**What is Emacs?**

Emacs is a text editor designed for operating systems and available Windows, and more. Users love Emacs because it features efficient commands for common but complex actions and for the plugins and configuration hacks that have developed around it for nearly 40 years. Because it's an old editor that was developed well before modern computer conventions and terminology existed (for instance, you "visit" instead of "opening" a file, and you "write" instead of "save," and so on), Emacs is often viewed as complex and even mysterious. As time has shown, however, once you learn the basics, you have a powerful, efficient, and extremely hackable editor for life. Learning Emacs can be difficult because it uses keyboard combinations fundamentally different from the way modern computers do. There's a method to this apparent madness, though, because Emacs is built to be flexible in how it's used. This includes devices without traditional keyboards and over networks that may not transmit modifier keys (such as **Ctrl** and **Alt**) correctly.By default, Emacs keybindings revolve mostly around the **Ctrl** and **Alt** keys. In the documentation, the **Ctrl** key is represented as **C** and the **Alt** key as **M** (because before the **Alt** key was called "Alt," it was called "Meta"). When you need to press the **Ctrl** key and another key together, the action is written as, for example, **C-x** (meaning **Ctrl+X**) or **C-c** (**Ctrl+C**). The same goes for Alt/Meta: if you're meant to press **Alt-X**, then the notation is **M-x**.

* **Who is Richard Stallman**

(nickname **RMS**) (born March 16, 1953) is both an acclaimed software freedom activist and software developer. In the 1980s, he founded the GNU project to create a free unix-like operating system, and has been the project's lead architect and organizer. Also in the 1980s, he founded the free software movement, and the Free Software Foundation. He co-founded the League for Programming Freedom

**How to open and save file**

A SAVE file contains a game save data created by The Sims 4, an open-ended life simulation game with virtual people called "Sims." It stores game progress, which includes character, building, and world information

* **What is a buffer and how to switch from one to the other**

.

The symbol switch-to-buffer is the first element of the list, so the Lisp interpreter will treat it as a function and carry out the instructions that are attached to it. But before doing that, the interpreter will note that other-buffer is inside parentheses and work on that symbol first. other-buffer is the first (and in this case, the only) element of this list, so the Lisp interpreter calls or runs the function. It returns another buffer. Next, the interpreter runs switch-to-buffer, passing to it, as an argument, the other buffer, which is what Emacs will switch to. If you are reading this in Info, try this now. Evaluate the expression

* **How to use the mark and the point to set the region**
* the procedure for doing this is the same in all emacs commands that work on regions of text. The “current region” is the set of characters from the “mark” to the current cursor position. The “mark” is an imaginary position marker established by the set-mark-command. The keystrokes for that command are either C-[spc] (hold the control key and type a space) or C-@ (hold the control and shift keys and type
* So to set up a region to operate on, you move the cursor to one end of the region, give the set-mark-command, then move the cursor to the other end of the region. Everything between the mark and the cursor constitutes the current region, and can be operated on by any region-based command. Some region commands of note are:
* **How to search forward and backward**

**How to invoke commands by name**

* The Emacs commands that are used often or that must be quick to type are bound to keys--short sequences of characters--for convenient use. Other Emacs commands that do not need to be brief are not bound to keys; to run them, you must refer to them by name.
* A command name is, by convention, made up of one or more words, separated by hyphens; for example, auto-fill-mode or manual-entry. The use of English words makes the command name easier to remember than a key made up of obscure characters, even though it is more characters to type.
* **How to undo**
* You can't undo some actions, such as clicking commands on the **File** tab or saving a file. If you can't undo an action, the **Undo** command changes to **Can't Undo**.
* To undo several actions at the same time, click the arrow next to **Undo**  select the actions in the list that you want to undo, and then click the list.

**How to cancel half-entered commands**

* As I spend more and more time in terminal sessions, it feels like I'm continually finding new commands that make my daily tasks more efficient. The GNU  command is one that really changed my work day.
* The GNU history command keeps a list of all the other commands that have been run from that terminal session, then allows you to replay or reuse those commands instead of retyping them. If you are an experienced terminal user, you know about the power of history, but for us dabblers or new sysadmin folks, history is an immediate productivity gain.
* First of all, the history command isn't actually a command. You can see this for yourself by looking for the command on your sy
* **How to quit Emacs**

In Emacs, key bindings are called keybindings because we like things to be plain. Some people in other possible worlds called them shortcuts but that is a very narrow view of what a keybinding is, since Emacs keybindings have a personnality of their own and can be a bit long sometimes.

The first time you encounter Control-x, you think: "Aha… so the x stands for eXiting!". Wrong.

**How to cut and paste lines and regions**

To cut or copy a portion of your document, you first have to select the text that is to be cut. There are a variety of ways to do this. The keyboard way and the mouse way.

* The keyboard way starts by setting the mark on the first character to be cut. Position the cursor on that character and type **C-<spacebar>**. That is hold down the Control key and hit the spacebar. You should see the confirmation message``Mark set'' in the command minibuffer. Then move the cursor to a position immediately after the last character to be cut. The region is implicitly ``selected''. The command **C-w** cuts the text and puts it in the invisible ``kill ring'', i.e. paste buffer. The command **M-w** copies the text to the kill ring, but doesn't delete it. To restore the text, use the command **C-y**. The text is copied to the position immediately before the cursor. If you made a mistake, you can also use **C-x u** to undo the cut.