

# The `texdate` Package, v1.0

Donald P. Goodman III

May 30, 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

<b>1</b>	<b>Introduction: The State of the <code>T<sub>E</sub>X</code></b>	<b>1</b>
<b>2</b>	<b>Printing and Setting the Date</b>	<b>2</b>
<b>3</b>	<b>Date Formats</b>	<b>3</b>
3.1	Preset Formats . . . . .	3
3.2	Custom Date Formats . . . . .	4
3.3	Number Format . . . . .	7
<b>4</b>	<b>Manipulating Dates</b>	<b>7</b>
4.1	Moving Dates Forward and Backward . . . . .	7
4.2	Saving and Restoring Dates . . . . .	8
<b>5</b>	<b>Convenience Macros</b>	<b>9</b>
<b>6</b>	<b>Language Specification</b>	<b>7</b>
<b>7</b>	<b>Implementation</b>	<b>10</b>

## 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 “2018-5-30” (which is the date on

which this document was compiled). L<sup>A</sup>T<sub>E</sub>X also has `\today`, which will produce the current date in the default American style: “May 30, 2018”. But that’s really about all there is.

Many packages have attempted to fill up this gap, some with excellent success; `datetime2` certainly 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 T<sub>E</sub>X-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 T<sub>E</sub>X for something for which T<sub>E</sub>X 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, ɀ, Ɂ, 10, 11, 12 . . . It uses the `dozenal` L<sup>A</sup>T<sub>E</sub>X package to do this. For more information, see <http://www.dozenal.org>.

## 2 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 can print that date with `\printdate`:

`\printdate`

---

```
\printdate
```

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

---

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

`\initcurrrdate`

---

```
\initcurrrdate
\printdate
```

```
Wednesday (Wed), May (May) 30, 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.

### 3 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.

#### 3.1 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 3.2, on page 4).

```
\printfdate{ISO}
```

`\printfdate{ISO}` will print the current date in the default ISO 8601 format, which is *yyymmdd*. In `texdate`’s formatting strings, this is `Ymd`; you’ll learn more about these in Section 3.2. 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
```

---

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 `\printdate{british}` and `\printdate{shbritish}`, along with alternate form `\printdate{shbritishdots}`,

```
\printdate{british}
\printdate{shbritish}
\printdate{shbritishdots}
```

---

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

\printdate{shbritish}

\printdate{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.

## 3.2 Custom Date Formats

All the custom formats described in Section 3.1 and printed with `\printdate` 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 `\printdate`.

```
\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}`.

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.

<i>Let.</i>	<i>Result</i>	<i>Ex.</i>
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 necessary	22

Table 1: Control codes for date formats

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 T<sub>E</sub>X-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: 11 July 2018
Y-m-d: 2018-07-11
a,\ d\ b\ Y: Wed, 11 Jul 2018
```

---

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. 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.

`\printfdate`

---

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

05-2018-30 (150)
Wednesday (Wed), May (May) 30, 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})

05-2018-30 (150)
```

---

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

### 3.3 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}
\printfddate{ISOext}
```

1202-10-21

---

## 4 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.

### 4.1 Moving Dates Forward and Backward

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: Wednesday (Wed), May (May) 30, 2018  
 8 days later: Thursday (Thu), June (Jun) 07, 2018  
 4 weeks later: Thursday (Thu), July (Jul) 05, 2018  
 4 months later: Monday (Mon), November (Nov) 05, 2018  
 14 days earlier: Monday (Mon), October (Oct) 22, 2018  
 8 weeks earlier: Monday (Mon), August (Aug) 27, 2018  
 2 months earlier: Wednesday (Wed), June (Jun) 27, 2018

---

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.

## 4.2 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 T<sub>E</sub>X will tolerate).

<code>\savedate</code>	<code>\savedate</code> takes a single argument, the <i>name</i> you'd like to give your saved date. This can be anything that T <sub>E</sub> X allows in a control sequence; best to stick with normal, seven-bit ASCII letters. You then access the saved date with
<code>\restoredate</code>	<code>\restoredate</code> , 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

```

Wednesday (Wed), May (May) 30, 2018  
 Wednesday (Wed), August (Aug) 22, 2018  
 Wednesday (Wed), May (May) 30, 2018  
 Saturday (Sat), June (Jun) 02, 2018  
 Wednesday (Wed), August (Aug) 22, 2018  
 Wednesday (Wed), May (May) 30, 2018



---

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

## 5 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							June 2018						
		01	02	03	04	05					01	02	
06	07	08	09	10	11	12	03	04	05	06	07	08	09
13	14	15	16	17	18	19	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23
27	28	29	30	31			24	25	26	27	28	29	30

  

August 2018							September 2018						
			01	02	03	04							01
05	06	07	08	09	10	11	02	03	04	05	06	07	08
12	13	14	15	16	17	18	09	10	11	12	13	14	15
19	20	21	22	23	24	25	16	17	18	19	20	21	22
26	27	28	29	30	31		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  $\text{\LaTeX}$  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							February 2018							March 2018						
01	02	03	04	05	06				01	02	03				01	02	03			
07	08	09	10	11	12	13	04	05	06	07	08	09	10	04	05	06	07	08	09	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24	18	19	20	21	22	23	24
28	29	30	31				25	26	27	28				25	26	27	28	29	30	31
April 2018							May 2018							June 2018						
01	02	03	04	05	06	07			01	02	03	04	05					01	02	
08	09	10	11	12	13	14	06	07	08	09	10	11	12	03	04	05	06	07	08	09
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23
29	30						27	28	29	30	31			24	25	26	27	28	29	30
July 2018							August 2018							September 2018						
01	02	03	04	05	06	07			01	02	03	04								01
08	09	10	11	12	13	14	05	06	07	08	09	10	11	02	03	04	05	06	07	08
15	16	17	18	19	20	21	12	13	14	15	16	17	18	09	10	11	12	13	14	15
22	23	24	25	26	27	28	19	20	21	22	23	24	25	16	17	18	19	20	21	22
29	30	31					26	27	28	29	30	31		23	24	25	26	27	28	29
October 2018							November 2018							December 2018						
01	02	03	04	05	06				01	02	03									01
07	08	09	10	11	12	13	04	05	06	07	08	09	10	02	03	04	05	06	07	08
14	15	16	17	18	19	20	11	12	13	14	15	16	17	09	10	11	12	13	14	15
21	22	23	24	25	26	27	18	19	20	21	22	23	24	16	17	18	19	20	21	22
28	29	30	31				25	26	27	28	29	30		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.

## 6 Language Specification

`texdate` does understand  $\text{\LaTeX}$  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.

```

\tmakeatletter
\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.

## 7 Implementation

```

1 \RequirePackage{modulus}%
2 \RequirePackage{padcount}%
3 \RequirePackage{iflang}%
4 \newcount\texd@loopi\texd@loopi=0%
5 \newcount\texd@mon\texd@mon=0%
6 \newcount\texd@dow\texd@dow=0%
7 \newcount\texd@dom\texd@dom=0%
8 \newcount\texd@yr\texd@yr=\year%
9 \newcount\texd@rdom\texd@rdom=\texd@dom\advance\texd@rdom by1%
10 \newcount\texd@rmon%
11 %% taken from dayofweek.tex, by Martin Minow of DEC;
12 %% included in TeXLive
13 \def\texd@nextdow#1#2#3{%
14 \newcount\texd@dow% Gets day of the week
15 \newcount\texd@leap% Leap year fingle
16 \newcount\texd@x% Temp register
17 \newcount\texd@y% Another temp register
18 \global\texd@leap=#2%
19 \global\advance\texd@leap by-14%
20 \global\divide\texd@leap by12%
21 \global\advance\texd@leap by#3%
22 \global\texd@dow=#2%
23 \global\advance\texd@dow by10%
24 \global\texd@y=\texd@dow%
25 \global\divide\texd@y by13%
26 \global\multiply\texd@y by12%
27 \global\advance\texd@dow by-\texd@y%
28 \global\multiply\texd@dow by13%
29 \global\advance\texd@dow by-1%
30 \global\divide\texd@dow by5%
31 \global\advance\texd@dow by#1%
32 \global\advance\texd@dow by77%
33 \global\texd@x=\texd@leap%
34 \global\texd@y=\texd@x%
35 \global\divide\texd@y by100%
36 \global\multiply\texd@y by100%
37 \global\advance\texd@x by-\texd@y%
38 \global\multiply\texd@x by5%
39 \global\divide\texd@x by4%

```

```

40 \global\advance\texd@dow by\texd@x%
41 \global\texd@x=\texd@leap%
42 \global\divide\texd@x by400%
43 \global\advance\texd@dow by\texd@x%
44 \global\texd@x=\texd@leap%
45 \global\divide\texd@x by100%
46 \global\multiply\texd@x by2%
47 \global\advance\texd@dow by-\texd@x%
48 \global\texd@x=\texd@dow%
49 \global\divide\texd@x by7%
50 \global\multiply\texd@x by7%
51 \global\advance\texd@dow by-\texd@x%
52 }
53 %% end taken from dayofweek.tex, by Martin Minow of DEC;
54 %% included in TeXLive
55 \def\texd@leapyear{%
56 }%
57 \def\texd@downname{%
58   \ifcase\texd@dow
59 \texd@firstdayname%
60 \or%
61 \texd@seconddayname%
62 \or%
63 \texd@thirddayname%
64 \or%
65 \texd@fourthdayname%
66 \or%
67 \texd@fifthdayname%
68 \or%
69 \texd@sixthdayname%
70 \or%
71 \texd@seventhdayname%
72 \fi%
73 }%
74 \def\texd@shdownname{%
75   \ifcase\texd@dow
76 \texd@firstdayshname%
77 \or%
78 \texd@seconddayshname%
79 \or%
80 \texd@thirddayshname%
81 \or%
82 \texd@fourthdayshname%
83 \or%
84 \texd@fifthdayshname%
85 \or%
86 \texd@sixthdayshname%
87 \or%
88 \texd@seventhdayshname%
89 \fi%

```

```

90 }%
91 \def\texd@nextmonth{%
92 \ifnum\texd@mon<11\global\advance\texd@mon by1\fi%
93 \ifnum\texd@mon=11\global\texd@mon=0\fi%
94 }%
95 \def\texd@lastmonth{%
96 \ifnum\texd@mon=0%
97 \global\texd@mon=11%
98 \global\advance\texd@yr by-1%
99 \fi%
100 \ifnum\texd@mon>0\global\advance\texd@mon by-1\fi%
101 }%
102 \def\texd@nextdate{%
103 \ifnum\texd@mon=11%
104 \ifnum\texd@dom=30%
105 \global\advance\texd@yr by1%
106 \global\texd@mon=0%
107 \global\texd@dom=0%
108 \fi%
109 \ifnum\texd@dom<30%
110 \global\advance\texd@dom by1%
111 \fi%
112 \else\ifnum\texd@mon=10%
113 \ifnum\texd@dom=29%
114 \global\advance\texd@mon by1%%
115 \global\texd@dom=0%
116 \fi%
117 \ifnum\texd@dom<29%
118 \global\advance\texd@dom by1%
119 \fi%
120 \else\ifnum\texd@mon=9%
121 \ifnum\texd@dom=30%
122 \global\advance\texd@mon by1%%
123 \global\texd@dom=0%
124 \fi%
125 \ifnum\texd@dom<30%
126 \global\advance\texd@dom by1%
127 \fi%
128 \else\ifnum\texd@mon=8%
129 \ifnum\texd@dom=29%
130 \global\advance\texd@mon by1%%
131 \global\texd@dom=0%
132 \fi%
133 \ifnum\texd@dom<29%
134 \global\advance\texd@dom by1%
135 \fi%
136 \else\ifnum\texd@mon=7%
137 \ifnum\texd@dom=30%
138 \global\advance\texd@mon by1%%
139 \global\texd@dom=0%

```

```

140 \fi%
141 \ifnum\texd@dom<30%
142 \global\advance\texd@dom by1%
143 \fi%
144 \else\ifnum\texd@mon=6%
145 \ifnum\texd@dom=30%
146 \global\advance\texd@mon by1%%
147 \global\texd@dom=0%
148 \fi%
149 \ifnum\texd@dom<30%
150 \global\advance\texd@dom by1%
151 \fi%
152 \else\ifnum\texd@mon=5%
153 \ifnum\texd@dom=29%
154 \global\advance\texd@mon by1%%
155 \global\texd@dom=0%
156 \fi%
157 \ifnum\texd@dom<29%
158 \global\advance\texd@dom by1%
159 \fi%
160 \else\ifnum\texd@mon=4%
161 \ifnum\texd@dom=30%
162 \global\advance\texd@mon by1%%
163 \global\texd@dom=0%
164 \fi%
165 \ifnum\texd@dom<30%
166 \global\advance\texd@dom by1%
167 \fi%
168 \else\ifnum\texd@mon=3%
169 \ifnum\texd@dom=29%
170 \global\advance\texd@mon by1%%
171 \global\texd@dom=0%
172 \fi%
173 \ifnum\texd@dom<29%
174 \global\advance\texd@dom by1%
175 \fi%
176 \else\ifnum\texd@mon=2%
177 \ifnum\texd@dom=30%
178 \global\advance\texd@mon by1%%
179 \global\texd@dom=0%
180 \fi
181 \ifnum\texd@dom<30%
182 \global\advance\texd@dom by1%
183 \fi%
184 \else\ifnum\texd@mon=1%
185 \ifnum\texd@leapyear=0%
186 \ifnum\texd@dom=27%
187 \global\advance\texd@mon by1%
188 \global\texd@dom=0%
189 \fi

```

```

190 \ifnum\texd@dom<27%
191 \global\advance\texd@dom by1%
192 \fi%
193 \else\ifnum\texd@leapyear=1%
194 \ifnum\texd@dom=28%
195 \global\advance\texd@mon by1%
196 \global\texd@dom=0%
197 \fi%
198 \ifnum\texd@dom<28%
199 \global\advance\texd@dom by1%
200 \fi%
201 \fi\fi%
202 \else\ifnum\texd@mon=0%
203 \ifnum\texd@dom=30%
204 \global\advance\texd@mon by1%%
205 \global\texd@dom=0%
206 \fi%
207 \ifnum\texd@dom<30%
208 \global\advance\texd@dom by1%
209 \fi%
210 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi%
211 \global\texd@rdom=\texd@dom\global\advance\texd@rdom by1%
212 \global\texd@rmon=\texd@mon\global\advance\texd@rmon by1%
213 \texd@setjnum%
214 \texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
215 }%
216 \def\texd@lastdate{%
217 \global\advance\texd@dom by-1%
218 \ifnum\texd@dom=0%
219 \global\advance\texd@mon by-1%
220 \ifnum\texd@mon=11%
221 \global\texd@dom=30%
222 \fi%
223 \ifnum\texd@mon=0%
224 \global\texd@mon=11%
225 \global\advance\texd@yr by-1%
226 \global\texd@dom=30%
227 \fi%
228 \ifnum\texd@mon=10%
229 \global\texd@dom=29%
230 \fi%
231 \ifnum\texd@mon=9%
232 \global\texd@dom=30%
233 \fi%
234 \ifnum\texd@mon=8%
235 \global\texd@dom=29%
236 \fi%
237 \ifnum\texd@mon=7%
238 \global\texd@dom=30%
239 \fi%

```



```

240 \ifnum\texd@mon=6%
241 \global\texd@dom=30%
242 \fi%
243 \ifnum\texd@mon=5%
244 \global\texd@dom=29%
245 \fi%
246 \ifnum\texd@mon=4%
247 \global\texd@dom=30%
248 \fi%
249 \ifnum\texd@mon=3%
250 \global\texd@dom=29%
251 \fi%
252 \ifnum\texd@mon=2%
253 \global\texd@dom=30%
254 \fi%
255 \ifnum\texd@mon=1%
256 \ifnum\texd@leapyear=0%
257 \global\texd@dom=27%
258 \else\ifnum\texd@leapyear=1%
259 \global\texd@dom=28%
260 \fi\fi%
261 \fi%
262 \ifnum\texd@mon=0%
263 \global\texd@dom=30%
264 \fi%
265 \fi%
266 \global\texd@rdom=\texd@dom\global\advance\texd@rdom by1%
267 \global\texd@rmon=\texd@mon\global\advance\texd@rmon by1%
268 \texd@setjnum%
269 \texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
270 }%
271 \def\texd@setrdom{\global\texd@rdom=\texd@dom\global\advance\texd@rdom by1}%
272 \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.
273 \newcount\texd@leapyear\texd@leapyear=0%
274 \def\texd@isleapyear{%
275 \global\texd@leapyear=0%
276 \modulo{\texd@yr}{4}%
277 \ifnum\remainder=0%
278 \modulo{\texd@yr}{100}%
279 \ifnum\remainder=0%
280 \global\texd@leapyear=0%
281 \fi\ifnum\remainder>0%
282 \global\texd@leapyear=1%
283 \fi%
284 \fi%
285 }%

```

Print the month names.

```
286 \def\texd@monthname{%
287 \ifnum\texd@mon=0%
288 \texd@firstmon%
289 \fi%
290 \ifnum\texd@mon=1%
291 \texd@secondmon%
292 \fi%
293 \ifnum\texd@mon=2%
294 \texd@thirdmon%
295 \fi%
296 \ifnum\texd@mon=3%
297 \texd@fourthmon%
298 \fi%
299 \ifnum\texd@mon=4%
300 \texd@fifthmon%
301 \fi%
302 \ifnum\texd@mon=5%
303 \texd@sixthmon%
304 \fi%
305 \ifnum\texd@mon=6%
306 \texd@seventhmon%
307 \fi%
308 \ifnum\texd@mon=7%
309 \texd@eighthmon%
310 \fi%
311 \ifnum\texd@mon=8%
312 \texd@ninthmon%
313 \fi%
314 \ifnum\texd@mon=9%
315 \texd@tenthmon%
316 \fi%
317 \ifnum\texd@mon=10%
318 \texd@eleventhmon%
319 \fi%
320 \ifnum\texd@mon=11%
321 \texd@twelfthmon%
322 \fi%
323 }%
324 \def\texd@shmonthname{%
325 \ifnum\texd@mon=0%
326 \texd@firstshmon%
327 \fi%
328 \ifnum\texd@mon=1%
329 \texd@secondshmon%
330 \fi%
331 \ifnum\texd@mon=2%
332 \texd@thirdshmon%
333 \fi%
334 \ifnum\texd@mon=3%
```

```

335 \texd@fourthshmon%
336 \fi%
337 \ifnum\texd@mon=4%
338 \texd@fifthshmon%
339 \fi%
340 \ifnum\texd@mon=5%
341 \texd@sixthshmon%
342 \fi%
343 \ifnum\texd@mon=6%
344 \texd@seventhshmon%
345 \fi%
346 \ifnum\texd@mon=7%
347 \texd@eighthshmon%
348 \fi%
349 \ifnum\texd@mon=8%
350 \texd@ninthshmon%
351 \fi%
352 \ifnum\texd@mon=9%
353 \texd@tenthshmon%
354 \fi%
355 \ifnum\texd@mon=10%
356 \texd@eleventhshmon%
357 \fi%
358 \ifnum\texd@mon=11%
359 \texd@twelfthshmon%
360 \fi%
361 }%

```

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.

```

362 \def\advancebydays#1{%
363 \texd@loopi=0%
364 \loop%
365 \ifnum\texd@loopi<#1%
366 \texd@nextdate%
367 \advance\texd@loopi by1%
368 \repeat%
369 }%
370 \def\regressbydays#1{%
371 \texd@loopi=0%
372 \loop%
373 \ifnum\texd@loopi<#1%
374 \texd@lastdate%
375 \advance\texd@loopi by1%
376 \repeat%
377 }%
378 \newcount\texd@loopj%
379 \def\advancebyweeks#1{%
380 \texd@loopi=0%
381 \texd@loopj=#1%

```

```

382 \multiply\texd@loopj by7%
383 \loop%
384 \ifnum\texd@loopi<\texd@loopj%
385 \texd@nextdate%
386 \advance\texd@loopi by1%
387 \repeat%
388 }%
389 \def\regressbyweeks#1{%
390 \texd@loopi=0%
391 \texd@loopj=#1%
392 \multiply\texd@loopj by7%
393 \loop%
394 \ifnum\texd@loopi<\texd@loopj%
395 \texd@lastdate%
396 \advance\texd@loopi by1%
397 \repeat%
398 }%
399 \def\advancebymonths#1{%
400 \texd@loopi=0%
401 \loop%
402 \ifnum\texd@loopi<#1%
403 \texd@nextmonth%
404 \advance\texd@loopi by1%
405 \repeat%
406 \texd@setrmon%
407 \initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@rdom}%
408 }%
409 \def\regressbymonths#1{%
410 \texd@loopi=0%
411 \loop%
412 \ifnum\texd@loopi<#1%
413 \texd@lastmonth%
414 \advance\texd@loopi by1%
415 \repeat%
416 \texd@setrmon%
417 \initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@rdom}%
418 }%

```

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

```

419 \def\printdate{%
420 \texd@dateformat%
421 }%
422 \def\printfdate#1{%
423 \texd@formatdateformat{#1}%
424 }%

```

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

```

425 \def\texd@expandloop#1{%
426 \texd@xloop#1\relax
427 }

```

```

428 \def\texdatenumformat#1{#1}
429 \def\texd@xloop#1{%
430 \ifx\relax#1%
431 \else%
432 \ifx#1d%
433 \setpadnum{2}\setpadchar{0}%
434 \padnum{\texdatenumformat{\the\texd@rdom}}%
435 \else\ifx#1e%
436 \setpadnum{2}\setpadchar{\hskip1ex}%
437 \padnum{\texdatenumformat{\the\texd@rdom}}%
438 \else\ifx#1a%
439 \texd@shdownname%
440 \else\ifx#1A%
441 \texd@downname%
442 \else\ifx#1b%
443 \texd@shmonthname%
444 \else\ifx#1h%
445 \texd@shmonthname%
446 \else\ifx#1B%
447 \texd@monthname%
448 \else\ifx#1w%
449 \the\texd@dow%
450 \else\ifx#1u%
451 \ifnum\texd@dow>0\leavevmode\texdatenumformat{\the\texd@dow}\fi%
452 \ifnum\texd@dow=0 7\fi%
453 \else\ifx#1Y%
454 \setpadnum{2}\setpadchar{0}%
455 \padnum{\texdatenumformat{\the\texd@yr}}%
456 \else\ifx#1m%
457 \texd@setrmon%
458 \setpadnum{2}\setpadchar{0}%
459 \leavevmode\padnum{\texdatenumformat{\the\texd@rmon}}%
460 \else\ifx#1j%
461 \texd@setjnum%
462 \setpadnum{3}\setpadchar{0}%
463 \leavevmode\padnum{\texdatenumformat{\the\texd@jnum}}%
464 \else\ifx#1C%
465 \quotient{\texd@yr}{100}%
466 \leavevmode\texdatenumformat{\the\intquotient}%
467 \else\ifx#1y%
468 \modulo{\texd@yr}{100}%
469 \leavevmode\texdatenumformat{\the\remainder}%
470 \else\ifx#1U%
471 \quotient{\texd@jnum}{7}
472 \setpadnum{2}\setpadchar{0}%
473 \leavevmode\padnum{\texdatenumformat{\the\intquotient}}%
474 \else\ifx#1V%
475 \quotient{\texd@jnum}{7}
476 \ifnum\texd@dow>0\advance\intquotient by1\fi%
477 \setpadnum{2}\setpadchar{0}%

```

```

478 \leavevmode\padnum{\texdatenumformat{\the\intquotient}}}%
479 \else\ifx#1W%
480 \quotient{\texd@jnum}{7}
481 \ifnum\texd@dow=0\advance\intquotient by-1\fi%
482 \setpadnum{2}\setpadchar{0}%
483 \leavevmode\padnum{\texdatenumformat{\the\intquotient}}}%
484 \else%
485 #1%
486 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi%
487 \expandafter\texd@xloop%
488 \fi%
489 }%
490 \def\texd@dateformat{%
491 \expandafter\expandafter\texd@expandloop{\setdateformat}%
492 }%
493 \def\texd@formatdateformat#1{%
494 \expandafter\expandafter\expandafter\expandafter\expandafter\expandafter\texd@expandloop{\csname#1
495 }%
496 \def\setdateformat{A{ }(a),\ B\ (b){ }d,\ Y}
497 \def\nameddateformat#1#2{%
498 \expandafter\def\csname texd@df#1\endcsname{#2}%
499 }%
500 \nameddateformat{american}{B\ d,\ Y}
501 \nameddateformat{shamerican}{m/d/Y}
502 \nameddateformat{ISO}{Ymd}
503 \nameddateformat{ISOext}{Y-m-d}
504 \nameddateformat{shbritish}{d/m/Y}
505 \nameddateformat{shbritishdots}{d.m.Y}
506 \nameddateformat{british}{d\ B\ Y}
507 \nameddateformat{d}{d}
508 \nameddateformat{e}{e}
509 \nameddateformat{B}{B}
510 \nameddateformat{b}{b}
511 \nameddateformat{h}{h}
512 \nameddateformat{m}{m}
513 \nameddateformat{A}{A}
514 \nameddateformat{a}{a}
515 \nameddateformat{w}{w}
516 \nameddateformat{u}{u}
517 \nameddateformat{Y}{Y}
518 \nameddateformat{j}{j}
519 \nameddateformat{C}{C}
520 \nameddateformat{y}{y}
521 \nameddateformat{U}{U}
522 \nameddateformat{V}{V}
523 \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.

```

524 \def\initcurrdate{%

```

```

525 \global\texd@mon=\month%
526 \global\advance\texd@mon by-1%
527 \global\texd@dom=\day%
528 \global\advance\texd@dom by-1%
529 \global\texd@yr=\year%
530 \texd@isleapyear%
531 \texd@setrdom%
532 \texd@setrmon%
533 \texd@setjnum%
534 \texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
535 }%
536 \def\initdate#1#2#3{%
537 \global\texd@yr=#1%
538 \global\texd@mon=#2%
539 \global\advance\texd@mon by-1%
540 \global\texd@dom=#3%
541 \global\advance\texd@dom by-1%
542 \global\texd@setrdom%
543 \global\texd@setrmon%
544 \texd@setjnum%
545 \texd@isleapyear%
546 \texd@nextdow{\the\texd@rdom}{\the\texd@rmon}{\the\texd@yr}%
547 }%

```

Now we define the macros for saving and restoring dates.

```

548 \def\savedate#1{%
549 \expandafter\edef\csname savedate#1\endcsname{\initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@dom}}%
550 }%
551 \def\restoredate#1{%
552 \csname savedate#1\endcsname%
553 }%

```

Convenience macros. First, \texdcal.

```

554 \newcount\texd@looptmp\texd@looptmp=0%
555 \def\texdcal#1#2{%
556 \global\texd@mon=#2%
557 \global\advance\texd@mon by-1%
558 \global\texd@yr=#1%
559 \global\texd@dom=0%
560 \texd@setrmon\texd@setrdom%
561 \initdate{\the\texd@yr}{\the\texd@rmon}{\the\texd@dom}%
562 \def\setdateformat{B\ Y}%
563 \begin{tabular}{rrrrrrr}
564 \multicolumn{7}{c}{\printdate} \\
565 \loop\ifnum\texd@dow>0\texd@lastdate\repeat%
566 \def\setdateformat{d}%
567 \ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
568 \def\setdateformat{d}\advancebydays{1}%
569 \ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&
570 \def\setdateformat{d}\advancebydays{1}%
571 \ifnum\texd@dom>8 {} \fi\ifnum\texd@dom<8\leavevmode\printdate\fi&

```

```

572 \def\setdateformat{d}\advancebydays{1}%
573 \ifnum\texdom>8 {} \fi\ifnum\texdom<8\leavevmode\printdate\fi&
574 \def\setdateformat{d}\advancebydays{1}%
575 \ifnum\texdom>8 {} \fi\ifnum\texdom<8\leavevmode\printdate\fi&
576 \def\setdateformat{d}\advancebydays{1}%
577 \ifnum\texdom>8 {} \fi\ifnum\texdom<8\leavevmode\printdate\fi&
578 \def\setdateformat{d}\advancebydays{1}%
579 \ifnum\texdom>8 {} \fi\ifnum\texdom<8\leavevmode\printdate\fi\\
580 \def\setdateformat{d}\advancebydays{1}\printdate &
581 \def\setdateformat{d}\advancebydays{1}\printdate &
582 \def\setdateformat{d}\advancebydays{1}\printdate &
583 \def\setdateformat{d}\advancebydays{1}\printdate &
584 \def\setdateformat{d}\advancebydays{1}\printdate &
585 \def\setdateformat{d}\advancebydays{1}\printdate &
586 \def\setdateformat{d}\advancebydays{1}\printdate \\
587 \def\setdateformat{d}\advancebydays{1}\printdate &
588 \def\setdateformat{d}\advancebydays{1}\printdate &
589 \def\setdateformat{d}\advancebydays{1}\printdate &
590 \def\setdateformat{d}\advancebydays{1}\printdate &
591 \def\setdateformat{d}\advancebydays{1}\printdate &
592 \def\setdateformat{d}\advancebydays{1}\printdate &
593 \def\setdateformat{d}\advancebydays{1}\printdate \\
594 \def\setdateformat{d}\advancebydays{1}\printdate &
595 \def\setdateformat{d}\advancebydays{1}\printdate &
596 \def\setdateformat{d}\advancebydays{1}\printdate &
597 \def\setdateformat{d}\advancebydays{1}\printdate &
598 \def\setdateformat{d}\advancebydays{1}\printdate &
599 \def\setdateformat{d}\advancebydays{1}\printdate &
600 \def\setdateformat{d}\advancebydays{1}\printdate \\
601 \def\setdateformat{d}\advancebydays{1}%
602 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
603 \def\setdateformat{d}\advancebydays{1}%
604 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
605 \def\setdateformat{d}\advancebydays{1}%
606 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
607 \def\setdateformat{d}\advancebydays{1}%
608 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
609 \def\setdateformat{d}\advancebydays{1}%
610 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
611 \def\setdateformat{d}\advancebydays{1}%
612 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
613 \def\setdateformat{d}\advancebydays{1}%
614 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi\\
615 \def\setdateformat{d}\advancebydays{1}%
616 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
617 \def\setdateformat{d}\advancebydays{1}%
618 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
619 \def\setdateformat{d}\advancebydays{1}%
620 \ifnum\texdom<14 {} \fi\ifnum\texdom>14\leavevmode\printdate\fi&
621 \def\setdateformat{d}\advancebydays{1}%

```



```

622 \ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
623 \def\setdateformat{d}\advancebydays{1}%
624 \ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
625 \def\setdateformat{d}\advancebydays{1}%
626 \ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi&
627 \def\setdateformat{d}\advancebydays{1}%
628 \ifnum\texd@dom<14 {} \fi\ifnum\texd@dom>14\leavevmode\printdate\fi\\
629 \end{tabular}
630 }%
631 \def\texdcalyear#1{%
632 \texd@yr=#1%
633 \texd@mon=0%
634 \texd@dom=0%
635 \texd@setrmon%
636 \texd@setrdom%
637 {\tabcolsep=3pt%
638 \begin{tabular}{ccc}
639 \texdcal{#1}{1} & \texdcal{#1}{2} & \texdcal{#1}{3} \\
640 \texdcal{#1}{4} & \texdcal{#1}{5} & \texdcal{#1}{6} \\
641 \texdcal{#1}{7} & \texdcal{#1}{8} & \texdcal{#1}{9} \\
642 \texdcal{#1}{10} & \texdcal{#1}{11} & \texdcal{#1}{12} \\
643 \end{tabular}
644 }%
645 }%

```

Calculate the day of the year (%j).

```

646 \newcount\texd@jnum\texd@jnum=0%
647 \def\texd@setjnum{%
648 \texd@jnum=0%
649 \ifnum\texd@mon>0\global\advance\texd@jnum by31\fi%
650 \ifnum\texd@mon>1%
651 \global\advance\texd@jnum by28%
652 \fi%
653 \ifnum\texd@mon>2\global\advance\texd@jnum by31\fi%
654 \ifnum\texd@mon>3\global\advance\texd@jnum by30\fi%
655 \ifnum\texd@mon>4\global\advance\texd@jnum by31\fi%
656 \ifnum\texd@mon>5\global\advance\texd@jnum by30\fi%
657 \ifnum\texd@mon>6\global\advance\texd@jnum by31\fi%
658 \ifnum\texd@mon>7\global\advance\texd@jnum by31\fi%
659 \ifnum\texd@mon>8\global\advance\texd@jnum by30\fi%
660 \ifnum\texd@mon>9\global\advance\texd@jnum by31\fi%
661 \ifnum\texd@mon>10\global\advance\texd@jnum by30\fi%
662 \global\advance\texd@jnum by\the\texd@dom%
663 \global\advance\texd@jnum by1%
664 }%

```

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

```

665 \def\texd@firstmon{January}%
666 \def\texd@firstshmon{Jan}%
667 \def\texd@secondmon{February}%

```

```

668 \def\textd@secondshmon{Feb}%
669 \def\textd@thirdmon{March}%
670 \def\textd@thirdshmon{Mar}%
671 \def\textd@fourthmon{April}%
672 \def\textd@fourthshmon{Apr}%
673 \def\textd@fifthmon{May}%
674 \def\textd@fifthshmon{May}%
675 \def\textd@sixthmon{June}%
676 \def\textd@sixthshmon{Jun}%
677 \def\textd@seventhmon{July}%
678 \def\textd@seventhshmon{Jul}%
679 \def\textd@eighthmon{August}%
680 \def\textd@eighthshmon{Aug}%
681 \def\textd@ninthmon{September}%
682 \def\textd@ninthshmon{Sep}%
683 \def\textd@tenthmon{October}%
684 \def\textd@tenthshmon{Oct}%
685 \def\textd@eleventhmon{November}%
686 \def\textd@eleventhshmon{Nov}%
687 \def\textd@twelfthmon{December}%
688 \def\textd@twelfthshmon{Dec}%
689 \def\textd@firstdayname{Sunday}%
690 \def\textd@firstdayshname{Sun}%
691 \def\textd@seconddayname{Monday}%
692 \def\textd@seconddayshname{Mon}%
693 \def\textd@thirddayname{Tuesday}%
694 \def\textd@thirddayshname{Tue}%
695 \def\textd@fourthdayname{Wednesday}%
696 \def\textd@fourthdayshname{Wed}%
697 \def\textd@fifthdayname{Thursday}%
698 \def\textd@fifthdayshname{Thu}%
699 \def\textd@sixthdayname{Friday}%
700 \def\textd@sixthdayshname{Fri}%
701 \def\textd@seventhdayname{Saturday}%
702 \def\textd@seventhdayshname{Sat}%
703 \AtBeginDocument{%
704 \IfLanguageName{spanish}{%
705 \def\textd@firstmon{enero}%
706 \def\textd@firstshmon{ene}%
707 \def\textd@secondmon{febrero}%
708 \def\textd@secondshmon{feb}%
709 \def\textd@thirdmon{marzo}%
710 \def\textd@thirdshmon{mar}%
711 \def\textd@fourthmon{abril}%
712 \def\textd@fourthshmon{abr}%
713 \def\textd@fifthmon{mayo}%
714 \def\textd@fifthshmon{may}%
715 \def\textd@sixthmon{junio}%
716 \def\textd@sixthshmon{jun}%
717 \def\textd@seventhmon{julio}%

```

```

718 \def\textd@seventhshmon{jul}%
719 \def\textd@eighthmon{agosto}%
720 \def\textd@eighthshmon{ago}%
721 \def\textd@ninthmon{septiembre}%
722 \def\textd@ninthshmon{sep}%
723 \def\textd@tenthmon{octubre}%
724 \def\textd@tenthshmon{oct}%
725 \def\textd@eleventhmon{noviembre}%
726 \def\textd@eleventhshmon{nov}%
727 \def\textd@twelfthmon{diciembre}%
728 \def\textd@twelfthshmon{dic}%
729 \def\textd@firstdayname{domingo}%
730 \def\textd@firstdayshname{dom}%
731 \def\textd@seconddayname{lunes}%
732 \def\textd@seconddayshname{lun}%
733 \def\textd@thirddayname{martes}%
734 \def\textd@thirddayshname{mar}%
735 \def\textd@fourthdayname{miercoles}%
736 \def\textd@fourthdayshname{mie}%
737 \def\textd@fifthdayname{jueves}%
738 \def\textd@fifthdayshname{jue}%
739 \def\textd@sixthdayname{viernes}%
740 \def\textd@sixthdayshname{vie}%
741 \def\textd@seventhdayname{sabado}%
742 \def\textd@seventhdayshname{sab}%
743 }{}%
744 \IfLanguageName{french}{%
745 \def\textd@firstmon{janvier}%
746 \def\textd@firstshmon{janv}%
747 \def\textd@secondmon{février}%
748 \def\textd@secondshmon{févr}%
749 \def\textd@thirdmon{mars}%
750 \def\textd@thirdshmon{mars}%
751 \def\textd@fourthmon{avril}%
752 \def\textd@fourthshmon{avr}%
753 \def\textd@fifthmon{mai}%
754 \def\textd@fifthshmon{mai}%
755 \def\textd@sixthmon{juin}%
756 \def\textd@sixthshmon{juin}%
757 \def\textd@seventhmon{juil}%
758 \def\textd@seventhshmon{juil}%
759 \def\textd@eighthmon{août}%
760 \def\textd@eighthshmon{août}%
761 \def\textd@ninthmon{septembre}%
762 \def\textd@ninthshmon{sept}%
763 \def\textd@tenthmon{octobre}%
764 \def\textd@tenthshmon{oct}%
765 \def\textd@eleventhmon{novembre}%
766 \def\textd@eleventhshmon{nov}%
767 \def\textd@twelfthmon{décembre}%

```

```

768 \def\textd@twelfthshmon{déc}%
769 \def\textd@firstdayname{dimanche}%
770 \def\textd@firstdayshname{dim}%
771 \def\textd@seconddayname{lundi}%
772 \def\textd@seconddayshname{lun}%
773 \def\textd@thirddayname{mardi}%
774 \def\textd@thirddayshname{mar}%
775 \def\textd@fourthdayname{mercredi}%
776 \def\textd@fourthdayshname{mer}%
777 \def\textd@fifthdayname{jeudi}%
778 \def\textd@fifthdayshname{jeu}%
779 \def\textd@sixthdayname{vendredi}%
780 \def\textd@sixthdayshname{ven}%
781 \def\textd@seventhdayname{samedi}%
782 \def\textd@seventhdayshname{sam}%
783 }{}%
784 \IfLanguageName{german}{%
785 \def\textd@firstmon{Januar}%
786 \def\textd@firstshmon{Jan}%
787 \def\textd@secondmon{Februar}%
788 \def\textd@secondshmon{Feb}%
789 \def\textd@thirdmon{März}%
790 \def\textd@thirdshmon{März}%
791 \def\textd@fourthmon{April}%
792 \def\textd@fourthshmon{Apr}%
793 \def\textd@fifthmon{Mai}%
794 \def\textd@fifthshmon{Mai}%
795 \def\textd@sixthmon{Juni}%
796 \def\textd@sixthshmon{Juni}%
797 \def\textd@seventhmon{Juli}%
798 \def\textd@seventhshmon{Juli}%
799 \def\textd@eighthmon{August}%
800 \def\textd@eighthshmon{Aug}%
801 \def\textd@ninthmon{September}%
802 \def\textd@ninthshmon{Sept}%
803 \def\textd@tenthmon{Oktober}%
804 \def\textd@tenthshmon{Okt}%
805 \def\textd@eleventhmon{November}%
806 \def\textd@eleventhshmon{Nov}%
807 \def\textd@twelfthmon{Dezember}%
808 \def\textd@twelfthshmon{Dez}%
809 \def\textd@firstdayname{Sonntag}%
810 \def\textd@firstdayshname{So}%
811 \def\textd@seconddayname{Montag}%
812 \def\textd@seconddayshname{Mo}%
813 \def\textd@thirddayname{Dienstag}%
814 \def\textd@thirddayshname{Di}%
815 \def\textd@fourthdayname{Mittwoch}%
816 \def\textd@fourthdayshname{Mi}%
817 \def\textd@fifthdayname{Donnerstag}%

```

```

818 \def\texd@fifthdayshname{Do}%
819 \def\texd@sixthdayname{Freitag}%
820 \def\texd@sixthdayshname{Fr}%
821 \def\texd@seventhdayname{Samstag}%
822 \def\texd@seventhdayshname{Sa}%
823 }{}%
824 }%

```

Happy T<sub>E</sub>Xing!