LibreOffice
Community

Math Guide

7.2

Copyright

This document is Copyright © 2021 by the LibreOffice Documentation Team. Contributors are listed below. You may distribute it and/or modify it under the terms of either the GNU General Public License (https://www.gnu.org/licenses/gpl.html), version 3 or later, or the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), version 4.0 or later.

All trademarks within this guide belong to their legitimate owners.

Contributors

To this edition

Jean Hollis Weber   Rafael Lima

To previous editions

Rafael Lima	Olivier Hallot	Jean Hollis Weber
Roman Kuznetsov	Dave Barton	Regina Henschel
Peter Schofield	Hazel Russman	Laurent Balland-Poirier
Jean Hollis Weber	Daniel Carrera	Agnes Belzunce
T. J. Frazier	Peter Kupfer	Ian Laurenson
Janet M. Swisher	Michele Zarri	Florian Reisinger
Christian Kühl	Jochen Schiffers	Frédéric Parrenin
Gisbert Friege (Dmaths)	Bernard Siaud	

LibreOffice 7.x Guide Series cover design by Rizal Mutaqin.

Feedback

Please direct any comments or suggestions about this document to the Documentation Team's mailing list: documentation@global.libreoffice.org. This mailing list is meant for comments on LibreOffice documentation and should not be used for other purposes as bug reporting or to question about LibreOffice functionalities.

✅ **Note**

> Everything you send to a mailing list, including your email address and any other personal information that is written in the message, is publicly archived and cannot be deleted.

Publication date and software version

Published [Month] 2021. Based on LibreOffice 7.2 Community.
Other versions of LibreOffice may differ in appearance and functionality.

Contents

Math Guide 7.2

Preface

Who is this book for?

LibreOffice Math is a formula (equation) editor that is an integral part of LibreOffice. Anyone who wants to learn how to insert formulas and equations using Math will find this guide valuable.

In LibreOffice, formulas can be inserted as objects into Writer, Impress, Draw, and Calc documents. Regardless of the document type, formula objects are edited using LibreOffice Math.

If you want an introduction to all of the LibreOffice components, you might like to read the *Getting Started Guide* first.

> ✅ **Note**
>
> LibreOffice Math is not capable of evaluating mathematical equations or performing calculations. For that purpose, you should use LibreOffice Calc; see the *Calc Guide*.

Where to get more help

This guide, the other LibreOffice user guides, the Help system, and user support systems assume that you are familiar with your computer and its operating system's basic functions such as starting a program, opening and saving files.

Help system

LibreOffice comes with an extensive online Help system. This is your first line of support. Windows and Linux users can choose to download and install the Help for use when not connected to the Internet; the offline Help is installed with the program on macOS.

To display the Help system, press *F1* or go to **Help > LibreOffice Help** on the Menu bar. If you do not have the offline help installed on your computer and you are connected to the Internet, your default browser will open the online Help pages on the LibreOffice website.

The Help menu also includes links to other LibreOffice information and support facilities. The options marked by a ‡ sign in the list below are only accessible if your computer is connected to the Internet.

- **What's This?** For quick tips when a toolbar is visible, place the mouse pointer over any of the icons to see a small box ("tooltip") with a brief explanation of the icon's function. For a more detailed explanation, select **Help > What's This?** and hold the pointer over the icon. In addition, you can choose whether to activate Extended Tips using **Tools > Options > LibreOffice > General**.

- **User Guides** ‡ Opens your default browser at the Documentation page of the LibreOffice website, https://documentation.libreoffice.org/en/english-documentation/. There you will find copies of user guides and other useful information.

- **Show Tip of the Day** Opens a small window with a random tip on how to use LibreOffice.

- **Search Commands** In Writer, Calc, Impress, and Draw (but not Base or Math), opens a window where you can type a few letters or the name of a Menu bar command, to quickly find where the command is located. Clicking on a command in the resulting list may open a relevant dialog or have other effects.

- **Get Help Online** ‡ Opens your default browser at the Ask LibreOffice forum of questions and answers from the LibreOffice community, https://ask.libreoffice.org/en/questions/.

- **Send Feedback** ‡ Opens your default browser at the Feedback page of the LibreOffice website, https://www.libreoffice.org/get-help/feedback/. From there you can report bugs, suggest new features and communicate with others in the LibreOffice community.

- **Restart in Safe Mode** Opens a dialog where you can restart LibreOffice and reset the software to its default settings.

- **Get Involved** ‡ Opens your default browser at the Get Involved page of the LibreOffice website, https://www.libreoffice.org/community/get-involved/. There you can choose a topic of interest to help improve the program.

- **Donate to LibreOffice** ‡ Opens your default browser at the Donation page of the LibreOffice website, https://donate.libreoffice.org/.

- **License Information** Outlines the licenses under which LibreOffice is made available.

- **Check for Updates** ‡ Opens a dialog and checks the LibreOffice website for updates to your version of the software.

- **About LibreOffice** Opens a dialog and displays information about the version of LibreOffice and the operating system you are using. This information will often be requested if you ask the community for help or assistance with the software. A button is provided to enable you to copy this information to the clipboard so that you can subsequently paste it into a forum post, an email, or a bug report. (On macOS, this item is under **LibreOffice** on the Menu bar.)

Other free online support

The LibreOffice community not only develops software, but provides free, volunteer-based support. In addition to the Help menu links above, other online community support options are available; see Table 1 and this web page: https://www.libreoffice.org/get-help/.

Table 1: Free support for LibreOffice users

Free LibreOffice support	
FAQs	Answers to frequently asked questions https://wiki.documentfoundation.org/Faq
Mailing lists	Free community support is provided by a network of experienced users https://www.libreoffice.org/get-help/mailing-lists/
Questions & Answers and Knowledge Base	Free community assistance is provided in a Question & Answer format. Search similar topics or ask a new question in https://ask.libreoffice.org/en/questions The service is available in several other languages; just replace /en/ with de, es, fr, ja, ko, nl, pt, tr, and many others in the web address above.
Native language support	The LibreOffice website in various languages https://www.libreoffice.org/community/nlc/ Mailing lists for native languages https://wiki.documentfoundation.org/Local_Mailing_Lists Information about social networking https://wiki.documentfoundation.org/Website/Web_Sites_services
Accessibility options	Information about available accessibility options https://www.libreoffice.org/get-help/accessibility/

Paid support and training

You can also pay for support through service contracts from a vendor or consulting firm specializing in LibreOffice. For information about certified professional support, see The Document Foundation's website: https://www.documentfoundation.org/gethelp/support/.

For schools, educational and research institutions, and large organizations, see
https://www.libreoffice.org/download/libreoffice-in-business/.

What you see may be different

LibreOffice runs on Windows, Linux, and macOS operating systems, each of which has several versions and can be customized by users (fonts, colors, themes, window managers).

Illustrations

The illustrations in this guide were taken from the Windows 10 operating system. Therefore, some illustrations will not look exactly like what you see on your computer display.

Also, some of the dialogs may be different because of the settings selected in LibreOffice. In some cases (mainly Open, Save, and Print dialogs), you can choose to use dialogs from your computer's operating system or dialogs provided by LibreOffice. To change which dialogs are used, go to **Tools > Options > LibreOffice > General** (on Windows and Linux) or **LibreOffice > Preferences > General** (on macOS) and select or deselect the option **Use LibreOffice dialogs**.

Icons

The LibreOffice community has created icons for several icon sets: Breeze, Colibre, Elementary, Karasa Jaga, Sifr, and Sakapura; some are also available in a dark version. As a user, you can select your own preferred set. The icons in this guide have been taken from a LibreOffice installation that has been set to display the default set of icons. The icons you see for some of the many tools available in LibreOffice may differ from the ones used in this guide.

To change the icon set used, go to **Tools > Options > LibreOffice > View**. In the *Icon style* section, choose from the drop-down list.

> ### ✅ Notes
>
> Some Linux distributions include LibreOffice as part of the installation and may not include all the icon sets mentioned above. You should be able to download other icon sets from the software repository for your Linux distribution if you wish to use them.
>
> The Galaxy, Oxygen, and Tango icon sets are no longer included as part of the standard installation package. You can download and install them as extensions from https://extensions.libreoffice.org/en/extensions/.

Using LibreOffice on macOS

Some keystrokes and menu items are different on macOS from those used in Windows and Linux. The table below gives some common substitutions for the instructions in this book. For a more detailed list, see the application Help.

Windows or Linux	macOS equivalent	Effect
Tools > Options menu selection	**LibreOffice > Preferences**	Access setup options
Right-click	*Control*+click or right-click depending on computer setup	Open a context menu
Ctrl (Control)	⌘ (Command)	Used with other keys
F11	⌘+*T*	Open the Styles deck in the Sidebar

What is new in LibreOffice Math 7.2 Community?

LibreOffice Math 7.2 Community includes many improvements not visible in the user interface, including support for MathML custom entities and improved fastparser.

This user guide has been updated from *Math Guide 7.0*. It covers some of the new features in LibreOffice 7.1 and 7.2 that are visible in the user interface, including:

- Math has improved color support, including MathML, and support for HTML (hex) colors as well as RGB and named colors. See "Changing formula color" on page 31.
- The code input box in the Formula Editor can now be scaled; see "Scaling of code in the Formula Editor input window" on page 53.
- New commands for frac and evaluate; see Appendix A, Commands Reference.

Math Guide 7.2

Chapter 1,
Creating and Editing Formulas

Introduction

Math is a formula editor included with LibreOffice that you can use to create or edit formulas (equations) in a symbolic form, within LibreOffice documents or as stand-alone objects. Example formulas are shown below:

$$\frac{df(x)}{dx} = \ln(x) + \tan^{-1}(x^2) \quad or \quad NH_3 + H_2O \rightleftharpoons NH_4^+ + OH^-$$

The Formula Editor in Math uses a markup language to represent formulas. This markup language is designed to be easily read wherever possible. For example, `a over b` using markup language produces the fraction $\frac{a}{b}$ when used in a formula.

Getting started

Using the Formula Editor, you can create a formula as a separate file for a formula library or insert formulas directly into a document using LibreOffice Writer, Calc, Impress, or Draw.

Formulas as separate documents or files

To create a formula as a separate document or file, use one of the following methods to open an empty formula document in LibreOffice Math (Figure 1).

- On the Menu bar, go to **File > New > Formula**.
- On the Standard toolbar, click the triangle to the right of the **New** icon and select **Formula**.
- From the Start Center, click **Math Formula**.
- From within LibreOffice Math, use the keyboard shortcut *Ctrl+N*.
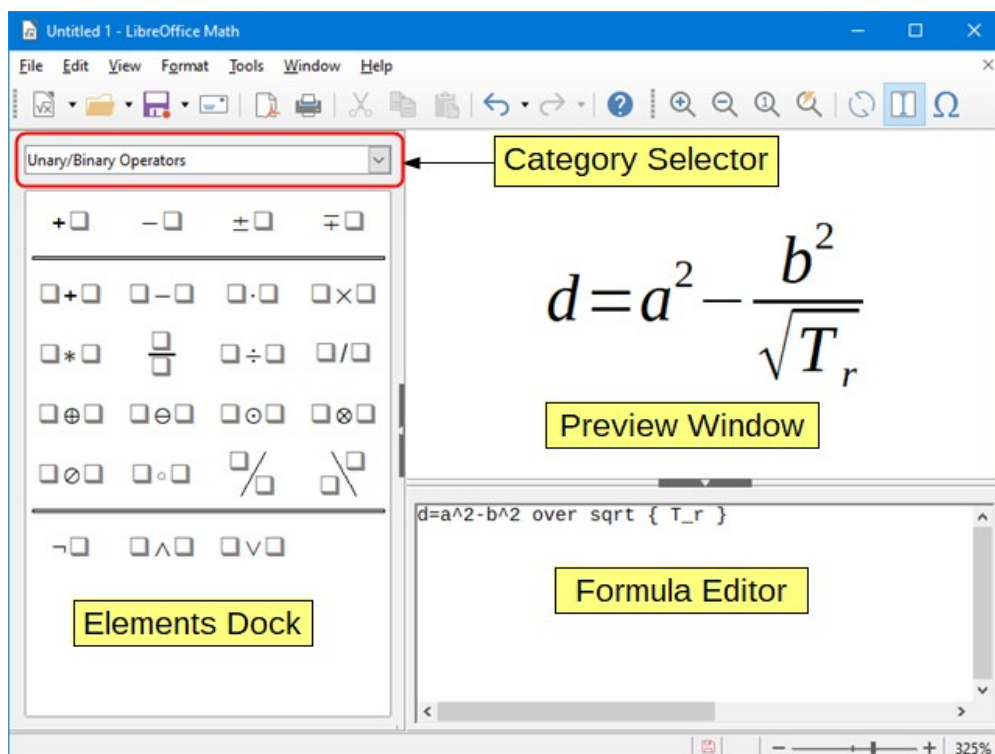- You can also launch Math from the command line using `libreoffice --math`

Figure 1: An empty formula document in Math

As you enter the markup language in the Formula Editor, the formula will appear in the Preview window during and after input of the markup language. The Elements dock to the left of the Preview window may also appear, if it has been selected in **View** on the Menu bar (Figure 2). This dock can be turned into a floating dialog, as shown in Figure 4.

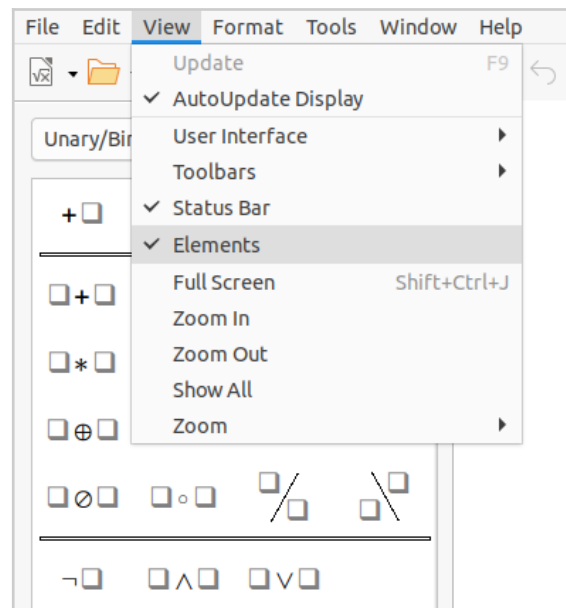For more information on creating formulas, see "Creating formulas" on page 13.

Figure 2: Enabling the Elements dock

Formulas in LibreOffice documents

To insert a formula into a LibreOffice document, open the document in Writer, Calc, Draw, or Impress. The LibreOffice component you are using affects how you position the cursor to insert the formula.

- In Writer, click in the paragraph where you want to insert the formula.
- In Calc, click in the spreadsheet cell where you want to insert the formula.
- In Draw and Impress, the formula is inserted into the center of the drawing or slide.

Then, go to **Insert > Object > Formula Object** on the Menu bar to open the Formula Editor. Alternatively, go to **Insert > Object > OLE Object** on the Menu bar to open the Insert OLE Object dialog, then select **Create new**, choose the **Object Type** "LibreOffice 7.2 Formula" and then click **OK** to open the Formula Editor.

The Elements dock to the left of the Preview window or the floating Elements dialog may also appear, if Elements has been selected in **View** on the Menu bar. For more information on creating formulas, see "Creating formulas" on page 13.

Figure 3 shows an example Writer document with the formula box selected ready for a formula to be entered.

When you have completed entering the markup language for the formula, close the Formula Editor by pressing the *Esc* key or by clicking an area outside the formula in the document. Double-click on the formula object in the document to open the Formula Editor again so that you can edit the formula.

Formulas are inserted as OLE objects into documents. You can change how the object is placed within the document, as with any OLE object. For more information on OLE objects, see Chapter 2, Formulas in Writer, Chapter 3, Formulas in Calc, Draw, and Impress, and the user guides for Writer, Calc, Draw, and Impress.
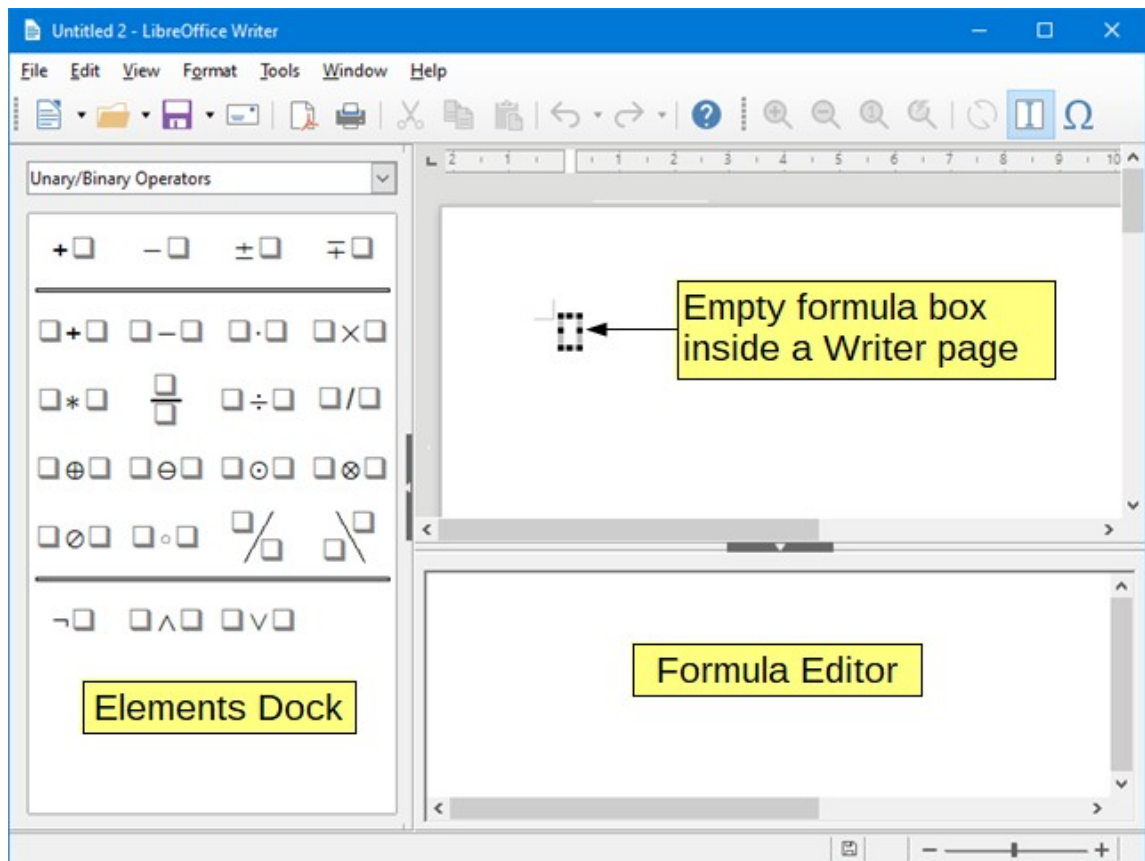
Figure 3: Empty formula in a Writer document

If you frequently insert formulas into documents, it is recommended to add the Formula button to the Standard toolbar or create a keyboard shortcut. See "Adding keyboard shortcuts" on page 47 for more information.

Creating formulas

You can create a formula using one of the following methods:

- In the Elements dock, select a category from the drop-down list, then a symbol.
- Right-click in the Formula Editor and select a category, then select a symbol in the context menu.
- Enter markup language directly in the Formula Editor.

✅ **Note**

Using the Elements dock or the context menus to create a formula provides a convenient way to learn the markup language used by LibreOffice Math.

ℹ️ **Tip**

When using the Elements dock, it is recommended to have *Extended Tips* selected in the LibreOffice Options. This will help you identify the categories and symbols you can use in the formula. Go to **Tools > Options > LibreOffice > General** on the Menu bar and select **Extended Tips** in the *Help* section.

Elements dock

The Elements dock (Figure 4) is a visual tool to help you create and edit formulas. It organizes mathematical symbols and operators using Categories in a drop-down list. The steps below describe how to enable the Elements dock and navigate its categories and symbols.

1) Go to **View > Elements** on the Menu bar to open the Elements dock.

2) Select the category you want to use in your formula from the drop-down list at the top of the Elements dock.

3) Select the symbol you want to use in your formula in the Elements dock. The symbols that are available change according to the selected category.

4) After choosing one of the symbols in the Elements dock, the Formula Editor will be updated with the Markup notation of the selected symbol.
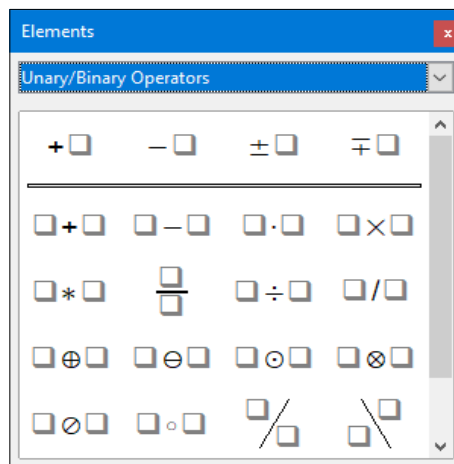
Figure 4: Elements dock as a floating dialog

> ✅ **Note**
>
> The Elements dock can either be a floating dialog, as shown in Figure 4, or positioned to the left of the Formula Editor, as shown in Figure 1 and Figure 3.

> ℹ️ **Tip**
>
> The Elements dock includes an Examples category which provides example formulas to use as a starting point for your formula or equation.

Context menu

The Formula Editor also provides a context menu to access categories and symbols when creating a formula. Right-click in the Formula Editor to open the context menu. Select a category and then select the markup example that you want to use in the sub-context menu. An example is shown in Figure 5.

> ✅ **Note**
>
> The Elements dock and the context menu contain only the most common commands that are used in formulas. To insert other symbols and commands not listed in the Elements dock and context menu, you have to enter them manually using the markup language. For a complete list of commands and symbols available in Math, see Appendix A, Commands Reference.

Figure 5: Context menu in Formula Editor

Markup language

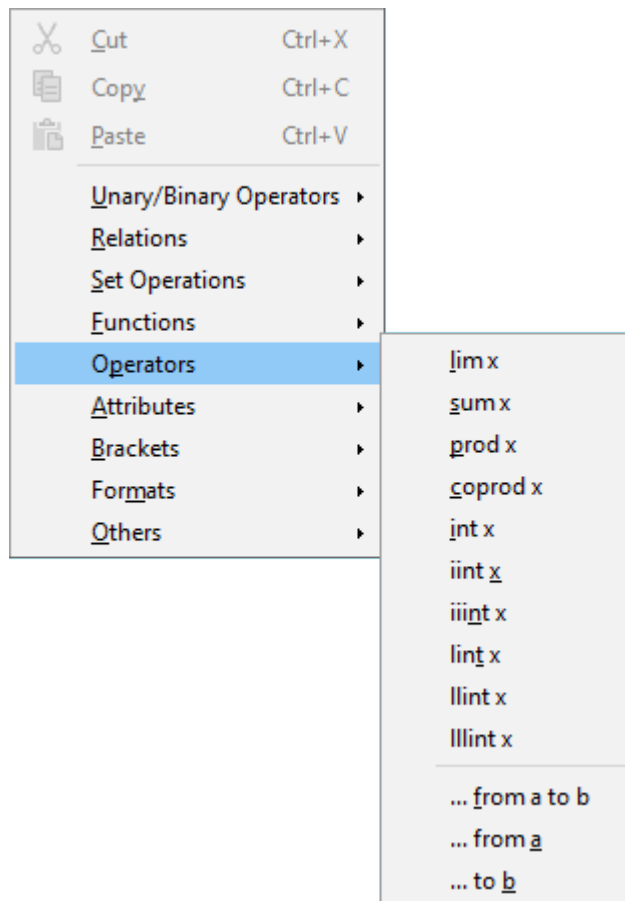Markup language is entered directly into the Formula Editor. For example, typing `5 times 4` into the Formula Editor creates the simple formula 5×4 . If you are experienced in using markup language, it can be the quickest way to enter a formula. Table 2 shows some examples of using markup language to enter commands. For a full list of commands that can be used in the Formula Editor, see *Appendix A, Commands Reference*.

Table 2: Example commands using markup language

Display	Command	Display	Command
$a = b$	`a = b`	$\sqrt{a}$	`sqrt {a}`
a^2	`a^2`	a_n	`a_n`
$\int f(x)dx$	`int f(x) dx`	$\sum a_n$	`sum a_n`
$a \leq b$	`a <= b`	∞	`infinity`
$a \times b$	`a times b`	$x \cdot y$	`x cdot y`

Greek characters

Using markup language

Greek characters are commonly used in formulas, but they cannot be entered into a formula using the Elements dock or the context menu. Use the English names of Greek characters in markup language when entering Greek characters into a formula. See *Appendix A, Commands Reference* for a list of characters that can be entered using markup language.

- For a lowercase Greek character, type a percentage % sign, then type the character name in lowercase using the English name. For example, typing %lambda creates the Greek character λ.

- For an UPPERCASE Greek character, type a percentage % sign, then type the character name in UPPERCASE using the English name. For example, typing %LAMBDA creates the Greek character Λ.

- For an *italic* Greek character, type a percentage % sign followed by the i character, then the English name of the Greek character in lower or UPPER case. For example, typing %iTHETA creates the *italic* Greek character Θ.

Symbols dialog

Greek characters can also be entered into a formula using the Symbols dialog.

1) Make sure your cursor is in the correct position in the Formula Editor.

2) Go to **Tools > Symbols** on the Menu bar, or click the **Symbols** icon in the Tools toolbar, to open the Symbols dialog (Figure 6)

3) Select *Greek* in the **Symbol set** drop-down list. For *italic* characters, select *iGreek* in the drop-down list.

4) Double-click the Greek character you want to insert, or select it and click **Insert**. When selected, the name of the character is shown below the symbol list.

5) Click **Close** when you have finished entering Greek characters into your formula.
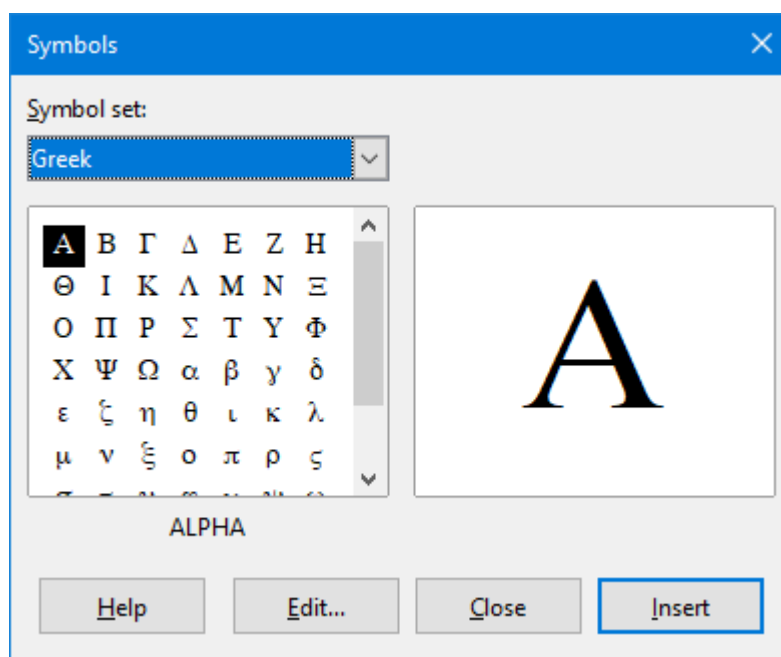
Figure 6: Symbols dialog

Formula examples

Example 1

The simple formula 5×4 can be created using LibreOffice Math as follows:

1) Make sure your cursor is flashing in the Formula Editor, then select the category **Unary/Binary Operators** and symbol **Multiplication** using one of the following methods:

   – In the Elements dock, select **Unary/Binary Operators** from the drop-down list and then select the **Multiplication** icon $\square \times \square$.

- Right-click in the Formula Editor and select **Unary/Binary Operators > a times b** from the context menu.
- Using markup language, type `5 times 4` in the Formula Editor.

  The first two methods place the formula text `<?> times <?>` in the Formula Editor and the symbol $\square \times \square$ appears in the document. For the third method, using markup language in the Formula Editor places the formula 5×4 directly into the document and there is no need to carry out the following steps.

2) Select the first placeholder `<?>` before the word `times` in the Formula Editor and replace it with the character `5`. The formula in your document updates automatically.

3) Select the second placeholder `<?>` after the word `times` in the Formula Editor and replace it with the character `4`. The formula in your document updates automatically.

> **Tip**
>
> To move forward from one placeholder to the next placeholder in a formula, press the *F4* key. To move backward from one placeholder to the previous placeholder in a formula, use the key combination *Shift+F4*.

> **Note**
>
> If necessary, you can prevent a formula in a document from updating automatically. Go to **View** on the Menu bar and deselect **AutoUpdate Display**. To then manually update a formula, press the *F9* key or select **View > Update** on the Menu bar.

Example 2

You want to enter the formula $\pi \simeq 3.14159$, where the value of pi is rounded to 5 decimal places. You know the name of the Greek character (pi), but do not know the markup associated with the `Is Similar Or Equal` symbol $\simeq$.

1) Make sure your cursor is flashing in the Formula Editor.

2) Enter `%pi` in the Formula Editor to enter the Greek character for pi (π).

3) Select the category **Relations** and symbol **Is Similar Or Equal** using one of the following methods:
   - In the Elements dock, select **Relations** in the drop-down list and then select the **Is Similar Or Equal** icon $\square \simeq \square$.
   - Right-click in the Formula Editor and select **Relations > a simeq b** in the context menu.

4) Delete the first placeholder `<?>` before the word `simeq` in the Formula Editor.

5) Select the second placeholder `<?>` after the word `simeq` in the Formula Editor and replace it with the characters `3.14159`. The formula $\pi \simeq 3.14159$ now appears in your document.

Editing formulas

How you edit a formula and switch into formula editing mode depends on whether the formula is in Math or another LibreOffice component.

1) In Math, double-click on a formula element in the formula that appears in the Preview window to select the formula element in the Formula Editor, or directly select a formula element in the Formula Editor.

In Writer, Calc, Impress, or Draw, double-click on the formula, or right-click on the formula and select **Edit** in the context menu, to open the Formula Editor and enter editing mode. The cursor is positioned at the start of the formula in the Formula Editor.

> ✅ **Note**
>
> If you cannot select a formula element using the cursor, click on the **Formula Cursor** icon in the Tools toolbar to activate the formula cursor.

2) Select the formula element you want to change using one of the following methods:
   – Click on the formula element in the Preview window, positioning the cursor at the beginning of the formula element in the Formula Editor, then select the formula element in the Formula Editor.
   – Double-click on the formula element in the Preview window to select the formula element in the Formula Editor.
   – Position the cursor in the Formula Editor at the formula element you want to edit, then select that formula element.
   – Double-click directly on the formula element in the Formula Editor to select it.
3) Make your changes to the formula element you have selected.
4) Go to **View > Update** on the Menu bar, or press the *F9* key, or click on the **Update** icon on the Tools toolbar to update the formula in the Preview window or your document.
5) In Math, save your changes to the formula after editing.

In Writer, Calc, Impress, or Draw, click anywhere in the document away from the formula to leave editing mode, then save the document to save your changes to the formula.

Formula layout

This section provides some advice on how to lay out complex formulas in Math or in your LibreOffice document.

Using braces

LibreOffice Math knows nothing about order of operation within a formula. You need to use braces (curly brackets) to define the order of operations. The following examples show how braces can be used in a formula.

Example 1

`2 over x + 1` gives the result $\dfrac{2}{x}+1$

Math has recognized that the 2 before and the x after the `over` command as belonging to the fraction, and has represented them accordingly. If you want $x+1$ rather than x to be the denominator, you must bracket them together using braces so that both will be placed there.

Inserting braces into `2 over {x + 1}` gives the result $\dfrac{2}{x+1}$ where $x+1$ is now the denominator.

Example 2

`– 1 over 2` gives the result $\dfrac{-1}{2}$

Math has recognized the minus sign as a prefix for the 1 and has therefore placed it in the numerator of the fraction. If you wish to show that the whole fraction is negative, with the minus sign in front of the fraction, you must put the fraction in braces to signify to Math that the characters belong together.

Adding braces into the markup language `– {1 over 2}` gives the result $-\dfrac{1}{2}$ and the whole fraction is now negative.

Example 3

When braces are used in markup language, they define the layout of the formula and are not displayed or printed. If you want to use braces within a formula, use the commands `lbrace` and `rbrace` within the markup language.

`x over {-x + 1}` gives the result $\dfrac{x}{-x+1}$

Replace the braces using the commands `lbrace` and `rbrace` in the markup language.

Write `x over lbrace –x + 1 rbrace` and the result is $\dfrac{x}{\{-x+1\}}$

Brackets (parentheses) and matrices

If you want to use a matrix in a formula, you have to use the `matrix` command. Below is a simple example of a 2 x 2 matrix.

$$\texttt{matrix \{ a \# b \#\# c \# d \}} \quad \Longrightarrow \quad \begin{matrix} a & b \\ c & d \end{matrix}$$

In matrices, rows are separated by two hashes (##) and entries within each row are separated by one hash (#).

Normally, when you use brackets within a matrix, the brackets do not scale as the matrix increases in size. The example below shows an a formula where the parentheses do not scale to the size of the resulting matrix.

$$\texttt{(matrix \{ a \# b \#\# c \# d \})} \quad \Longrightarrow \quad (\begin{smallmatrix} a & b \\ c & d \end{smallmatrix})$$

To overcome this problem, Math provides scalable brackets that grow in size to match the size of the matrix. The commands `left(` and `right)` have to be used to create scalable brackets around a matrix. The following example shows how to create a matrix with scalable parentheses.

$$\texttt{left(matrix \{ a \# b \#\# c \# d \} right)} \quad \Longrightarrow \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Scalable brackets can also be used with any element of a formula, such as fraction, square root, and so on.

To create a matrix where some values are empty, you can use the grave accent (`) so that Math will put a small space in that position, as shown in the example below:

$$\texttt{left(matrix \{ 1 \# 2 \# 3 \#\#}$$
$$\texttt{4 \# ` \# 6 \} right)} \quad \Longrightarrow \quad \begin{pmatrix} 1 & 2 & 3 \\ 4 & & 6 \end{pmatrix}$$

> **ℹ️ Tip**
>
> Use the commands `left[` and `right]` to obtain square brackets. A list of all brackets available within Math can be found in *Appendix A, Commands Reference*.

> **ℹ️ Tip**
>
> If you want all brackets to be scalable, go to **Format > Spacing** on the Menu bar to open the Spacing dialog. Click on **Category**, select *Brackets* from the drop-down list. and then select the option **Scale all brackets**.

Unpaired brackets

When using brackets in a formula, Math expects that for every opening bracket there will be a closing one. If you forget to add a closing bracket, Math places an inverted question mark next to where the closing bracket should have been placed. For example, `lbrace a; b` will result in ¿ because the right bracket `rbrace` is missing.

This inverted question mark disappears when all the brackets are paired. The previous example could be fixed to `lbrace a; b rbrace`, resulting in $\{a;b\}$. However, there are cases where an unpaired bracket is necessary and for that you have the following options.

Non scalable brackets

A backslash \ is placed before a non scalable bracket to indicate that the subsequent character should not be regarded as a bracket, but rather as a literal character.

For example, the unpaired brackets in the formula `[ a; b [` would result in an inverted question mark because Math expects that `[` will be closed by `]`. To fix the error, use the backslash and insert `\ [ a; b \ [` into the Formula Editor to obtain $[a;b[$ as the result.

Scalable brackets

To create unpaired scalable brackets or braces in a formula, the markup commands `left`, `right`, and `none` can be used.

Example

The following formula uses the `stack` command to create a two-line formula and adds a bracket only to the right side of the stack.

```
abs x = left lbrace stack {x "for" x
>= 0 # -x "for" x < 0} right none
```

$$|x| = \begin{cases} x \text{ for } x \geq 0 \\ -x \text{ for } x < 0 \end{cases}$$

This effect can be achieved by using the `left lbrace` command combined with the `right none` command. The first command indicates that the left bracket is a `lbrace` whereas the second command tells Math that the right bracket will be `none`, meaning that no bracket will be added to the right side of the formula.

Recognizing functions

In the basic installation of Math, Math outputs functions in normal characters and variables in italic characters. However, if Math fails to recognize a function, you can tell Math that you have just entered a function. Entering the markup command `func` before a function forces Math to recognize the following text as a function.

For a full list of functions within Math, see Appendix A, Commands Reference.

Some Math functions have to be followed by a number or a variable. If these are missing, Math places an inverted question mark where the missing number or variable should be. To remove the inverted question mark and correct the formula, you have to enter a number, a variable, or a pair of empty brackets as a placeholder.

> **ⓘ Tip**
>
> You can navigate through errors in a formula using the key *F3* to move to the next error or the key combination *Shift+F3* to move to the previous error.

Formulas over multiple lines

Suppose you want to create a formula that requires more than one line, for example $\begin{matrix} x=3 \\ y=1 \end{matrix}$.

Your first reaction would be to press the *Enter* key. However, if you do that, the markup language in the Formula Editor goes to a new line, but the resulting formula does not have two lines. To add a new line to the formula you need to use the markup command `newline`.

Example

The examples below illustrate how the markup command `newline` can be used to add a new line to a formula. The first one simply adds an *Enter* to break the line in the markup language, which does not result in adding a new line. The second example uses the newline command, yielding a two-line formula.

Markup Language	*Resulting Formula*
x = 3 y = 1	$x=3\ y=1$
x = 3 newline y = 1	$\begin{matrix} x=3 \\ y=1 \end{matrix}$

It is not possible in Math to create multiple line formulas when a line ends with an equals sign and you want to continue the formula on a new line without completing the term on the right side of the equals sign. If you require a multiple line formula to have an equals sign at the end of a line without a term after the equals sign, then use empty quotes "" or empty braces {} or the space characters grave ` or tilde ˜.

By default, the alignment of a multiple line formula is center aligned. For more information on alignment using the equals sign, see "Aligning formulas using equals sign" on page 24.

Spacing within formulas

Spacing between the element in a formula is not set by using space characters in the markup language. To add spaces into a formula, use one of the following options:

- Grave ` to add a small space.
- Tilde ~ for a large space.
- Add space characters between quotes " ". These spaces will be considered as text.

Any spaces at the end of a line in the markup language are ignored by default. For more information, see "Formula spacing" on page 53.

Adding limits to summations and integrals

The `sum` and `int` commands, used for summations and integrals respectively, can take the parameters `from` and `to` if you want to set the lower and upper limits. The parameters `from` and

`to` can be used singly or together as shown by the following examples. For more information on the sum and integral commands, see "Operators" on page 62.

Examples

Here are some examples of how to add upper and lower bounds to summations and integrals. Note that you can use brackets to explicitly define which parts of the formula correspond to the desired bounds.

Markup Language	*Resulting Formula*
`sum from k = 1 to n a_k`	$\displaystyle\sum_{k=1}^{n} a_k$
`sum to infinity 2^{-n}`	$\displaystyle\sum^{\infty} 2^{-n}$
`sum from{ i=1 } to{ n } sum from{ j=1; i <> j } to{ m } x_ij`	$\displaystyle\sum_{i=1}^{n}\sum_{j=1;i\neq j}^{m} x_{ij}$
`int from 0 to x f(t) dt`	$\displaystyle\int_{0}^{x} f(t)\,dt$
`int_0^x f(t) dt`	$\int_{0}^{x} f(t)\,dt$
`int from Re f`	$\displaystyle\int_{\Re} f$

Writing derivatives

When writing derivatives, you have to tell Math that it is a fraction by using the `over` command. The `over` command is combined with the character `d` for a total derivative or the `partial` command for a partial derivative to achieve the effect of a derivative. Braces `{}` are used in each side of the elements to surround them and make the derivative as shown by the following examples.

Examples

Below are a few examples of how you can use Math to write derivatives:

Markup Language	*Resulting Formula*
`{df} over {dx}`	$\dfrac{df}{dx}$
`{partial f} over {partial y}`	$\dfrac{\partial f}{\partial y}$
`{partial^2 f} over {partial t^2}`	$\dfrac{\partial^2 f}{\partial t^2}$

✅ **Note**

To write function names with primes, as is normal in school notation, you must first add the symbols to the catalog. See "Catalog customization" on page 50 for more information.

Markup language characters as normal characters

Characters that are used as controls in markup language cannot be entered directly as normal characters. These characters are: %, {, }, &, |, _, ^ and ". For example, you cannot write `2% = 0.02` in markup language and expect the same characters to appear in your formula. To overcome this limitation in markup language, use one of the following methods:

- Use double quotes to mark that character as text, for example `2"%"= 0.02` will appear in your formula as $2\%=0.02$. However, this method cannot be used for the double-quote character itself, see "Text in formulas" below.
- Add the character to the Math Catalog, for example the double quote character.
- Use commands, for example `lbrace` and `rbrace` give you literal braces $\{\}$.

> ✅ **Note**
>
> The Special Characters dialog used by other LibreOffice components is not available in Math. If you regularly require special characters in Math, then it is recommended to add the characters to the Math Catalog. See "Catalog customization" on page 50 for more information.

Text in formulas

To include text in a formula, you have to enclose any text in double-quotes, for example `x " for " x >= 0` in markup language will create the formula $x \text{ for } x \geq 0$. All characters, except double quotes, can be used in text.

However, if you require double quotes in formula text, then you have to create your text with double quotes in LibreOffice Writer, then copy and paste the text into the Formula Editor as shown in Figure 7.
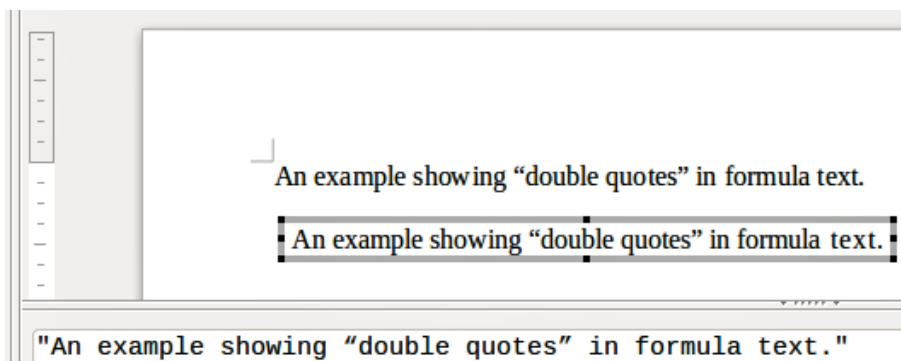
Figure 7: Example of double quotes in formula text

The font used for text in a formula will be the default font that has been set in the Fonts dialog. For more information on how to change fonts used in formulas, see "Changing formula appearance" on page 24.

By default, text alignment is left-justified in formulas. For more information on how to change text alignment, see "Adjusting formula alignment" on page 30.

Formatting text in formulas

Formatting commands are not interpreted within text used in formulas. If you want to use formatting commands within formula text, then you must break up the text using double quotes in the Formula Editor.

Example

Enter the following in the Formula Editor:

```
"In " color blue bold "isosceles" "triangles, the base angles are
equal"
```

This markup creates the following text in a formula:
 In **isosceles** triangles, the base angles are equal

This example shows how to use the markup commands `color` and `bold` to format the word "isosceles". Note that the commands are applied to the immediately subsequent text within double quotes. Hence, the formatting commands do not apply to the remainder of the text in the example because it is in a separate quote block.

The command `color` needs to be followed by the name of the desired color or to its RGB or hexadecimal value. Refer to "Attributes" on page 64 for a list of predefined color names.

Aligning formulas using equals sign

LibreOffice Math does not have a command for aligning formulas on a particular character. However, you can use a matrix to align formulas on a character and this character is normally the equals sign (=). In addition, you can use the markup commands `alignr`, `alignl` and `alignc` to set the alignment of each value inside the matrix to the right, left or center, respectively.

Example

The example below uses a matrix to align formulas on the equals sign. Notice how the alignment commands align the contents of each position in the matrix.

```
matrix{ alignr x+y # {}={} # alignl 2
## alignr x # {}={} # alignl 2-y }
```
$$\begin{aligned} x+y &= 2 \\ x &= 2-y \end{aligned}$$

> ✅ **Note**
>
> The empty braces each side of the equals sign are necessary because the equals sign is a binary operator and requires an expression on each side. You can use spaces, or ` or ~ characters each side of the equals sign, but braces are recommended as they are easier to see within the markup language.

You can reduce the spacing on each side of the equals sign if you change the inter-column spacing of the matrix. See "Adjusting formula spacing" on page 28 for more information.

Changing formula appearance

This section describes how to change the font or font size in a selected formula and how to change the default font or font size.

> ✅ **Note**
>
> If you have already inserted formulas into your document and you change the **default** font or font size, only formulas inserted after the change in default font or font size will use the new default settings. You have to individually change the font or font size of formulas already inserted if you want these formulas to use the same font or font size as the default settings.

Formula font size

Current formula font size

To change the font size used for a formula already inserted in Math or another LibreOffice component:

1) Click in the markup language in the Formula Editor.

2) Go to **Format > Font size** on the Menu bar to open the Font Sizes dialog (Figure 8).

3) Select a different font size using the *Base size* spinner or type a new font size in the *Base Size* box.

4) Click **OK** to save your changes and close the dialog. An example result when you change font size is shown below.

Example

Default font size 12pt: $\pi \simeq 3.14159$

After font size change to 18pt: $\pi \simeq 3.14159$

Figure 8: Font Sizes dialog

Default formula font size

To change the default font size used for all formulas in Math or another LibreOffice component:

1) Before inserting any formulas in your document, go to **Format > Font size** on the Menu bar to open the Font Sizes dialog (Figure 8).

2) Select a different font size using the Base size spinner or type a new font size in the Base Size box.

3) Click **Default** and confirm your changes to the base size font. Any formulas created from this point on will use the new base size font for formulas.

4) Click **OK** to save your changes and close the Font Sizes dialog.

Font size options

The Font Sizes dialog (Figure 8) specifies the font sizes for a formula. Select a base size and all elements of the formula will be scaled in relation to this base.

- **Base size** – all elements of a formula are proportionally scaled to the base size. To change the base size, select or type in the desired point (pt) size. You can also use other units of measure or other metrics, which are then automatically converted to points.
- **Relative Sizes** – in this section, you can determine the relative sizes for each type of element with reference to the base size.
  - *Text* – select the size for text in a formula relative to the base size.
  - *Indexes* – select the relative size for the indexes in a formula in proportion to the base size.
  - *Functions* – select the relative size for names and other function elements in a formula in proportion to the base size.
  - *Operators* – select the relative size of the mathematical operators in a formula in proportion to the base size.
  - *Limits* – select the relative size for the limits in a formula in proportion to the base size. This is used to determine the size for commands as `from` and `to` used in summations and integrals.
- **Default** – click this button to save any changes as a default for all new formulas. A confirmation message appears before saving any changes.

Formula fonts

Current formula fonts

To change the fonts used for the current formula in Math or another LibreOffice component:

1) Click in the markup language in the Formula Editor.
2) Go to **Format > Fonts** on the Menu bar to open the Fonts dialog (Figure 9).
3) Select a new font for each of the various options from the drop-down lists.
4) If the font you want to use does not appear in the drop-down list, click **Modify** and select the option from the context menu to open a fonts dialog. Select the font you want to use and click **OK** to add it to the drop-down list for that option.
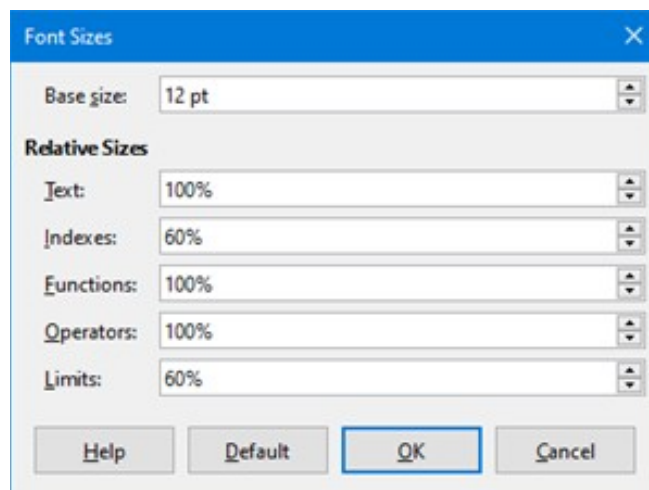5) Click **OK** to save your changes and close the Fonts dialog.

Default formula fonts

To change the default fonts used for all formulas in Math or another LibreOffice component:

1) Before inserting any formulas in your document, go to **Format > Fonts** on the Menu bar to open the Fonts dialog (Figure 9).
2) Select a new font for each the various options from the drop-down lists.
3) If the font you want to use does not appear in the drop-down list, click **Modify** and select the option from the context menu to open a fonts dialog. Select the font you want to use and click **OK** to add it to the drop-down list for that option.
4) Click **Default** and confirm your changes to the fonts. Any formulas created from this point on will use the new font for formulas.
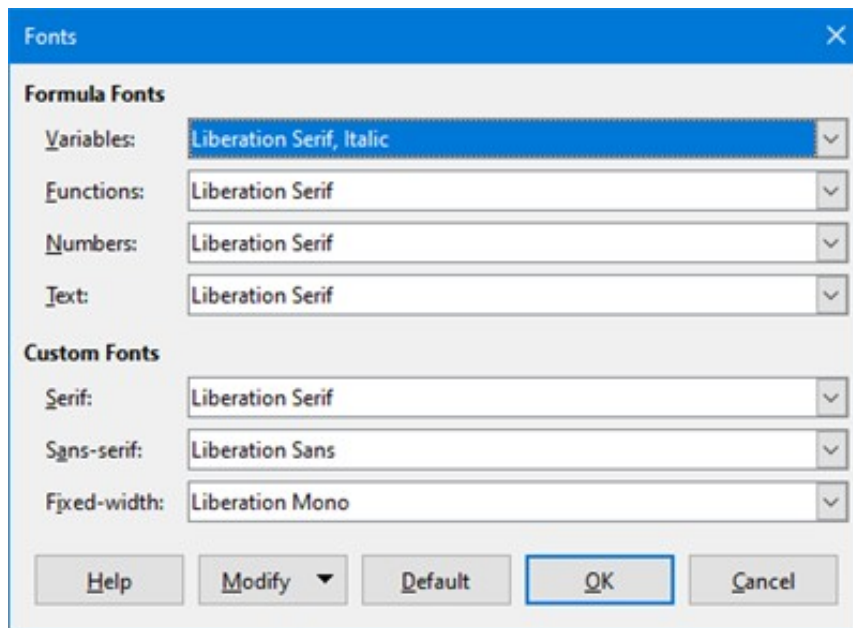5) Click **OK** to save your changes and close the Fonts dialog.

Figure 9: Fonts dialog

Formula font options

Defines the fonts that can be applied to formula elements.

- **Formula Fonts** – defines the fonts used for the variables, functions, numbers and inserted text that form the elements of a formula.

  – *Variables* – selects the fonts for the variables in your formula. For example, in the formula x=SIN(y), x and y are variables and will reflect the assigned font.

  – *Functions* – selects the fonts for names and properties of functions. For example, the functions in the formula x=SIN(y) are =SIN( ).

  – *Numbers* – selects the fonts for the numbers in a formula.

  – *Text* – defines the fonts for the text in a formula.

- **Custom Fonts** – this section of the Fonts dialog (Figure 9) defines fonts to format text components in a formula. The three basic fonts Serif, Sans-serif and Fixed-width are available. Other fonts can be added to each standard installed basic font using the **Modify** button. Every font installed on a computer system is available for use.

  – *Serif* – specifies the font to be used for the font serif format. Serifs are the small "guides" that can be seen, for example, at the bottom of a capital A when the Times serif font is used. Using serifs is quite helpful since it guides the eye of a reader in a straight line and can speed up reading.

  – *Sans* – specifies the font to be used for sans font formatting.

  – *Fixed* – specifies the font to be used for fixed font formatting.

- **Modify** – click one of the options from the context menu to access the Fonts dialog, where the font and attributes can be defined for the respective formula and for custom fonts.

- **Default** – click this button to save any changes as a default for all new formulas. A confirmation message appears before saving any changes.

> ✅ **Note**
>
> When a new font is selected for a formula, the old font remains in the list alongside the new one and can be selected again.

> ✅ **Note**
>
> Variables should be written in *italics*, so make sure that the *Italic* option is selected. For the font you want to use. For all other elements, use the basic form of a font. The style can be easily altered in the formula itself by using the commands `italic` or `bold` to set these characteristics and `nitalic` or `nbold` to unset them.

Adjusting formula spacing

Use the Spacing dialog (Figure 10) to determine the spacing between formula elements. The spacing is specified as a percentage in relation to the defined base size for font sizes.
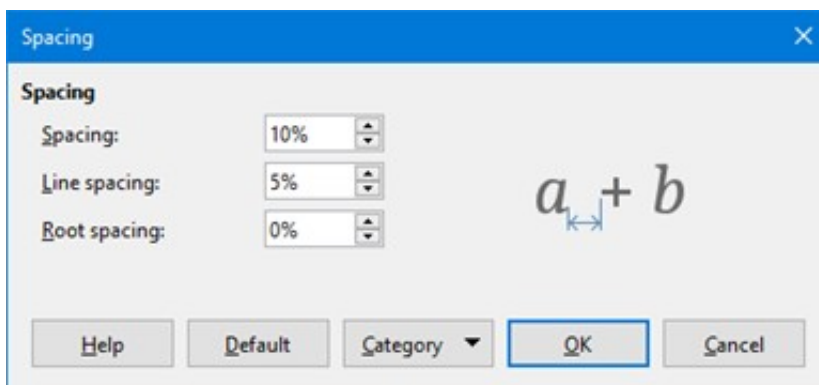

Figure 10: Spacing dialog

Current formula spacing

To change the spacing used for the current formula in Math or another LibreOffice component:

1) Click in the markup language in the Formula Editor.
2) Go to **Format > Spacing** on the Menu bar to open the Spacing dialog (Figure 10).
3) Click **Category** and select one of the options from the drop-down list. The options in the Spacing dialog change according to the category selected.
4) Enter new values for the spacing category and click **OK**.
5) Check the result in your formula. If not to your satisfaction, repeat the above steps.

Default formula spacing

To change the default spacing used for all formulas in Math or another LibreOffice component:

1) Before inserting any formulas in your document, go to **Format > Spacing** on the Menu bar to open the Spacing dialog (Figure 10).
2) Click **Category** and select one of the options from the drop-down list. The options in the Spacing dialog change according to the category selected.
3) Click **Default** and confirm your changes to the formula spacing. Any formulas created from this point on will use the new spacing for formulas.
4) Click **OK** to save your changes and close the Spacing dialog.

> ✅ **Note**
>
> If you have already inserted formulas into your document and you change the spacing, only formulas inserted after the change in spacing will use the new default settings. You have to individually change the spacing of formulas already inserted if you want these formulas to use the same spacing as the default settings.

Spacing options

Use Category in the Spacing dialog (Figure 10) to determine the formula element for which you would like to specify the spacing. The appearance of the dialog depends on the selected category. A preview window shows you which spacing is modified through the respective boxes.

- **Category** – click this button to select the category for which you want to change the spacing.
- **Spacing** – defines the spacing between variables and operators, between lines, and between root signs and radicals.
    - *Spacing* – determines the spacing between variables and operators
    - *Line Spacing* – determines the spacing between lines.
    - *Root Spacing* – determines the spacing between the root sign and radicals.
- **Indexes** – defines the spacing for superscript and subscript indexes.
    - *Superscript* – determines the spacing for superscript indexes.
    - *Subscript* – determines the spacing for subscript indexes.
- **Fractions** – defines the spacing between the fraction bar and the numerator or denominator.
    - *Numerator* – determines the spacing between the fraction bar and the numerator.
    - *Denominator* – determines the spacing between the fraction bar and the denominator.
- **Fraction Bars** – defines the excess length and line weight of the fraction bar.
    - *Excess length* – determines the excess length of the fraction line.
    - *Weight* – determines the weight of the fraction line.
- **Limits** – defines the spacing between the sum symbol and the limit conditions.
    - *Upper limit* – determines the spacing between the sum symbol and the upper limit.
    - *Lower limit* – determines the spacing between the sum symbol and the lower limit.
- **Brackets** – defines the spacing between brackets and the content.
    - *Excess size (left/right)* – determines the vertical distance between the upper edge of the contents and the upper end of the brackets.
    - *Spacing* – determines the horizontal distance between the contents and the upper end of the brackets.
    - *Scale all brackets* – scales all types of brackets. If you then enter `( a over b )` in the Formula Editor, the brackets will surround the whole height of the argument. You normally achieve this effect by entering left `( a over b right )`.
    - *Excess size* – adjusts the percentage excess size. At 0% the brackets are set so that they surround the argument at the same height. The higher the entered value is, the larger the vertical gap between the contents of the brackets and the external border of the brackets. The field can only be used in combination with Scale all brackets.
- **Matrices** – defines the relative spacing for the elements in a matrix.
    - *Line spacing* – determines the spacing between matrix elements in a row.
    - *Column spacing* – determines the spacing between matrix elements in a column.
- **Symbols** – defines the spacing of symbols in relation to variables
    - *Primary height* – defines the height of the symbols in relation to the baseline.
    - *Minimum spacing* – determines the minimum distance between a symbol and variable.

- **Operators** – defines the spacing between operators and variables or numbers.
  - *Excess size* – determines the height from the variable to the operator upper edge.
  - *Spacing* – determines the horizontal distance between operators and variables.
- **Borders** – adds a border to a formula. This option is particularly useful if you want to integrate the formula into a text file in LibreOffice Writer by combining mathematical formulas and text inside the same paragraph.
  - *Left* – the left border is positioned between the formula and background.
  - *Right* – the right border is positioned between the formula and background.
  - *Top* – the top border is positioned between the formula and background.
  - *Bottom* – the bottom border is positioned between the formula and background.
- **Preview Field** – displays a preview of the current selection.
- **Default** – saves any changes as default settings for all new formulas. A security response will appear before saving these changes.

✅ Note

If you are adding Math formulas into a LibreOffice Writer document, you may notice spacing between the formula and the surrounding text even if you set all formula borders to zero. This is because Math formulas are OLE objects embedded into a frame that, by default, adds 0.2 cm to the left and right of the Math object.

You can remove that extra spacing in two ways. The first way is to edit the object properties by right-clicking the Math object and then going to **Properties > Wrap** and adjust the **Spacing** fields to the desired values. The second way is to modify the spacing options of the default Formula frame style or create a new style. To learn more on how to change the spacing of Math objects, refer to "Object spacing" on page 38. For more information on how to edit styles in general, refer to the Styles chapters in the *Writer Guide*.

Example

In the following example, Math objects are combined with text by removing all spacing both from the formula and the frame style surrounding it. Because spacing is set to zero, the spacing between text and formulas is achieved by simply adding a space character using the keyboard space bar.

"Let c_{ij} be a cost matrix and $x_{ij} \in \{0\,;1\}$ be a decision variable defined for $i = 1, 2, ..., m$ origins and for $j = 1, 2, ..., n$ destinations".

Adjusting formula alignment

The alignment settings determine how formula elements located above one another are aligned horizontally relative to each other.

✅ Note

It is not possible to align formulas on a particular character and formula alignment does not apply to text elements. Text elements are always aligned to the left.

Current formula alignment

To change the alignment used for the current formula in Math or another LibreOffice component:

1) Click in the markup language in the Formula Editor.

2) Go to **Format > Alignment** on the Menu bar to open the Alignment dialog (Figure 11).

3) Select either *Left*, *Centered*, or *Right* for horizontal alignment.

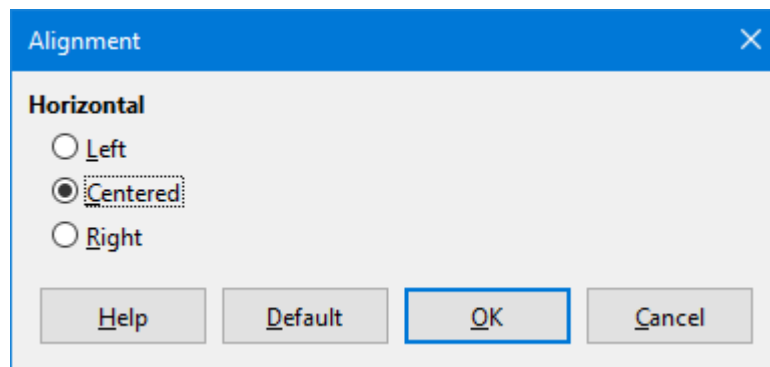4) Click **OK** and check the result in your formula. If not to your satisfaction, repeat the above steps.

Figure 11: Alignment dialog

> ✅ **Note**
>
> Regardless of the alignment option selected in the Alignment dialog, it is possible to align sections of a formula using the commands `alignl`, `alignc` and `alignr`. For example, they can be useful to align formulas in matrices. These commands also work for text elements.

Default formula alignment

To change the default alignment used for all formulas in Math or another LibreOffice component:

1) Before inserting any formulas in your document, go to **Format > Alignment** on the Menu bar to open the Alignment dialog (Figure 11).

2) Select either *Left*, *Centered*, or *Right* for horizontal alignment.

3) Click **Default** and confirm your changes to the formula alignment. Any formulas created from this point on will use the new alignment for formulas.

4) Click **OK** and check the result in your formula. If not to your satisfaction, repeat the above steps.

> ✅ **Note**
>
> If you have already inserted formulas into a document and you change the formula alignment, only formulas inserted after the change in alignment will use the new default settings. You have to individually change the alignment of formulas already inserted if you want them to use the same alignment as the default settings.

Changing formula color

You can change the color of characters used in a formula using markup language: the command `color` followed by a color name, RGB value, or HTML hex value.

The `color` command only works on the formula element immediately after the color name, RGB value, or hex value. For example, entering the markup language `color red 5 times 4` gives the result 5×4. Only the number 5 is colored red.

To change the color of the whole formula, you have to enclose the whole formula within brackets. For example, entering the markup language `color red {5 times 4}` gives the result 5×4.

Named colors

For information on the named colors available in Math, see "Attributes" on page 64. These colors are listed in the *Attributes* section of the Elements dock (Figure 4 on page 14).

RGB values

To use custom colors defined by RGB (Red, Green and Blue) values ranging from 0 to 255, use the `color rgb R G B` markup command, where R, G, and B correspond to the Red, Green, and Blue values of the desired color.

> **Example**
>
> In this example, the term "decision variable" uses the color defined by the RGB values 160, 82, 45.
>
> ```
> "Let " x_ij " be a " color rgb 160 82 45 "decision variable " "in
> the problem under consideration"
> ```
>
> Math will create the following output:
>
> Let x_{ij} be a decision variable in the problem under consideration

Hex values

Math now supports HTML colors, defined by a hex number. Use the `color hex 000000` markup command, where 00000 is the corresponding hex number. For example, the command `color hex FF0000 decision` creates the output *decision*.

Background color

It is not possible to select a background color for formulas in LibreOffice Math. The background color for a formula is by default the same color as the document or frame that the formula has been inserted into. In LibreOffice Writer, you can use object properties to change the background color for a formula. For more information, see "Background and borders" on page 39.

Formula library

If you regularly insert the same formulas into your documents, you can create a formula library using formulas that you have created using the Formula Editor. Individual formulas can be saved as separate files using the ODF format for formulas (file type .odf), or in MathML format (file type .mml).

You can use either LibreOffice Math, Writer, Calc, Draw, or Impress to create formulas and build up your formula library.

Using Math

1) Create a folder on your computer to contain your formulas and give the folder a memorable name, for example Formula Library.

2) In LibreOffice, go to **File > New > Formula** on the Menu bar, or click on **Math Formula** in the Start Center to open LibreOffice Math and create your formula using the Formula Editor. See "Formulas as separate documents or files" on page 11 for more information.

3) Go to **File > Save As** on the main Menu bar or use the keyboard shortcut *Ctrl+Shift+S* to open a Save As dialog.

4) Navigate to the folder you have created for your formula library.

5) Type a memorable name for your formula in the **File name** box.

6) Select in the **File type** drop-down list either *ODF Formula (.odf)* or *MathML 2.0 (.mml)* as the file type for the formula.

7) Click **Save** to save the formula and close the Save As dialog.

> ✅ **Note**
>
> MathML 2.0 stands for Mathematical Markup Language (MathML) Version 2.0.
> MathML is an XML format that describes mathematical notation and enables the use
> of mathematics in the World Wide Web. If you want to learn more about the MathML
> format, visit its official website at https://www.w3.org/TR/MathML2/overview.html

Using Writer, Calc, Draw, or Impress

1) Create a folder on your computer to contain your formulas and give the folder a memorable name, for example Formula Library.

2) Open a document using Writer, Calc, Draw, or Impress.

3) Go to **Insert > Object > Formula Object** on the Menu bar to open the Formula Editor and create a formula. See "Formulas in LibreOffice documents" on page 12 for more information.

4) Right-click on the formula object and select **Save Copy as** in the context menu to open a Save As dialog.

5) Navigate to the folder you have created for your formula library.

6) Type a memorable name for your formula in the **File name** box.

7) Select in the **File type** drop-down list either *ODF Formula (.odf)* or *MathML 2.0 (.mml)* as the file type for the formula.

8) Click **Save** to save the formula and close the Save As dialog.

Using your formula library

You cannot insert a formula from your library into a document by dragging and dropping using the mouse, nor by using **Insert > File** on the Menu bar. You must insert a formula from your library into a document as an OLE object.

1) Open your document in Writer, Calc, Draw, or Impress.

2) Go to **Insert > Object > OLE Object** on the Menu bar to open the Insert OLE Object dialog.

3) Select the option **Create from file**.

4) Click **Search** to open your file browser dialog.

5) Navigate to the folder you have created for your formula library.

6) Select the formula you want to insert and click **Open**, or double-click on the formula you want to insert.

7) Click **OK** to insert your formula as an OLE object in the document and close the OLE Object dialog.

Math Guide 7.2

Chapter 2,
Formulas in Writer

Introduction

When a formula is inserted into a document, LibreOffice Writer inserts the formula into a frame and treats the formula as an OLE object. Double-clicking on an inserted formula will open the Formula Editor in LibreOffice Math, allowing you to edit the formula. For more information on creating and editing formulas, please refer to Chapter 1, Creating and Editing Formulas.

This chapter explains what options you can change for each individual formula within a Writer document. Please refer to the chapters on styles in the *Writer Guide* for information on how to change the default settings for frame styles for OLE objects.

Automatic formula numbering

Automatic numbering of formulas for cross-reference purposes can only be carried out in LibreOffice Writer. The easiest way to add numbered formulas in sequence is to use the AutoText entry **fn** (for '**formula numbered**').

Numbering

1) Start a new line in your document.

2) Type *fn* and then press the *F3* key. A two column table with no borders is inserted into your document with the left column containing a sample formula and the right column containing a reference number, as shown below.

$$E = mc^2 \tag{1}$$

3) Delete the sample formula and insert your formula as an object in the left column.

4) Alternatively, you can first insert your formula into the document, then carry out Steps 1 and 2 above replacing the sample formula with your formula.

Cross-referencing

1) Click in your document where you want the cross-reference to appear.

2) Go to **Insert > Cross-reference** on the Menu bar to open the Fields dialog (Figure 12).

3) Click on the **Cross-references** tab, then select *Text* in the **Type** section.

4) In the **Selection** section, select the formula number you want to refer to.

5) In the **Insert reference to** section, select *Reference* and click **Insert**.

6) When you have finished creating cross-references, click **Close** to close the Fields dialog.

> ### *i* Tip
>
> To insert the cross-reference number without parentheses, select *Numbering* instead of *Reference* in the **Insert reference to** section.

> ### ✔ Note
>
> If you want to use square parentheses instead of round ones, or if you want the cross-reference number to be separated from the formula by tabs instead of using a table, then refer to the *Writer Guide*.
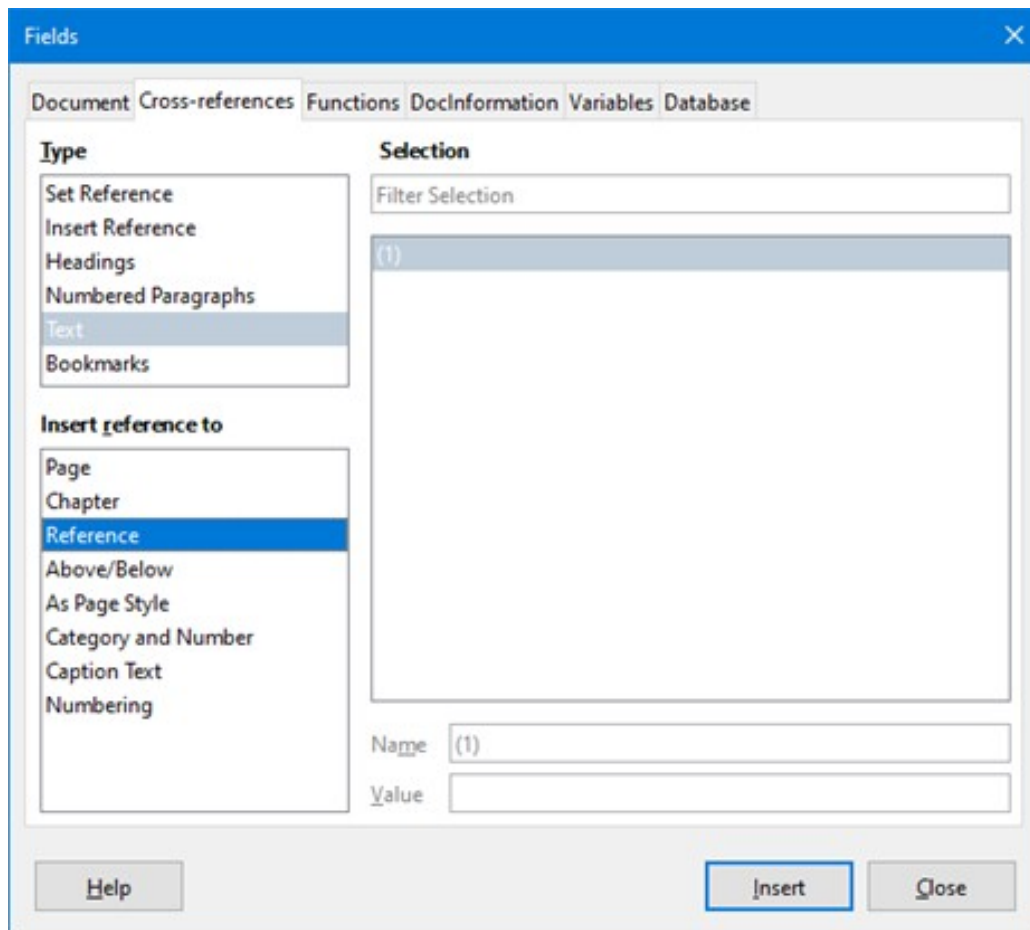
Figure 12: Fields Dialog – Cross-references tab

Anchoring formulas

A formula is treated as an object within Writer and its default anchoring is **To character** within a paragraph when it is inserted into a document. To change the anchoring of a formula object:

1) Right-click on the selected formula object and select **Anchor** from the context menu.

2) Select a new anchoring option from the context sub-menu. The anchoring positions available are **To page**, **To paragraph**, **To character**, or **As character**.

3) Alternatively, right-click on the selected formula object and select **Properties** from the context menu, or go to **Format > Frame and Object > Properties** on the Menu bar to open the Object dialog (Figure 13).

4) Make sure the **Type** tab is selected and select a new anchoring position from the *Anchor* section.

5) Click **OK** to save your changes and close the Object dialog.

✔ **Note**

> The anchoring options are not available in the Object dialog when you are making changes to the options available for frame styles. For more information on how to modify frame styles, please refer to the chapters on styles in the *Writer Guide*.
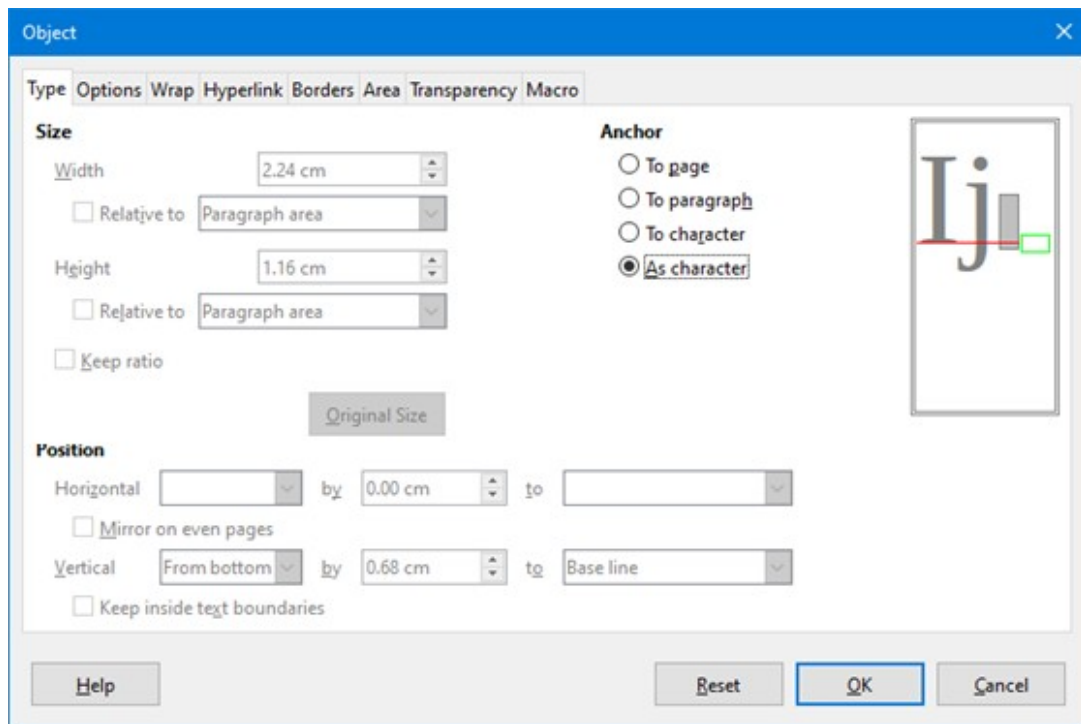
Figure 13: Object dialog – Type tab with Anchor options

Vertical alignment

The normal default setting for vertical alignment for formula objects is to use the text base line as a reference. This default setting can be changed by modifying the Formula frame style; see the chapters on styles in the *Writer Guide* for more information.

To change the vertical alignment position of an individual formula object (assuming that the **As character** anchoring option is selected):

1) Right-click on the selected formula object and select **Properties** in the context menu, or go to **Format > Frame and Object > Properties** to open the Object dialog (Figure 13).

2) Make sure the **Type** tab is selected and select a new alignment position in the drop-down list in the *Position* section. The vertical alignment options available are **Top**, **Bottom**, **Center**, or **From bottom**.

3) If necessary, type in the text box a plus or minus value for vertical alignment. This option is only available if **From bottom** vertical alignment has been selected.

4) Select the type of text alignment from the drop-down list in the *Position* section. The text alignment options available are **Base line**, **Character**, and **Row**.

5) Click **OK** to save your changes and close the Object dialog.

> ✅ **Note**
>
> If the *Position* section in the Object dialog is grayed out and not available, then go to **Tools > Options > LibreOffice Writer > Formatting Aids** and uncheck the option *Math baseline alignment*. This setting is stored with the document and applies to all formulas within it. Any new documents created will also use this setting for *Math baseline alignment*.

Object spacing

A formula object, when inserted into a Writer document, has spacing on each side of the object. The default value used for spacing is set within the frame style for formula objects and can be changed by modifying the Formula frame style. See the chapters on styles in the *Writer Guide* for more information.

To individually adjust the spacing for each formula object within your document:

1) Create your formula in your Writer document.

2) Right-click on your selected formula object and select **Properties** in the context menu, or go to **Format > Frame and Object > Properties** to open the Object dialog.

3) Click on the **Wrap** tab to open the Wrap page in the Object dialog (Figure 14).

4) In the **Spacing** section, enter the spacing value for *Left*, *Right*, *Top*, and *Bottom* spacing.

5) Click **OK** to save your changes and close the Object dialog.
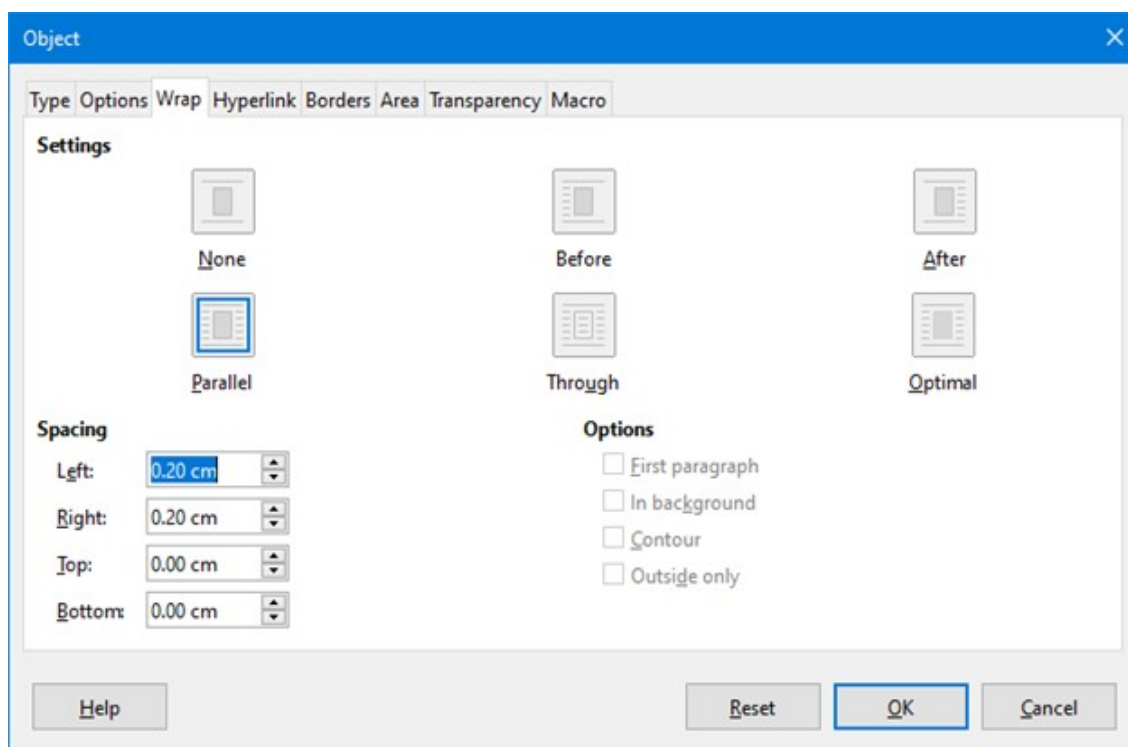

Figure 14: Object dialog – Wrap tab

> ✅ **Note**
>
> Object wrapping options will be grayed out if the object is anchored *As Character*. To enable all wrapping options, you need to anchor the object *To Page*, *To Paragraph*, or *To Character*. To change the anchor settings, right-click the formula object and use the Anchor context menu to select the desired option.

Text mode

In large formulas placed within a line of text, the formula elements can often be higher than the text height. Therefore, to make large formulas easier to read, it is recommended to always insert large formulas into a separate paragraph of their own so that it is separated from text.

However, if it is necessary to place a large formula within a line of text, double-click on the formula to open the Formula Editor and then go to **Format > Text Mode** on the Menu bar. The Formula Editor will try to shrink the formula to fit the text height. The numerators and denominators of fractions are shrunk, and the limits of integrals and sums are placed beside the integral/sum sign, as shown in the following example.

> **Example**
> A formula in a separate paragraph:
>
> $$\sum_{i=2}^{5} i^2$$
>
> and the same formula embedded into a line of text using text mode format: $\sum_{i=2}^{5} i^2$
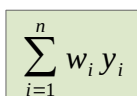
Background and borders

The default setting for background (area fill) and borders for formula objects is set by the Formula frame style. To change the default setting for the Formula frame style, refer to the chapters on styles in the *Writer Guide*. However, for individual formulas in your document, you can change the background and borders.

> ✅ **Note**
>
> The size of the frame that a formula is placed in when inserted into a document cannot be changed. The frame size for a formula object depends on the setting of the formula font size. See "Formula font size" on page *25* for more information.

Below is an example of a formula using background color and borders.

$$\sum_{i=1}^{n} w_i\, y_i$$

Note that the background color and border properties are part of the Math OLE object and hence cannot be defined using markup language.

Backgrounds

1) In your document, select the formula where you want to change the background.
2) Right-click on the formula and select **Properties** from the context menu, or go to **Format > Frame and Object > Properties** on the Menu bar to open the object dialog.
3) Click on the **Area** tab and use the buttons at the top to select the type of fill you want to use for your formula (Figure 15).
4) Select the options you want to use for your formula background. The options change depending on the type of fill selected.
5) Click **OK** to save your changes and close the Object dialog.
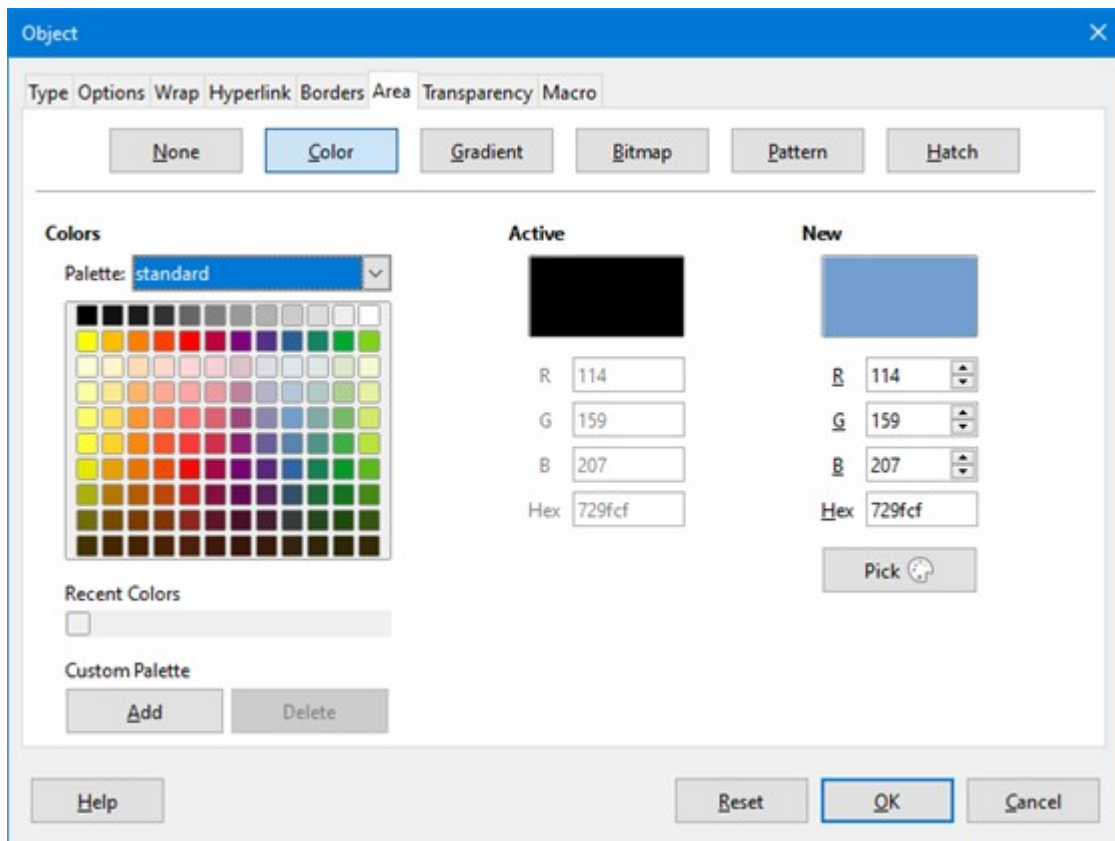
Figure 15: Object dialog – Area tab

Borders

1) In your document, select the formula where you want to change the borders.

2) Right-click on the formula and select **Properties** from the context menu, or go to **Format > Frame and Object > Properties** on the Menu bar to open the object dialog.

3) Click on the **Borders** tab and select the options you want to use for your formula borders (Figure 16). Click **OK** to save your changes and close the Object dialog.
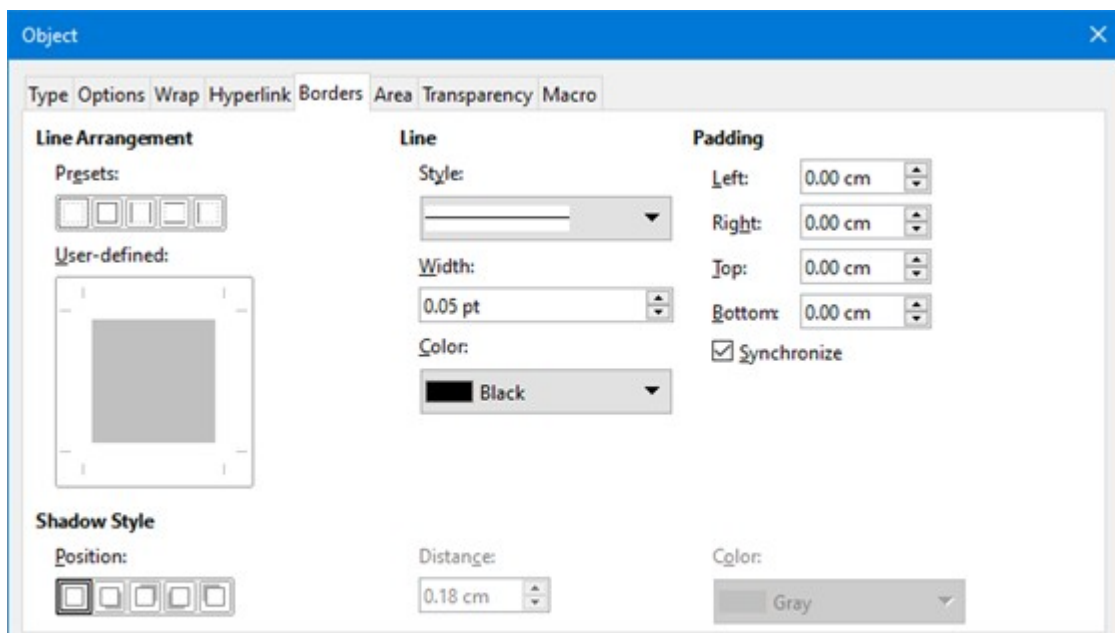
Figure 16: Object dialog – Borders tab

Quick insertion of formulas

If you know the markup language for your formula, you can quickly insert it into your Writer document without opening the Formula Editor:

1) Enter the formula markup language into your document at the position where you want the formula to be placed.

2) Select the markup language.

3) Go to **Insert > Object** on the Menu bar and select **Formula Object** to create a formula from the selected markup language.

4) Alternatively you can use the key combination *Ctrl + Insert* to open the Insert OLE Object dialog and then select **Formula**.

Math Guide 7.2

Chapter 3,
Formulas in Calc, Draw, and
Impress

Introduction

In Calc, Draw, and Impress, formulas are inserted as OLE objects without any background (area fill) or borders. See Chapter 1, Creating and Editing Formulas for more information on inserting formulas into a LibreOffice component.

Each formula object is inserted into a spreadsheet, drawing, or slide as follows:

- In Calc, formulas are inserted into a selected cell in a spreadsheet with no style assigned to the formula object.

- In Draw and Impress, formulas are inserted into a central position on your drawing or slide and, by default, are assigned the drawing object style *Object with no fill and no line*. For more information on how to modify or assign drawing object styles, see the *Draw Guide* or the *Impress Guide*.

Anchoring formulas

Calc

A formula object can be anchored into a spreadsheet as **To Page** (default setting), or as **To Cell**. To change the anchoring type of formulas in a Calc spreadsheet:

1) Select the formula object in the spreadsheet.

2) Right-click on the formula and select **Anchor > To Page** or **To Cell** in the context menu.

3) Alternatively, go to **Format > Anchor** on the Menu bar and select **To Page** or **To Cell**.

> ### ⓘ Tip
>
> If you insert a formula into a Calc spreadsheet and it appears out of scale, you can fix it by right-clicking the formula object and then selecting the **Original Size** option in the context menu.

Draw and Impress

When a formula is inserted into a drawing or slide, it is inserted as a floating OLE object and is not anchored to any particular position in the drawing or slide.

Formula object properties

Formula objects in Calc, Draw, and Impress can be modified just like any other object that has been placed in a spreadsheet, drawing, or presentation, with the exception of formula object size and changing the format of any text within a formula. For more information on how to change object properties, see the *Calc Guide*, *Draw Guide* and *Impress Guide*. For more information on formula object size and formatting formula text, see Chapter 1, Creating and Editing Formulas.

The following points will help you select which dialog to use if you want to change the properties of formula objects.

- For formula backgrounds, use the various options in the tabs of the Area dialog.

- For formula borders, use the various options in the Line dialog. Note that formula borders are separate from cell borders in a Calc spreadsheet.

- To accurately re-position a formula object, use the various options in tabs of the Position and Size dialog.

- In Draw and Impress, you can arrange, align, group, flip, convert, break, combine, and edit points of formula objects.
- You cannot change the text attributes of a formula object. The text used in a formula is set when you create the formula in the Formula Editor.
- Formula object size is set by the formula font size when the formula is created in the Formula Editor. The formula object size is protected in the Position and Size dialog, but this can be deselected if you wish. However, this is not recommended as resizing a formula object using the Position and Size dialog could lead to distortion of the formula making it difficult to read.

Formulas in charts

A chart in a Calc spreadsheet is itself an OLE object, therefore, you cannot use the Formula Editor to create and insert a formula directly into a chart. However, you can create both the Chart and Math objects separately and later copy and paste the Math formula into the Chart object:

1) Create the chart using LibreOffice Calc. For a complete reference on how to create charts, see Chapter 3 in the *Calc Guide*.
2) Click at any cell in your spreadsheet so that the Chart is no longer selected.
3) Insert a Math Formula object by clicking **Insert > Object > Formula**.
4) Type the desired formula into the Formula Editor.
5) After editing the formula, select the Math Formula object and press *Ctrl+C* to copy the Formula object to the clipboard.
6) Double-click the chart object to start editing the chart and press *Ctrl+V* to paste the Formula object into the chart.
7) Now you can position the object anywhere you want inside the chart.

Figure 17 shows an example of a Calc chart with a Math Formula object inside it, which in this example is the $\sin(x)$ function.
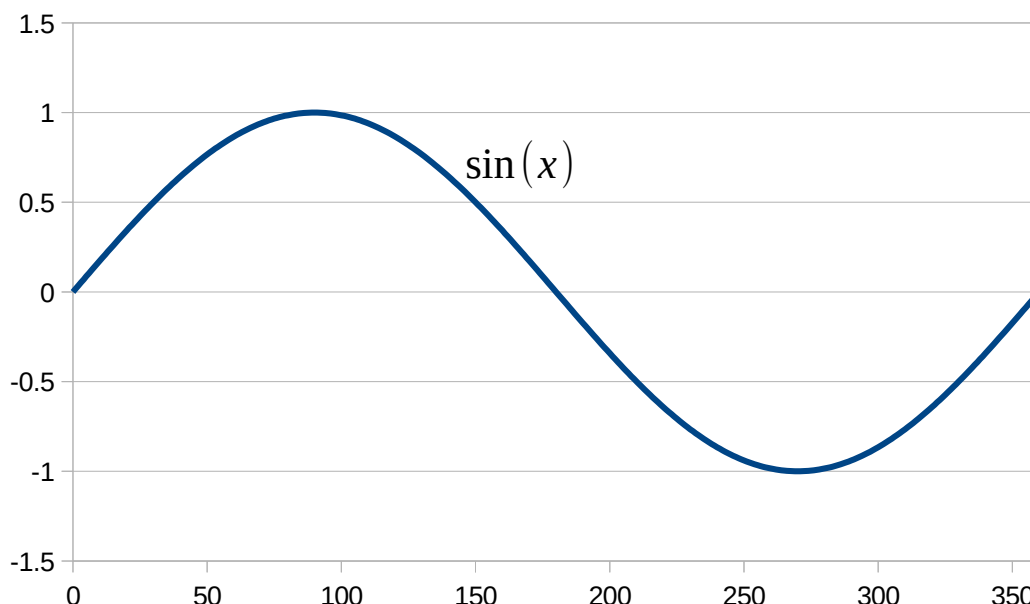
Figure 17: Calc chart with a Math object inside it

To change the formula, you must repeat the whole process of creating, copying, and pasting the Formula object into the chart.

Chemical formulas

The primary purpose of Math is to create mathematical formulas, but it can also be used to write chemical formulas. However, in chemical formulas, the chemical symbols are normally written in uppercase using non-italic characters. The following table shows some examples of chemical formulas.

Construction	Example	Markup Language
Molecules	H_2SO_4	`H_2 SO_4`
Isotopes	$^{238}_{92}U$	`U lsub 92 lsup 238`
Ions	SO_4^{2-} or SO_4^{2-}	`SO_4^{2-{}}` *or* `SO_4^{2"-"}`

To create chemical formulas using Math, change the font used for variables to a non-italic font. For more information on how to change fonts in a formula, see "Formula fonts" on page 26.

For reversible reactions in chemical formulas, there is no symbol for a double arrow in Math. If you have a font available with correct symbols for use in chemical formulas, then you can add these symbols to the Catalog. See "Catalog customization" on page 50 for more information on how to add symbols to the Catalog.

Math Guide 7.2

Chapter 4,
Customization

Introduction

This chapter explains how you can customize LibreOffice Math to suit the way you create formulas for use in LibreOffice documents. Also, refer to Chapter 14, Customizing LibreOffice, in the *Getting Started Guide* for more general information on how to customize LibreOffice.

Floating dialogs

The Formula Editor and Elements dock can cover a large part of your document. To help create more space and/or allow you to move either the Formula Editor or Elements dock out of the way, you can turn both of them into floating dialogs.

1) Position the cursor on the frame.
2) Hold down the *Ctrl* key and double-click. This turns the Formula Editor into the Commands dialog (Figure 18) and the Elements dock into the Elements dialog (Figure 19).
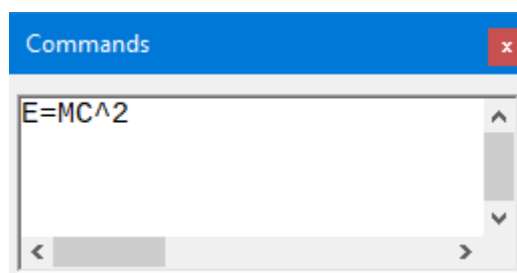
Figure 18: Commands dialog
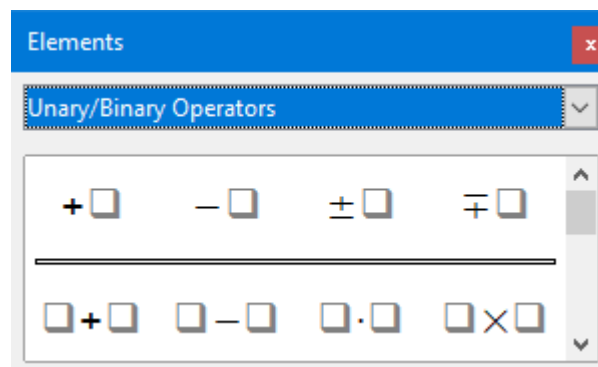
Figure 19: Elements dialog

To return the Commands dialog and Elements dialog to their default positions:

1) Position the cursor on the frame of the dialog, NOT the title bar at the top of the dialog.
2) Hold down the *Ctrl* key and double-click.

Adding keyboard shortcuts

You can add keyboard shortcuts to LibreOffice to make creating documents much easier and to match your workflow. Below is an example of how to add a keyboard shortcut for inserting a Math formula into a LibreOffice document.

> ⚠ **Caution**
>
> When selecting new keyboard shortcuts, make sure that you do not select a shortcut that is already in use by LibreOffice or your computer system.

Example keyboard shortcut

1) Go to **Tools > Customize** on the Menu bar to open the Customize dialog (Figure 20).
2) Click on the **Keyboard** tab to access the options available for adding keyboard shortcuts.
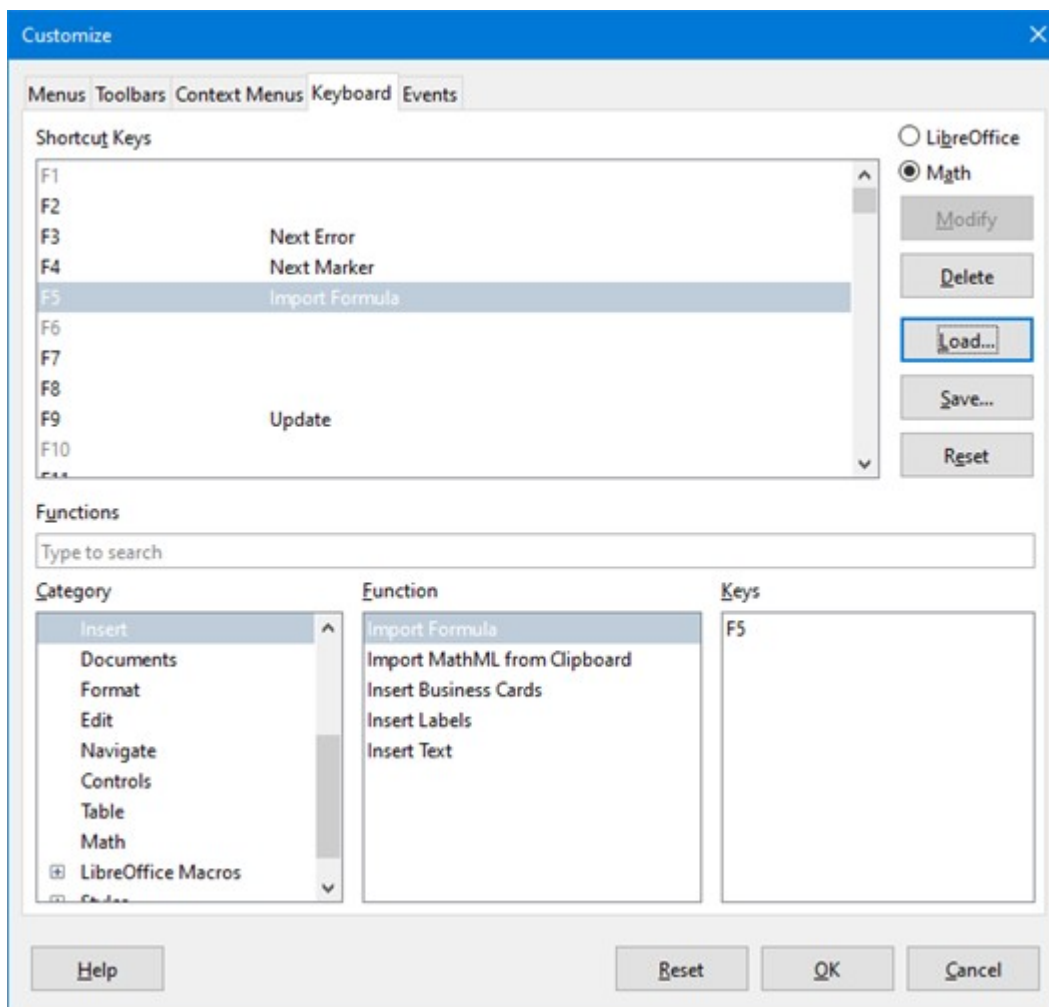
Figure 20: Customize dialog – Keyboard tab

3) Select the level of the new keyboard shortcut.
   – Select *LibreOffice* if you want the new keyboard shortcut to be set for all components of LibreOffice.
   – Select *Math* if you want the new keyboard shortcut to be set for Math only.
4) In the **Category** list, select *Insert***.**
5) In the **Function list**, select *Import Formula***.**
6) In the **Shortcut Keys** list, select the key or keyboard combination that you want to use for your new keyboard shortcut.
7) Click **Modify** and your keyboard shortcut will appear in the **Keys** list.
8) If necessary, continue to add keyboard shortcuts using the above steps.
9) Click **OK** to save your keyboard shortcuts and close the Customize dialog.

> **ℹ️ Tip**
>
> If you frequently add formulas into Writer documents, you can set a shortcut to quickly create Math Formulas. This can be done by following the steps described above to assign a shortcut to the function *Formula* in the category *Insert*.
>
> Be aware that there are two Formula commands available in the Customize dialog in Writer and, although they have the same name, they correspond to different LibreOffice commands. As shown in Figure 21, the first Formula command is used to insert a calculated field in Writer, whereas the second Formula command is the one used to insert new Math objects.
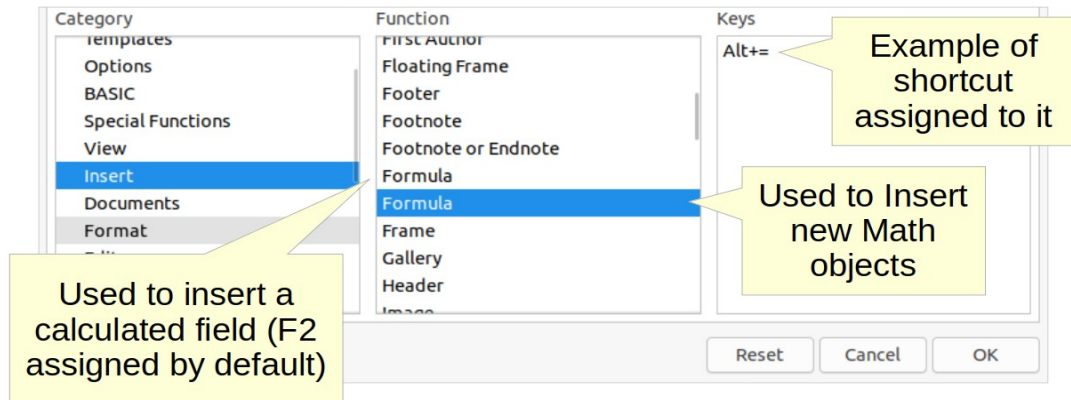
Figure 21: Assigning a shortcut to insert new Math objects in Writer

Keyboard customize options

The options for customizing keyboard shortcuts are as follows:

- **LibreOffice** – displays shortcut keys that are common to all LibreOffice components.
- **Math**, **Writer**, **Calc**, **Draw**, or **Impress** – displays the keyboard shortcuts for the open and current LibreOffice component.
- **Shortcut Keys** – lists the shortcut keys and the associated commands. To assign or modify the shortcut key for the command selected in the *Function* list, click a shortcut in this list, and then click **Modify**.
- **Functions** – type function name in the text field for search it in Function list.
- **Function** – lists the function categories and the LibreOffice functions that you can assign shortcut keys to.
    - *Category* – lists the available function categories.
    - *Function* – select a function that you want to assign a shortcut key to, select a key combination in the Shortcut Keys list, and then click Modify. If the selected function already has a shortcut key, it is displayed in the Keys list.
    - *Keys* – displays the shortcut keys that are assigned to the selected function.
- **Modify** – assigns the key combination selected in the Shortcut Keys list to the command selected in the Function list.
- **Delete** – deletes the selected element or elements without requiring confirmation.
- **Load** – replaces the shortcut key configuration with one that was previously saved.
- **Save** – saves the current shortcut key configuration, so that you can load it later.
- **Reset** – resets modified values back to the default values.

Catalog customization

If you regularly use a symbol that is not available in Math, you can add it to the Symbols dialog (Figure 22) by using the Edit Symbols dialog (Figure 23).

You can add symbols to a symbol set, edit symbol sets, or modify symbol notations. You can also define new symbol sets, assign names to symbols, or modify existing symbol sets.

Adding symbols

1) Go to **Tools > Symbols** on the Menu bar or click on the Symbols icon on the Tools toolbar to open the Symbols dialog (Figure 22).

2) Click the **Edit** button to open Edit Symbols dialog (Figure 23).

3) Select a font in the *Font* drop-down list.

4) Select a symbol character that you want to add in the preview box. You may have to scroll down in the preview box to locate the symbol you want to use. The small right preview box displays the new symbol.

5) In the *Symbol* box, type a memorable name for the symbol you are adding.

6) In the *Symbol set* box, select a symbol set in the drop-down list to add your new symbol to, or type a new name to create a new symbol set for your new symbol.

7) If required, select a font style from the *Style* drop-down list – Standard, Italic, Bold, or Bold Italic.

8) Click **Add**, then click **OK** to close the Edit Symbols dialog. The new symbol and, if created, new symbol set are now available for use.
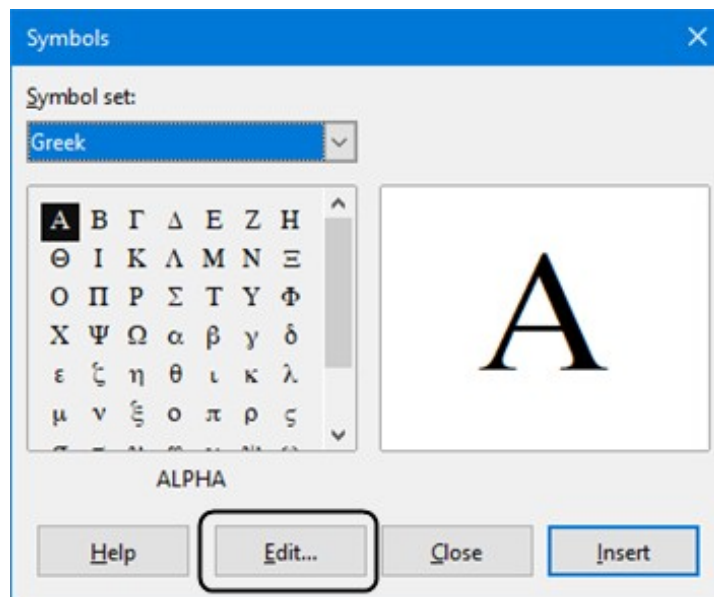
Figure 22: Symbol dialog
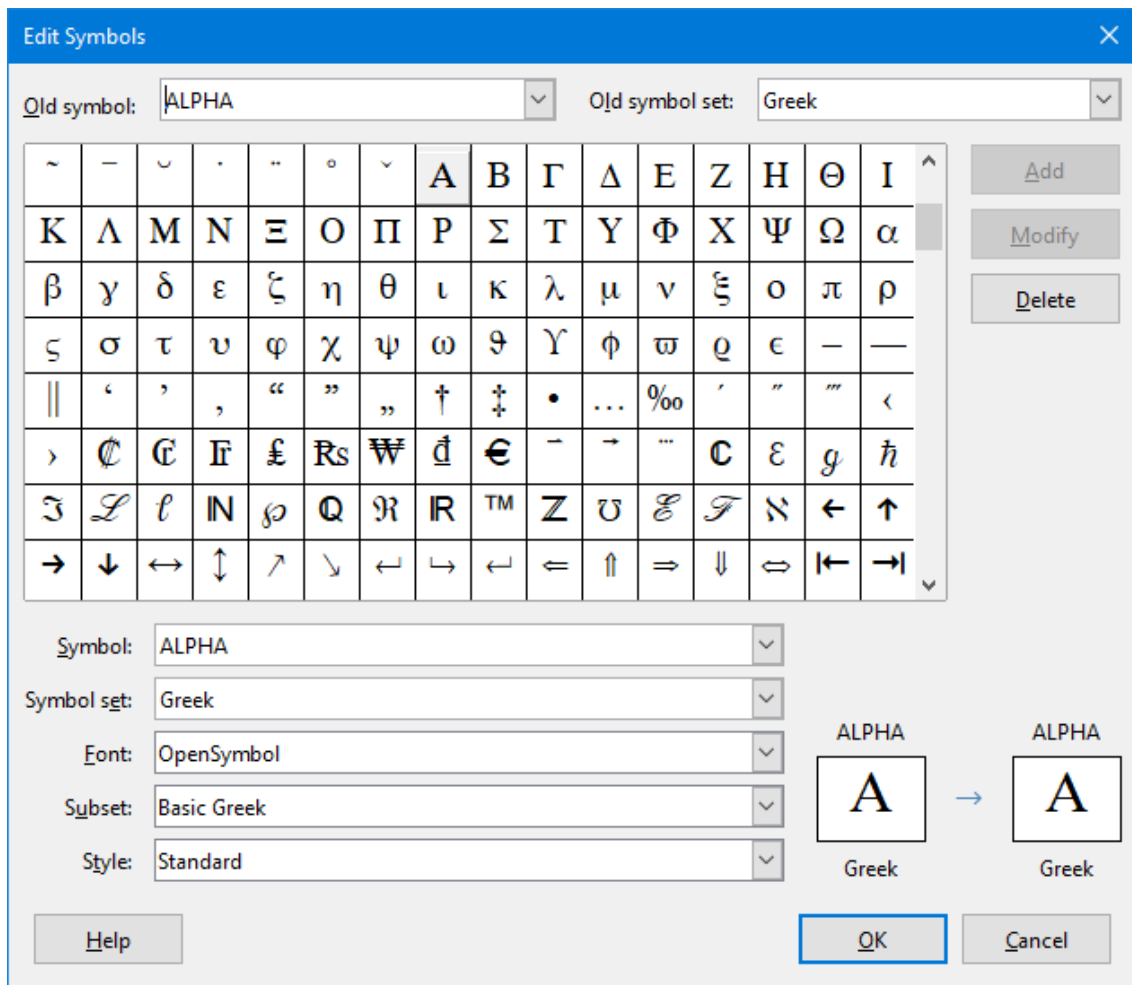
Figure 23: Edit Symbols dialog

> ✅ **Note**
>
> When a new symbol is added to the catalog, you can type a percentage sign (%)
> followed by the new name into the markup language in the Formula Editor and your
> new symbol will appear in the formula. Remember that symbol names are case
> sensitive, for example, `%prime` is a different symbol to `%Prime`.

> ✅ **Note**
>
> There are numerous free fonts available that contain several symbols if you cannot
> find a symbol to use in the fonts already installed on your computer. For example,
> the STIX font was developed specially for writing mathematical and technical texts.
> Also, the DejaVu and Lucida fonts have a wide range of symbols.

> ✅ **Note**
>
> By default, only those user-defined symbols that actually occur in the document are
> stored with it. Sometimes it is useful to embed all the user-defined symbols, so that
> when the document is transferred to another computer it can be edited by another
> person. Go to **Tools > Options > LibreOffice Math > Settings** on the Menu bar and
> uncheck **Embed only used symbols (smaller file size)** in the *Miscellaneous
> Options* setion.

Editing symbols

Modifying symbol names

To change the name of a symbol:

1) In the Edit Symbols dialog (Figure 23), select the symbol name you want to change from the *Old symbol* drop-down list. The symbol appears in the left preview pane at the bottom of the dialog.

2) Type a new name for the symbol in the *Symbol* text box, or select a new name in the *Symbol* drop-down list. The new symbol name appears above the right preview pane at the bottom of the Edit Symbols dialog.

3) Click **Modify** and the symbol name is changed.

4) Click **OK** to close the Edit Symbols dialog.

Moving symbols

To move a symbol from one symbol set to another:

1) In the Edit Symbols dialog (Figure 23), in the *Old symbol set* drop-down list, select the symbol set where the symbol you want to move is located.

2) Select the symbol name you want move in the *Old symbol* drop-down list. The symbol appears in the left preview pane at the bottom of the Edit Symbols dialog.

3) In the *Symbol set* drop-down list, select the symbol set that you want to move the symbol to. The new symbol set name appears below the right preview pane at the bottom of the Edit Symbols dialog.

4) Click **Modify** and the symbol is moved to the new symbol set.

5) Click **OK** to close the Edit Symbols dialog.

Deleting symbols

To delete a symbol from a symbol set:

1) In the Edit Symbols dialog (Figure 23), in the *Old symbol set* drop-down list, select the symbol set from which you want to delete the symbol.

2) Select the symbol name you want to delete in the *Old symbol* drop-down list. The symbol appears in the left preview pane at the bottom of the Edit Symbols dialog (Figure 23).

3) Click **Delete** and the symbol is deleted from the symbol set without any confirmation.

4) Click **OK** to close the Edit Symbols dialog.

✅ Note

The only way you can delete a symbol set is by deleting all of the symbols in that set. When you delete the last symbol from a set, the set is also deleted.

Options for editing symbols

The fields in the Edit Symbols dialog (Figure 23) are as follows:

- **Old symbol** – select the name of the current symbol. The symbol, the name of the symbol, and the set that the symbol belongs to are displayed in the left preview pane at the bottom of the dialog.

- **Old symbol set** – this list box contains the name of the current symbol set. You can also select a different symbol set using this list box.

- **Symbol** – lists the names for the symbols in the current symbol set. Select a name from the list or type a name for a newly added symbol.

- **Symbol set** – this list box contains the names of all existing symbol sets. You can modify a symbol set or create a new one.

- **Font** – displays the name of the current font and enables you to select a different font.

- **Subset** – if you selected a non-symbol font in the Font list box, you can select a Unicode subset in which to place your new or edited symbol. When a subset has been selected, all symbols belonging to this subset of the current symbol set are displayed in the Symbol list.

- **Style** – the current style of font is displayed. You can change the font style by selecting one from the list box.

- **Add** – click this button to add the symbol shown in the right preview window to the current symbol set. It will be saved under the name displayed in the Symbol list. You must specify a name under Symbol or Symbol set to be able to use the Add button. Names cannot be used more than once.

- **Modify** – click this button to replace the name of the symbol shown in the left preview window (the old name is displayed in the Old symbol list) with the new name you have entered in the Symbol list.

- **Delete** – click this button to remove the symbol shown in the left preview window from the current symbol set. There will be confirmation of deletion. Deleting the last remaining symbol of a symbol set also deletes the symbol set.

- **Cancel** – click this button at any time to close the dialog without saving any of the changes.

Formula spacing

The grave accent (`) inserts an additional small space and the tilde (~) inserts an additional large space into formulas. However, in the basic installation of LibreOffice, these symbols are ignored when they occur at the end of a formula. If you are working with running text in a formula, it may be necessary to include spacing at the end of formulas as well. This customization is only required when you are working with a Math document and is not required when you are inserting a formula into another LibreOffice component.

To add spacing at the end of formula in Math, go to **Tools > Options > LibreOffice Math > Settings** on the Menu bar and uncheck **Ignore ~ and ` at the end of the line** in the *Miscellaneous Options* section.

> ✅ **Note**
>
> To customize LibreOffice Math settings, you need to start LibreOffice Math and then navigate to **Tools > Options > LibreOffice Math > Settings**.

Scaling of code in the Formula Editor input window

By default, the code in the Formula Editor input window (Figure 1) is scaled to 100%. You can change the scaling factor by going to **Tools > Options > LibreOffice Math > Settings** on the Menu bar and changing the percent in the **Scaling code input window** box in the *Miscellaneous Options* section. For example, if you change the scaling to 150%, the code in the input window will appear larger. This setting affects the Formula Editor in all LibreOfice components. However, the LibreOffice Math section in the Options dialog can only be accessed from within the Math component. It is not available when you are editing a formula in a LibreOffice document.

Extensions

If you create formulas frequently in your documents, you can customize LibreOffice by adding extensions that are designed to help you create formulas. Extensions are easily installed using the Extension Manager. For more information on how to install extensions, see Chapter 14, Customizing LibreOffice, in the *Getting Started Guide*.

A commonly used extension is **Formatting of All Math Formulas**. It allows you to format all Math formulas in a Writer, Calc, Draw, or Impress document. With it you can change the font names and font sizes of all formulas in the document. For more information on this extension, go to https://extensions.libreoffice.org/en/extensions/show/formatting-of-all-math-formulas.

Math Guide 7.2

Chapter 5,
Exporting and Importing

MathML format

In addition to exporting documents as PDFs, as described in the *Getting Started Guide*, LibreOffice offers the possibility of exporting formulas in the MathML format. This allows you or another person to insert formulas into documents that were created in other software, for example, Microsoft Office or an internet browser.

> ✅ **Note**
>
> Some internet browsers do not fully support the MathML format and your formula may not display correctly.

If you are working on a Math document, go to **File > Save as** on the Menu bar or use the keyboard combination *Ctrl+Shift+S* to open the Save as dialog. Select MathML from the list of available file formats in File type to save your formula as MathML.

If you are working in another LibreOffice component, right-click on the formula object and select **Save copy as** from the context menu to open the Save as dialog. Select MathML from the list of available file formats in File type to save your formula object as MathML.

In Math you can also import MathML formulas. Use **Tools > Import MathML from clipboard** on the Menu bar.

Microsoft file formats

To control how formulas in Microsoft format are imported and exported using LibreOffice, go to **Tools > Options > Load/Save > Microsoft Office** on the Menu bar and select or deselect the options for *MathType to LibreOffice Math or reverse*.

- [L]: Load and convert the object
- [S]: Convert and save the object

[L]: Load and convert the object

Select this option if Microsoft OLE objects are to be converted into the specified LibreOffice OLE objects when a Microsoft document is opened in LibreOffice. For formulas, any embedded MathType objects must not exceed the MathType 3.1 specifications to be successfully loaded and converted. Information on MathType format can be found at https://docs.wiris.com/en/mathtype/office_tools/microsoft_office.

If a document containing OMML formulas has been saved in .docx format and then converted to the older .doc format, then any OMML objects are converted into graphics, which will be displayed in LibreOffice as graphics.

[S]: Convert and save the object

Select this option if LibreOffice OLE objects are to be converted and saved in Microsoft file format. LibreOffice converts any formulas into a format that can be read and modified by Microsoft Equation Editor and MathType.

When this option is not selected, the formula is treated as an OLE object on conversion into a .doc format and remains linked to LibreOffice. A double-click on the object in Microsoft Office will attempt to launch LibreOffice.

Math Guide 7.2

Appendix A,
Commands Reference

Introduction

This appendix lists the operators and commands that are available for use in the Formula Editor when you are creating formulas.

The more common operators and commands can be entered by clicking on the appropriate icon in the Elements dock. Where there is no icon displayed in the following tables for the Elements dock, then you have to enter the operator or command into the Formula Editor using markup language.

> ✅ **Note**
>
> In the markup language shown in the following tables, you have to replace the place holder <?> with the value you want to use in your formulas.

Unary/binary operator commands

To access the icons used for entering markup language for unary/binary operator commands into the Formula Editor, select *Unary/Binary Operators* from the drop-down list at the top of the Elements dock.

Table 3: Unary/binary commands

Operation	Markup language	Example formula
Plus	`+`	$+1$
Minus	`-`	-1
Plus/minus	`+-` or `plusminus`	± 1
Minus/plus	`-+` or `minusplus`	∓ 1
Addition	`<?> + <?>`	$A+B$
Subtraction	`<?> - <?>`	$A-B$
Multiplication (Dot)	`<?> cdot <?>`	$A \cdot B$
Multiplication	`<?> times <?>`	$A \times B$
Multiplication (*)	`<?> * <?>`	$A * B$
Division (Fraction)	`{<?>} over {<?>}`	$\dfrac{A}{B}$
Frac (Fraction)	`frac {<?>} {<?>}`	$\dfrac{A}{B}$
Division	`<?> div <?>`	$A \div B$
Division (Slash)	`<?> / <?>` or `<?> slash <?>`	A/B
Division (Wideslash)	`{<?>} wideslash {<?>}`	$A\big/B$
Division (Counter Wideslash)	`<?> widebslash <?>`	$A\big\backslash B$
Concatenate	`<?> circ <?>`	$A \circ B$
Boolean NOT	`neg <?>`	$\neg A$
Boolean AND	`<?> and <?>` or `<?> & <?>`	$A \wedge B$
Boolean OR	`<?> or <?>`	$A \vee B$

Backslash	`<?> bslash <?>`	$A \setminus B$
Slash in circle	`<?> odivide <?>`	$A \oslash B$
Small multiply symbol in circle	`<?> odot <?>`	$A \odot B$
Subtract symbol in circle	`<?> ominus <?>`	$A \ominus B$
Add symbol in circle	`<?> oplus <?>`	$A \oplus B$
Multiply symbol in circle	`<?> otimes <?>`	$A \otimes B$
User defined binary operator	`<?> boper ????? <?>`	$A \, binOp \, B$
User defined unary operator	`uoper ????? <?>`	$unOp \, B$

Relation commands

To access the icons used for entering markup language for relations commands into the Formula Editor, select *Relations* from the drop-down list at the top of the Elements dock.

Table 4: Relation commands

Operation	Markup language	Example formula
Is equal	`<?> = <?>`	$A = B$
Is not equal	`<?> <> <?>` or `<?> neq <?>`	$A \neq B$
Is less than	`<?> < <?>` or `<?> lt <?>`	$A < B$
Is less than or equal to	`<?> <= <?>`	$A \leq B$
Is less than or equal to	`<?> leslant <?>`	$A \leqslant B$
Is greater than	`<?> > <?>` or `<?> gt <?>`	$A > B$
Is greater than or equal to	`<?> >= <?>`	$A \geq B$
Is greater than or equal to	`<?> geslant <?>`	$A \geqslant B$
Is much less than	`<?> << <?>` or `<?> ll <?>`	$A \ll B$
Is much greater than	`<?> >> <?>` or `<?> gg <?>`	$A \gg B$
Is approximately equal	`<?> approx <?>`	$A \approx B$
Is similar to	`<?> sim <?>`	$A \sim B$
Is similar to or equal	`<?> simeq <?>`	$A \simeq B$
Is congruent to	`<?> equiv <?>`	$A \equiv B$
Is proportional to	`<?> prop <?>`	$A \propto B$
Is parallel to	`<?> parallel <?>`	$A \parallel B$
Is orthogonal to	`<?> ortho <?>`	$A \perp B$
Divides	`<?> divides <?>`	$A \mid B$
Does not divide	`<?> ndivides <?>`	$A \nmid B$
Toward	`<?> toward <?>`	$A \rightarrow B$
Double arrow left	`<?> dlarrow <?>`	$A \Leftarrow B$
Double arrow left and right	`<?> dlrarrow <?>`	$A \Leftrightarrow B$
Double arrow right	`<?> drarrow <?>`	$A \Rightarrow B$

Precedes	`<?> prec <?>`	$A \prec B$
Succeeds	`<?> succ <?>`	$A \succ B$
Precedes or equal to	`<?> preccurlyeq <?>`	$A \preccurlyeq B$
Succeeds or equal to	`<?> succcurlyeq <?>`	$A \succcurlyeq B$
Precedes or equivalent to	`<?> precsim <?>`	$A \precsim B$
Succeeds or equivalent to	`<?> succsim <?>`	$A \succsim B$
Not precedes	`<?> nprec <?>`	$A \nprec B$
Not succeeds	`<?> nsucc <?>`	$A \nsucc B$
Definition	`<?> def <?>`	$A \stackrel{def}{=} B$
Corresponding symbol image of	`<?> transl <?>`	$A \multimapdotbothA B$
Corresponding symbol original of	`<?> transr <?>`	$A \multimapdotbothB B$

Set operation commands

To access the icons used for entering markup language for set operation commands into the Formula Editor, select *Set Operations* from the drop-down list at the top of the Elements dock.

Table 5: Set operation commands

Operation	*Markup language*	*Example formula*
Is in	`<?> in <?>`	$A \in B$
Is not in	`<?> notin <?>`	$A \notin B$
Owns	`<?> owns <?>` or `<?> ni <?>`	$A \ni B$
Intersection	`<?> intersection <?>`	$A \cap B$
Union	`<?> union <?>`	$A \cup B$
Difference	`<?> setminus <?>` or `<?> bslash <?>`	$A \setminus B$
Quotient set (slash) between characters	`<?> slash <?>`	A / B
Subset	`<?> subset <?>`	$A \subset B$
Subset or equal to	`<?> subseteq <?>`	$A \subseteq B$
Superset	`<?> supset <?>`	$A \supset B$
Superset or equal to	`<?> supseteq <?>`	$A \supseteq B$
Not subset	`<?> nsubset <?>`	$A \not\subset B$
Not subset or equal to	`<?> nsubseteq <?>`	$A \nsubseteq B$
Not superset	`<?> nsupset <?>`	$A \not\supset B$
Not superset or equal to	`<?> nsupseteq <?>`	$A \nsupseteq B$
Empty set	`emptyset`	$\emptyset$
Aleph (cardinal numbers)	`aleph`	$\aleph$
Natural numbers set	`setN`	$\mathbb{N}$

Integers set	`setZ`	$\mathbb{Z}$
Set of rational numbers	`setQ`	$\mathbb{Q}$
Real numbers set	`setR`	$\mathbb{R}$
Complex numbers set	`setC`	$\mathbb{C}$

Functions

To access the icons used for entering markup language for function commands into the Formula Editor, select *Functions* from the drop-down list at the top of the Elements dock.

Table 6: Function commands

Operation	Markup language	Example formula
Absolute value	`abs{<?>}`	$\lvert A \rvert$
Factorial	`fact{<?>}`	$A!$
Square root	`sqrt{<?>}`	$\sqrt{A}$
Nth root	`nroot{<?>}{<?>}`	$\sqrt[A]{B}$
Power	`<?>^{<?>}`	A^{B}
Natural exponential function	`func e^{<?>}`	e^{A}
Natural logarithm	`ln(<?>)`	$\ln(A)$
Exponential function	`exp(<?>)`	$\exp(A)$
Logarithm	`log(<?>)`	$\log(A)$
Sine	`sin(<?>)`	$\sin(A)$
Cosine	`cos(<?>)`	$\cos(A)$
Tangent	`tan(<?>)`	$\tan(A)$
Cotangent	`cot(<?>)`	$\cot(A)$
Hyperbolic sine	`sinh(<?>)`	$\sinh(A)$
Hyperbolic cosine	`cosh(<?>)`	$\cosh(A)$
Hyperbolic tangent	`tanh(<?>)`	$\tanh(A)$
Hyperbolic cotangent	`coth(<?>)`	$\coth(A)$
Inverse sine or arcsine	`arcsin(<?>)`	$\arcsin(A)$
Inverse cosine or arccosine	`arccos(<?>)`	$\arccos(A)$
Inverse tangent or arctangent	`arctan(<?>)`	$\arctan(A)$
Inverse cotangent or arccotangent	`arccot(<?>)`	$\text{arccot}(A)$
Inverse or area hyperbolic sine	`arsinh(<?>)`	$\text{arsinh}(A)$
Inverse or area hyperbolic cosine	`arcosh(<?>)`	$\text{arcosh}(A)$
Inverse or area hyperbolic tangent	`artanh(<?>)`	$\text{artanh}(A)$
Inverse or area hyperbolic cotangent	`arcoth(<?>)`	$\text{arcoth}(A)$
Back epsilon	`backepsilon`	$\backepsilon$

Subscript	`<?> sub <?>`	A_B

Operators

To access the icons used for entering markup language for operator commands into the Formula Editor, select *Operators* from the drop-down list at the top of the Elements dock.

Table 7: Operator commands

Operation	Markup language	Example formula
Limes	`lim <?>`	$\lim A$
Limes subscript bottom	`lim from{<?>} <?>`	$\lim_{A} B$
Limes superscript top	`lim to{<?>} <?>`	$\lim^{A} B$
Limes sup/sub script	`lim from{<?>} to{<?>} <?>`	$\lim_{A}^{B} C$
Limes inferior	`liminf <?>`	$\lim \inf A$
Limes superior	`limsup <?>`	$\lim \sup A$
Sum	`sum <?>`	$\sum a$
Sum subscript bottom	`sum from{<?>} <?>`	$\sum_{A} B$
Sum superscript top	`sum to{<?>} <?>`	$\sum^{A} B$
Sum sup/sub script	`sum from{<?>} to{<?>} <?>`	$\sum_{A}^{B} C$
Product	`prod <?>`	$\prod A$
Product subscript bottom	`prod from{<?>} <?>`	$\prod_{A} B$
Product superscript top	`prod to{<?>} <?>`	$\prod^{A} B$
Product sup/sub script	`prod from{<?>} to{<?>} <?>`	$\prod_{A}^{B} C$
Coproduct	`coprod <?>`	$\coprod A$
Coproduct subscript bottom	`coprod from{<?>} <?>`	$\coprod_{A} B$
Coproduct superscript top	`coprod to{<?>} <?>`	$\coprod^{A} B$
Coproduct sup/sub script	`coprod from{<?>} to{<?>} <?>`	$\coprod_{A}^{B} C$
Integral	`int <?>`	$\int A$
Integral subscript bottom	`int from{<?>} <?>`	$\int_{A} B$

Integral superscript top	`int to{<?>} <?>`	$\displaystyle\int^{A} B$
Integral sup/sub script	`int from{<?>} to{<?>} <?>`	$\displaystyle\int_{A}^{B} C$
Double integral	`iint <?>`	$\displaystyle\iint A$
Double integral subscript bottom	`iint from{<?>} <?>`	$\displaystyle\iint_{A} B$
Double integral superscript top	`iint to{<?>} <?>`	$\displaystyle\iint^{A} B$
Double integral sup/sub script	`iint from{<?>} to{<?>} <?>`	$\displaystyle\iint_{A}^{B} C$
Triple integral	`iiint <?>`	$\displaystyle\iiint A$
Triple integral subscript bottom	`iiint from{<?>} <?>`	$\displaystyle\iiint_{A} B$
Triple integral superscript top	`iiint to{<?>} <?>`	$\displaystyle\iiint^{A} B$
Triple integral sup/sub script	`iiint from{<?>} to{<?>} <?>`	$\displaystyle\iiint_{A}^{B} C$
Curve integral	`lint <?>`	$\displaystyle\oint A$
Curve integral subscript bottom	`lint from{<?>} <?>`	$\displaystyle\oint_{A} B$
Curve integral superscript top	`lint to{<?>} <?>`	$\displaystyle\oint^{A} B$
Curve integral sup/sub script	`lint from{<?>} to{<?>} <?>`	$\displaystyle\oint_{A}^{B} C$
Double curve integral	`llint <?>`	$\displaystyle\oiint A$
Double curve integral subscript bottom	`llint from{<?>} <?>`	$\displaystyle\oiint_{A} B$
Double curve integral superscript top	`llint to{<?>} <?>`	$\displaystyle\oiint^{A} B$
Double curve integral sup/sub script	`llint from{<?>} to{<?>} <?>`	$\displaystyle\oiint_{A}^{B} C$
Triple curve integral	`lllint <?>`	$\displaystyle\oiiint A$
Triple curve integral subscript bottom	`lllint from{<?>} <?>`	$\displaystyle\oiiint_{A} B$
Triple curve integral superscript top	`lllint to{<?>} <?>`	$\displaystyle\oiiint^{A} B$
Triple curve integral sup/sub script	`lllint from{<?>} to{<?>} <?>`	$\displaystyle\oiiint_{A}^{B} C$

> ✅ **Note**
>
> For customized operators, use the command `oper` followed by your custom
> operator. For example, entering `oper OP from 0 to 1 A` will give the formula
>
> $$\underset{0}{\overset{1}{OP}}\, A\ .$$

Attributes

To access the icons used for entering markup language for attribute commands into the Formula
Editor, select *Attributes* from the drop-down list at the top of the Elements dock, or click the
Attributes icon in the Elements dialog.

Table 8: Attribute commands

Operation	Markup language	Example formula
Acute accent	`acute <?>`	$\acute{A}$
Grave accent	`grave <?>`	$\grave{A}$
Breve	`breve <?>`	$\breve{A}$
Circle	`circle <?>`	$\mathring{A}$
Dot	`dot <?>`	$\dot{A}$
Double dot	`ddot <?>`	$\ddot{A}$
Triple dot	`dddot <?>`	$\dddot{A}$
Line above	`bar <?>`	$\bar{A}$
Vector arrow	`vec <?>`	$\vec{A}$
Harpoon	`harpoon <?>`	$\vec{A}$
Tilde	`tilde <?>`	$\tilde{A}$
Circumflex	`hat <?>`	$\hat{A}$
Reverse circumflex	`check <?>`	$\check{A}$
Large vector arrow	`widevec {<?>}`	$\overrightarrow{AB}$
Large harpoon	`wideharpoon <?>`	$\overrightarrow{AB}$
Large tilde	`widetilde {<?>}`	$\widetilde{AB}$
Large circumflex	`widehat {<?>}`	$\widehat{AB}$
Line over	`overline {<?>}`	$\overline{AB}$
Line below	`underline {<?>}`	$\underline{AB}$
Line through	`overstrike {<?>}`	$\cancel{AB}$
Transparent (blank placeholder to create space)	`phantom {<?>}`	$A\quad B$
Bold font	`bold <?>`	$\mathbf{AB}$
Not bold font	`nbold <?>`	AB
Italic font	`ital <?> or italic <?>`	AB
Not italic font	`nitalic <?>`	AB

Resize	`size <?> {<?>}`	*AB*
Change font	`font <?> {<?>}`	*AB*
Color black	`color black {<?>}`	*AB*
Color blue	`color blue {<?>}`	*AB*
Color green	`color green {<?>}`	*AB*
Color red	`color red {<?>}`	*AB*
Color cyan	`color cyan {<?>}`	*AB*
Color aqua (same as cyan)	`color aqua {<?>}`	*AB*
Color magenta	`color magenta {<?>}`	*AB*
Color fuchsia (same as magenta	`Color fuchsia {<?>}`	*AB*
Color yellow	`color yellow {<?>}`	*AB*
Color gray	`color gray {<?>}`	*AB*
Color lime	`color lime {<?>}`	*AB*
Color maroon	`color maroon {<?>}`	*AB*
Color navy	`color navy {<?>}`	*AB*
Color olive	`color olive {<?>}`	*AB*
Color purple	`color purple {<?>}`	*AB*
Color silver	`color silver {<?>}`	*AB*
Color teal	`color teal {<?>}`	*AB*
RGB colors	`color rgb R G B {<?>}`	*AB*
Hexadecimal colors	`color hex 000000 {<?>}`	*AB*
Color coral	`color coral {<?>}`	*AB*
Color midnight	`color midnightblue {<?>}`	*AB*
Color crimson	`color crimson {<?>}`	*AB*
Color violet	`color violet {<?>}`	*AB*
Color orange	`color orange {<?>}`	*AB*
Color orangered	`color orangered {<?>}`	*AB*
Color seagreen	`color seagreen {<?>}`	*AB*
Color indigo	`color indigo {<?>}`	*AB*
Color hotpink	`color hotpink {<?>}`	*AB*
Color lavender	`color lavender {<?>}`	*AB*

✔ **Note**

The font command changes the font type. The first placeholder is replaced with the name of the font and the second placeholder is replaced with your formula values or text. The default fonts are Serif, Sans, or Fixed. If you have added custom fonts to Math (see "Formula fonts" on page 26), then you can replace the first placeholder with the custom font name.

Brackets

To access the icons used for entering markup language for bracket commands into the Formula Editor, select *Brackets* from the drop-down list at the top of the Elements dock.

Table 9: Bracket commands

Operation	Markup language	Example formula
Group brackets (not displayed in formula)	`{<?>}`	AB
Round brackets	`(<?>)`	$\left(AB\right)$
Left round bracket only	`\(<?>`	$\left(A\right.$
Right round bracket only	`<?>\)`	$A\left.\right)$
Square brackets	`[<?>]`	$\left[AB\right]$
Left square bracket only	`\[<?>`	$\left[A\right.$
Right square bracket only	`<?>\]`	$A\left.\right]$
Double square brackets	`ldbracket <?> rdbracket`	$⟦AB⟧$
Left double square bracket only	`\ldbracket <?>`	$⟦A$
Right double square bracket only	`<?> \rdbracket`	$A⟧$
Braces	`lbrace <?> rbrace`	$\{AB\}$
Left brace only	`\lbrace <?>`	$\{A$
Right brace only	`<?> \rbrace`	$A\}$
Angle brackets	`langle <?> rangle`	$⟨AB⟩$
Left angle bracket only	`\langle <?>`	$⟨A$
Right angle bracket only	`<?> \rangle`	$A⟩$
Operator brackets	`langle <?> mline <?> rangle`	$⟨AB\vert CD⟩$
Ceiling (upper half square brackets)	`lceil <?> rceil`	$\lceil AB\rceil$
Left ceiling only	`\lceil <?>`	$\lceil A$
Right ceiling only	`<?> \rceil`	$A\rceil$
Floor (lower half square brackets)	`lfloor <?> rfloor`	$\lfloor AB\rfloor$
Left floor only	`\lfloor <?>`	$\lfloor A$
Right floor only	`<?> \rfloor`	$A\rfloor$
Single lines	`lline <?> rline`	$\vert AB\vert$
Left single line only	`\lline <?>`	$\vert A$
Right single line only	`<?> \rline`	$A\vert$
Double lines	`ldline <?> rdline`	$\Vert AB\Vert$
Left double line only	`\ldline <?>`	$\Vert A$
Right double line only	`<?> \rdline`	$A\Vert$

Scalable round brackets	`left ( <?> right )`	$\left(\dfrac{A}{B}\right)$
Scalable left round bracket only	`left ( <?> right none`	$\left(\dfrac{A}{B}\right.$
Scalable right round bracket only	`left none <?> right )`	$\left.\dfrac{A}{B}\right)$
Scalable square brackets	`left [ <?> right ]`	$\left[\dfrac{A}{B}\right]$
Scalable left square bracket only	`left [ <?> right none`	$\left[\dfrac{A}{B}\right.$
Scalable right square bracket only	`left none <?> right ]`	$\left.\dfrac{A}{B}\right]$
Scalable double square brackets	`left ldbracket <?> right rdbracket`	$\left[\!\left[\dfrac{A}{B}\right]\!\right]$
Scalable left double square bracket only	`left ldbracket <?> right none`	$\left[\!\left[\dfrac{A}{B}\right.\right.$
Scalable right double square bracket only	`left none <?> right rdbracket`	$\left.\left.\dfrac{A}{B}\right]\!\right]$
Scalable braces	`left lbrace <?> right rbrace`	$\left\{\dfrac{A}{B}\right\}$
Scalable left brace only	`left lbrace {<?>} right none`	$\left\{\dfrac{A}{B}\right.$
Scalable right brace only	`left none {<?>} right rbrace`	$\left.\dfrac{A}{B}\right\}$
Scalable angle brackets	`left langle <?> right rangle`	$\left\langle\dfrac{A}{B}\right\rangle$
Scalable left angle bracket only	`left langle {<?>} right none`	$\left\langle\dfrac{A}{B}\right.$
Scalable right angle bracket only	`left none {<?>} right rangle`	$\left.\dfrac{A}{B}\right\rangle$
Scalable operator brackets	`left langle <?> mline <?> right none`	$\left\langle\dfrac{A}{B}\middle|\dfrac{C}{D}\right\rangle$
Scalable left operator bracket	`left langle <?> mline <?> right none`	$\left\langle\dfrac{A}{B}\middle|\dfrac{C}{D}\right.$
Scalable right operator bracket	`left none <?> mline <?> right rangle`	$\left.\dfrac{A}{B}\middle|\dfrac{C}{D}\right\rangle$
Scalable ceiling (upper half square brackets)	`left lceil <?> right rceil`	$\left\lceil\dfrac{A}{B}\right\rceil$

Scalable left ceiling	`left lceil <?> right none`	$\left\lceil \dfrac{A}{B} \right.$
Scalable right ceiling	`left none <?> right rceil`	$\left. \dfrac{A}{B} \right\rceil$
Scalable floor (lower half square brackets)	`left lfloor <?> right rfloor`	$\left\lfloor \dfrac{A}{B} \right\rfloor$
Scalable left floor	`left lfloor <?> right none`	$\left\lfloor \dfrac{A}{B} \right.$
Scalable right floor	`left none <?> right rfloor`	$\left. \dfrac{A}{B} \right\rfloor$
Scalable single lines	`left lline <?> right rline`	$\left\lvert \dfrac{A}{B} \right\rvert$
Scalable left single line	`left lline <?> right none`	$\left\lvert \dfrac{A}{B} \right.$
Scalable right single line	`left none <?> right rline`	$\left. \dfrac{A}{B} \right\rvert$
Scalable double lines	`left ldline <?> right rdline`	$\left\lVert \dfrac{A}{B} \right\rVert$
Scalable left double line	`left ldline <?> right none`	$\left\lVert \dfrac{A}{B} \right.$
Scalable right double line	`left none <?> right rdline`	$\left. \dfrac{A}{B} \right\rVert$
Scalable top brace	`{<?>} overbrace {<?>}`	$\overbrace{ABC}^{D}$
Scalable bottom brace	`{<?>} underbrace {<?>}`	$\underbrace{ABC}_{D}$

Formats

To access the icons used for entering markup language for format commands into the Formula Editor, select *Formats* from the drop-down list at the top of the Elements dock.

Table 10: Format commands

Operation	Markup language	Example formula
Power (superscript right)	`<?>^{<?>}` or `<?>sup{<?>}` or `<?>rsup{<?>}`	20^{10}
Subscript right	`<?>_{<?>}` or `<?>sub{<?>}` or `<?>rsub{<?>}`	20_{10}
Superscript left	`<?>lsup{<?>}`	$^{10}20$
Subscript left	`<?>lsub{<?>}`	$_{10}20$
Superscript center	`<?>csup{<?>}`	$\overset{10}{20}$

Subscript center	`<?>csub{<?>}`	$\begin{smallmatrix}20\\{\scriptstyle 10}\end{smallmatrix}$
New line	`newline`	123 *CD*
Small gap	`` ` ``	123 456
Blank or large gap	`~`	123 456
No space (suppresses space between elements)	`nospace {<?>}`	12+4+5
Align left	`alignl <?>`	*ABC* *D*
Align center	`alignc <?>`	*ABC* *D*
Align right	`alignr <?>`	*ABC* *D*
Vertical stack (2 elements)	`binom{<?>}{<?>}`	*AB* 12
Vertical stack (more than 2 elements)	`stack{<?> # <?> # <?>}`	*AB* 12 *CD*
Matrix stack	`matrix{<?> # <?> ## <?> # <?>}`	*AB* 12 34 *CD*

> ✅ **Note**
>
> By default, text characters are aligned to the center in a formula. Use the `alignl` and `alignr` commands to align text characters in a formula to the left or right when a formula has more than one line. You can also use the align commands in conjunction with the stack commands to align multi-line formulas on a specific formula element, for example, the equals sign (=).

Others

To access the icons used for entering markup language for other commands into the Formula Editor, select *Others* from the drop-down list at the top of the Elements dock, or click the *Others* icon in the Elements dialog.

Table 11: Other commands

Operation	*Markup language*	*Example formula*
Placeholder	`<?>`	
Infinity	`infinity` or `infty`	∞
Partial derivative or set margin	`partial`	∂
Nabla vector operator	`nabla`	∇
Existential quantifier, there is at least one	`exists`	$\exists$
Existential quantifier, there does not exist	`notexists`	$\nexists$
Universal quantifier, for all	`forall`	$\forall$

H-bar constant	`hbar`	$\hbar$
Lambda bar	`lambdabar`	λ
Real part of a complex number	`re`	$\Re$
Imaginary part of a complex number	`im`	$\Im$
Weierstrass p function	`wp`	$\wp$
Laplace transformation	`laplace`	$\mathscr{L}$
Left arrow	`leftarrow`	←
Right arrow	`rightarrow`	→
Up arrow	`uparrow`	↑
Down arrow	`downarrow`	↓
Ellipsis	`dotslow`	⋯
Math axis ellipsis	`dotsaxis`	⋯
Vertical ellipsis	`dotsvert`	⋮
Upward diagonal ellipsis	`dotsup` or `dotsdiag`	⋰
Downward diagonal ellipsis	`dotsdown`	⋱
Back epsilon	`backepsilon`	϶

Greek characters

Greek characters are normally entered into a formula using the Symbols dialog (see "Greek characters" on page 15 for more information). However, if you know the name of the Greek character, you can type a percentage sign (%) followed by the name of the Greek character into the Formula Editor.

For uppercase Greek characters, type the name using uppercase characters, for example %BETA B (see Table 12 for a full list of uppercase Greek characters). For lowercase Greek characters, type the name using lowercase characters, for example %beta β (see Table 13 for a full list of lowercase Greek characters)

Italic Greek characters can also be entered into a formula by typing the character i after the percentage sign (%) and before the name of the Greek character, for example, %iPI $\mathit{\Pi}$ for uppercase and %ipi for lowercase π . See Table 14 for a full list of italic uppercase Greek characters and Table 15 for a full list of italic lowercase Greek characters.

✅ **Note**

> In localized versions of LibreOffice, the names of Greek characters are also localized. If a document is not localized to the same language, then the names of Greek characters in the following tables may not work correctly. If this happens, you can use the Symbols dialog to select the required Greek character. When selected in the Symbols dialog, the name of the Greek character is displayed at the bottom of the Symbols dialog in the correct language. See "Greek characters" on page 15 for more information on Greek characters and the Symbols dialog.

Table 12: Uppercase Greek characters

%ALPHA	Α	%BETA	Β	%GAMMA	Γ	%DELTA	Δ
%EPSILON	Ε	%ZETA	Ζ	%ETA	Η	%THETA	Θ
%IOTA	Ι	%KAPPA	Κ	%LAMBDA	Λ	%MU	Μ
%NU	Ν	%XI	Ξ	%OMICRON	Ο	%PI	Π
%RHO	Ρ	%SIGMA	Σ	%TAU	Τ	%UPSILON	Υ
%PHI	Φ	%CHI	Χ	%PSI	Ψ	%OMEGA	Ω

Table 13: Lowercase Greek characters

%alpha	α	%beta	β	%gamma	γ	%delta	δ
%varepsilon	ε	%zeta	ζ	%eta	η	%theta	θ
%iota	ι	%kappa	κ	%lambda	λ	%mu	μ
%nu	ν	%xi	ξ	%omicron	ο	%pi	π
%rho	ρ	%varsigma	ς	%sigma	σ	%tau	τ
%upsilon	υ	%varphi	φ	%chi	χ	%psi	ψ
%omega	ω	%vartheta	ϑ	%phi	ϕ	%varpi	ϖ
%varrho	ϱ	%epsilon	ϵ				

Table 14: Uppercase italic Greek characters

%iALPHA	*Α*	%iBETA	*Β*	%iGAMMA	*Γ*	%iDELTA	*Δ*
%iEPSILON	*Ε*	%iZETA	*Ζ*	%iETA	*Η*	%iTHETA	*Θ*
%iIOTA	*Ι*	%iKAPPA	*Κ*	%iLAMBDA	*Λ*	%iMU	*Μ*
%iNU	*Ν*	%iXI	*Ξ*	%iOMICRON	*Ο*	%iPI	*Π*
%iRHO	*Ρ*	%iSIGMA	*Σ*	%iTAU	*Τ*	%iUPSILON	*Υ*
%iPHI	*Φ*	%iCHI	*Χ*	%iPSI	*Ψ*	%iOMEGA	*Ω*

Table 15: Lowercase italic Greek characters

%ialpha	*α*	%ibeta	*β*	%igamma	*γ*	%idelta	*δ*
%ivarepsilon	*ε*	%izeta	*ζ*	%ieta	*η*	%itheta	*θ*
%iiota	*ι*	%ikappa	*κ*	%ilambda	*λ*	%imu	*μ*
%inu	*ν*	%ixi	*ξ*	%iomicron	*ο*	%ipi	*π*
%irho	*ρ*	%ivarsigma	*ς*	%isigma	*σ*	%itau	*τ*
%iupsilon	*υ*	%ivarphi	*φ*	%ichi	*χ*	%ipsi	*ψ*
%iomega	*ω*	%ivartheta	*ϑ*	%iphi	*ϕ*	%ivarpi	*ϖ*
%ivarrho	*ϱ*	%iepsilon	*ϵ*				

Special characters

Special characters are normally entered into a formula using the Symbols dialog (see "Symbols dialog" on page 16 and "Catalog customization" on page 50 for more information). However, if you know the name of the special character, you can type a percentage sign (%) followed by the name of the special character into the Formula Editor. Table 16 Shows the full list of special characters that are available in LibreOffice.

> **✅ Note**
>
> In localized versions of LibreOffice, the names of special characters are also localized. If a document is not localized to the same language, then the names of special characters in the following table may not work correctly. If this happens, you can use the Symbols dialog to select the required special character. When selected in the Symbols dialog, the name of the special character is displayed at the bottom of the Symbols dialog in the correct language.

Table 16: Special characters

|			
`%perthousand`	‰	`%tendto`	→
`%noelement`	∉	`%infinite`	∞
`%angle`	∡	`%and`	∧
`%or`	∨	`%notequal`	≠
`%identical`	≡	`%strictlylessthan`	≪
`%strictlygreaterthan`	≫		

Reserved words

A reserved word is a word that is used by LibreOffice as a command in the Math markup language and it cannot be used as a variable name or entered into a formula as text. However, if you want to use a reserved word as text in a formula then you must place the reserved word between double quotes.

For example, consider that you want to use the word "purple" in a formula and you do not want the formula elements to be changed to the color purple. If you simply type `A purple B` in the Formula Editor, the resulting formula is $A \, ¿ \, B$, which is incorrect and the error is shown by the inverted question mark. To correct this error, type `A "purple" B` in the Formula Editor using double quotes each side of the word purple and the resulting correct formula is $A \, \text{purple} \, B$.

If you want the word "purple" written in italic, as you would normally have with variables, you can type `A italic{ "purple" } B` in the formula editor, which will result in $A \, purple \, B$.

The reserved words used in Math are listed in the above tables for commands in markup language.
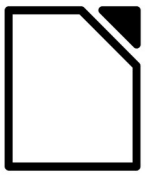
Math Guide

Using the formula editor

About this book:

Math is a simple formula editor that lets you lay out and display mathematical, chemical, electrical, or scientific formulas quickly in standard written notation.

It is most commonly used in Writer for text documents, but it can also be used with other types of documents or stand-alone.

If you have never used LibreOffice before, or you want an introduction to all of its components, you might like to read the *Getting Started Guide* first.

About the authors:

This book was written by volunteers from the LibreOffice community.

A PDF version of this book can be downloaded free from: https://documentation.libreoffice.org

About LibreOffice:

LibreOffice is the free, libre, and open source personal productivity suite from The Document Foundation. It runs on Windows, macOS, and GNU/Linux. Support and documentation is free from our large, dedicated community of users, contributors and developers.

You too can get involved with volunteer work in many areas: development, quality assurance, documentation, translation, user support, and more.

You can download the free Community release of LibreOffice from:
https://libreoffice.org/download/

For commercial use: You are encouraged to work with a certified LibreOffice professional for your business needs, from deployment or conversion off other office suites, staff training, or custom development needs.

Information on professional support services at:
https://www.libreoffice.org/get-help/professional-support/

7.2

Fantastic Community - Fun Project - Free Office Suite