

L^AT_EX and Friends

Branching

M. R. C. van Dongen

ucc

January 18, 2012

Branching

Variables

[The `ifthen` Package](#)[The `calc` Package](#)[Looping](#)[Tail Recursion](#)

Acronyms & Abbreviations

[About this Document](#)

```
\newcounter{<name>}
\setcounter{<name>}{<value>}
\stepcounter{<name>}
\addtocounter{<name>}{<increment>}
\the<name>
\newcounter{<slave>}[<master>]
```

L^AT_EX Input

```
\newcounter{ans}  
\setcounter{ans}{9}  
\addtocounter{ans}{11}  
\stepcounter{ans}  
\addtocounter{ans}{\theans}
```

The answer to the ultimate
question of life, the universe,
and everything is `\theans`.

L^AT_EX Output

The answer to the ultimate question of life, the universe, and everything is 42.

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

Decision Making

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

- L^AT_EX does not support decision making.
- To make decisions you need T_EX or package such as `ifthen`.

Branching

Variables

The `ifthen` PackageThe `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

```
\newif\if<switch>  
\<switch>true  
\<switch>>false  
\if<switch><then clause>\fi  
\if<switch><then clause>\else<else clause>\fi
```

Example

Branching

Variables

The `ifthen` PackageThe `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

L^AT_EX Usage

```
\newif\ifnotes
\notesttrue

\begin{document}
\section{\ifnotes Lecture Notes%
         \else Presentation%
         \fi}
...
\end{document}
```

Unit	Name	Equivalent
pt	point	
pc	pica	1 pc = 12 pt
in	inch	1 in = 72.27 pt
bp	big point	72 bp = 1 in
cm	centimetre	2.54 cm = 1 in
mm	millimetre	10 mm = 1 cm
dd	didôt point	1157 dd = 1238 pt
cc	cicero	1 cc = 12 dd
sp	scaled point	65536 sp = 1 pt

- Each length unit represents its own length.
- When L^AT_EX expects a length, writing `1<unit>` results in the length of the unit `<unit>`.
 - For example `1mm` gives you the length of one millimetre.
- Write `<constant><unit>` to multiply `<unit>` and `<constant>`.
 - For example, `101in` is equivalent to `256.54cm`.

Length Variables

- Length variables hold length values.
- You write them just as control sequences.
- Given variable `<len>`, `2<len>` gives you twice its current value.
- There are two kinds of lengths: *rigid* and *rubber*.

rigid A rigid length always has the same size.

rubber A rubber length is a combination of natural length and elasticity.

- Values may stretch or shrink depending on the situation.
- Useful for stretching/shrinking inter-word space and so on.
- Multiplying a rubber length makes it rigid.

Branching

Variables

The `ifthen` PackageThe `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

Existing Length Commands

```
\parindent  
\textwidth  
\textheight  
\parskip  
\baselineskip
```

Branching

Variables

The `ifthen` PackageThe `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

Length-related Commands

```
\newlength{<command>}
\setlength{<command>}{<length>}
\addtolength{<command>}{<length>}
\settoheight{<command>}{<stuff>}
\settodepth{<command>}{<stuff>}
\settoheight{<command>}{<stuff>}
\settodepth{<command>}{<stuff>}
```

The `ifthen` Package

```
\newboolean{<bool>}
```

Define variable. May fail.

```
\provideboolean{<bool>}
```

Define variable. Can't fail.

```
\setboolean{<bool>}{<value>}
```

Assign `<value>`.

[Branching](#)[Variables](#)[The `ifthen` Package](#)[The `calc` Package](#)[Looping](#)[Tail Recursion](#)[Acronyms & Abbreviations](#)[About this Document](#)

```
\ifthenelse{<test>}{<then clause>}{<else clause>}
```

- Two-way branching statement.

```
<boolean>
<number1><op><number2>
\lengthtest{<dimen1><op><dimen2>}
\isodd{<number>}
\isundefined{<command>}
\equal{<string1>}{<string2>}
\boolean{<bool>}
<test1><command><test2>
<negation><test>
\(<test>\)
```

The `\ifthenelse` Command

L^AT_EX Input

```
\begin{document}
  \ifthenelse
    {\isodd{\value{page}}}
    {Odd page.}
    {Even page.}
\end{document}
```

L^AT_EX Output

Odd page.

The `\whiledo` Command

```
\whiledo{<test>}{<statement>}
```

Implements `while` statement.

L^AT_EX Input

```
\newcounter{count}
\setcounter{count}{3}
$\thecount = 0
\whiledo
  {\not\(\thecount=0\)}%
  {+ 1 \addtocounter{count}{-1}}$.
```

L^AT_EX Output

3 = 0 + 1 + 1 + 1.

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

The `calc` Package

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

- The `calc` package extends T_EX and L^AT_EX's arithmetic.
- Makes counter/length commands accept infix expressions.
- Also provides additional useful commands.

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document



Loop Commands

Comma-separated Version

```
\@for \var:=<list>\do \command
```

L^AT_EX Input

```
\@for \var:=1,two\do{%  
  (\var)%  
}
```

L^AT_EX Output

(1)(two)

More Loop Commands

Token Sequence Version

```
\@tfor\var :=<list>\do \command
```

L^AT_EX Input

```
\newcommand*\swop[2]{#2#1}  
\@tfor\var:=1\swop\do{%  
  \var23%  
}
```

L^AT_EX Output

12332

More Looping

while Loop With Condition Based on Switch

```
\@whilesw<switch>\fi{<statements>}
```

L^AT_EX Input

```
\newif\iffirst\firsttrue  
\newif\ifsecond\secondfalse  
\@whilesw\iffirst\fi{%  
  X\ifsecond\firstfalse%  
  \else\secondtrue\fi%  
}
```

L^AT_EX Output

XX

L^AT_EX Usage

```
\def\apply#1{%
  \def\Apply##1{%
    \ifx##1\endApply%
      \breakApply% terminate recursion
    \fi%
    #1{##1}% Apply command to next item.
    \Apply% Tail recursive call.
  }%
  \Apply%
}
\def\breakApply#1\Apply{\fi}%
\def\twice#1{#1#1}

\apply\twice a{bc}d\endApply
```

[Branching](#)[Variables](#)[The `ifthen` Package](#)[The `calc` Package](#)[Looping](#)[Tail Recursion](#)[Acronyms & Abbreviations](#)[About this Document](#)

Bibliography

Branching

Variables

The `ifthen` Package

The `calc` Package

Looping

Tail Recursion

Acronyms & Abbreviations

About this Document

AMS American Mathematical Society

API Application Programming Interface

APL A Programming Language

CTAN Comprehensive T_EX Archive Network

CD Compact Disk

FAQ Frequently Asked Question

GUI Graphical User Interface

IDE Integrated Development Environment

ISBN International Standard Book Number

SI Système International d'Unités/International System of Units

OS Operating System

TUG T_EX Users Group

URL Uniform Resource Locator

WYSIWYG What You See is What You Get

About this Document

- This document was created with `pdflatex`.
- The L^AT_EX document class is `beamer`.
- The main font is *T_EX Gyre Heros Condensed*.
 - You may obtain the font from <http://www.gust.org.pl>.