# Package innerscript v. 1.4a Implementation

Conrad Kosowsky

May 2025

kosowsky.latex@gmail.com

For easy, off-the-shelf use, type the following in your
document preamble and compile using LuaLaTeX:

`\usepackage{innerscript}`

**Overview**

The innerscript package optionally modifies four aspects of TeX's automatic math formatting to improve typesetting: (1) it adds extra space around relation and operation symbols in superscripts and subscripts; (2) it removes extra space around `\left`-`\right` delimiter pairs; (3) it adds extra space after right delimiters in certain situations; and (4) it forces `\left` and `\right` delimiters to completely cover their contents. Using LuaLaTeX is required.

---

This file documents the code for the innerscript package. It is not a user guide! If you are looking for instructions on how to load innerscript and what it can do for your document, see `innerscript-user-guide.pdf`, which is included with the innerscript installation and is available on CTAN. Section 1 begins with the package declaration and option processing. Sections 2 and 3 contain the code to adjust spacing between symbols in `\scriptstyle` and `\scriptscriptstyle` respectively, and Section 4 contains the code for adjusting the spacing around `\mathinner` subformulas. Section 5 handles spacing between `\mathclose` and `\mathord` atoms, and section 6 fills out the resizable delimiter heights. Version history appears at the end of the document.

## 1 Setup

We begin the implementation by declaring the package and defining a general informational macro. The first 60 lines of `innerscript.sty` are comments.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{innerscript}[2025/05/08 v. 1.4a]
\def\IS@info#1{\wlog{Package innerscript Info: #1}}
```

Check whether `\Umathordordspacing` is defined. If not, issue an error and `\endinput`. We normally can't print multiple spaces at once, so within a group, we change the space catcode to 12.

```
\ifx\Umathordordspacing\@undefined
```

---

Acknowledgements: Thanks to Sheldon Axler and Clea F. Rees for pointing a bug in a previous version of innerscript.

```
\bgroup
\catcode`\ =12\relax
\def\NoLuaTeXError{\GenericError{}%
{\MessageBreak\MessageBreak
Package innerscript error:%
\MessageBreak\MessageBreak
*******************\MessageBreak
*                 *\MessageBreak
*   CANNOT LOAD   *\MessageBreak
*   INNERSCRIPT   *\MessageBreak
*  LuaTeX Needed  *\MessageBreak
*                 *\MessageBreak
*******************\MessageBreak\@gobbletwo}%
{See the innerscript package documentation for explanation.}%
{I need LuaTeX to make the innerscript package work. It\MessageBreak
looks like the current engine is something else, so I\MessageBreak
can't load the package file. To use innerscript, please\MessageBreak
typeset with LuaLaTeX. To continue without innerscript,\MessageBreak
press return.}}%
\expandafter\egroup
\NoLuaTeXError
  \AtEndOfPackage{\IS@info{Failed to load \on@line!}}
  \expandafter\endinput % we \endinput with a balanced conditional
\fi
```

The conditionals will encode package option information.

```
\newif\ifIS@script
\newif\ifIS@scriptscript
\newif\ifIS@legacyscript
\newif\ifIS@legacyscriptscript
\newif\ifIS@inner
\newif\ifIS@close
\newif\ifIS@cover
```

And set default options.

```
\IS@scripttrue
\IS@scriptscripttrue
\IS@legacyscriptfalse
\IS@legacyscriptscriptfalse
\IS@innertrue
\IS@closetrue
\IS@covertrue
```

Now define and process the package options.

```
\DeclareOption{script}             {\IS@scripttrue
  \IS@legacyscriptfalse}
\DeclareOption{scriptscript}       {\IS@scriptscripttrue
  \IS@legacyscriptscriptfalse}
\DeclareOption{legacy-script}      {\IS@scripttrue
```

```
  \IS@legacyscripttrue}
\DeclareOption{legacy-scriptscript}{\IS@scriptscripttrue
  \IS@legacyscriptscripttrue}
\DeclareOption{inner}              {\IS@innertrue}
\DeclareOption{close}              {\IS@closetrue}
\DeclareOption{cover}              {\IS@covertrue}
\DeclareOption{no-script}          {\IS@scriptfalse
  \IS@legacyscriptfalse}
\DeclareOption{no-scriptscript}    {\IS@scriptscriptfalse
  \IS@legacyscriptscriptfalse}
\DeclareOption{no-inner}           {\IS@innerfalse}
\DeclareOption{no-close}           {\IS@closefalse}
\DeclareOption{no-cover}           {\IS@coverfalse}
\ProcessOptions*
```

If the user requested a legacy option for super or subscripts, issue a warning. We store the warning in `\@tempa`, and if both `\ifIS@legacyscript` and `\ifIS@legacyscriptscript` and false, we turn `\@tempa` into `\relax`. The `\@gobbletwo` eats `\on@line` and a period from `\PackageWarning`.

```
\def\@tempa{%
  \AtEndOfPackage{\PackageWarning{innerscript}
  {I'm ignoring your request for\MessageBreak
  legacy spacing\on@line\MessageBreak
  and using the default space\MessageBreak
  adjustments instead. (The\MessageBreak
  package options for legacy\MessageBreak
  spacing are deprecated.)\@gobbletwo}}}
\ifIS@legacyscript
\else
  \ifIS@legacyscriptscript
  \else
    \let\@tempa\relax
  \fi
\fi
\@tempa
```

## 2 Superscripts and Subscripts

First we set the spacing for `\scriptstyle` and `\scriptscriptstyle` atoms. Each space adjustment has the form

> `\Umath`⟨*classes*⟩`spacing`⟨*style*⟩`\muexpr`⟨*skip*⟩`*`⟨*factor*⟩`\relax`.

The ⟨*classes*⟩ are a pair of choices from `ord`, `op`, `bin`, `rel`, `open`, `close`, `punct`, and `inner`, and the ⟨*style*⟩ is either `\scriptstyle`, `\scriptscriptstyle`, or the cramped version of each style. The ⟨*skip*⟩ is the amount of muglue that typically appears between the ⟨*classes*⟩ while in `\textstyle` or `\displaystyle`. The package takes that muglue and scales it by 0.6

**Table 1: Space Inserted between Atoms**

| Consecutive Atom Types | Default space added |
|---|---|
| `\mathord\mathop` | `\thinmuskip` |
| `\mathord\mathbin` | `\medmuskip` |
| `\mathord\mathrel` | `\thickmuskip` |
| `\mathord\mathinner` | `\thinmuskip` |
| `\mathop\mathord` | `\thinmuskip` |
| `\mathop\mathop` | `\thinmuskip` |
| `\mathop\mathrel` | `\thickmuskip` |
| `\mathop\mathinner` | `\thickmuskip` |
| `\mathbin\mathord` | `\medmuskip` |
| `\mathbin\mathop` | `\medmuskip` |
| `\mathbin\mathopen` | `\medmuskip` |
| `\mathbin\mathinner` | `\medmuskip` |
| `\mathrel\mathord` | `\thickmuskip` |
| `\mathrel\mathop` | `\thickmuskip` |
| `\mathrel\mathopen` | `\thickmuskip` |
| `\mathrel\mathinner` | `\thickmuskip` |
| `\mathclose\mathop` | `\thinmuskip` |
| `\mathclose\mathbin` | `\medmuskip` |
| `\mathclose\mathrel` | `\thickmuskip` |
| `\mathclose\mathinner` | `\thinmuskip` |
| `\mathpunct\mathord` | `\thinmuskip` |
| `\mathpunct\mathop` | `\thinmuskip` |
| `\mathpunct\mathrel` | `\thinmuskip` |
| `\mathpunct\mathopen` | `\thinmuskip` |
| `\mathpunct\mathclose` | `\thinmuskip` |
| `\mathpunct\mathpunct` | `\thinmuskip` |
| `\mathpunct\mathinner` | `\thinmuskip` |
| `\mathinner\mathord` | `\thinmuskip` |
| `\mathinner\mathop` | `\thinmuskip` |
| `\mathinner\mathbin` | `\medmuskip` |
| `\mathinner\mathrel` | `\thickmuskip` |
| `\mathinner\mathopen` | `\thinmuskip` |
| `\mathinner\mathpunct` | `\thinmuskip` |
| `\mathinner\mathinner` | `\thinmuskip` |

for superscripts and subscripts and by 0.4 for second-order superscripts and subscripts. Table 1 shows the usual amounts of space between consecutive atom types. To actually code it, we define `\@tempa` to expand to various space-adjustment commands depending on package options.

```
\ifIS@script         % change \scriptstyle
  \IS@info{Adjusting space for \string\scriptstyle.}
  \IS@info{Adjusting space for \string\crampedscriptstyle.}
  \ifIS@scriptscript % change \scriptscriptstyle
    \IS@info{Adjusting space for \string\scriptscriptstyle.}
    \IS@info{Adjusting space for \string\crampedscriptscriptstyle.}
```

In this branch, `\@tempa` expands to space-adjustment commands for all four styles of superscripts and subscripts. The `#1` argument is a `\Umath` spacing primitive, and `#2` is a muglue register.

```
    \def\@tempa#1#2{%
      #1\scriptstyle             \muexpr #2 * 6 / 10\relax
      #1\crampedscriptstyle      \muexpr #2 * 6 / 10\relax
      #1\scriptscriptstyle       \muexpr #2 * 4 / 10\relax
      #1\crampedscriptscriptstyle\muexpr #2 * 4 / 10\relax}
  \else              % do not change \scriptscriptstyle
    \IS@info{No space changes for \string\scriptscriptstyle.}
    \IS@info{No space changes for \string\crampedscriptscriptstyle.}
```

Now change just first-order superscripts and subscripts.

```
    \def\@tempa#1#2{%
      #1\scriptstyle             \muexpr #2 * 6 / 10\relax
      #1\crampedscriptstyle      \muexpr #2 * 6 / 10\relax}
  \fi
\else                  % do not change \scriptstyle
  \IS@info{No space changes for \string\scriptstyle.}
  \IS@info{No space changes for \string\crampedscriptstyle.}
  \ifIS@scriptscript % change \scriptscriptstyle
    \IS@info{Adjusting space for \string\scriptscriptstyle.}
    \IS@info{Adjusting space for \string\crampedscriptscriptstyle.}
```

In this case, change just second-order superscripts and subscripts.

```
    \def\@tempa#1#2{%
      #1\scriptscriptstyle       \muexpr #2 * 4 / 10\relax
      #1\crampedscriptscriptstyle\muexpr #2 * 4 / 10\relax}
  \else              % do not change \scriptscriptstyle
    \IS@info{No space changes for \string\scriptscriptstyle.}
    \IS@info{No space changes for \string\crampedscriptscriptstyle.}
```

And here change nothing.

```
    \let\@tempa\@gobbletwo
  \fi
\fi
```

Now actually make the space changes as applicable.

```
  \@tempa{\Umathordopspacing}     {\thinmuskip}
  \@tempa{\Umathordbinspacing}    {\medmuskip}
  \@tempa{\Umathordrelspacing}    {\thickmuskip}
  \@tempa{\Umathordinnerspacing}  {\thinmuskip}
  \@tempa{\Umathopordspacing}     {\thinmuskip}
  \@tempa{\Umathopopspacing}      {\thinmuskip}
  \@tempa{\Umathoprelspacing}     {\thickmuskip}
  \@tempa{\Umathopinnerspacing}   {\thickmuskip}
  \@tempa{\Umathbinordspacing}    {\medmuskip}
  \@tempa{\Umathbinopspacing}     {\medmuskip}
  \@tempa{\Umathbinopenspacing}   {\medmuskip}
  \@tempa{\Umathbininnerspacing}  {\medmuskip}
  \@tempa{\Umathrelordspacing}    {\thickmuskip}
  \@tempa{\Umathrelopspacing}     {\thickmuskip}
  \@tempa{\Umathrelopenspacing}   {\thickmuskip}
  \@tempa{\Umathrelinnerspacing}  {\thickmuskip}
  \@tempa{\Umathcloseopspacing}   {\thinmuskip}
  \@tempa{\Umathclosebinspacing}  {\medmuskip}
  \@tempa{\Umathcloserelspacing}  {\thickmuskip}
  \@tempa{\Umathcloseinnerspacing}{\thinmuskip}
  \@tempa{\Umathpunctordspacing}  {\thinmuskip}
  \@tempa{\Umathpunctopspacing}   {\thinmuskip}
  \@tempa{\Umathpunctrelspacing}  {\thinmuskip}
  \@tempa{\Umathpunctopenspacing} {\thinmuskip}
  \@tempa{\Umathpunctclosespacing}{\thinmuskip}
  \@tempa{\Umathpunctpunctspacing}{\thinmuskip}
  \@tempa{\Umathpunctinnerspacing}{\thinmuskip}
  \@tempa{\Umathinnerordspacing}  {\thinmuskip}
  \@tempa{\Umathinneropspacing}   {\thinmuskip}
  \@tempa{\Umathinnerbinspacing}  {\medmuskip}
  \@tempa{\Umathinnerrelspacing}  {\thickmuskip}
  \@tempa{\Umathinneropenspacing} {\thinmuskip}
  \@tempa{\Umathinnerpunctspacing}{\thinmuskip}
  \@tempa{\Umathinnerinnerspacing}{\thinmuskip}
```

## 3 Mathclose Spacing

In `\textstyle` and `\displaystyle`, we add half a `\thinmuskip` between `\mathclose` and `\mathord` atoms. As with the `script` and `scriptscript` options, we scale the spacing by 0.6 and 0.4 for `\scriptstyle` and `\scriptscriptstyle` respectively. Even though `close` comes after `inner` in the options list in the user guide, it is important to put it before `inner` in the implementation. We should set all interatom spacing before changing the spacing around `\mathinner` subformulas because the `inner` spacing changes reference values of other `\Umath` spacing primitives.

```
\ifIS@close
  \IS@info{Adding space after closing grouping symbols.}
  \Umathcloseordspacing\displaystyle        \muexpr\thinmuskip / 2\relax
  \Umathcloseordspacing\textstyle           \muexpr\thinmuskip / 2\relax
  \Umathcloseordspacing\crampeddisplaystyle \muexpr\thinmuskip / 2\relax
  \Umathcloseordspacing\crampedtextstyle    \muexpr\thinmuskip / 2\relax
  \ifIS@script
```

```
    \Umathcloseordspacing\scriptstyle       \muexpr\thinmuskip * 3 / 10\relax
    \Umathcloseordspacing\crampedscriptstyle\muexpr\thinmuskip * 3 / 10\relax
  \fi
  \ifIS@scriptscript
    \Umathcloseordspacing\scriptscriptstyle \muexpr\thinmuskip / 5\relax
    \Umathcloseordspacing\crampedscriptscriptstyle
      \muexpr\thinmuskip / 5\relax
  \fi
\else
  \IS@info{No changes to space after delimiters.}
\fi
```

## 4 Mathinner Subformulas

Now set the spacing for `\mathinner` subformulas. Prior to version 1.4, innerscript spaced `\mathinner` subformulas under the `inner` option like `\mathord` atoms, but as of version 1.4, innerscript treats these subformulas like `\mathopen` on the left and `\mathclose` on the right. Every space adjustment in this section has the form

> `\Umath`⟨*classes*⟩`spacing`⟨*style*⟩`=\the\Umath`⟨*other classes*⟩`spacing`⟨*style*⟩`\relax`.

In most cases, the ⟨*classes*⟩ are a pair of `inner` and one of `ord`, `op`, `bin`, `rel`, `open`, `close`, or `punct`, and the ⟨*other classes*⟩ are the same except with `inner` replaced with `open` or `close` depending on whether it appears on the left or the right. We can also have ⟨*classes*⟩ be `innerinner`, in which case ⟨*other classes*⟩ is `closeopen`. The ⟨*style*⟩ is one of the eight math style commands: `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`, and the cramped versions. This section comes after `script` and `close` in the implementation because the spacing here depends on the spacing of other math classes.

```
\ifIS@inner
  \IS@info{Adjusting space around \string\mathinner\space subformulas.}
```

First, we loop through the math classes and let `\@i` be a math class.

```
  \@tfor\@i:=\displaystyle\crampeddisplaystyle
    \textstyle\crampedtextstyle
    \scriptstyle\crampedscriptstyle
    \scriptscriptstyle\crampedscriptscriptstyle
    \do{%
```

For each style, first set `\Umathinnerinnerspacing`, and then loop through the math classes. We simultaneously set the spacing for atom pairs where `inner` appears either first or second. Using `\expanded` solves a problem where the `\Umath`⟨*atom*⟩`inner` primitives don't behave properly if they appear on both sides of an assignment. I am unclear if this is a bug in LuaTeX?

```
      \expanded{\Umathinnerinnerspacing\@i=\the\expandafter
        \Umathcloseopenspacing\@i\relax}
      \@for\@j:=ord,op,bin,rel,open,close,punct\do{%
        \expanded{%
```

```
          \csname Umath\@j innerspacing\endcsname\@i=\the
            \csname Umath\@j openspacing\expandafter\endcsname\@i\relax
          \csname Umathinner\@j spacing\endcsname\@i=\the
            \csname Umathclose\@j spacing\expandafter\endcsname\@i\relax}}}
\else
  \IS@info{No changes to space for \string\mathinner\space subformulas.}
\fi
```

# 5 Delimiter Heights

Finally, filling out the delimiter heights is easy. We set `\delimiterfactor` to 1000.

```
\ifIS@cover
  \IS@info{Setting delimiters to full height.}
  \delimiterfactor\@m
\else
  \IS@info{No changes to delimiter heights.}
\fi
\wlog{}
```

And that’s a wrap!

# Version History

**1.0** . . . . . . . . . . . . . . . . . . . . . . February 2021
—initial release

**1.1** . . . . . . . . . . . . . . . . . . . . . . February 2021
—bug fix for `\mathinner` spacing
—added `\IS@skip` muglue register

**1.2** . . . . . . . . . . . . . . . . . . . . . November 2023
—redesigned spacing for options `script` and `scriptscript`
—no more `\IS@skip`—options `script` and `scriptscript` now use `\the`
—added option `close`
—added option `cover`
—separated implementation and user guide
—added `\scalemu`

**1.3** . . . . . . . . . . . . . . . . . . . . . . . . August 2024
—bug fix involving `\cramedisplaystyle` typo

**1.4** . . . . . . . . . . . . . . . . . . . . . . . . . . April 2025
—refactored package code
—`legacy-` package options for superscripts and subscripts now same as default options
—use `\muexpr` instead of `\scalemu`
—`inner` option changed to look more like `\mathopen` and `\mathclose` atoms

**1.4a** . . . . . . . . . . . . . . . . . . . . . . . . . May 2025
—bug fix involving `inner` adjustments not registering properly

# Index