Meta CS

せみ
蝉
'Cicada'

19 Apr 2021

Abstract

LaTeX commands for describing LaTeX commands and switches, providing *(a)* somewhat customisable argument delimiters, `{}`, `[]`; *(b)* different levels of emphasis, **`\command`**, `\command` and `\command` ; and *(c)* the ability to print and run commands: `\textit {text}` ↦ *text*.

*Acknowledgements: Macro names `\marg` , `\oarg` and `\meta` from ltxdoc, and `latexcode` environment from doctools package. All latex coding inspiration in this document derives ultimately from hundreds of contributions on tex stackexchange, and from package authors and maintainers over the decades.

Contents

1 Control Sequence Meta Commands

Meta command	Result
\cs{⟨*macro-name*⟩}	\test
\marg{⟨*mandatory argument*⟩}	{⟨*test*⟩}
\margv{⟨*mandatory argument value*⟩}	{test}
\oarg{⟨*optional argument*⟩}	[⟨*test*⟩]
\oargv{⟨*optional argument value*⟩}	[test]
\meta{⟨*meta value*⟩}	⟨*test*⟩

2 \cc – run a switch

Run #1 as the control sequence \#1

\cc{⟨*no-slash cs-name*⟩}: run the zero-argument command (switch): \#1

The code

```
{ \cc{itshape}\cc{large} 123 }
```

produces

123

3 \cd – run a 1-argument command

Run as \#1{#2}

\cd{⟨*cs-name*⟩}{⟨*argument*⟩} : run the one-argument command: \#1{#2}

Command	**\cd**{textit}{Sample}
resolves to	**\textit**{Sample}
which produces	*Sample*

A command can be demonstrated inline (using \cdr), like so:
`\cd {textit}{Sample}` ↦ *Sample*.
A command can also be demonstated in display mode (using \cdrq), like so:

`\cd {textit}{Sample}` ↦ *Sample*

For \cdr, see §8, and for \cdrq, see §9.

c SAMPLE$^{\text{squared}}$ π

If	`$\cc {pi}$` ↦ π
and	`$\pi $` ↦ π
therefore	`$\cc {pi}$` ≡ `$\pi $`

`\cd {textsc}{Sample}\cd {textsuperscript}{squared}` ↦ SAMPLE$^{\text{squared}}$
x x x x x x x x x x x x x x x x x x x x x x `\cd {textsc}{Sample}\cd {textsuperscript}{squared}` ↦ SAMPLE$^{\text{squared}}$ x x x x x x x x x x x x x x x x x

`\cm`

↑ formatted as a header

\cm == ← a custom format

4 \cm – inline maths

\cm{⟨*inline maths*⟩}: $\frac{\pi^2}{2}$

5 \cmd – display maths

\cmd{⟨*display maths*⟩}:

$$e = mc^2$$

If	`\cm {\frac {\pi ^2}{2}}` ↦ $\frac{\pi^2}{2}$
and	`$\frac {\pi ^2}{2}$` ↦ $\frac{\pi^2}{2}$
therefore	`\cm {\frac {\pi ^2}{2}}` ≡ `$\frac {\pi ^2}{2}$`

More generally, `\cm {...}` does `$...$`, and `\cmd {...}` does `\[...\]`.

6 \css – add a \

\css: adds a \: \css {abc} ↦ \abc

\css{makebox}\oargcss{width}\margcss{text}

produces

\makebox[⟨*width*⟩]{⟨*text*⟩}

:: \makebox[⟨*width*⟩]{⟨*text*⟩} :: same output using mcs{}{}{}{}{}

e.g., \makebox[4cm]{The cat sat on the mat.}

Using \mcsv {}{}{}{}{}, where the second argument is the option:

\makebox[4cm]{The cat sat on the mat.}

Unemphasised **Meta command**	**Result**
\css{⟨*cs-name*⟩}	\css {test} ↦ \test
\margcss{⟨*mandatory argument*⟩}	{⟨*test*⟩}
\margvcss{⟨*mandatory argument value*⟩}	{test}
\oargcss{⟨*optional argument*⟩}	[⟨*test*⟩]
\oargvcss{⟨*optional argument value*⟩}	[test]
\meta{⟨*meta value*⟩}	⟨*test*⟩

7 \cssom etc – show and run

\cssom: show and run command+opt+arg

\makebox[4cm]{The cat sat on the mat.}↦ The cat sat on the mat.

Using cdr:

\cssom {makebox}{4cm}{The cat sat on the mat.} ↦ \makebox[4cm]{The cat sat on the mat.}↦ The cat sat on the mat.

\cssm

\cssm: show and run command+arg

\textit{Some text}↦*Some text*

\fbox{This is framed text.}↦ This is framed text.

\cssmm

\cssmm: show and run command+arg+arg

\textcolor{red}{This is red.}↦This is red.

\cssmmm

\cssmmm: show and run command+arg+arg+arg

\fcolorbox{brown}{blue!20}{Some text in a framed box.}↦ Some text in a framed box.

8 \cdr – generic print and run

Print \#1 inline, and run \#1.

9 \cdrq – print in quotation and run

Print \#1 in display format, and run \#1.

10 \cdrd – print

Print \#1 inline.

This command takes options.

Special Case: When option [format=section] is active, and a second option `[`⟨*descriptive text*⟩`]` is specified (this will become the underlying submacro's argument #1), and the argument (will become #2) has no backslash (only #2, not \#2), the command prints `\#2 -- #1` as the section heading (and therefore table of contents) and sets the label as `\label{sec:#2}`.

The heading for this section was produced with

```
\cdrd [format=section][print]{cdrd}
```

List of `format=` options:

Option	Example
⟨*None*⟩	`\cdrd {\xyz }` ↦ `\xyz`
head	`\cdrd [format=head]{\xyz }` ↦ $\boxed{\texttt{\textbackslash xyz}}$
custom	`\cdrd [format=custom][\bfseries \tiny \sffamily ]{\xyz }` ↦ \xyz
section	See outside the table.
quote	`\cdrd [format=quote]{xyz demo This is a quote}`: See outside the table.
listing	<in development>
general	`\cdrd [format=general]{[format=xyz]}` ↦ `[format=xyz]`
detok	`\cdrd [format=detok]{\xyz }` ↦ `\xyz`

The `[format=section]` output does not fit in a table.
`\cdrd [format=section]{xyz demo}` ↦

11 \xyz demo –

Note that the label for §11 has been set as

```
\label {sec:xyz demo}
```

that is, with a space between `xyz` and `demo`.

```
xyz demo This is a quote
```

xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

Calling the listing environment from inside a command doesn't quite work:

```
atexcodex

qwerty: The listing environment never closes when used inside a command.

abc
```

[**latexcode** env from DOCTOOLS^{2012} package, adapted]

So for a listing, use the \begin {latexcode} . . . \end {latexcode} environment directly, with those commands on separate lines:

```
xxx
```

[**latexcode** env from DOCTOOLS^{2012} package, adapted]

The \cdr command can take abitrary code.

Example start »»

```
\bigskip \begin {quotation} \begin {tabular}{ll} \rowcolor {\mytheadcolour
} Option & Example \\ \hline None & \cdr {\cdrd {\xyz }}\\ Header &
\cdr {\cdrd [format=head]{\xyz }}\\ Custom & \cdr {\cdrd [format=custom][\bfseries
\tiny \sffamily ]{\xyz }}\\ Section & See outside the table. \\ \hline
\end {tabular} \end {quotation} ↦
```

Option	Example
None	`\cdrd {\xyz }` ↦ `\xyz`
Header	`\cdrd [format=head]{\xyz }` ↦ $\boxed{\texttt{\textbackslash xyz}}$
Custom	`\cdrd [format=custom][\bfseries \tiny \sffamily ]{\xyz }` ↦ **\xyz**
Section	See outside the table.

«« Example end

12 Empty arguments

{⟨⟩}
[⟨⟩]
{}
[]

13 Formatting

Delimiters can be reformatted to some extent.

```
    {⟨delimiter test⟩}
    \renewcommand \margdelimleftchar {+++}
    +++⟨delimiter test⟩}
    \renewcommand \margdelimrightformat {\itshape \Huge
 }
    +++⟨delimiter test⟩}
    \margreset
    {⟨delimiter test⟩}
```

Alternatively, just type the new code in the command definition directly.

List of some customisable components.

marg left

Item	Default value
\margdelimformat	\ttfamily \large \color {red}
\margdelimleftformat	\margdelimformat
\margdelimleftchar	\c _left_brace_str

\margdelimleft	{
	\margdelimleftformat
	\margdelimleftchar
	}

marg right

Item	Default value
\margdelimformat	\ttfamily \large \color {red}
\margdelimrightformat	\margdelimformat
\margdelimrightchar	\c _right_brace_str

\margdelimright	{
	\margdelimrightformat
	\margdelimrightchar
	}

oarg left

Item	Default value
\oargdelimformat	\ttfamily \large
\oargdelimleftformat	\oargdelimformat
\oargdelimleftchar	[

\oargdelimleft	{
	\oargdelimleftformat
	\oargdelimleftchar
	}

oarg right

Item	Default value
\oargdelimformat	\ttfamily \large
\oargdelimrightformat	\oargdelimformat
\oargdelimrightchar	]

\oargdelimright	{
	\oargdelimrightformat
	\oargdelimrightchar
	}

group
Current font text
\setmainfont {Noto Serif}
Current font text xxx
\addfontfeature {Colour=brown,Scale=2}
XXX
endgroup xxx

14 Listing using latexcode environment

```
    \documentclass{article}
    \usepackage{doctools}
    \begin{document}
    \begin{latexcode}
    (code)    % comment
\cdr{
\bigskip
\begin{quotation}
\begin{tabular}{ll}
\rowcolor{\mytheadcolour}
Option & Example \\
\hline
None & \cdr{\cdrd{\xyz}}\\
Header & \cdr{\cdrd[format=head]{\xyz}}\\
Custom & \cdr{\cdrd[format=custom][\bfseries\tiny\sffamily]{\xyz}}\\
Section & See outside the table. \\
\hline
\end{tabular}
\end{quotation}
}

    \ end{latexcode}%<--- note the space
    \end{document}
```

[**latexcode** env from DOCTOOLS^{2012} package, adapted]