

JUMBO BLOCK ENGINEERING SUITE

Text

User Manual · Version 1.0.0

© 2026 ZANNI GROUP · Brauhof 12, 44866 Bochum, Germany · <https://jumboblock.org>

Table of Contents

1. Overview
2. Supported Files and Languages
3. Getting Started: New, Open and Save
4. Editing and Auto-Indent
5. Find, Replace and Go to Line
6. Encoding, BOM and Line Endings
7. Display: Font, Word Wrap, Light/Dark and Focus Mode
8. Language (German/English)
9. Local Session and Privacy
10. Large-File Mode, Limitations and Notes
11. Quick Reference: Keys and Buttons

1. Overview

JUMBO BLOCK Text is a local text and code editor of the JUMBO BLOCK Engineering Suite. It is used to create, open, edit and save text files, source code and configuration files – for example Python scripts, web pages, JSON and XML files, Markdown documents, shell scripts, INI files or SQL queries.

Text runs entirely locally in the browser. Files are not uploaded to a server, no user account is required, and the application contains no third-party program libraries. The editor core with line numbers, syntax highlighting, auto-indent and find and replace was developed entirely in-house.

The application is designed for desktop, tablet and smartphone and can be reached via the shared menu of the Engineering Suite. The regular layout follows the suite's 1400 px grid; a focus mode is available for concentrated work.

2. Supported Files and Languages

Text opens any text file. Syntax highlighting is available for the following languages; it is chosen from the file extension when a file is opened and can be changed at any time with the "Language" selection:

- Python (.py, .pyw, .pyi): keywords, strings including multi-line docstrings, comments, numbers, decorators and function and class names.
- JavaScript (.js, .mjs, .cjs, .jsx; TypeScript files are shown like JavaScript), JSON (.json, .dbm, .geojson), HTML (.html, .htm) with embedded CSS and JavaScript, and CSS (.css).
- XML (.xml, .svg, .xsd, .xsl, .gpx, .kml and GAEB files .x80 to .x84) and Markdown (.md, .markdown).
- Shell (.sh, .bash, .zsh), INI / Config (.ini, .cfg, .conf, .toml, .properties, .env, .yaml, .yml) and SQL (.sql).
- Plain text: all other files, for example .txt, .log or .csv, are shown without highlighting.

Highlighting is deliberately lightweight and works line by line with state carry-over so that large files stay responsive. It is not a full language parser; in rare special cases (for example regular expressions in JavaScript or here-documents in shell scripts) the colouring may differ from the actual language semantics. The file content itself is never changed by this.

3. Getting Started: New, Open and Save

"New" creates an empty document named "untitled.txt". "Open" (Ctrl+O) selects a file via the operating system dialog. Alternatively, a file can be dragged directly onto the editor area. If a file appears to be binary (for example an image or an Office document), Text asks before opening it.

The file name is shown at the top right of the header. A leading dot "●" indicates unsaved changes. Before unsaved changes are discarded by "New" or "Open", Text asks for confirmation; when the browser tab is closed, the browser warns as well.

Depending on the browser, there are two ways of saving:

- Direct saving: in browsers with the File System Access API (for example Chrome or Edge on the desktop), "Open" opens the file with write access. "Save" (Ctrl+S) then writes directly to the original file. A "↔" after the file name indicates this connection. The browser may ask for permission once. For new documents, a save dialog appears on the first save.

- Download: in all other browsers, “Save” stores the file as a download under the current file name, usually in the browser’s download folder.

“Save as...” (Ctrl+Shift+S) saves under a new name or to a new location. Without the File System Access API, the new file name is requested and the file is downloaded. If the file extension changes, Text adjusts the highlighting language automatically.

4. Editing and Auto-Indent

Line numbers are shown to the left of the text; the line containing the cursor is subtly highlighted. Clicking a line number selects the whole line. “All” or Ctrl+A selects the entire text. “↶” and “↷” or Ctrl+Z and Ctrl+Y undo or redo changes; indenting, commenting and replacing can be undone in the same way. Opening another file starts a new history.

Indentation is set under “Indent”: 2, 4 or 8 spaces or a tab with a width of 4 or 8. When a file is opened, Text detects the indentation used in the file and applies it to this document.

- Tab inserts an indent, with spaces up to the next indent level. If several lines are selected, all selected lines are indented; Shift+Tab outdents them again.
- Enter keeps the indentation of the current line. In Python, the next line is indented one level after a colon and outdented one level after return, pass, break, continue or raise.
- In JavaScript, JSON, CSS, shell and SQL, the next line is indented after “{”, “[” or “(”; if the closing bracket directly follows the cursor, it is moved to its own line. A closing bracket at the start of an empty line outdents automatically. In HTML and XML, the next line is indented after an opening tag.
- In Markdown, bullet lists, numbered lists and task lists are continued. Pressing Enter in an empty list item ends the list.
- Backspace within leading spaces removes a whole indent level.

Ctrl+/ (on German keyboards also Ctrl+#) comments or uncomments the current line or all selected lines. Depending on the language, #, //, --, /* ... */ or <!-- ... --> is used. JSON and “Plain text” have no comment markers.

Because Tab is used for indenting inside the editor, press Esc and then Tab to leave the editor with the keyboard.

5. Find, Replace and Go to Line

Ctrl+F or the magnifier opens the find bar; selected text is used as the search term. The search runs while you type. Enter or F3 jumps to the next match, Shift+Enter or Shift+F3 to the previous one. The indicator “3 / 17” shows the current match and the total; the current match is highlighted in colour.

- “Aa” matches case, “W” finds whole words only, “.” treats the search term as a regular expression (JavaScript syntax).
- Ctrl+H opens the bar with the “Replace with” field. “Replace” replaces the current match and jumps to the next one, “Replace all” replaces all matches at once. With regular expressions, \$1, \$2 ... stand for groups, \$<name> for named groups, \$& for the whole match and \$\$ for a dollar sign.
- All replacements can be undone with Ctrl+Z. Up to 100,000 matches are counted.

Ctrl+G or “↵#” opens “Go to line”. Enter a line number such as “120” or line and column such as “120:5”.

6. Encoding, BOM and Line Endings

When a file is opened, Text detects the character encoding automatically: a byte order mark (BOM) identifies UTF-8, UTF-16 LE or UTF-16 BE. Without a BOM, UTF-16 is recognised by typical byte patterns; valid UTF-8 is read as UTF-8, everything else as ISO-8859-1. The detected encoding, an existing BOM and the line ending are shown in the status bar.

- “Encoding” defines how the file is written when saving: UTF-8, UTF-16 LE, UTF-16 BE or ISO-8859-1. “BOM” determines whether a byte order mark is written (not for ISO-8859-1). When switching to UTF-16, the BOM is enabled automatically.
- If a file was detected incorrectly, “↺” reloads the original file with the selected encoding. This is possible while the original data is in memory, i.e. after opening or saving in the current session.
- If the text contains characters that cannot be represented in ISO-8859-1 (for example €, typographic quotes or emojis), Text warns when saving. You can replace these characters with “?” or cancel; the first affected character is then selected.
- Windows-1252 files are read as ISO-8859-1. Bytes 0x80 to 0x9F (in Windows-1252 among others €, „ and “) appear as control characters and are preserved byte for byte when saving in ISO-8859-1.

The line ending (LF, CRLF or CR) is detected as well and kept when saving. If a file contains mixed line endings, the status bar shows “(mixed)”; when saving, all lines are written uniformly with the selected line ending. A changed encoding, BOM or line ending counts as a modification and is indicated with “●”.

7. Display: Font, Word Wrap, Light/Dark and Focus Mode

“Font” offers common programming fonts, for example Cascadia Code, JetBrains Mono, Fira Code, Consolas, SF Mono, Menlo, DejaVu Sans Mono, Liberation Mono or Courier New. Text uses only fonts installed on the device and does not load fonts from the internet. Fonts that are not installed are marked “(not installed)”; a fallback font is then used automatically. “System (monospace)” selects the best available font. Ligatures are deliberately disabled so that cursor and characters line up exactly.

- A– and A+ change the font size between 9 and 36 px.
- “ Wrap” or Alt+Z turns word wrap on or off. With word wrap, line numbers still refer to the actual lines of the file.
- “ ” switches between light and dark display. “ ” enables focus mode, which hides the menu, header, settings and status bar; Esc ends it.
- “ ” shows or hides the settings bar. On smartphones it is hidden initially.

Text remembers font, font size, word wrap, default indentation, light/dark and the local session option locally in the browser.

8. Language (German/English)

The interface is bilingual. The language is selected with the DE and EN buttons in the shared menu of the Engineering Suite and remembered for all suite applications. Labels, hints, status messages and the status bar switch immediately; the file content remains unchanged.

9. Local Session and Privacy

If “Keep session locally” is enabled (default), Text stores the current text together with file name, language, encoding, line ending and cursor position shortly after each change in the browser’s local storage (IndexedDB). On the next start, the last session is restored – including changes that have not been saved yet. If the browser supports it, the connection for direct saving is kept as well; the browser then asks for permission again on the next save.

If the option is turned off, Text deletes the stored session immediately. On shared devices, turning the option off is recommended. The local session does not replace saving the file: clearing browser data or using a private window removes it.

No content, file names or usage data are transmitted to servers. Once the page has loaded, Text does not require an internet connection.

10. Large-File Mode, Limitations and Notes

From about 2 million characters or 60,000 lines, Text switches automatically to large-file mode, indicated in the status bar. Syntax highlighting, current-line highlighting and match highlighting are then disabled; line numbers are only shown without word wrap. Editing, find, replace, go to line and saving continue to work.

- Performance with very large files depends on the device and browser. Files of many megabytes can be opened but are not recommended for everyday editing.
- Text edits one file at a time. A print function is not included.
- With very wide characters (for example East Asian characters or emojis), automatic horizontal scrolling to the cursor may be imprecise; the text itself is not affected.
- Direct saving to the original file requires the File System Access API. In other browsers, the file is downloaded and may have to be moved to the target location manually.
- Important files should be backed up independently of the editor.

11. Quick Reference: Keys and Buttons

Key / Button	Meaning
New	Create a new, empty document
Open / Ctrl+O	Open a file (also via drag and drop)
Save / Ctrl+S	Save – directly to the original file or as a download
Save as... / Ctrl+Shift+S	Save under a new name or to a new location
↶ / Ctrl+Z	Undo
↷ / Ctrl+Y	Redo
All / Ctrl+A	Select the entire text

Key / Button	Meaning
Magnifier / Ctrl+F	Find
Ctrl+H	Find and replace
F3 / Shift+F3	Next / previous match
↵# / Ctrl+G	Go to line (line or line:column)
 Wrap / Alt+Z	Turn word wrap on/off
Tab / Shift+Tab	Indent / outdent
Ctrl+/ or Ctrl+#	Comment / uncomment lines
Click on line number	Select the whole line
	Reload original file with the selected encoding
A- / A+	Decrease / increase font size
	Light / dark display
	Turn focus mode on/off
	Show/hide the settings bar
Esc	Close find or go-to bar, end focus mode; Esc followed by Tab leaves the editor