

latexindent.pl

Version 3.0

Chris Hughes

February 1, 2017

`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 commands, together with 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. All user options are customisable via the switches in the YAML interface.

Contents

0.0.1	The remaining code blocks	1
0.0.2	Summary	3

Listings

LISTING 1: <code>pgfkeys1.tex</code>	2	LISTING 5: <code>psforeach1.tex</code>	2
LISTING 2: <code>pgfkeys1.tex</code> default output	2	LISTING 6: <code>psforeach1.tex</code> default output	2
LISTING 3: <code>child1.tex</code>	2	LISTING 7: <code>noAdditionalIndentGlobal</code>	3
LISTING 4: <code>child1.tex</code> default output	2	LISTING 8: <code>indentRulesGlobal</code>	3

0.0.1 The remaining code blocks

Referencing the different types of code blocks in ?? on page ??, we have a few code blocks yet to cover; these are very similar to the `commands` code block type covered comprehensively in ?? on page ??, but a small discussion defining these remaining code blocks is necessary.

keyEqualsValuesBraces `latexindent.pl` defines this type of code block by the following criteria:

- it must immediately follow either `{` OR `[` OR `,` with comments and blank lines allowed;
- then it has a name made up of the characters detailed in ?? on page ??;
- then an `=` symbol;
- then at least one set of curly braces or square brackets (comments and line breaks allowed throughout).

An example is shown in Listing 1, with the default output given in Listing 2.

and contributors! (See ?? on page ??.) For all communication, please visit [1].



LISTING 1: pgfkeys1.tex

```
\pgfkeys{/tikz/.cd,
start coordinate/.initial={0,
\vertfactor},
}
```

LISTING 2: pgfkeys1.tex default output

```
\pgfkeys{/tikz/.cd,
\start coordinate/.initial={0,
\ \ \ \ \vertfactor},
}
```

In Listing 2, note that the maximum indentation is three tabs, and these come from:

- the `\pgfkeys` command's mandatory argument;
- the `start coordinate/.initial` key's mandatory argument;
- the `start coordinate/.initial` key's body, which is defined as any lines following the name of the key that include its arguments. This is the part controlled by the `body` field for `noAdditionalIndent` and friends from ??.

namedGroupingBracesBrackets This type of code block is mostly motivated by `tikz`-based code; we define this code block as follows:

- it must immediately follow either *horizontal space* OR *one or more line breaks* OR `{` OR `[` OR `$`;
- the name may contain the characters detailed in ?? on page ??;
- then at least one set of curly braces or square brackets (comments and line breaks allowed throughout).

A simple example is given in Listing 3, with default output in Listing 4.

LISTING 3: child1.tex

```
\coordinate
child[grow=down]{
edge from parent[antiparticle]
node[above=3pt]{ $ C $ }
}
```

LISTING 4: child1.tex default output

```
\coordinate
child[grow=down]{
\edge from parent[antiparticle]
\node[above=3pt]{ $ C $ }
}
```

In particular, `latexindent.pl` considers `child`, `parent` and `node` all to be `namedGroupingBracesBrackets`¹. Referencing Listing 4, note that the maximum indentation is two tabs, and these come from:

- the `child`'s mandatory argument;
- the `child`'s body, which is defined as any lines following the name of the `namedGroupingBracesBrackets` that include its arguments. This is the part controlled by the `body` field for `noAdditionalIndent` and friends from ??.

UnNamedGroupingBracesBrackets occur in a variety of situations; specifically, we define this type of code block as satisfying the following criteria:

- it must immediately follow either `{` OR `[` OR `,` OR `&` OR `)` OR `(` OR `$`;
- then at least one set of curly braces or square brackets (comments and line breaks allowed throughout).

An example is shown in Listing 5 with default output give in Listing 6.

LISTING 5: psforeach1.tex

```
\psforeach{\row}{%
{
{3,2.8,2.7,3,3.1}},%
{2.8,1,1.2,2,3},%
}
```

LISTING 6: psforeach1.tex default output

```
\psforeach{\row}{%
\{
\ \ \ \ {3,2.8,2.7,3,3.1}},%
\{2.8,1,1.2,2,3},%
}
```

¹ You may like to verify this by using the `-tt` option and checking `indent.log`!



Referencing Listing 6, there are *three* sets of unnamed braces. Note also that the maximum value of indentation is three tabs, and these come from:

- the `\psforeach` command's mandatory argument;
- the *first* un-named braces mandatory argument;
- the *first* un-named braces *body*, which we define as any lines following the first opening `{` or `[` that defined the code block. This is the part controlled by the *body* field for `noAdditionalIndent` and friends from ??.

Users wishing to customise the mandatory and/or optional arguments on a *per-name* basis for the `UnNamedGroupingBracesBrackets` should use `always-un-named`.

`filecontents` code blocks behave just as `environments`, except that neither arguments nor items are sought.

0.0.2 Summary

Having considered all of the different types of code blocks, the functions of the fields given in Listings 7 and 8 should now make sense.

LISTING 7: `noAdditionalIndentGlobal`

```

247 noAdditionalIndentGlobal:
248   environments: 0
249   commands: 1
250   optionalArguments: 0
251   mandatoryArguments: 0
252   ifElseFi: 0
253   items: 0
254   keyEqualsValuesBraces: 0
255   namedGroupingBracesBrackets: 0
256   UnNamedGroupingBracesBrackets: 0
257   specialBeginEnd: 0
258   afterHeading: 0
259   filecontents: 0

```

LISTING 8: `indentRulesGlobal`

```

263 indentRulesGlobal:
264   environments: 0
265   commands: 0
266   optionalArguments: 0
267   mandatoryArguments: 0
268   ifElseFi: 0
269   items: 0
270   keyEqualsValuesBraces: 0
271   namedGroupingBracesBrackets: 0
272   UnNamedGroupingBracesBrackets: 0
273   specialBeginEnd: 0
274   afterHeading: 0
275   filecontents: 0

```