

# Das `datenum.sty` Paket v0.03

Jörg-Michael Schröder

2022/01/27\*

## Zusammenfassung

Dieses Paket stellt Kommandos zur Verfügung, mit denen sich ein Datum in eine natürliche Zahl umwandeln läßt. Umgekehrt kann auch aus einer Zahl ein Datum errechnet werden. Außerdem gibt es Kommandos zur Inkrementierung und Dekrementierung eines Datums. Schaltjahre und die Gregorianische Kalenderreform werden berücksichtigt.

## Inhaltsverzeichnis

<b>1</b>	<b>Startjahr</b>	<b>1</b>
<b>2</b>	<b>Zähler</b>	<b>1</b>
<b>3</b>	<b>Makros</b>	<b>2</b>
3.1	Makros mit Standardzählern . . . . .	2
3.2	Makros mit eigenen Zählern . . . . .	2
3.3	Monatsname . . . . .	2
3.4	Wochentag . . . . .	2
<b>4</b>	<b>Sprachunterstützung</b>	<b>3</b>
<b>5</b>	<b>Beispiele</b>	<b>3</b>
<b>6</b>	<b>Sonstiges</b>	<b>4</b>

## 1 Startjahr

Der Start der Zählung wird mit `\setstartyear{year}` festgelegt (Standard 1800). Der 01.01. des Startjahres bekommt die Nummer 1. Der Wert von `startyear` muß mindestens 1 sein darf aber nicht größer sein als das Jahr eines Datums für das eine Nummer berechnet werden soll. Je größer die Differenz von Datum und Startjahr ist, desto länger dauern die Berechnungen. Die richtige Berechnung der Wochentage ist nur garantiert, wenn `startyear` auf 1800, 1900 oder 2000 gesetzt ist. Ausgegeben wird das Jahr mit `\thestartyear`.

## 2 Zähler

Es werden fünf Zähler definiert

`datenumber`: Nummer des Tages

`dateyear`: Jahr

`datemonth`: Monat

---

\*Das ursprüngliche Datum war 2001/08/06 (version 0.02). Diese Version behebt einen kleinen Fehler in `datenumberfrench.ldf` und in `datenumberspanish.ldf`.

`dateday`: Tag

`datedayname`: Wochentag: 1–7 entspricht Montag–Sonntag

Diese Zähler werden von den meisten Makros verwendet. Es gibt aber auch Makros denen eigene Zähler übergeben werden können.

## 3 Makros

### 3.1 Makros mit Standardzählern

Bei diesen Makros werden alle oben genannten Zähler, der Wochentag und der Monatsname aktualisiert.

`\setdatanumber{year}{month}{day}`: Setzt den Zähler `datanumber` auf den Wert, der dem Datum entspricht.

`\setdatebynumber{number}`: Setzt die Zähler `dateyear`, `datemonth` und `dateday` auf die Werte, die der Nummer entsprechen.

`\nextdate`: Setzt die Zähler `dateyear`, `datemonth` und `dateday` auf das nächste Datum.

`\prevdate`: Setzt die Zähler `dateyear`, `datemonth` und `dateday` auf das vorherige Datum.

`\setdate{year}{month}{day}`: Setzt die Zähler `dateyear`, `datemonth` und `dateday` auf die übergebenen Werte.

`\setdatetoday`: Setzt die Zähler `dateyear`, `datemonth` und `dateday` auf das aktuelle Datum.

`\datemonthname`: Gibt den Namen des Monats aus (Siehe Abschnitt 3.3).

`\datedayname`: Gibt den Namen des Wochentages aus (Siehe Abschnitt 3.4).

`\datedate`: Gibt das Datum aus, daß den Zählern `dateyear`, `datemonth` und `dateday` entspricht.

### 3.2 Makros mit eigenen Zählern

Bei diesen Makros werden weder die oben genannten Zähler geändert (es sei denn, ihre Namen werden hier übergeben), noch werden der Wochentag und der Monatsname aktualisiert.

`\setmydatanumber{numbercount}{year}{month}{day}`: Setzt den Zähler `numbercount` auf den Wert, der dem Datum entspricht.

`\setmydatebynumber{number}{yearcount}{monthcount}{daycount}`: Setzt die Zähler `yearcount`, `monthcount` und `daycount` auf die Werte, die der Nummer entsprechen.

`\mynextdate{yearcount}{monthcount}{daycount}`: Setzt die Zähler `yearcount`, `monthcount` und `daycount` auf das nächste Datum.

`\mynextdate{yearcount}{monthcount}{daycount}`: Setzt die Zähler `yearcount`, `monthcount` und `daycount` auf das vorherige Datum.

### 3.3 Monatsname

Der Monatsname wird von den Makros aus Abschnitt 3.1 mit dem Aufruf `\setmonthname{number}` aktualisiert. Der Name steht mit `\datemonthname` zur Verfügung.

### 3.4 Wochentag

Makros aus Abschnitt 3.1 definieren das Kommando `\datedayname` entsprechend dem Datum. Man kann mit `\setdayname{number}` den Wochentag auch explizit setzen, wobei 1 Montag und 7 Sonntag entspricht. Auch mit der laufenden Nummer eines Tages läßt sich der Wochentag setzen: `\setdaynamebynumber{number}`. Das Ganze funktioniert richtig, wenn `startyear` auf 1800, 1900 oder 2000 gesetzt ist.

## 4 Sprachunterstützung

Bisher werden die Sprachoptionen `english`, `USenglish` (Standard), `french`, `spanish`, `german` und `ngerman` unterstützt.

Um eine Sprache auszuwählen, schreibe `\dateselectlanguage{language}`. Für andere Sprachen: Erstelle eine Datei `datenumbermylanguage.ldf`. Kopiere den Inhalt von `datenumberdummy.ldf`. Ersetze jedes Vorkommen von „dummy“ durch „mylanguage“ und ändere die Namen für Monat und Wochentag. Nach `\usepackage{datenumber}` füge `\input{datenumbermylanguage.ldf}` in das Dokument ein.

## 5 Beispiele

```
\setdate{2002}{1}{1}%  
\thedataumber
```

Ergebnis: 73780

```
\setdatetoday  
\addtocounter{datenumber}{10}%  
\setdatebynumber{\thedataumber}%  
In 10 Tagen ist der \datedate
```

Ergebnis: In 10 Tagen ist der 6. Februar 2022

```
\newcounter{dateone}\newcounter{datetwo}%
```

```
\newcommand{\daydifftoday}[3]{%  
  \setmydatenumber{dateone}{\the\year}{\the\month}{\the\day}%  
  \setmydatenumber{datetwo}{#1}{#2}{#3}%  
  \addtocounter{datetwo}{-\thedataone}%  
  \thedataetwo  
}
```

Bis Weihnachten sind es noch `\daydifftoday{\the\year}{12}{25}` Tage.

Ergebnis: Bis Weihnachten sind es noch 332 Tage.

```

\newcommand{\sd}{%
\ifcase\thedatedayname \or
    Mo.\or Di.\or Mi.\or Do.\or
    Fr.\or Sa.\or So.\fi
}%

```

```

\newcommand{\pnext}{%
\ifnum\value{dateday}<10 0\fi
\thedateday.%
\ifnum\value{datemonth}<10 0\fi
\thdatemonth.\thdateyear
\nextdate
}

```

```

\setdate{2001}{9}{29}%
\[\begin{tabular}{lll}
\sd & \pnext & Abc\\
\sd & \pnext & Def\\
\sd & \pnext & Ghi\\
\sd & \pnext & Jkl\\
\end{tabular}\]

```

Ergebnis:

Sa.	29.09.2001	Abc
So.	30.09.2001	Def
Mo.	01.10.2001	Ghi
Di.	02.10.2001	Jkl

## 6 Sonstiges

- Test auf Schaltjahr

```

Das Jahr \the\year\ ist
\ifleapyear{\the\year} ein \else kein \fi Schaltjahr.
Ergebnis: Das Jahr 2022 ist kein Schaltjahr.

```

- Test auf gültiges Datum

```

Das Datum 29.2.1900 ist
\ifvaliddate{1900}{2}{29} g"ultig\else nicht g"ultig\fi.
Ergebnis: Das Datum 29.2.1900 ist nicht gültig.1

```

---

<sup>1</sup>Es gibt einige Programme, die dazu eine andere Meinung haben. Für mehr Information über Schaltjahre und den 5. Oktober 1582 suche nach „Gregorian calendar“.