Prefix commands:

C-x Mainly used for file, buffer and window manipulation
C-c Usually particular to a mode or context
C-h Mainly used for help commands

Some definitions

File: The real file in the file system.
Buffer: Internal chunk of data. Usually points (visits) a file.
Window: View of the buffer. Screen can be split into several windows.

Manipulating files

C-x C-f `find-file`: Opens a file and change current window to it.
C-x C-s `save-buffer`: Writes current buffer to disk.
C-x k `kill-buffer`: Prompts for a buffer and kills it. Writing of unsaved changes is optional.

Manipulating Buffers

C-x b `switch-to-buffer`: Prompts and switch to a buffer in the same window(It can be a new buffer).
C-x C-b `list-buffers`: Lists buffers with some properties like name, modified date, size, etc.
C-x k `kill-buffer`: Prompts for a buffer and kills it. Unsaved changes writing are optional.
C-x C-q `vc-toggle-read-only` Toggles between read-only and read-write mode. If the file is version controlled it is checked out for you.

Manipulating Windows

C-v `scroll-down`: Scroll down one screen.
M-v `scroll-up`: Scrolls up one screen (opposite of `scroll-up`).
C-x o `other-window`: Changes to other window.
C-x 0 `delete-window`: Delete current window.

C-x 1 `delete-other-windows`: Delete other windows.
C-x 2 `split-window-vertically`: Splits current window in two vertically.
C-x 3 `split-window-horizontally`: Splits current window in two horizontally.
C-M-v `scroll-other-window`: scroll other window (the one that `other-window` will change to).

More basics

To enter : emacs
To enter with a buffer: emacs file.ext
To exit : C-x C-c `save-buffer-kill-emacs`

The screen

The mode line

Located at the bottom, in reverse video.

Shows info about

The state of the buffer: *modified*, ~~unmodified~~ or %read-only%.
The name of the buffer, the visible amount or its content offset in percentage.

The minibuffer

Used by emacs to show messages, also called echo-line.

Sometimes it shows random messages of what Emacs is doing, if you encounter in the situation that it prompts you something you don't know you can always press:

C-g `keyboard-quit` to leave that prompt.

Help

C-h : Default prefix for help. Type C-h C-h for a list of commands.
C-h t : Runs the tutorial, good for beginners.
C-h i : Info. Enters the built in Info hypertext documentation reader.

Infinite Undo

C-_ or C-x u : Undo
TODO : Redo

Arguments to commands

Type C-u plus the argument before typing the actual command. Since they go before the actual command they are called: *Prefix Arguments*.
- is for negative arguments like -1
It is possible to cancel current input with C-g

Inserting control sequences

C-q `quoted-insert` inserts control sequences like ESC, C, M into the text rather than recognize them as commands. Three digits octal ASCII code is also possible.

Moving around

moving through text highly depends on the mode emacs is in (C mode, text mode, etc).
Note: The *f* for forward and *b* for backward mnemonic will reoccur. Also *a* for beginning and *e* for end of things.

C-f `forward-char`. Next character (to the right).
C-b `backward-char`. Previous character (to the left).
M-f `forward-word`. Next word.
M-b `backward-word`. Previous word.

C-n `next-line`. Next line (go down).
C-p `previous-line`. Previous line (go up).

C-a `beginning-of-line`. Beginning of current line.
C-e `end-of-line`. End of current line.

Sentences

M-a `backward-sentence`. Beginning of current sentence.
M-e `forward-sentence`. End of current sentenceMoves to the end of the current sentence.

Paragraphs

M-{ backward-paragraph. Beginning of current paragraph.
M-} forward-paragraph. End of current paragraph.

Pages

C-x [backward-page. Beginning of current page.
C-x] forward-page. End of current paragraph.

Pages are separated by formfeed characters (C-l) in most modes.

Buffers

M-< l beginning-of-bufer. Start of the buffer.
M-> end-of-buffer. End of the buffer.

Deleting Killing and Yanking

First, some definitions...

Delete: is simply the deletion we know.
Kill: is like cutting. Deleted text is saved in the *kill ring* for future usage. When killing the *kill ring* acts like a FIFO stack. Its maximum depth is defined by kill-ring-max variable (defaults to 30).
Yank: is pulling some text from the *kill ring*. When yanking things back the *kill ring* acts like a ring (you can yanking around).

C-d delete-char deletes char to the right.
DEL delete-backward-char deletes char to the left.
M-d kill-word kills word to the right.
M-DEL backward-kill-word kills word to the left.

C-k kill-line. Kills to the end of the current line, not including the newline.
C-u 0 C-k kill-line. Kills to the beginning of the current line, not including the newline.
C-u -1 C-k Kills to the beginning of the line including the newline before the line as well.

Remember that C-u is for giving prefix arguments, in this case 0 and/or 1, to the command which in this case is C-k `kill line`.

Sentences

M-k `kill-sentence`. Kills to the end of current sentence, including any newline within the sentence.
C-u -1 M-k `kill-sentence`. Kills to the beginning of the current sentence, including any newlines within the sentence.

Yanking

C-y `yank` Yanks what is in the kill-ring. Remember the kill-ring acts like a ring when yanking so we can move to the next spot with M-y and replace the just yanked text. Subsequents M-y will work until any other command is executed (insertion, anything) breaking the cycle.

Copying and Moving Text

There is no extra commands to learn, copying can be done by killing and immediately yanking. Moving is equivalent to killing, moving the cursor and yanking.

Search and Replace

C-s `isearch-forwards` searches the text forwards incrementally.
C-r `isearch-backward` searches the text backwards incrementally.

To go to the next result press C-s and C-r to go back. When searching backwards it is the opposite. To stop the search press ret.

TODO: word search

TODO: regex search

Replacement

M-% `query-replace` Prompts for a couple of strings (target and replacement) for the replacement and in each iteration prompts for an action. Some options are:

- `SPC` : Perform this replacement.
- `DEL` : Don't perform this replacement.
- `RET` : Terminate `query-replace` without performing this replacement.
- `ESC` : Same as `RET`.
- `.` : Perform this replacement but then terminate the `query-replace`.
- `!` : Perform this replacement and all the rest in the buffer without querying (i.e. unconditionally).

TODO: Regex search and replace

`replace-string` Replaces unconditionally all occurrences.
`replace-regexp` Replaces unconditionally all occurrences.
`query-replace-regexp` Prompts an action for each occurrence.