24}
\printfdate{british}

\printfdate{shbritish}

\printfdate{shbritishdots}
```

```
24 June 2019
24/06/2019
24.06.2019
```

This is enough to cover the standards of most places in the world. However, if you want something different, you can easily create it with format strings.

4.2 Custom Date Formats

All the custom formats described in Section 4.1 and printed with `\printfdate` are created using the same general mechanism described in this section. We will begin by discussing a way to generically change the presentation of all dates called with the basic `\printdate`, then move on to creating custom date formats that can be printed by name with `\printfdate`.

\setdateformat

The macro `\setdateformat` holds the formatting string for the date. It's not *completely* arbitrary, because none of the characters used to produce specific parts

of the date can be used in the string itself; however, it's pretty flexible despite that limitation.

The default date format string, quite unsuitable for real work, includes most of the possible control characters, and is `A{ }(a),\ B\ (b){ }d,\ Y`. Note that spaces have to be preserved by either *bracing* them or *escaping* them; that is, to put a space in your format string, use either "`\ `" or "`{ }`".

Table 1 on page 5 shows the control characters, an explanation of their meaning, and an example of each. They assume the date 4 June 2019, selected by `\initdate{2019}{6}{4}`.

Let.	*Result*	*Ex.*
d	Numeric day of the month; 0-padded to two digits if necessary	04
e	Numeric day of the month; space-padded to two spaces if necessary	4
B	Full name of the month	June
b	Abbreviated name of the month	Jun
h	Abbreviated name of the month; same as `b`	Jun
m	Number of month, with January as 1; 0-padded to two digits if necessary	06
A	Full name of the weekday	Tuesday
a	Abbreviated name of the weekday	Tue
w	Numeric value of weekday, with Sunday as 0	2
u	ISO numeric value of weekday, with Monday as 1 and Sunday as 7	2
Y	Number of the current year	2019
j	Numeric day of the year, starting on a constant count from 1 Jan; 0-padded to three digits if necessary	155
C	Century; essentially, the first two digits of the year	20
y	The year, in only two digits	19
U	Week number of the year, starting at 0, with the week starting on Sunday; 0-padded to two digits if necessary	22
V	ISO week number of the year, starting at 1, with the week starting on Monday; 0-padded to two digits if necessary	23
W	Week number of the year, starting at 0, with the week starting on Monday; 0-padded to two digits if necesary	22

Table 1: Control codes for date formats

For folks not familiar with the *control characters* concept, the essential idea is that you format some information with a certain "string," called the "format

string." The format string contains some characters which are meaningless as far as formatting goes, and are passed through unchanged; and some characters which will be replaced with certain information. In other words, assume that we have a format string consisting of the following characters: `a b c d e`. `c` is a control character, and represents the information "zzz"; the other characters are not control characters.

`a b c d e` → a b zzz d e

Anyone who has used GNU `date` or BSD `date` will recognize these control characters, though of course in those programs a `%` character would be necessary, as well. `texdate` duplicates the behavior of these programs as closely as my TEX-pertise allows.

```
\initcurrdate
\advancebyweeks{6}
\def\setdateformat{d\ B\ Y}
|d\ B\ Y|:  \printdate\par
\def\setdateformat{Y-m-d}
|Y-m-d|:  \printdate\par
\def\setdateformat{a,\ d\ b\ Y}
|a,\ d\ b\ Y|:  \printdate\par
```

`d\ B\ Y`: 19 January 2019
`Y-m-d`: 2019-01-19
`a,\ d\ b\ Y`: Sat, 19 Jan 2019

We can meddle with this however we like, except that these control characters (the ones that turn into elements of the date) cannot be included literally.

`\nameddateformat` You can also define *named date formats*:

`\nameddateformat {`⟨*name*⟩`} {`⟨*format-string*⟩`}`

Perhaps I want a peculiar date format, with the month, followed by the year, followed by the day of the month, followed by the day of the year in parentheses. `\printfdate` My format string should be `m-Y-d\ (j)`. I'll then want to use the `\printfdate` command with its single argument, which is the name of the date format I want to use.

```
\initcurrdate
\nameddateformat{weird}{m-Y-d\ (j)}
\printfdate{weird}\par
\printdate
```

12-2018-08 (342)
Saturday (Sat), December (Dec) 08, 2018

It's worth nothing that all of the control characters also have a formatted print string that can be called by name. So one could duplicate the above `weird` date format the hard way, by using these each individually:

```
\initcurrdate
\printfdate{m}-\printfdate{Y}-\printfdate{d} (\printfdate{j})
```

12-2018-08 (342)

These seems a bit convoluted, but perhaps you want to wrap it in a macro?

4.3 Number Format

`\texdatenumformat`

Any command which will work on a TEX count register can be inserted into the `\texdatenumformat` command, which will be applied to all the numbers which `texdate` outputs. For example, if you are using the `dozenal` package:

```
\def\texdatenumformat#1{\dozens{#1}}
\initdate{2018}{12}{25}
\printfdate{ISOext}
```

1202-10-21

---

5 Manipulating Dates

`texdate` goes well beyond merely printing and setting dates; you can manipulate them in many ways. The original purpose of the package was to allow LATEX to print calendar sheets and things of that nature without resorting to an external program, or loading some enormous package, so it needed the ability to move forward and backward by given increments. So we have that.

5.1 Moving Dates Forward and Backward

`\advancebydays`
`\advancebyweeks`
`\advancebymonths`
`\regressbydays`
`\regressbyweeks`
`\regressbymonths`

You can advance the date by a certain number of days, weeks, or months. The macros are named, unsurprisingly, `\advancebydays`, `\advancebyweeks`, and `\advancebymonths`, each of which takes one argument, which is the number of that unit you wish to advance the date by. The corresponding commands `\regressbydays`, `\regressbyweeks`, and `\regressbymonths` also exist.

```
\initcurrdate
Current date:  \printdate\par
\advancebydays{8}
8 days later:  \printdate\par
\advancebyweeks{4}
4 weeks later:  \printdate\par
\advancebymonths{4}
4 months later:  \printdate\par
\regressbydays{14}
14 days earlier:  \printdate\par
\regressbyweeks{8}
8 weeks earlier:  \printdate\par
\regressbymonths{2}
2 months earlier:  \printdate\par
```

Current date: Saturday (Sat), December (Dec) 08, 2018
8 days later: Sunday (Sun), December (Dec) 16, 2018
4 weeks later: Sunday (Sun), January (Jan) 13, 2019
4 months later: Monday (Mon), May (May) 13, 2019
14 days earlier: Monday (Mon), April (Apr) 29, 2019
8 weeks earlier: Monday (Mon), March (Mar) 04, 2019
2 months earlier: Friday (Fri), January (Jan) 04, 2019

Note that `\advancebymonths` does not validate the date, so it's possible that you'll end up with something impossible, such as 31 September. It's best to watch the results of this one carefully.

Both the `\advancebys` and the `\regressbys` should be given positive numbers; negative numbers will just confuse them.

5.2 Saving and Restoring Dates

Sometimes you may wish to save a date, change the internal date, use that internal date for a while, then restore the old date. `texdate` makes it possible to save and use as many dates as you want (or, at any rate, as many as TEX will tolerate).

`\savedate` `\savedate` takes a single argument, the *name* you'd like to give your saved date. This can be anything that TEX allows in a control sequence; best to stick with normal, seven-bit ASCII letters. You then access the saved date with `\restoredate` `\restoredate`, which takes that same name as its argument.

```
\initcurrdate
\printdate\par
\savedate{current}
\advancebyweeks{12}
\printdate\par
\savedate{advanced}
```

```
\restoredate{current}
\printdate\par
\advancebydays{3}
\printdate\par
\restoredate{advanced}
\printdate\par
\restoredate{current}
\printdate\par
```

Saturday (Sat), December (Dec) 08, 2018
Saturday (Sat), March (Mar) 02, 2019
Saturday (Sat), December (Dec) 08, 2018
Tuesday (Tue), December (Dec) 11, 2018
Saturday (Sat), March (Mar) 02, 2019
Saturday (Sat), December (Dec) 08, 2018

You can also retrieve your saved date directly; rather than calling `\restoredate`, you can call `\savedate<name>`, without the angle brackets. That's the name that `texdate` uses internally, and calls with `\restoredate` to get your information back.

6 Convenience Macros

`texdate` offers a few macros for tasks which its author anticipates will likely be common. For example, to produce a small monthly calendar, consider using the `\texdcal` macro, which takes two arguments: the year and the month of the calendar you're seeking to create:

```
\begin{center}
\begin{tabular}{cc}
\texdcal{2018}{5} &
\texdcal{2018}{6} \\
\texdcal{2018}{8} &
\texdcal{2018}{9} \\
\end{tabular}
\end{center}
```

May 2018

		01	02	03	04	05
06	07	08	09	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June 2018

					01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

August 2018

			01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September 2018

						01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Notice that `\texdcal` does the right thing when there a month goes into an extra week: it simply prints another week. It also correctly refuses to print the days of a week which do not belong to the requested month.

`\texdcalyear` `\texdcalyear` will produce one of these calendars for an entire year, in three columns; the year chosen is the argument given to the macro. Because the margins of the LaTeX standard classes are much too large (or rather, the paper sizes are much too large; the text blocks are rather nicely proportioned), `\texdcalyear` prints this calendar in a small size, with very small space between columns.

```
\footnotesize%
\begin{center}
\texdcalyear{2018}
\end{center}
```

January 2018

	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

February 2018

				01	02	03
04	05	06	07	08	09	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

March 2018

				01	02	03
04	05	06	07	08	09	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April 2018

01	02	03	04	05	06	07
08	09	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May 2018

		01	02	03	04	05
06	07	08	09	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June 2018

					01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July 2018

01	02	03	04	05	06	07
08	09	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August 2018

			01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September 2018

						01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

October 2018

	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

November 2018

				01	02	03
04	05	06	07	08	09	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

December 2018

						01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Obviously, it uses `\texdcal` internally to do this, so the definition of `\texdcalyear` is much simpler than that of `\texdcal`.

Just as obviously, these yearly calendars could easily be formatted in many different ways; so many, in fact, that attempting to make the macros flexible enough for meaningful customization would be prohibitively difficult. More fruitful results can be obtained by reading the macros themselves (they are truly not very difficult) and customizing them oneself.

7 Language Specification

`texdate` does understand LaTeX language specifications, using Heiko Oberdiek's `iflang` package, which should work for both `babel` and `polyglossia`. Built-in are only English (the default), Spanish, French, and German. However, it's pretty simple to customize the month name and weekday name strings by defining a few commands, so if you need a different language, you just need to redefine a few strings.

Each string begins with the prefix `\texd@`, then the English ordinal string for the order in which it comes, with January being the first month and Sunday being the first weekday; e.g., `\texd@first`. Then comes `sh` if it's an abbreviation; e.g., `\texd@firstsh`. Finally comes the string `mon` if it's a month, or `name` if it's a weekday name. Below is the complete list, for German.

```
\makeatletter
\def\texd@firstmon{Januar}
\def\texd@firstshmon{Jan}
\def\texd@secondmon{Februar}
\def\texd@secondshmon{Feb}
\def\texd@thirdmon{März}
\def\texd@thirdshmon{März}
\def\texd@fourthmon{April}
\def\texd@fourthshmon{Apr}
\def\texd@fifthmon{Mai}
\def\texd@fifthshmon{Mai}
\def\texd@sixthmon{Juni}
\def\texd@sixthshmon{Juni}
\def\texd@seventhmon{Juli}
\def\texd@seventhshmon{Juli}
\def\texd@eighthmon{August}
\def\texd@eighthshmon{Aug}
\def\texd@ninthmon{September}
\def\texd@ninthshmon{Sept}
\def\texd@tenthmon{Oktober}
\def\texd@tenthshmon{Okt}
\def\texd@eleventhmon{November}
\def\texd@eleventhshmon{Nov}
\def\texd@twelfthmon{Dezember}
\def\texd@twelfthshmon{Dez}
\def\texd@firstdayname{Sonntag}
\def\texd@firstdayshname{So}
\def\texd@seconddayname{Montag}
\def\texd@seconddayshname{Mo}
\def\texd@thirddayname{Dienstag}
\def\texd@thirddayshname{Di}
\def\texd@fourthdayname{Mittwoch}
\def\texd@fourthdayshname{Mi}
\def\texd@fifthdayname{Donnerstag}
\def\texd@fifthdayshname{Do}
\def\texd@sixthdayname{Freitag}
\def\texd@sixthdayshname{Fr}
\def\texd@seventhdayname{Samstag}
\def\texd@seventhdayshname{Sa}
\makeatother
```

Doing something like this for your desired language, *after* you've loaded `texdate`, will localize all the strings involved.

8 Plain TEX Usage

I was asked recently, quite unexpectedly, whether `texdate` could be used with plain TEX. My initial thought was an obvious "yes," since it's implemented entirely with TEX primitives; however, the matter wasn't quite that simple. The package file does use some LATEX-specific macros, all related to the packaging itself; and it uses a `padcount` macro which doesn't work with plain TEX. Also, according to LATEX convention, it uses `@` as a letter in control sequences willy-nilly, and TEX balks at such craziness. Finally, a small change in the code (due to deep TEX magic involving `\outer` that is best left unspoken) needed to be made. This done, however, the package *can* (mostly) be used in plain TEX. Here's how.

The following must be included in your document in order to prevent TEX from choking on our LATEX packaging macros:

```
\def\NeedsTeXFormat#1[#2]{}
\def\ProvidesPackage#1[#2]{}
\def\RequirePackage#1{}
\def\AtBeginDocument#1{}
```

This simply defines these macros to do nothing, which is how TEX prefers packaging macros to work. Then, you need to tell TEX that `@` can, in fact, be part of the name of a control sequence:

```
\catcode`@=11
```

This, again, is some deep TEX magic best left undiscussed for the benefit of those not interested. There's plenty of information around if you really want it. Finally, we need to input the packages that `texdate` needs, and tell TEX not to use the `padcount` macro that it doesn't like, by redefining it to simply spit out its own parameter:

```
\input modulus.sty
\input padcount.sty
\input texdate.sty
\def\padnum#1{#1}
```

These things done, `texdate` will work almost entirely with plain TEX, except that (obviously) the padding options won't have any effect. So, if plain TEX is your preference, go for it.

9 Implementation

```
1 \RequirePackage{modulus}%
2 \RequirePackage{padcount}%
3 \RequirePackage{iflang}%
```

```
\newcount\texd@loopi\texd@loopi=0%
\newcount\texd@mon\texd@mon=0%
\newcount\texd@dow\texd@dow=0%
\newcount\texd@dom\texd@dom=0%
\newcount\texd@yr\texd@yr=\year%
\newcount\texd@rdom\texd@rdom=\texd@dom\advance\texd@rdom by1%
\newcount\texd@rmon%
%% taken from dayofweek.tex, by Martin Minow of DEC;
%% included in TeXLive
\newcount\texd@dow% Gets day of the week
\newcount\texd@leap% Leap year fingaler
\newcount\texd@x% Temp register
\newcount\texd@y% Another temp register
\def\texd@nextdow#1#2#3{%
\global\texd@leap=#2%
\global\advance\texd@leap by-14%
\global\divide\texd@leap by12%
\global\advance\texd@leap by#3%
\global\texd@dow=#2%
\global\advance\texd@dow by10%
\global\texd@y=\texd@dow%
\global\divide\texd@y by13%
\global\multiply\texd@y by12%
\global\advance\texd@dow by-\texd@y%
\global\multiply\texd@dow by13%
\global\advance\texd@dow by-1%
\global\divide\texd@dow by5%
\global\advance\texd@dow by#1%
\global\advance\texd@dow by77%
\global\texd@x=\texd@leap%
\global\texd@y=\texd@x%
\global\divide\texd@y by100%
\global\multiply\texd@y by100%
\global\advance\texd@x by-\texd@y%
\global\multiply\texd@x by5%
\global\divide\texd@x by4%
\global\advance\texd@dow by\texd@x%
\global\texd@x=\texd@leap%
\global\divide\texd@x by400%
\global\advance\texd@dow by\texd@x%
\global\texd@x=\texd@leap%
\global\divide\texd@x by100%
\global\multiply\texd@x by2%
\global\advance\texd@dow by-\texd@x%
\global\texd@x=\texd@dow%
\global\divide\texd@x by7%
\global\multiply\texd@x by7%
\global\advance\texd@dow by-\texd@x%
}
%% end taken from dayofweek.tex, by Martin Minow of DEC;
```

```
%% included in TeXLive
\def\texd@leapyear{%
}%
\def\texd@downame{%
  \ifcase\texd@dow
\texd@firstdayname%
\or%
\texd@seconddayname%
\or%
\texd@thirddayname%
\or%
\texd@fourthdayname%
\or%
\texd@fifthdayname%
\or%
\texd@sixthdayname%
\or%
\texd@seventhdayname%
\fi%
}%
\def\texd@shdowname{%
  \ifcase\texd@dow
\texd@firstdayshname%
\or%
\texd@seconddayshname%
\or%
\texd@thirddayshname%
\or%
\texd@fourthdayshname%
\or%
\texd@fifthdayshname%
\or%
\texd@sixthdayshname%
\or%
\texd@seventhdayshname%
\fi%
}%
\def\texd@nextmonth{%
\ifnum\texd@mon<11\global\advance\texd@mon by1\fi%
\ifnum\texd@mon=11\global\texd@mon=0\fi%
}%
\def\texd@lastmonth{%
\ifnum\texd@mon=0%
\global\texd@mon=11%
\global\advance\texd@yr by-1%
\fi%
\ifnum\texd@mon>0\global\advance\texd@mon by-1\fi%
}%
\def\texd@nextdate{%
\ifnum\texd@mon=11%
```

```
\ifnum\texd@dom=30%
\global\advance\texd@yr by1%
\global\texd@mon=0%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=10%
\ifnum\texd@dom=29%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<29%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=9%
\ifnum\texd@dom=30%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=8%
\ifnum\texd@dom=29%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<29%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=7%
\ifnum\texd@dom=30%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=6%
\ifnum\texd@dom=30%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=5%
\ifnum\texd@dom=29%
```

```
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<29%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=4%
\ifnum\texd@dom=30%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=3%
\ifnum\texd@dom=29%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<29%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=2%
\ifnum\texd@dom=30%
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@mon=1%
\ifnum\texd@leapyear=0%
\ifnum\texd@dom=27%
\global\advance\texd@mon by1%
\global\texd@dom=0%
\fi
\ifnum\texd@dom<27%
\global\advance\texd@dom by1%
\fi%
\else\ifnum\texd@leapyear=1%
\ifnum\texd@dom=28%
\global\advance\texd@mon by1%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<28%
\global\advance\texd@dom by1%
\fi%
\fi\fi%
\else\ifnum\texd@mon=0%
\ifnum\texd@dom=30%
```

```
\global\advance\texd@mon by1%%
\global\texd@dom=0%
\fi%
\ifnum\texd@dom<30%
\global\advance\texd@dom by1%
\fi%
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi%
\global\texd@rdom=\texd@dom\global\advance\texd@rdom by1%
\global\texd@rmon=\texd@mon\global\advance\texd@rmon by1%
\texd@setjnum%
\texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
}%
\def\texd@lastdate{%
\global\advance\texd@dom by-1%
\ifnum\texd@dom=0%
\global\advance\texd@mon by-1%
\ifnum\texd@mon=11%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=0%
\global\texd@mon=11%
\global\advance\texd@yr by-1%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=10%
\global\texd@dom=29%
\fi%
\ifnum\texd@mon=9%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=8%
\global\texd@dom=29%
\fi%
\ifnum\texd@mon=7%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=6%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=5%
\global\texd@dom=29%
\fi%
\ifnum\texd@mon=4%
\global\texd@dom=30%
\fi%
\ifnum\texd@mon=3%
\global\texd@dom=29%
\fi%
\ifnum\texd@mon=2%
\global\texd@dom=30%
```

```
\fi%
\ifnum\texd@mon=1%
\ifnum\texd@leapyear=0%
\global\texd@dom=27%
\else\ifnum\texd@leapyear=1%
\global\texd@dom=28%
\fi\fi%
\fi%
\ifnum\texd@mon=0%
\global\texd@dom=30%
\fi%
\fi%
\global\texd@rdom=\texd@dom\global\advance\texd@rdom by1%
\global\texd@rmon=\texd@mon\global\advance\texd@rmon by1%
\texd@setjnum%
\texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
}%
\def\texd@setrdom{\global\texd@rdom=\texd@dom\global\advance\texd@rdom by1}%
\def\texd@setrmon{\global\texd@rmon=\texd@mon\global\advance\texd@rmon by1}%
```

We have to deal with leap years somehow. We have the counter `\texd@leapyear`, which is 0 if it's not a leap year and 1 if it is. Then we have `\texd@isleapyear`, which sets the counter appropriately.

```
\newcount\texd@leapyear\texd@leapyear=0%
\def\texd@isleapyear{%
\global\texd@leapyear=0%
\modulo{\texd@yr}{4}%
\ifnum\remainder=0%
\modulo{\texd@yr}{100}%
\ifnum\remainder=0%
\global\texd@leapyear=0%
\fi\ifnum\remainder>0%
\global\texd@leapyear=1%
\fi%
\fi%
}%
```

Print the month names.

```
\def\texd@monthname{%
\ifnum\texd@mon=0%
\texd@firstmon%
\fi%
\ifnum\texd@mon=1%
\texd@secondmon%
\fi%
\ifnum\texd@mon=2%
\texd@thirdmon%
\fi%
\ifnum\texd@mon=3%
\texd@fourthmon%
\fi%
```

```
\ifnum\texd@mon=4%
\texd@fifthmon%
\fi%
\ifnum\texd@mon=5%
\texd@sixthmon%
\fi%
\ifnum\texd@mon=6%
\texd@seventhmon%
\fi%
\ifnum\texd@mon=7%
\texd@eighthmon%
\fi%
\ifnum\texd@mon=8%
\texd@ninthmon%
\fi%
\ifnum\texd@mon=9%
\texd@tenthmon%
\fi%
\ifnum\texd@mon=10%
\texd@eleventhmon%
\fi%
\ifnum\texd@mon=11%
\texd@twelfthmon%
\fi%
}%
\def\texd@shmonthname{%
\ifnum\texd@mon=0%
\texd@firstshmon%
\fi%
\ifnum\texd@mon=1%
\texd@secondshmon%
\fi%
\ifnum\texd@mon=2%
\texd@thirdshmon%
\fi%
\ifnum\texd@mon=3%
\texd@fourthshmon%
\fi%
\ifnum\texd@mon=4%
\texd@fifthshmon%
\fi%
\ifnum\texd@mon=5%
\texd@sixthshmon%
\fi%
\ifnum\texd@mon=6%
\texd@seventhshmon%
\fi%
\ifnum\texd@mon=7%
\texd@eighthshmon%
\fi%
```

```
\ifnum\texd@mon=8%
\texd@ninthshmon%
\fi%
\ifnum\texd@mon=9%
\texd@tenthshmon%
\fi%
\ifnum\texd@mon=10%
\texd@eleventhshmon%
\fi%
\ifnum\texd@mon=11%
\texd@twelfthshmon%
\fi%
}%
```

Here we define the `\advancebys`, so that you can add move the internal date forward by a given number of units. Does *not* print the date.

```
\def\advancebydays#1{%
\texd@loopi=0%
\loop%
\ifnum\texd@loopi<#1%
\texd@nextdate%
\advance\texd@loopi by1%
\repeat%
}%
\def\regressbydays#1{%
\texd@loopi=0%
\loop%
\ifnum\texd@loopi<#1%
\texd@lastdate%
\advance\texd@loopi by1%
\repeat%
}%
\newcount\texd@loopj%
\def\advancebyweeks#1{%
\texd@loopi=0%
\texd@loopj=#1%
\multiply\texd@loopj by7%
\loop%
\ifnum\texd@loopi<\texd@loopj%
\texd@nextdate%
\advance\texd@loopi by1%
\repeat%
}%
\def\regressbyweeks#1{%
\texd@loopi=0%
\texd@loopj=#1%
\multiply\texd@loopj by7%
\loop%
\ifnum\texd@loopi<\texd@loopj%
\texd@lastdate%
```

```
\advance\texd@loopi by1%
\repeat%
}%
\def\advancebymonths#1{%
\texd@loopi=0%
\loop%
\ifnum\texd@loopi<#1%
\texd@nextmonth%
\advance\texd@loopi by1%
\repeat%
\texd@setrmon%
\initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@rdom}%
}%
\def\regressbymonths#1{%
\texd@loopi=0%
\loop%
\ifnum\texd@loopi<#1%
\texd@lastmonth%
\advance\texd@loopi by1%
\repeat%
\texd@setrmon%
\initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@rdom}%
}%
```

Print the date, either with the default format or a named format.

```
\def\printdate{%
\texd@dateformat%
}%
\def\printfdate#1{%
\texd@formatdateformat{#1}%
}%
```

This defines the date format. We need some helper macros to flip through each character one at a time.

```
\def\texd@expandloop#1{%
\texd@xloop#1\relax
}
\def\texdatenumformat#1{#1}
\def\texd@xloop#1{%
\ifx\relax#1%
\else%
\ifx#1d%
\setpadnum{2}\setpadchar{0}%
\padnum{\texdatenumformat{\the\texd@rdom}}%
\else\ifx#1e%
\setpadnum{2}\setpadchar{\hskip1ex}%
\padnum{\texdatenumformat{\the\texd@rdom}}%
\else\ifx#1a%
\texd@shdowname%
\else\ifx#1A%
\texd@downame%
```

```
\else\ifx#1b%
\texd@shmonthname%
\else\ifx#1h%
\texd@shmonthname%
\else\ifx#1B%
\texd@monthname%
\else\ifx#1w%
\the\texd@dow%
\else\ifx#1u%
\ifnum\texd@dow>0\leavevmode\texdatenumformat{\the\texd@dow}\fi%
\ifnum\texd@dow=0 7\fi%
\else\ifx#1Y%
\setpadnum{2}\setpadchar{0}%
\padnum{\texdatenumformat{\the\texd@yr}}%
\else\ifx#1m%
\texd@setrmon%
\setpadnum{2}\setpadchar{0}%
\leavevmode\padnum{\texdatenumformat{\the\texd@rmon}}%
\else\ifx#1j%
\texd@setjnum%
\setpadnum{3}\setpadchar{0}%
\leavevmode\padnum{\texdatenumformat{\the\texd@jnum}}%
\else\ifx#1C%
\quotient{\texd@yr}{100}%
\leavevmode\texdatenumformat{\the\intquotient}%
\else\ifx#1y%
\modulo{\texd@yr}{100}%
\leavevmode\texdatenumformat{\the\remainder}%
\else\ifx#1U%
\quotient{\texd@jnum}{7}
\setpadnum{2}\setpadchar{0}%
\leavevmode\padnum{\texdatenumformat{\the\intquotient}}%
\else\ifx#1V%
\quotient{\texd@jnum}{7}
\ifnum\texd@dow>0\advance\intquotient by1\fi%
\setpadnum{2}\setpadchar{0}%
\leavevmode\padnum{\texdatenumformat{\the\intquotient}}%
\else\ifx#1W%
\quotient{\texd@jnum}{7}
\ifnum\texd@dow=0\advance\intquotient by-1\fi%
\setpadnum{2}\setpadchar{0}%
\leavevmode\padnum{\texdatenumformat{\the\intquotient}}%
\else%
#1%
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi%
\expandafter\texd@xloop%
\fi%
}%
\def\texd@dateformat{%
\expandafter\expandafter\texd@expandloop{\setdateformat}%
```

```
}%
\def\texd@formatdateformat#1{%
\expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\texd@expandloop{\csnam
}%
\def\setdateformat{A{ }(a),\ B\ (b){ }d,\ Y}
\def\nameddateformat#1#2{%
\expandafter\def\csname texd@df#1\endcsname{#2}%
}%
\nameddateformat{american}{B\ d,\ Y}
\nameddateformat{shamerican}{m/d/Y}
\nameddateformat{ISO}{Ymd}
\nameddateformat{ISOext}{Y-m-d}
\nameddateformat{shbritish}{d/m/Y}
\nameddateformat{shbritishdots}{d.m.Y}
\nameddateformat{british}{d\ B\ Y}
\nameddateformat{d}{d}
\nameddateformat{e}{e}
\nameddateformat{B}{B}
\nameddateformat{b}{b}
\nameddateformat{h}{h}
\nameddateformat{m}{m}
\nameddateformat{A}{A}
\nameddateformat{a}{a}
\nameddateformat{w}{w}
\nameddateformat{u}{u}
\nameddateformat{Y}{Y}
\nameddateformat{j}{j}
\nameddateformat{C}{C}
\nameddateformat{y}{y}
\nameddateformat{U}{U}
\nameddateformat{V}{V}
\nameddateformat{W}{W}
```

Initialize the date to the current date, or to an arbitrary date, entered in the order year, month, and day of month.

```
\def\initcurrdate{%
\global\texd@mon=\month%
\global\advance\texd@mon by-1%
\global\texd@dom=\day%
\global\advance\texd@dom by-1%
\global\texd@yr=\year%
\texd@isleapyear%
\texd@setrdom%
\texd@setrmon%
\texd@setjnum%
\texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
}%
\def\initdate#1#2#3{%
\global\texd@yr=#1%
\global\texd@mon=#2%
```

```
\global\advance\texd@mon by-1%
\global\texd@dom=#3%
\global\advance\texd@dom by-1%
\global\texd@setrdom%
\global\texd@setrmon%
\texd@setjnum%
\texd@isleapyear%
\texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
}%
```

Now we define the macros for saving and restoring dates.

```
\def\savedate#1{%
\expandafter\edef\csname savedate#1\endcsname{\initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd
}%
\def\restoredate#1{%
\csname savedate#1\endcsname%
}%
```

Convenience macros. First, `\texdcal`.

```
\newcount\texd@looptmp\texd@looptmp=0%
\def\texdcal#1#2{%
\global\texd@mon=#2%
\global\advance\texd@mon by-1%
\global\texd@yr=#1%
\global\texd@dom=0%
\texd@setrmon\texd@setrdom%
\initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@rdom}%
\def\setdateformat{B\ Y}%
\begin{tabular}{rrrrrrr}
\multicolumn{7}{c}{\printdate} \\
\loop\ifnum\texd@dow>0\texd@lastdate\repeat%
\def\setdateformat{d}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi\\
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
```

```
\def\setdateformat{d}\advancebydays{1}\printdate \\
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate \\
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate &
\def\setdateformat{d}\advancebydays{1}\printdate \\
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi\\
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
\def\setdateformat{d}\advancebydays{1}%
\ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi\\
\end{tabular}
}%
\def\texdcalyear#1{%
\texd@yr=#1%
\texd@mon=0%
\texd@dom=0%
\texd@setrmon%
```

```
\texd@setrdom%
{\tabcolsep=3pt%
\begin{tabular}{ccc}
\texdcal{#1}{1} & \texdcal{#1}{2} & \texdcal{#1}{3} \\
\texdcal{#1}{4} & \texdcal{#1}{5} & \texdcal{#1}{6} \\
\texdcal{#1}{7} & \texdcal{#1}{8} & \texdcal{#1}{9} \\
\texdcal{#1}{10} & \texdcal{#1}{11} & \texdcal{#1}{12} \\
\end{tabular}
}%
}%
```

Calculate the day of the year (`%j`).

```
\newcount\texd@jnum\texd@jnum=0%
\def\texd@setjnum{%
\texd@jnum=0%
\ifnum\texd@mon>0\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>1%
\global\advance\texd@jnum by28%
\fi%
\ifnum\texd@mon>2\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>3\global\advance\texd@jnum by30\fi%
\ifnum\texd@mon>4\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>5\global\advance\texd@jnum by30\fi%
\ifnum\texd@mon>6\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>7\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>8\global\advance\texd@jnum by30\fi%
\ifnum\texd@mon>9\global\advance\texd@jnum by31\fi%
\ifnum\texd@mon>10\global\advance\texd@jnum by30\fi%
\global\advance\texd@jnum by\the\texd@dom%
\global\advance\texd@jnum by1%
}%
```

Language strings. I've only got English here right now, but additiona languages would be trivial to add, either in a particular document, or in a separate package.

```
\def\texd@firstmon{January}%
\def\texd@firstshmon{Jan}%
\def\texd@secondmon{February}%
\def\texd@secondshmon{Feb}%
\def\texd@thirdmon{March}%
\def\texd@thirdshmon{Mar}%
\def\texd@fourthmon{April}%
\def\texd@fourthshmon{Apr}%
\def\texd@fifthmon{May}%
\def\texd@fifthshmon{May}%
\def\texd@sixthmon{June}%
\def\texd@sixthshmon{Jun}%
\def\texd@seventhmon{July}%
\def\texd@seventhshmon{Jul}%
\def\texd@eighthmon{August}%
\def\texd@eighthshmon{Aug}%
\def\texd@ninthmon{September}%
```

```
\def\texd@ninthshmon{Sep}%
\def\texd@tenthmon{October}%
\def\texd@tenthshmon{Oct}%
\def\texd@eleventhmon{November}%
\def\texd@eleventhshmon{Nov}%
\def\texd@twelfthmon{December}%
\def\texd@twelfthshmon{Dec}%
\def\texd@firstdayname{Sunday}%
\def\texd@firstdayshname{Sun}%
\def\texd@seconddayname{Monday}%
\def\texd@seconddayshname{Mon}%
\def\texd@thirddayname{Tuesday}%
\def\texd@thirddayshname{Tue}%
\def\texd@fourthdayname{Wednesday}%
\def\texd@fourthdayshname{Wed}%
\def\texd@fifthdayname{Thursday}%
\def\texd@fifthdayshname{Thu}%
\def\texd@sixthdayname{Friday}%
\def\texd@sixthdayshname{Fri}%
\def\texd@seventhdayname{Saturday}%
\def\texd@seventhdayshname{Sat}%
\AtBeginDocument{%
\IfLanguageName{spanish}{%
\def\texd@firstmon{enero}%
\def\texd@firstshmon{ene}%
\def\texd@secondmon{febrero}%
\def\texd@secondshmon{feb}%
\def\texd@thirdmon{marzo}%
\def\texd@thirdshmon{mar}%
\def\texd@fourthmon{abril}%
\def\texd@fourthshmon{abr}%
\def\texd@fifthmon{mayo}%
\def\texd@fifthshmon{may}%
\def\texd@sixthmon{junio}%
\def\texd@sixthshmon{jun}%
\def\texd@seventhmon{julio}%
\def\texd@seventhshmon{jul}%
\def\texd@eighthmon{agosto}%
\def\texd@eighthshmon{ago}%
\def\texd@ninthmon{septiembre}%
\def\texd@ninthshmon{sep}%
\def\texd@tenthmon{octubre}%
\def\texd@tenthshmon{oct}%
\def\texd@eleventhmon{noviembre}%
\def\texd@eleventhshmon{nov}%
\def\texd@twelfthmon{diciembre}%
\def\texd@twelfthshmon{dic}%
\def\texd@firstdayname{domingo}%
\def\texd@firstdayshname{dom}%
\def\texd@seconddayname{lunes}%
```

```
\def\texd@seconddayshname{lun}%
\def\texd@thirddayname{martes}%
\def\texd@thirddayshname{mar}%
\def\texd@fourthdayname{miercoles}%
\def\texd@fourthdayshname{mie}%
\def\texd@fifthdayname{jueves}%
\def\texd@fifthdayshname{jue}%
\def\texd@sixthdayname{viernes}%
\def\texd@sixthdayshname{vie}%
\def\texd@seventhdayname{sabado}%
\def\texd@seventhdayshname{sab}%
}{}%
\IfLanguageName{french}{%
\def\texd@firstmon{janvier}%
\def\texd@firstshmon{janv}%
\def\texd@secondmon{février}%
\def\texd@secondshmon{févr}%
\def\texd@thirdmon{mars}%
\def\texd@thirdshmon{mars}%
\def\texd@fourthmon{avril}%
\def\texd@fourthshmon{avr}%
\def\texd@fifthmon{mai}%
\def\texd@fifthshmon{mai}%
\def\texd@sixthmon{juin}%
\def\texd@sixthshmon{juin}%
\def\texd@seventhmon{juil}%
\def\texd@seventhshmon{juil}%
\def\texd@eighthmon{août}%
\def\texd@eighthshmon{août}%
\def\texd@ninthmon{septembre}%
\def\texd@ninthshmon{sept}%
\def\texd@tenthmon{octobre}%
\def\texd@tenthshmon{oct}%
\def\texd@eleventhmon{novembre}%
\def\texd@eleventhshmon{nov}%
\def\texd@twelfthmon{décembre}%
\def\texd@twelfthshmon{déc}%
\def\texd@firstdayname{dimanche}%
\def\texd@firstdayshname{dim}%
\def\texd@seconddayname{lundi}%
\def\texd@seconddayshname{lun}%
\def\texd@thirddayname{mardi}%
\def\texd@thirddayshname{mar}%
\def\texd@fourthdayname{mercredi}%
\def\texd@fourthdayshname{mer}%
\def\texd@fifthdayname{jeudi}%
\def\texd@fifthdayshname{jeu}%
\def\texd@sixthdayname{vendredi}%
\def\texd@sixthdayshname{ven}%
\def\texd@seventhdayname{samedi}%
```

```
\def\texd@seventhdayshname{sam}%
}{}%
\IfLanguageName{german}{%
\def\texd@firstmon{Januar}%
\def\texd@firstshmon{Jan}%
\def\texd@secondmon{Februar}%
\def\texd@secondshmon{Feb}%
\def\texd@thirdmon{März}%
\def\texd@thirdshmon{März}%
\def\texd@fourthmon{April}%
\def\texd@fourthshmon{Apr}%
\def\texd@fifthmon{Mai}%
\def\texd@fifthshmon{Mai}%
\def\texd@sixthmon{Juni}%
\def\texd@sixthshmon{Juni}%
\def\texd@seventhmon{Juli}%
\def\texd@seventhshmon{Juli}%
\def\texd@eighthmon{August}%
\def\texd@eighthshmon{Aug}%
\def\texd@ninthmon{September}%
\def\texd@ninthshmon{Sept}%
\def\texd@tenthmon{Oktober}%
\def\texd@tenthshmon{Okt}%
\def\texd@eleventhmon{November}%
\def\texd@eleventhshmon{Nov}%
\def\texd@twelfthmon{Dezember}%
\def\texd@twelfthshmon{Dez}%
\def\texd@firstdayname{Sonntag}%
\def\texd@firstdayshname{So}%
\def\texd@seconddayname{Montag}%
\def\texd@seconddayshname{Mo}%
\def\texd@thirddayname{Dienstag}%
\def\texd@thirddayshname{Di}%
\def\texd@fourthdayname{Mittwoch}%
\def\texd@fourthdayshname{Mi}%
\def\texd@fifthdayname{Donnerstag}%
\def\texd@fifthdayshname{Do}%
\def\texd@sixthdayname{Freitag}%
\def\texd@sixthdayshname{Fr}%
\def\texd@seventhdayname{Samstag}%
\def\texd@seventhdayshname{Sa}%
}{}%
}%
```

Happy TEXing!