

An Emacs tutorial generated by its authors is available online by typing `[control]-H t` from within an editing session. It is not good. This tutorial is written for an audience that is assumed to have used word processors before but may never have used a text editor. It should be used in conjunction with the Emacs reference handout.

- `[meta]-<`, `[meta]->` and `[control]-@` require that you use `[shift]` while pressing the second key. None of the `[control]` or `[meta]` sequences using alphabet letters require `[shift]`.
- `[control]-G` is your “quit” command from most Emacs functions—it is the first thing you should try when the behavior of the editor seems quite different from what is described here.
- **If your emacs sessions begins with the window split horizontally into two buffers, press `[control]-X 1` to request one visible buffer.**

## Emacs Tutorial

### Creating or Opening a File.

1. At the Unix prompt (`%` or `$`), type the Unix command

```
xemacs animals.text
```

or

```
emacs animals.text
```

and then press the `[return]` key. If you do not have X11 properly enabled, the Emacs window will take over your terminal window (which is OK for this tutorial but something you will want to get fixed soon). With properly configured X11, Emacs opens a new window. Your Emacs window should look something like this:

```
-----XEmacs:  animals.text (Text Fill)--All-----
(New file)
```

The line just above the last line will actually be reverse video or shaded on the Emacs window. That is the status line: it tells you that you are running XEmacs (in this example), that the filename you are editing is `animals.text`, that you are in Text mode with automatic paragraph filling, and that all of the file (which is currently empty) is shown on the current window. Actual contents of the file are above the status line. The very last line that currently says `(New file)` is where you will occasionally type editing commands—it is called the **minibuffer**.

2. Your cursor will initially be at the top of the file. Everything you type will be entered as text into your file. Type the line

```
This small file is full of animals.
```

Don't worry about mistakes. You will learn how to correct them later. If you wish to correct mistakes just after you type them, use the `[delete]` key, which backs up over letters and erases them.

3. Press the `return` key to move to the next line and continue adding text to your file, pressing the `return` key after each line:

```
The small brown dog ate a small meal.
The small horse ran under the elephant without any problem.
The small grey fox jumped over the large white aardvark.
The small lion was intimidated by the giant mouse.
```

4. To end the editing session, type `control-X control-C`. The line below the status line will prompt you with

```
Save file /filepath/animals.text? (list of options)
```

Press `Y` to confirm that the file should be saved. The Emacs window should disappear (or revert to a terminal window if you did not have X11 working). You can confirm that you now have a file called `animals.text` by using the Unix directory command `ls`.

5. To open and edit the file you have just created, type the command

```
xemacs animals.text
```

(If this is the last file you worked on, you can re-edit it with the Unix shortcut `!x` that means “re-run the last command that started with `x`.”)

If you have followed the directions so far, the XEmacs window will look like this:

```
This small file is full of animals.
The small brown dog ate a small meal.
The small horse ran under the elephant without any problem.
The small grey fox jumped over the large white aardvark.
The small lion was intimidated by the giant mouse.

-----XEmacs:  animals.text (Text Fill)--All-----
```

Note that your cursor begins on the first character of the first line of the file. You can use the arrow keys to move around in the text area of the file. The arrow keys will not let you move beyond the area of the window that actually contains your file. That is, you will not be able to move to the status line using arrow keys, nor can you move past the ends of your typed lines.

6. Use the arrow keys or `control-n` to move to the last line of the file. Then use `control-E` to move to the end of that line. Press `return` to start a new line.
7. Add some text by inserting a file. Press `control-x i` (or select the Insert File ... option on the File menu). A prompt `Insert file:` will appear in the minibuffer, along with the complete path to your current directory. Press `delete` until your path is gone from the minibuffer and type in the following file reference: `~hanson/Geog605/xemacs.1`

Your editing window should look like this (the \*\* on the status line indicates the the file has changed during this editing session). The last six lines in the main buffer came from the file you just inserted.

```
This small file is full of animals.
The small brown dog ate a small meal.
The small horse ran under the elephant without any problem.
The small grey fox jumped over the large white aardvark.
The small lion was intimidated by the giant mouse.
The zippy zebra drank two sarsaparillas.
The small beetle insulted the zebra.
The zebra left in a huff and a taxi.
The small beetle asked the barman for a grasshopper.
The beetle drank too many grasshoppers for such a small bug.
This is the last line of the file and I'm not lion.

---*-XEmacs:  animals.text (Text Fill)--All-----
```

Familiarize yourself with the “Moving the Cursor” section of the reference manual. Using the arrow keys should be familiar, but check out the various `control` and `meta` functions. Use `control-A` to go to the beginning of a line, and `control-E` to go to the end of a line. Try `meta-→` and `meta-←` or `control-→` and `control-←` and see that the cursor moves right and left by words instead of by characters. You can use `meta-<` to go to the beginning of the file, and `meta->` to go to the end of the file. (You don't have paragraphs yet, but `meta-↑` and `meta-↓` will go up and down by paragraphs, meaning they will jump to the next blank line.)

## Erasing, Moving, and Copying lines of text.

8. Move to the beginning of the sixth line (The zippy zebra...) and press `control-K` four times. Note what happens each time you press `control-K`: first the text on a line disappears, then the line itself disappears. Pressing `control-K` four times thus completely “kills” two lines.
9. Press `control-Y` to “yank” the lines back, and see that they have returned.
10. Move the beginning of the file and press `control-Y` again. You can see that you have copied the two lines at the top of the file. You can repeat `control-Y` as many times as you want.

The sequence you have just completed copied a two-line block from the middle of the file to the top. If you had not performed the first `control-Y` before moving the cursor to the top, you would have just moved the text. If you just wish to erase the text, use `control-K` without subsequently using `control-Y`.

11. Another way to kill a large block of text and yank it elsewhere is to mark the beginning and end of a block. Move to the top of the file where you have just copied text. “Mark” the beginning of the file using `control-@`. Mark set will appear in the mini-buffer. Move the cursor down to the beginning of the This small file ... line. The marked area should be shaded. Press `control-W` and notice that everything between your “mark” and the current cursor position has been “wiped” out. Move to the bottom of the file and type `control-Y` to yank the block of text back into a new position.

The preceding “kill” and “yank” method of moving or copying lines will typically be faster than copy-and-paste or cut-and-paste methods, once you are used to it. However, copy, cut, and paste with the mouse are available, as in most software, from the Edit menu.

## Correcting errors and making changes in your file.

12. To add characters anywhere in your file, simply move to the point at which you wish to insert text and start typing. Emacs is in “insert” mode, wherein text underneath the cursor is constantly shifted to the right as you type in new text on top of it.
13. To add lines after a line, go to the end of the line, `[control]-E`, and press `[return]`. To add lines before a line, go the beginning of the line (`[control]-A`) and press `[control]-O` (“oh”, not “zero”). Pressing `[return]` anywhere within a line will break the line into two lines.
14. Move to the top line of the file, and move to the end of the line. Press `[return]` to start a new second line. Insert the following lines by inserting the file `~/hanson/Geog605/xemacs.2`

```
The Bery green tomxto fell on the small beatle.
the grey oul chased the the small rabbbbit.
the beelte hda enuff excitement for oneday.
```

These should now be the second through fourth lines of the file, as in:

```
This small file is full of animals.
The Bery green tomxto fell on the small beatle.
the grey oul chased the the small rabbbbit.
the beelte hda enuff excitement for oneday.
The small brown dog ate a small meal.
The small horse ran under the elephant without any problem.
The small grey fox jumped over the large white aardvark.
The small lion was intimidated by the giant mouse.
The zippy zebra drank two sarsaparillas.
The small beetle insulted the zebra.
The zebra left in a huff and a taxi.
The small beetle asked the barman for a grasshopper.
The beetle drank too many grasshoppers for such a small bug.
This is the last line of the file and I'm not lion.
The zippy zebra drank two sarsaparillas.
The small beetle insulted the zebra.

---*-XEmacs: animals.text (Text Fill)--All-----
```

15. There are two ways to delete characters:
  - Move before a character to be deleted, and press `[control]-D`. (If using a block cursor, it will cover the character about to be deleted this way.)
  - Move after a character to be deleted, and press `[delete]`. Use these to correct the misspellings `Bery`, `tomxto` and `beatle` in the second line. When you delete a character, the remaining text shifts left on the line to fill the space.
16. On the third and fourth lines of the file, initial words need to be capitalized. Move to the beginning of each word and press `[meta]-C` which capitalizes the word. You should also experiment with `[meta]-U` which changes the entire word to uppercase instead of just the first letter, and `[meta]-L` which changes the entire word to lowercase.

17. On the third line you have a double occurrence of `the`. To delete an entire word you may use `[meta]-D` to delete an entire word to the right of the cursor or `[meta]-delete` to delete an entire word to the left of the cursor. (`[meta]` works on these keys just as with arrows—operating on words instead of on characters.)
18. The misspellings `beelte` and `hda` contain “transpositions,” a very common typing error. Place the cursor between the `l` and `t` in `beelte` and press `[control]-T`. The two letters will switch places. (You can also place the cursor between two words and switch them using `[meta]-T`. Experiment with this.)
19. Finish correcting the spelling errors in lines two through four using the techniques you have learned.

## Replacing Text Strings

20. Move to the top of the file and press `[meta]-X`. That is your signal to Emacs that you wish to type a more complicated command that is not represented by a simple `[control]` or `[meta]` sequence. The bottom line of the Emacs window is called the minibuffer, and it is used to type commands. Text you type in the minibuffer does not become part of your file, but is part of your interaction with Emacs. Your cursor will move down to the minibuffer, where you can type the command `replace string` and press `[return]`.
21. After the `Replace string:` prompt appears in the mini-buffer, type `beetle` and press `[return]`.

22. After the

`Replace string beetle with:`

prompt in the mini-buffer, type `cricket` and press `[return]`. You should now see the word `Done` in the mini-buffer, and every occurrence of `beetle` in the file has been changed into `cricket`.

23. Move to the top of the file and press `[meta]-X`, type `query replace` into the mini-buffer and press `[return]`. After the

`Query replace:`

appears in the minibuffer, type `cricket` and press `[return]`. When the

`Query replace cricket with:`

prompt appears, type `beetle` and press `[return]`. The cursor in text will move to the first occurrence of `cricket` and the mini-buffer will say

`Query replacing cricket with beetle:`

24. Press `Y`. The first occurrence of `cricket` will be replaced with `beetle` and the cursor will move to the second occurrence of `cricket`.
25. Press `N`. The second occurrence of `cricket` will be left unchanged and the cursor will move to the third occurrence of `cricket`.
26. You can move through every occurrence of `cricket` pressing `Y` or `N` as needed. You may also press `[control]-G` to abort the process, or `!` (exclamation point) to tell Emacs to change all remaining occurrences of `cricket` into `beetle` without further checking. Query Replace is also available as the Replace command under the Edit menu.

## Incremental Search

27. The “incremental search” feature looks for text strings. Rather than waiting for you to type a complete text string, it starts looking as soon as you type a character and refines the search as you continue typing. To begin the process, go to the top of the file using `[meta]-<`, and press `[control]-S`. `I-search:` should appear in the mini-buffer.

Some X11 windows may stop responding to input at this point, because by default in some old terminal emulators these programs `[control]-S` means “stop transmission.” You may usually press `[control]-Q` to resume transmission, but you will not be able to use forward incremental search without changing these key definitions in your terminal emulation program.

You can search for the word `grasshopper` in your text by typing it one letter at a time. Type `g` and note that the cursor has moved to the second line, after the word `green`, which contains the first occurrence of `g` in the file. The `g` appears in the mini-buffer. Type `r` and see that the cursor stays at `green`. Type `a` and the cursor will move to `grasshopper` on line 12—the first occurrence of `gra` in the file.

28. Suppose `grasshoppers` in the next line is what you really wanted to find. Press `[control]-S` again and the cursor will move to the next occurrence of `gra`. Press `[esc]` to terminate the incremental search. The prompt in the mini-buffer will disappear, and your cursor will be left at the word `grasshoppers` where you can continue editing.
29. `[control]-R` allows you to search backwards. Try searching backwards from your current position for the word `fox`, noting where the cursor stops after you type each letter. Both forward and backward incremental searches are also available under the Edit menu.

## Termination and files.

30. Now you should leave Emacs and save the file using `[control]-X [control]-C`. If you type `ls` at the Unix prompt again, your files will include both `animals.text` and `animals.text~`. The version with `~` appended is a backup file—it contains the file you began the last editing session with (remember that you saved the file and restarted back on page 2). Emacs will automatically replace backup files each time you re-edit a file. If for any reason you abort an Emacs session or exit without saving a file, you might find files with `#` surrounding their names, such as `#animals.text#`. These are “autosave” files that Emacs generates occasionally during an editing session to protect you from system crashes and catastrophic errors. Emacs removes these files automatically if you save a file and exit normally at the end of an editing session.
31. You will have no further use for `animals.text` or its backup files in this course, so you might use them to experiment with some of the file handling commands discussed in your Unix handout. You should particularly become familiar with `ls`, `cp`, `mv`, `rm`, and `cat`.