Q Search

72

💬 Daily ways GNU Hyperbole helps me stay in flow ...   ✕ CLOSE

Posted by 7 days ago 🏵️

Daily ways GNU Hyperbole helps me stay in flow and reduces cognitive load

1. Just FYI, I am the author of GNU Hyperbole. See www.gnu.org/s/hyperbole for what it is). People often ask why they should use GNU Hyperbole . It is a large package because information navigation and management is a big thing. But Hyperbole hides this complexity internally, giving you easy key and button presses or keyboard menus to handle all sorts of tasks.

2. Although you can find elements of some of these things in many different packages, nothing brings them together and makes them effortless to accomplish via a single package like Hyperbole.

3. Herein I list things I do quite frequently with Hyperbole just to give you a taste of the possibilities. I hope some will spark an interest to try it out and expand your Emacs universe. I often:

4. Display URLs, pathnames with environment variables in them, or Tramp remote paths with {M-RET}.

5. Navigate table of contents and textual menus in many modes by pressing {M-RET}.

6. Follow cross-references in Markdown, Info, Texinfo and Org mode with {M-RET}.

7. Jump to Emacs bookmarks or edit Org agenda items with {M-RET}.

8. Jump to my personal global buttons where I store quick abbreviations which display frequently used directories or invoke arbitrary code actions I have defined.

9. Compose mail by pressing {M-RET} on an email address.

0. When using Windows Subsystem for Linux, I jump to Windows formatted paths, even those on Windows shares by pressing {M-RET}. No need to translate path formats any more regardless of which operating system you run Emacs on.

1. Make quit-window restore my window configuration to exactly the way it was before I invoked Emacs help whenever I press {q}.

2. Embed brace-delimited key series (multiple key sequences strung together) in my documents that are executed whenever I press {M-RET} on them.

3. Choose a buffer line to put at the top of a window via {M-RET} at the end of a line or at the window bottom with {C-u M-RET}. {C-l} has adopted something similar in recent years but you have to cycle through positions with that rather than a single press.

4. Jump to source code lines from grep -n outputs or stack traces in shell mode with {M-RET}.

5. Jump from an identifier use in code to its source definition with {M-RET}.

6. Select groupings delimited by parens, brackets or braces in any mode with {M-RET} on the opening or closing delimiter. This also works with HTML and XML start and end tags, for quick selection.

7. {C-c RET} lets me select bigger and bigger syntactical units in many modes by repeatedly pressing it. I can go from a word, to a symbol, to a line, to a paragraph and beyond very easily.

8. Together with Ace Window and Dired, quickly place buffers in windows wherever I like and then save window configurations for rapid recall later. I can name these configurations or use a window configuration ring similar to the kill ring, all accessible via Hyperbole quick key menu.

9. Create rapid layouts of window grids with {C-c @}, letting me see a bunch of existing buffers at once, either ones I've marked in Dired, recently used ones or those of a specific mode.

0. Rapidly grow, shrink and rearrange windows/buffers/frames with HyControl, Hyperbole's scriptable window and frame manager. I can move frames on screen by dragging with my middle mouse button depressed on each frame's bottommost modeline. I can clone the exact size and contents of a window to a new frame by just dragging my middle mouse button from the window to outside

new frame by just dragging my middle mouse button from the window to outside of any frame.

1. Create hierarchical, auto-numbered outlines in the Koutliner or use it to brainstorm any sort of list I need. Now I can embed Org tables in Koutlines too (in the latest git branch of Hyperbole) and toggle this minor mode on and off by pressing {M-RET} on one of the table's | vertical dividers. I export Koutlines to HTML, or Emacs/Org outline files when needed.

2. Manage my contacts with HyRolo, allowing rapid full-text and logical search across any number of contact files, each of which is an Emacs outline of hierarchical contact records, e.g. people within an organization. {C-x 4 r} from anywhere looks up anything I need quickly; a prefix argument limits the number of matching results returned to a maximum. {q} then makes it disappear for minimal workflow interruption. {C-h h r a} adds a new entry.

3. Jump to the original message and discussion in GNUS for any Emacs or Hyperbole bug wherever I see bug#1234 by pressing {M-RET} on it.

4. Perform highly targeted web searches with the Hyperbole Find/Web menu.

5. Use my own custom Hyperbole Helm menu that exposes many of the useful Helm commands that are hidden or hard-to-find by the default configuration when you load the package.

6. Use my own custom implicit button type that allows me to load and use a standard release of Hyperbole but if I try to edit one of the files from here, it automatically instead edits the equivalent one in my local Git repo of Hyperbole, preventing editing of the wrong file.

More posts from the emacs community

Continue browsing in r/emacs →