

Emacs

The Emacs icon, in Applications or on the Dock, is a set of wildebeast horns turned on its side to look like a script E. Emacs is a **text editor**, not a word processor. It is primarily intended to create text files that are read by other programs on the computer, either as data files to be manipulated or as program source code files to be interpreted or compiled. Emacs does not embed any hidden characters in your file to do formatting, except for line ending characters and blank spaces.

To start the Emacs editor, click the icon on the dock, and when the application is open, use the File menu as usual to start a new file or open an existing one. You will also see the opens for saving a copy of an existing file under a new name and inserting another existing file in this one – the usual options for a File menu.

Moving through a file

Generally, arrows, pageup and pagedown, mouse clicks and scrolls, and the items under the Edit do exactly what you are used to from other word processor and editor programs. Among what Emacs does “special” are a lot of sequences obtained by holding down the `control` key with a letter or arrow.

- `Home` Move to the beginning of a line.
- `End` Move to the end of a line.
- `control-Home` Move to the beginning of a file.
- `control-End` Move to the end of a file.
- `control-→`, `control-←` Move forward or backward by *words* instead of characters.
- `control-↑`, `control-↓` Move up or down by *paragraphs* instead of lines (paragraphs are assumed to be separated by blank lines).
- `control-L` Refresh screen, center on cursor.
- `Delete` Left Delete. Delete character to the left (backspace and delete).
- `control-D` Right Delete. Delete character to the right of the cursor. (When using a “block” cursor, this will be the character that appears to be *under* the cursor.)

Moving and Copying

The usual mouse drag to mark a block and use of the Edit menu will work in Emacs. However, it also has a special set of “kill” or “wipe” and “yank” commands that are quite handy to learn for line-based editing, which applies to data files and programs.

- `control-K` Delete (**kill**) everything on line to right of cursor.
- `control-Y` Recover (**yank**) everything in most recently killed block. Undelete.
- `control-@` Mark beginning or end of block to kill. (“Mark set” should appear at bottom.)
Alternately: drag the mouse with the left mouse button pressed.
- `control-W` Delete everything (**wipe**) between current cursor position and where the mark was set.

Note that “killing” a block of text with a series of `control-K` or a pair of `control-@` – `control-W` commands, and then “yanking” or undeleting that text in one or more other locations is an equivalent to the usual Cut, Copy, and Paste commands from the Edit menu.

Miscellaneous

- `control-T` Transpose letters (switch positions of the letters on each side of the cursor).
- `control-G` Quit whatever Emacs function you may be in the middle of.
- `control-S` Incremental search forward. `esc` to terminate search.)
- `control-R` Incremental search backward.

Buffers and files

What you are editing in an Emacs window is not the file itself, but a copy of that file in memory, known as a **buffer**. The changes you make in a buffer do not affect the file until you explicitly **save** the buffer (`control-X control-S` or **Save** in the **File** menu).

While you are editing *filename* in a buffer, Emacs will occasionally make an “autosave” copy called *#filename#*. When you finish with a file and properly exit, Emacs will delete the autosave file, but if you abort the editing session these will not be deleted. When you save a file and exit, the version of *filename* that existed when you started the session will be saved and renamed to *filename~*.

Split windows

Emacs has the ability to split a window into two buffers, either horizontally or vertically, which can be useful for comparing two versions of a program or looking at a data file while trying to figure out how to successfully `split` its lines into useful items. These commands all involve pressing `control-X` followed by a digit.

- `control-X 2` Split the window horizontally.
- `control-X 3` Split the window vertically.
- `control-X 1` Remove splits (also useful if a large error message appears and splits the screen).

After removing splits, you may find yourself in the wrong buffer – Emacs can keep multiple files open in different buffers that do not all have to be visible. Use the **Buffers** menu to move between buffers.

Python mode, major modes

If you start Emacs with a file whose names ends in `.py` then Emacs assumes you are editing modern Python source code.

- `tab` This key works in all modes, but in Python mode it is set to indent an entire line to the same level as the previous line in order to help with Python’s block structures. Use `Shift Tab` to move left to the previous indent level to end a block.
- `)] }` After typing a right parenthesis, square bracket, or curly bracket, the cursor will temporarily, for a about a second, jump to the matching left bracket. This is a very handy code checking tool, as Python requires brackets to appear as matched pairs in most circumstances.

When in Python mode, the editor should automatically perform syntax coloring, in which different types of Python syntax (keywords, variable names, character strings, comments) appear in different colors. If this does not work for you, it may need to be turned on under the **Options** menu.

Emacs has editing modes for most programming languages, and it automatically enters them if it detects the standard filetype for that language (i.e., `.py` for Python, `.sh` for shell scripts, and so on). These modes are called “major modes” and will be indicated in the status line near the bottom of the buffer.