



`\dante_tutorial:nn { expl3 } { 2022 }`

Eine Einführung in die \LaTeX 3-Programmierung

Marei Peischl (marei@peiTeX.de)

2022-06-23

Tutorial im Rahmen der DANTE Sommertagung 2022, Magdeburg

Vielleicht ein bisschen verrückt ...

I presume this is one of several dozen bugs that would arise over the years if anyone were foolish enough to try allowing “_” in command names.

Leslie Lamport[2]

- Einheitliche Interfaces für Funktionen und Variablen
- Modernisierung der Syntax
- Vereinfachung der Expansionskontrolle

⇒ Mehr Möglichkeiten.

Wieso sollte ich in \LaTeX programmieren?

Wieso sollte ich in \LaTeX programmieren?

Weil es geht!

Wieso sollte ich in \LaTeX programmieren?

Weil es geht!

Automatisierung – Effizienz – Flexibilität

Aber ich verwende doch schon `ifthen[1]`.

Aber ich verwende doch schon ifthen[1].

CTAN
Comprehensive T_EX Archive Network

Marel Logout Settings Help

Cover Upload Browse Search

Location: CTAN Topics Conditional compilation

Conditional compilation

This topic contains packages for conditional typesetting.

codesection Provides an environment that may be conditionally included.	graphicscache Cache includegraphics calls.	multiaudience Several versions of output from the same source.	tagging Document configuration with tags.
comment Selectively include/exclude portions of text.	ifallfalse Compare a string against a set of other strings.	optional Facilitate optional printing of parts of a document.	to-be-determined Highlight text passages that need further work.
comment_io A Python script to comment and uncomment lines.	ifthen Conditional commands in L ^A T _E X documents.	phfcc Convenient inline commenting in collaborative documents.	version Conditionally include text.
comment-pln Comment macros for use in Plain T _E X.	ifthenx Extra tests for \ifthenelse.	renditions Multiple versions from the same content.	versions Optionally omit pieces of text.
CondiT_EXt Define and manage conditional content.	luacensor Securely redact sensitive information using Lua.	scredact A tool for redacting sources.	xcomment Allows selected environments to be included/excluded.
			xifthen Extended conditional commands.



Anwendungsbeispiel: Schleifen T_EX vs. expl3

Beispiel (plain T_EX)

```
\newcount\cnt
\cnt=1
\loop
  \the\cnt
  \advance \cnt +1
\ifnum \cnt<11
\repeat
```

Anwendungsbeispiel: Schleifen T_EX vs. expl3

Beispiel (plain T_EX)

```
\newcount\cnt
\cnt=1
\loop
  \the\cnt
  \advance \cnt +1
\ifnum \cnt<11
\repeat
```

Beispiel (expl3)

```
\int_step_inline:nn { 10 } {#1}
```

**Wiederholung/Begleitinfos
zu ein paar T_EXnischen Konzepte**

Category Codes

0	Escape character	8	Subscript
1	Beginning of group	9	Ignored character
2	End of group	10	Space
3	Math shift	11	Letter
4	Alignment tab	12	Other character
5	End of Line	13	Active character
6	Parameter	14	Comment character
7	Superscript	15	Invalid Character

Die Sonderrolle des @-Zeichens

```
\makeatletter
```

```
\makeatother
```

Die Sonderrolle des @-Zeichens

```
\makeatletter  
\makeatother
```

Beispiel

```
\makeatletter  
\newcommand*{\cmd}{\@ifstar\@cmdstar\@cmd}  
\newcommand*{\@cmd}{ohne *}  
\newcommand*{\@cmdstar}{mit *}  
\makeatother
```

expl3 – L^AT_EX 3 programming layer

- Leerzeichen und Zeilenumbrüche im Code sind nur dafür da um Zeichenketten zu begrenzen
- Leerzeilen sind kein Absatzumbruch
- Doppelpunkt (:) und Unterstrich (_) sind Bestandteil von Makronamen
- Es gibt einen Unterschied zwischen Funktionen und Variablen

Syntax umschalten

`\ExplSyntaxOn`

`\ExplSyntaxOff`

Funktionen

\Modul_Beschreibung:Argumente

Variablen

\Gültigkeit_Modul_Beschreibung_Typ

Variablen

\Gültigkeit_Modul_Beschreibung_Typ

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

Variablen: Speichern Werte

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

Gültigkeit

constant, global, local

Variablen: Speichern Werte

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

Modul

Benannt nach Paket/Bundle. Registrierung [3]

Variablen: Speichern Werte

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

intern?

_ oder __

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

Beschreibung

Was genau speichert die Variable wofür.

Variablen: Speichern Werte

Variable: Beispiel

```
\l__siunitx_complex_sign_tl
```

Datentyp

Wie werden die Daten gespeichert.

expl3 Datentypen (Beispiele)

<code>bool</code>	Boolean
<code>box</code>	Box
<code>clist</code>	Comma list
<code>coffin</code>	„Box with handles“
<code>dim</code>	Dimesnion
<code>fp</code>	Floating Points (double precision)
<code>int</code>	Integer
<code>prop</code>	Property List (KeyVal)
<code>seq</code>	Sequence
<code>skip</code>	Glue
<code>stream</code>	Input/Output Streams
<code>t1</code>	Tokenlist

Funktion

\Module_Beschreibung: Argumentenspezifikation

Funktion: Beispiel

```
\int_set:Nn
```

Argumente: 1 Token pro Argument

Großbuchstaben: Tokens

Kleinbuchstaben: gruppiertes Argument

Argumentenspezifikation

Token	Name/Beschreibung
N/n	No manipulation
c	Csname
V/v	Value
o	expand Once
x	eXhaustive expansion
e	Exhaustive expansion, allerdings kann das Makro selbst expandierbar sein (langsam)
f	Full expansion

Argumentenspezifikation II

Token	Name/Beschreibung
TF/T/F	True/False;
p	Parameter wie bei T _E X Definitionen
w	Weird

Längen in $\text{\LaTeX}3$
(Argumententyp N/n)

```
\dim_new:N
```



```
\dim_new:N  
\dim_const:Nn
```

```
\dim_new:N  
\dim_const:Nn  
\dim_if_exist:NTF
```

```
\dim_add:Nn
```

```
\dim_set:Nn
```

```
\dim_set_eq:NN
```

```
\dim_sub:Nn
```

```
\dim_add:Nn
```

```
\dim_set:Nn
```

```
\dim_set_eq:NN
```

```
\dim_sub:Nn
```

```
\dim_gadd:Nn
```

```
\dim_gset:Nn
```

```
\dim_gset_eq:NN
```

```
\dim_gsub:Nn
```

```
\dim_compare:nTF
```

Vergleichsoperatoren

Gleichheit = oder ==

Kleiner <

Größer >

Kleinergleich <=

Größergleich >=

Ungleich !=

```
\dim_compare:nTF
```

Vergleichsoperatoren

Gleichheit	= oder ==		
Kleiner	<	Größer	>
Kleinergleich	<=	Größergleich	>=
Ungleich	!=		

Beispiel

```
\dim_compare:nTF { \l_tmpa_dim = \l_tmpb_dim } {true} {false}
```

```
\dim_show:N
```

```
\dim_show:n
```

```
\dim_log:N
```

```
\dim_log:n
```

25.1 Creating and initialising dim variables

```
\dim_new:N <dimension>
```

```
\dim_new:c
```

Creates a new *<dimension>* or raises an error if the name is already taken. The declaration is global. The *<dimension>* is initially equal to 0 pt.

```
\dim_const:Nn <dimension> {<dimension expression>}
```

```
\dim_const:cn
```

New: 2012-03-05

Creates a new constant *<dimension>* or raises an error if the name is already taken. The value of the *<dimension>* is set globally to the *<dimension expression>*.

```
\dim_zero:N <dimension>
```

```
\dim_zero:c
```

```
\dim_gzero:N
```

```
\dim_gzero:c
```

Sets *<dimension>* to 0 pt.

```
\dim_zero_new:N <dimension>
```

```
\dim_zero_new:c
```

```
\dim_gzero_new:N
```

```
\dim_gzero_new:c
```

New: 2012-01-07

Ensures that the *<dimension>* exists globally by applying `\dim_new:N` if necessary, then applies `\dim_(g)zero:N` to leave the *<dimension>* set to zero.

csname/endcsname
(Argumententyp c)

```
\csname name\endcsname
```

Beispiel

```
\label{frame:csname}
```

```
\expandafter\meaning\csname r@frame:csname\endcsname
```

```
macro:->{22}{42}{csname/endcsname\\(Argumententyp c)}{Doc-Start}}
```

```
\dim_set:Nn \l_tmpa_dim { 1cm }  
\dim_set:cn { l_tmpb_dim }  
\dim_set_eq:cc { l_tmpa_dim } { l_tmpb_dim }
```

Expansionskontrolle

Befehl wird durch seine Bedeutung ersetzt

```
\newcommand*{\Variable}{def}
```

```
\meaning\Variable\
```

```
\newcommand*{\Funktion}[1]{Funktion mit Argument (#1)}
```

```
\meaning\Funktion
```

Befehl wird durch seine Bedeutung ersetzt

```
\newcommand*\Variable}{def}
```

```
\meaning\Variable\
```

```
\newcommand*\Funktion}[1]{Funktion mit Argument (#1)}
```

```
\meaning\Funktion
```

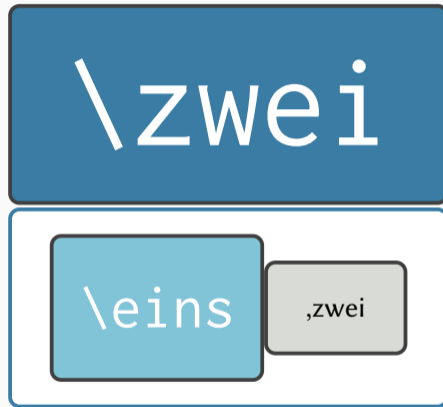
```
macro:->def
```

```
macro:#1->Funktion mit Argument (#1)
```

```
\newcommand\eins{eins}
```



```
\newcommand\zwei{\eins, zwei}
```



Mehrstufige Expansion

```
\newcommand \eins {eins}
```

```
\newcommand \zwei {\eins ,zwei}
```

```
\newcommand \drei {\eins ,\zwei ,drei}
```



`\drei`

Mehrstufige Expansion

```
\newcommand \eins {eins}
```

```
\newcommand \zwei {\eins ,zwei}
```

```
\newcommand \drei {\eins ,\zwei ,drei}
```

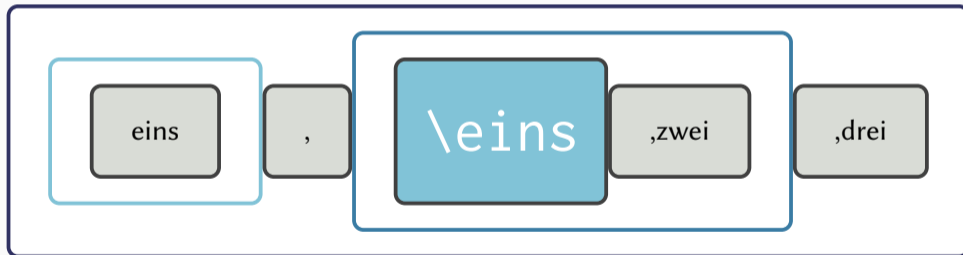


Mehrstufige Expansion

```
\newcommand \eins {eins}
```

```
\newcommand \zwei {\eins ,zwei}
```

```
\newcommand \drei {\eins ,\zwei ,drei}
```

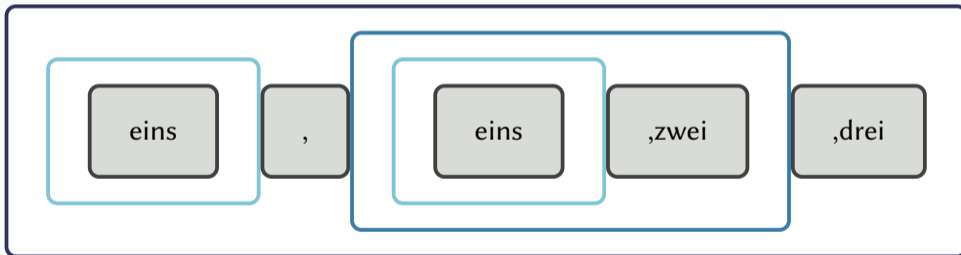


Mehrstufige Expansion

```
\newcommand \eins {eins}
```

```
\newcommand \zwei {\eins ,zwei}
```

```
\newcommand \drei {\eins ,\zwei ,drei}
```



l3keys – Key-Value Syntax ohne Pakete

```
\MyModuleSetup{key=value}
```

```
\MyModuleMacro[key=value]{argument}
```

```
\keys_define:nn {module} {KeyVal setup}
```

KeyVal setup

```
key .code:n = { Code; value will be inserted with #1 },  
key .default:n = default value,  
key .initial:n = initial value,
```

```
\keys_set:nn {module} {KeyVal list}
```

Schlüssel als Paket-/Klassenoptionen:

```
\RequirePackage{13keys2e}
```

```
\ProcessKeysOptions{module}
```

```
\ProcessKeysPackageOptions{module}
```

l3keys choice example

```
\keys_define:nn {ptxcd/pub} {  
  custommargins .choice:,  
  custommargins / true .code:n = {  
    \bool_gset_true:N \g_ptxcd_custommargins_bool  
    \bool_gset_false:N \g_ptxcd_geometry_bool  
  },  
  % [andere mögliche Werte]  
  custommargins .initial:n = false,  
  custommargins .default:n = true,  
}
```


Fazit:

Nur keine Panik.

Fazit:

Nur keine Panik.

Und falls doch:

Einladung: BayT_EX 03.09.2022 Waldkirchen

- Samstag Vormittag: Brennereiführung bei Penninger
- Samstag Nachmittag: Vorträge im Bürgerhaus
- Abend: Social (Grillen oder Biergarten)
- Sonntag: Wanderung

Wir freuen uns über Workshop/Vortragseinreichungen.

Weitere Infos folgen.

Anmeldung bis 18.08.2022 über baytex-stammtisch-anmeldung-2022@KplusS-Ing.de

Literatur

- [1] David Carlisle und Leslie Lamport. *The ifthen package*. 13. Apr. 2022. URL: <https://ctan.org/pkg/ifthen>.
- [2] Frank Mittelbach. „ $\text{\LaTeX}3$ architecture and current work in progress“. In: *TUG Conference 2011*. 20. Okt. 2011. URL: <https://www.latex-project.org/publications/2011-FMi-TUG-LaTeX3-architecture-slides.pdf>.
- [3] Joseph Wright. *Registering expl3 module*. 4. Nov. 2012. URL: <https://www.texdev.net/2012/11/04/registering-expl3-module/>.