

# latexindent.pl

## Version 3.0

Chris Hughes \*

January 18, 2017

### Abstract

`latexindent.pl` is a Perl script that indents `.tex` (and other) files according to an indentation scheme that the user can modify to suit their taste. Environments, including those with alignment delimiters (such as `tabular`), and commands, including those that can split braces and brackets across lines, are *usually* handled correctly by the script. Options for `verbatim`-like environments and indentation after headings (such as `chapter`, `section`, etc) are also available. The script also has the ability to modify line breaks, and add comment symbols.

### Contents

0.0.1	special code blocks . . . . .	1
-------	-------------------------------	---

### Listings

LISTING 1:	<code>displayMath-noAdd.yaml</code> . . . . .	1
LISTING 2:	<code>displayMath-indent-rules.yaml</code> . . . . .	1
LISTING 3:	<code>special1.tex</code> using Listing 1 . . . . .	2
LISTING 4:	<code>special1.tex</code> using Listing 2 . . . . .	2
LISTING 5:	<code>special-noAdd-glob.yaml</code> . . . . .	2
LISTING 6:	<code>special-indent-rules-global.yaml</code> . . . . .	2
LISTING 7:	<code>special1.tex</code> using Listing 5 . . . . .	2
LISTING 8:	<code>special1.tex</code> using Listing 6 . . . . .	2

#### 0.0.1 special code blocks

Let's use the example from ?? on page ?? which has default output shown in ?? on page ??.

It is recommended to specify `noAdditionalIndent` and `indentRules` in the 'scalar' form only for these type of code blocks, although the 'field' form would work, assuming that body was specified. Examples are shown in Listings 1 and 2.

LISTING 1: displayMath-noAdd.yaml	LISTING 2: displayMath-indent-rules.yaml
1 noAdditionalIndent:	1 indentRules:
2 displayMath: 1	2 displayMath: "\t\t\t"

After running the following commands,

```
cmh:~$ latexindent.pl special1.tex -local displayMath-noAdd.yaml
cmh:~$ latexindent.pl special1.tex -l displayMath-indent-rules.yaml
```

\*and contributors! (See ?? on page ??.) For all communication, please visit [[latexindent-home](#)].



we receive the respective output given in Listings 3 and 4; note that in Listing 3, the `displayMath` code block has *not* received any additional indentation, while in Listing 4, the `displayMath` code block has received three tabs worth of indentation.

LISTING 3: `special1.tex` using Listing 1

The function `$ f $` has formula

```
\[
f(x)=x^2.
\]
```

If you like splitting dollars,

```
$
g(x)=f(2x)
$
```

LISTING 4: `special1.tex` using Listing 2

The function `$ f $` has formula

```
\[
      f(x)=x^2.
\]
```

If you like splitting dollars,

```
$
      g(x)=f(2x)
$
```

We may specify `noAdditionalIndentGlobal` and `indentRulesGlobal` as in Listings 5 and 6.

LISTING 5:  
`special-noAdd-glob.yaml`

```
1 noAdditionalIndentGlobal:
2   specialBeginEnd: 1
```

LISTING 6:  
`special-indent-rules-global.yaml`

```
1 indentRulesGlobal:
2   specialBeginEnd: " "
```

Upon running the following commands

```
cmh:~$ latexindent.pl special1.tex -local special-noAdd-glob.yaml
cmh:~$ latexindent.pl special1.tex -l special-indent-rules-global.yaml
```

we receive the outputs in Listings 7 and 8; notice that in Listing 7 neither of the special code blocks have received indentation, while in Listing 8 both code blocks have received a single space of indentation.

LISTING 7: `special1.tex` using Listing 5

The function `$ f $` has formula

```
\[
f(x)=x^2.
\]
```

If you like splitting dollars,

```
$
g(x)=f(2x)
$
```

LISTING 8: `special1.tex` using Listing 6

The function `$ f $` has formula

```
\[
  f(x)=x^2.
\]
```

If you like splitting dollars,

```
$
  g(x)=f(2x)
$
```