

Quicknotes - Quick Reference
Revision: 2026-05-10T20:30:00

A lightweight Markdown note-taking app with tabs, syntax-highlighted preview, clickable hyperlinks, an integrated Windows-command runner, and PDF export. Single-file Python script, standard library only -- no pip install needed.

XYWRITE-STYLE COMMAND LINE

The gray strip at the bottom-left of the window is a short, named-command line modeled on the old DOS word processor XyWrite. Type a command and press Enter to run it; chain multiple commands with ';'.

F5	Blank the command line and focus it
F10	Toggle focus between editor and cmdline
Up / Down	Recall command history
Tab	Complete the verb at the cursor by prefix (built-ins + library routines). Repeated Tab cycles candidates.

GUI cmdline option (Options -> Use GUI command line):
when ON, F5 / F10 open a popup dialog instead of focusing the bottom strip. Same Enter / Up / Down / Tab / Esc bindings inside.

Examples:

SA	Save the active tab
NE notes.md	Create / open notes.md in a new tab
SE /todo/	Find next 'todo' (case-insensitive)
CH /foo/bar/	Replace foo with bar from cursor on
CV /foo/bar/	Selective (verify) replace, prompts Y/N/A/Q per match
X dir	Run a Windows command silently
XY dir	Run it and insert the output below

See Help -> Command Reference (or type COMMANDS) for the full list with descriptions, modifier syntax, and more examples.

TABS

Each open note lives in its own tab. The tab strip is at the top of the editor area. Tab labels show the note title, with a '*' prefix when there are edits since the last manual save. Autosave does NOT clear the marker -- it stays until you press Ctrl+S or use File -> Save. Closing the last tab opens a fresh blank one so you always have somewhere to type.

Ctrl+N	New blank tab (start typing -- the file is created on first keystroke)
Ctrl+W	Close active tab
Ctrl+Tab	Cycle to next tab
Ctrl+Shift+Tab	Cycle to previous tab

Middle-click	Close that tab
Sidebar click	Open in a new tab (or focus an existing tab if already open)

KEYBOARD SHORTCUTS

File

Ctrl+N	New note (new blank tab)
Ctrl+O	Open file (defaults to notes folder; can navigate anywhere). The Open cascade also lists up to 10 recently opened files for quick re-access.
Ctrl+S	Save active tab (force, bypass debounce)
Ctrl+Shift+S	Save As: copy to a new filename
Ctrl+W	Close active tab
Ctrl+Del	Delete active note (and close its tab)

View

Ctrl+P	Toggle Markdown preview
--------	-------------------------

Tools (run Windows commands)

Ctrl+R	Run selection in dialog (auto-runs)
Ctrl+Alt+R	Run selection inline, insert output
Ctrl+Shift+R	Same as Ctrl+Alt+R (alternate)
Ctrl+Enter	Run current line, insert output below
F5 / Alt+Enter	Run current line SILENTLY (no dialog, no insert; output discarded, exit code shown in status bar)

Options

Shift+F11	Toggle line numbers
-----------	---------------------

Edit

Ctrl+F	Find & Replace (modeless)
Ctrl+D	Duplicate current line below
Ctrl+Shift+K	Delete current line
F8	Insert date/time using the most-recently picked format (defaults to %c; persisted across launches)

Insert Date & Time submenu offers 8 strftime masks (locale default, ISO 8601, date-only, time-only, US m/d/y, and a long human-friendly form). The selected mask is inserted at the cursor position.

Tabs / focus

Ctrl+Tab	Next tab
Ctrl+Shift+Tab	Previous tab
Shift+Tab	Move focus forward through fields (plain Tab still inserts a tab in the editor)

MENUS

File	New, Save, Save As, Save selected text as, Export as PDF (plain text), Export as styled PDF (full Markdown via Edge headless), Close note, Delete note, Change notes folder, Open folder in Explorer, Exit.
Edit	Find & Replace... (finds inside current note; distinct from the sidebar 'Find note...' field which filters the note list). Duplicate line (Ctrl+D), Delete line (Ctrl+Shift+K). Insert Date & Time submenu (F8 reuses last-picked strftime mask).
Convert	Text-transformation actions. Currently: Text Encoding... -> pick a target encoding (UTF-8/16, ASCII transliterate, Latin-1, Windows-125x, Mac Roman, CP437/850, Shift-JIS, GBK, Big5, EUC-KR, ...). The note's text is round-tripped through the encoding; chars that can't fit are replaced with '?'. ASCII (transliterate) first folds typographic punctuation (curly quotes -> straight, em/en dash -> -- / -, ellipsis -> ..., NBSP -> space) and then runs NFKD to turn accented letters into their plain equivalents (cafe, naive, Zurich). A confirmation dialog reports how many characters will change. Undoable with Ctrl+Z.
View	Toggle Markdown preview.
Tools	Run command... (full dialog), Run selected text as command, Run selection inline (insert output), Run current line (insert output below), Run current line (silent), Run current line in dialog... .
Options	Fonts (family/style/size), Line numbers, Dark theme, Always on top. All persisted.
Scripts	Auto-populated list of .qns files in your scripts folder; click any to run it. Includes 'Run script...' (file picker) and 'Open scripts folder'. The CMLine command RUN <name> does the same thing.
Help	Usage / Quick Reference (this window), Command Reference (full list of every command you can type in the bottom command line, with descriptions and modifier syntax), About.

EDITOR RIGHT-CLICK

Cut / Copy / Paste, plus the same Run-command actions

as the Tools menu (selection, current line, dialog).

COMMAND LINE (XyWrite-style)

The gray strip at the bottom-left of the window accepts short, named commands -- file ops (NE, SA, AB, SAAS), cursor moves (BF, LU, DN, NW, PW), search (SE/SEA/SEB/SEBA with /W and /T modifiers), global replace (CH/CHA), selective replace (CV/CVA, prompts Y/N/A/Q per match), Windows commands (X, XY), PDF export (PDF, PDFS), and more. Multiple commands chain with ';'.

F5	Blank the command line and focus it
F10	Toggle focus between editor and cmdline
Up / Down	Recall command history
Tab	Complete the verb at the cursor by prefix (built-ins + library routines). Repeated Tab cycles candidates.

GUI cmdline option (Options -> Use GUI command line):
when ON, F5 / F10 open a popup dialog instead of focusing the bottom strip. Same Enter / Up / Down / Tab / Esc bindings inside.

See Help -> Command Reference for the full list with descriptions, modifier syntax, and examples.

MARKDOWN

Headings	# H1 ## H2 ### H3
Emphasis	**bold** <i>*italic*</i> / <u><i>_italic_</i></u> 'code'
Lists	- bullet 1. numbered
Rule	--- on its own line
Link	[text](url)
Bare URL	https://... is auto-linked in the preview
Hyperlinks	in the preview are LIVE: hover shows a hand cursor; click opens in your default browser.
Code block	triple-backtick + language label, e.g. '''python ... ''' Supported: python (py), javascript (js/ts/ jsx/tsx/node), bash (sh/shell/zsh/bat/cmd), json, sql (psql/mysql/postgres), c, cpp (c++/cxx/cc/hpp), csharp (cs/c#), autoit (au3), zig.

RUN-COMMAND BEHAVIOR

Dialog runs (Ctrl+R, Alt+Enter) auto-execute on open and stream output into the dialog's output pane. Click 'Insert output into note' to paste it back as a fenced code block.

Inline runs (Ctrl+Alt+R, F5/Ctrl+Enter) drop a fenced

[running...] placeholder right after your selection or current line, then replace it with the captured output and exit code when the command finishes. Multi-line selections execute one line at a time so each command's output is reported separately.

DISPLAY OPTIONS (Options menu)

- Fonts Pick any installed font, choose style (normal/bold/italic) and size (6-72). Applies to every tab's editor.
- Tab width Number of spaces per Tab character (1-16). Pixel width is recomputed when the font changes.
- Line numbers Gutter on the left of every editor. Wrapped lines correctly show one number per logical line.
- Dark theme Toggles light <-> dark colors across all tabs (editor, preview, line numbers, sidebar) and the syntax-highlight palette.
- Always on top Keeps the window above other windows.
- PDF page numbers When on, Export as styled PDF adds a centered 'page X of Y' footer. When off, the PDF has no header or footer.
- Confirm before running commands When on, every command-execution path (Tools menu, Ctrl+R / Ctrl+Enter / F5 / Alt+Enter / Ctrl+Alt+R, plus the Run dialog's Run button) prompts you with the exact command(s) before running. Default off.
- Keep selection through navigation When on, mouse clicks and arrow / Home / End / PageUp / PageDown move the cursor without clearing the active selection. Typing a printable char still REPLACES the selection (default Tk behavior); Shift + arrows / drag still extend it; XD or YD in the command line cancels it explicitly. Default off; persisted across launches.
- Hyperlink safety: only http(s):// and mailto: links in the preview are clickable. file://, custom handlers, and other schemes are silently ignored.
- Editor colors Pick a custom foreground or background color via a color picker; persists across launches and overrides the active theme

across the editor, preview, sidebar, and the line-number gutter. Secondary tints (code-block background, syntax-highlight colors, selection highlight) stay theme-controlled so the picture stays balanced. 'Reset editor colors to theme' clears the active overrides (theme shows through), while 'Reset to custom colors' re-applies the last saved custom colors. The saved colors persist even after you reset to theme.

EXPORT

File -> Export as PDF...

Plain-text PDF written by a hand-rolled minimal PDF writer. No external programs needed. Body is in monospace Courier with automatic page breaks.

File -> Export as styled PDF...

Renders the note to HTML (always black on white, independent of the in-app light/dark theme), then runs Microsoft Edge or Chrome in headless mode to produce a fully styled PDF with real fonts, headings, lists, and syntax-highlighted code blocks. Falls back gracefully if no browser is found.

NOTE FILES

One .md file per note with a small YAML-style header:

title: My note

tags: work, todo

created: 2026-04-25T10:23:00

(note body)

Files are plain text; you can edit them in any editor or sync the folder via OneDrive/Dropbox/Git.

AUTOSAVE & DRAFTS

Saves fire 800 ms after you stop typing. A draft tab becomes a real file the moment you put anything in title, tags, or body -- no need to choose a name first. Renaming a note in the Title field renames the file on disk on the next autosave.

PERSISTENCE

Saved to the active config file across launches:

- Notes folder
- Window position and size

- Font family / style / size
- Tab width
- Last-used Insert Date & Time format
- Custom editor foreground / background colors
- PDF page-numbers on/off
- Confirm-before-running-commands on/off
- Keep-selection-through-navigation on/off
- Recent-files list (up to 10)
- Line-numbers visibility
- Dark-theme on/off
- Always-on-top on/off

COMMAND-LINE OPTIONS

```
python quicknotes.py
    Default: portable mode -- config and notes folder
    live next to the script.
python quicknotes.py --notes-dir DIR
    Use DIR for this run only; NOT saved to the config.
    Useful for one-off sessions.
python quicknotes.py --user-profile
    Old behavior: config in ~/.<brand>.json, notes in
    ~/Documents/<Brand>/. The window title shows
    '(user profile)' in this mode.
python quicknotes.py --portable
    No-op alias for the default. Kept for backward
    compatibility with older shortcuts.
python quicknotes.py --help
    Print these options in the terminal and exit.
```

REBRANDING

Window title, default Documents subfolder, and the config filename all derive from this script's name. Rename quicknotes.py to mynotes.py and you get window title 'Mynotes', notes in ~/Documents/Mynotes, config at ~/.mynotes.json.