

PENULTIMATE EDITOR

The next to last editor you will need.

USER MANUAL

Version 0.13

A single-file, offline programmer's editor built on CodeMirror 6
HTML • JavaScript • CSS

Copyright © 2026 Steven Davis. All rights reserved.

Contents

Contents.....	2
1. Introduction.....	4
Who this manual is for.....	4
Key characteristics.....	4
Getting started.....	4
2. The Interface.....	5
The menu bar.....	5
The editor panel and toolbar.....	5
The status bar.....	5
The lint results panel.....	5
3. The File Menu.....	6
File commands.....	6
Re-interpreting the current buffer.....	6
4. The Edit Menu.....	7
Undo, redo, and the clipboard.....	7
Selection.....	7
Commenting and indentation.....	7
5. The Search Menu.....	8
Using the search panel.....	8
The Go To Line dialog.....	8
6. The View Menu.....	9
Folding.....	9
Font size.....	9
Word wrap.....	9
7. The Format Menu.....	10
Pretty-printing.....	10
The Indentation dialog.....	10
8. The Tools Menu.....	11
9. The Help Menu.....	12
About Penultimate Editor.....	12
10. Linting.....	13
Running the linter.....	13
What the JavaScript linter checks.....	13

HTML linting.....	13
Working with the results.....	13
11. Working with Languages.....	14
How the mode is chosen.....	14
HTML documents.....	14
Language-aware features.....	14
12. Keyboard Shortcuts.....	15
Files.....	15
Editing.....	15
Selection.....	15
Search and navigation.....	15
Format and tools.....	15
View and indentation.....	16
13. Tips and Notes.....	17

1. Introduction

Penultimate Editor is a programmer's file editor that runs entirely inside a single HTML file. It is built on CodeMirror 6 and provides full language support for HTML (including embedded JavaScript and CSS), standalone JavaScript, and CSS. Everything you need—syntax highlighting, code folding, bracket matching, find and replace, linting, and Crockford-style pretty-printing—is contained in that one file, with no installation and no network connection required.

Who this manual is for

This manual is a complete reference to every menu, command, dialog, and toolbar control in Penultimate Editor v0.13. It is written for anyone who edits source files and wants a lightweight, dependable tool that works even on locked-down computers where installing software or opening browser developer tools is not permitted.

Key characteristics

- **Single file.** The entire application is one self-contained HTML document. There is nothing to install.
- **Fully offline.** It runs from the local file system (a file:// address) and never contacts the network.
- **Real language parsers.** Highlighting, folding, and bracket matching are powered by genuine Lezer grammars, not regular-expression approximations.
- **Three languages.** HTML, JavaScript, and CSS, with HTML documents understanding their embedded script and style blocks.
- **Built-in linting.** A JavaScript analyzer flags common mistakes without any external tooling—useful where developer consoles are unavailable.

Getting started

Opening the editor takes a single step:

- Double-click the Penultimate Editor HTML file, or open it in any modern web browser.

The editor loads immediately. A ready-to-edit source area appears with the menu bar across the top, the editor toolbar just above the text, and a status bar along the bottom. You can begin typing right away, or use File › Open... to load an existing file.

2. The Interface

The Penultimate Editor window is divided into four regions, top to bottom: the menu bar, the editor panel header with its toolbar, the source editor itself, and the status bar. A lint results panel appears on demand.

The menu bar

The menu bar runs across the top of the window. At its left edge is the brand marker—the letters PE with a small v0.13 next to them—followed by seven menus: File, Edit, Search, View, Format, Tools, and Help. Click any menu label to open its dropdown. At the far right of the menu bar is a live document-statistics readout showing the current line and character counts (for example, “128 lines, 3,904 chars”).

Some menu entries display their keyboard shortcut on the right. Where a shortcut uses “mod,” that means the Ctrl key on Windows and Linux, and the Command key on macOS. Menu items that toggle a setting—such as the active language mode or word wrap—show a checkmark when active.

The editor panel and toolbar

Below the menu bar, the editor panel is titled “Source Editor.” Its header carries a toolbar with the most frequently used controls so they are always within reach:

- **Indent selector.** A dropdown offering 2, 4, 6, or 8 spaces, or a Tab character. The default is 4 spaces.
- **Pretty-Print button.** Reformats the whole document in Crockford style.
- **Lint button.** Analyzes the document and opens the lint results panel.
- **Find / Replace button.** Opens the search panel.
- **Go To button.** Opens the Go To Line dialog.

The status bar

The status bar along the bottom of the window is split in two:

- **Left side.** A colored status dot and a status message (for example, “Ready”) that reflect the editor’s current state.
- **Right side.** The current language mode (“Mode: HTML”), the current indentation setting (“Indent: 4 spaces”), and a saved/dirty indicator that reads “Saved” when there are no unsaved changes.

The lint results panel

When you run Lint, a results panel opens. Its header reads “Lint Results” and offers an “Errors only” checkbox to hide warnings, along with Copy, Download, and Close buttons. Below the header is the list of findings, each color-coded by severity, followed by a summary line. See Section 10, Linting, for full details.

3. The File Menu

The File menu handles creating, opening, and saving documents, and lets you re-interpret the current buffer as a different language.

File commands

Command	Shortcut	What it does
New	mod+N	Starts a new, empty document. If the current document has unsaved changes you are prompted first.
Open...	mod+O	Opens a file from disk. Accepted types include .html, .htm, .js, .mjs, .cjs, .css, and .txt.
Save	mod+S	Saves the current document, writing it back out through the browser's download mechanism.
Save As...	mod+Shift+S	Saves the current document under a new name.

Re-interpreting the current buffer

Below the save commands, three entries let you change how the current text is parsed and highlighted without reloading it from disk. The active mode is marked with a checkmark.

Command	Shortcut	What it does
Treat as HTML	—	Re-interprets the current buffer as HTML, enabling embedded JavaScript and CSS support.
Treat as JavaScript	—	Re-interprets the current buffer as standalone JavaScript.
Treat as CSS	—	Re-interprets the current buffer as CSS.

These commands are useful when a file's extension does not match its contents, or when you paste code into a new document and want the correct language features applied immediately. When a file is opened, the editor also detects the language automatically from the file name and contents.

4. The Edit Menu

The Edit menu covers undo and redo, the clipboard, selection, commenting, and indentation.

Undo, redo, and the clipboard

Command	Shortcut	What it does
Undo	mod+Z	Reverses the most recent change.
Redo	mod+Y	Re-applies a change that was undone. (mod+Shift+Z also works.)
Cut	mod+X	Removes the selected text and places it on the clipboard.
Copy	mod+C	Copies the selected text to the clipboard.
Paste	mod+V	Inserts the clipboard contents at the cursor.

Selection

Command	Shortcut	What it does
Select All	mod+A	Selects the entire document.
Select Word	mod+D	Selects the word under the cursor; used again, extends the selection to the next matching occurrence. Double-clicking also selects a word.
Select Line	mod+Shift+L	Selects the current line. Triple-clicking also selects a line.
Expand Selection	mod+Shift+E	Grows the selection outward to the next enclosing syntactic region—word, then expression, then block, and so on.

Commenting and indentation

Command	Shortcut	What it does
Toggle Comment	mod+/**	Comments or uncomments the current line or selection, using the correct comment syntax for the active language.
Indent More	mod+]	Increases the indentation of the selected lines by one indent unit.
Indent Less	mod+[	Decreases the indentation of the selected lines by one indent unit.

The size of one indent unit is set by the indent selector in the toolbar and by Format › Indentation....

5. The Search Menu

The Search menu provides finding, replacing, and navigation commands.

Command	Shortcut	What it does
Find / Replace...	mod+F	Opens the search panel for finding and replacing text. The panel reports a live match count and shows “No matches” when a search term is not found.
Find Occurrences	mod+Shift+F	Opens the search panel pre-filled with the word under the cursor, so you can immediately step through every occurrence.
Go To Line...	mod+G	Opens a dialog that jumps the cursor to a specific line number.
Go To Matching Bracket	—	Moves the cursor to the bracket that matches the one at the current position.

Using the search panel

When the search panel is open, it shows the number of matches for the current term and reports how many replacements were made after a Replace All. Pressing Escape, or clicking Close, dismisses the panel. Opening the panel clears the previous find text and places the cursor in the find field, ready for a new search. To replace, use mod+H (see the Keyboard Shortcuts reference).

The Go To Line dialog

The Go To Line dialog shows the valid range of line numbers (for example, “Line (1–128)”) and accepts a number. Pressing Enter, or clicking Go, jumps to that line; the value is clamped to the valid range so an out-of-range entry lands on the first or last line rather than failing. Cancel closes the dialog without moving.

6. The View Menu

The View menu controls code folding, font size, and word wrap—everything that affects how the document is displayed without changing its contents.

Folding

Command	Shortcut	What it does
Fold All	—	Collapses every foldable region in the document, showing only the top-level structure.
Unfold All	—	Expands every folded region so the whole document is visible.

Individual blocks can also be folded and unfolded with `mod+Shift+[` and `mod+Shift+]`.

Font size

Command	Shortcut	What it does
Increase Font Size	<code>mod+=</code>	Makes the editor text one step larger.
Decrease Font Size	<code>mod+-</code>	Makes the editor text one step smaller.
Reset Font Size	—	Returns the editor text to its default size.

Word wrap

Command	Shortcut	What it does
Toggle Word Wrap	—	Turns word wrapping on or off. A checkmark shows when wrapping is on. Word wrap is on by default, so long lines are visible without horizontal scrolling.

7. The Format Menu

The Format menu reformats the document and configures indentation.

Command	Shortcut	What it does
Pretty-Print	mod+Shift+P	Reformats the entire document in Crockford style, using the current indent setting.
Indentation...	—	Opens a dialog to choose the indent width.

Pretty-printing

Pretty-Print rewrites the whole document with consistent, Crockford-style formatting at the indentation width you have configured. It is careful to preserve things that should not be reflowed:

- Blank lines between functions are kept, so logical grouping survives.
- Template literals are preserved verbatim, so their internal whitespace is never disturbed.
- In HTML documents, script and style blocks are preserved as written.

The Indentation dialog

The Indentation dialog lets you choose 2, 4, 6, or 8 spaces, or a Tab character, as the indent unit. Apply confirms the choice and Cancel dismisses it without changing anything. This setting is mirrored by the indent selector in the toolbar and is shown in the status bar. It governs Indent More, Indent Less, the Tab key, and Pretty-Print.

8. The Tools Menu

The Tools menu runs the linter and closes its results panel.

Command	Shortcut	What it does
Lint	mod+L	Analyzes the current document for problems and opens the lint results panel.
Close Lint Panel	—	Closes the lint results panel.

Linting is described in full in Section 10.

9. The Help Menu

The Help menu provides the keyboard shortcuts reference and version information.

Command	Shortcut	What it does
Keyboard Shortcuts...	—	Opens a dialog listing every keyboard shortcut, grouped by category.
About...	—	Shows the About dialog with the version and a summary of features.

About Penultimate Editor

The About dialog identifies the application as Penultimate Editor V0.13, “the next to last editor you will need,” and describes it as a programmer’s file editor built on CodeMirror 6 with full support for HTML (including embedded JavaScript and CSS), standalone JavaScript, and CSS. It lists the language features—syntax highlighting, code folding, bracket matching, indentation, and selection-match highlighting, all powered by real Lezer parsers—and the editor features: find/replace, find occurrences, go to line, toggle comment, Crockford-style pretty-print, lint, indent/outdent, font size controls, word wrap, and file open/save.

10. Linting

The linter is designed for environments where a browser's developer console is not available. It analyzes your code and reports problems directly in the editor, so you can catch mistakes without any external tools.

Running the linter

Run the linter from Tools › Lint, the Lint toolbar button, or mod+L. The lint results panel opens with the findings.

What the JavaScript linter checks

The JavaScript analysis was strengthened in v0.13. It flags, among other things:

- Mismatched braces, parentheses, and brackets.
- Assignment used where a comparison was likely intended (assignment in a condition).
- Unreachable code.
- Errors in `typeof` comparisons.
- Double semicolons.
- Duplicate keys within an object literal.
- Outstanding `TODO` and `FIXME` markers.

HTML linting

For HTML documents, the linter extracts the embedded scripts from the raw HTML and analyzes them, while taking care not to raise false positives on content that legitimately sits inside script or style blocks.

Working with the results

- **Errors only.** Tick this checkbox to hide warnings and show only errors.
- **Severity colors.** Each finding is color-coded by severity, and a summary line totals the findings.
- **Copy.** Copies the results to the clipboard as text.
- **Download.** Saves the results to a text file.
- **Close.** Closes the panel (also available as Tools › Close Lint Panel).

11. Working with Languages

Penultimate Editor supports three languages, each backed by a real Lezer parser: HTML, JavaScript, and CSS. The active language—the mode—determines how the document is highlighted, folded, commented, and pretty-printed. The current mode is shown in the status bar.

How the mode is chosen

- When you open a file, the mode is detected automatically from the file name and contents.
- You can override the detected mode at any time with the File menu's Treat as HTML, Treat as JavaScript, and Treat as CSS commands. The active mode carries a checkmark.

HTML documents

In HTML mode the editor understands embedded JavaScript inside script blocks and embedded CSS inside style blocks, applying the correct highlighting and structure to each. Pretty-Print preserves those embedded blocks, and the linter can extract and analyze embedded scripts.

Language-aware features

Because the modes use genuine grammars rather than pattern matching, the following features are accurate to each language:

- Syntax highlighting.
- Code folding that follows the real block structure.
- Bracket matching driven by the parse tree.
- Toggle Comment using the language's own comment syntax.
- Selection-match highlighting for the current selection's other occurrences.

12. Keyboard Shortcuts

The complete shortcut reference below matches the Keyboard Shortcuts dialog under the Help menu. Throughout, “mod” is Ctrl on Windows and Linux and Command on macOS, and “Shift” is the Shift key.

Files

Shortcut	Action
mod N	New file
mod O	Open file
mod S	Save
mod Shift S	Save As

Editing

Shortcut	Action
mod Z	Undo
mod Y / mod Shift Z	Redo
mod /	Toggle comment
mod]	Indent more
mod [	Indent less

Selection

Shortcut	Action
mod D	Select word / next occurrence
mod Shift L	Select line
mod Shift E	Expand selection
Double-click	Select word
Triple-click	Select line

Search and navigation

Shortcut	Action
mod F	Find / Replace
mod Shift F	Find Occurrences (word under cursor)
mod G	Go to line
mod H	Replace

Format and tools

Shortcut	Action
mod Shift P	Pretty-print

Shortcut	Action
mod L	Lint

View and indentation

Shortcut	Action
mod = / mod -	Font size increase / decrease
Tab	Insert indent
Shift Tab	Outdent
mod Shift [	Fold block
mod Shift]	Unfold block

13. Tips and Notes

- **Match the mode to the content.** If highlighting looks wrong after pasting code into a new document, use the File menu's Treat as... commands to set the right language.
- **Lint before you save.** On computers without a developer console, a quick mod+L is the fastest way to catch mismatched brackets, assignment-in-condition mistakes, and stray TODO markers.
- **Set your indent first.** Choose your indent width in the toolbar or Format › Indentation... before pretty-printing, since Pretty-Print uses the current setting.
- **Step through occurrences quickly.** Place the cursor in a word and press mod+Shift+F to open the search panel already populated with that word.
- **Expand Selection is structural.** mod+Shift+E grows the selection along the parse tree—word, expression, block—which is often faster than dragging.
- **Watch the saved indicator.** The right side of the status bar shows “Saved” when there are no unsaved changes, so you always know where you stand.
- **It stays offline.** The editor runs from the local file system and needs no network connection, which makes it safe to use on restricted or air-gapped machines.