Emacs Commands List

C = Control
M = Meta = Alt

Basics
C-x C-f "find" file i.e. open/create a file in buffer
C-x C-s save the file
C-x C-w write the text to an alternate name
C-x C-v find alternate file
C-x i insert file at cursor position
C-x b create/switch buffers
C-x C-b show buffer list
C-x k kill buffer
C-z suspend emacs
C-X C-c close down emacs

Basic movement
C-f forward char
C-b backward char
C-p previous line
C-n next line
M-f forward one word
M-b backward one word
C-a beginning of line
C-e end of line
C-v one page up
M-v scroll down one page
M-> beginning of text
M-< end of text

Editing
M-n repeat the following command n times
C-u repeat the following command 4 times
C-u n repeat n times
C-d delete a char
M-d delete word
M-Delete delete word backwards
C-k kill line

C-Space Set beginning mark (for region marking for example)
C-R "kill" (delete) the marked region
M-W copy the marked region
C-y "yank" (paste) the copied/killed region/line
M-y yank earlier text (cycle through kill buffer)
C-x C-x exchange cursor and mark

C-t transpose two chars
M-t  transpose two words
C-x  C-t  transpose lines
M-u  make letters uppercase in word from cursor position to end
M-c  simply make first letter in word uppercase
M-l  opposite to M-u

Important
C-q  quit the running/entered command
C-x  u  undo previous action
M-x  revert-buffer RETURN (insert like this) undo all changes since last save
M-x  recover-file RETURN Recover text from an autosave-file
M-x  recover-session RETURN if you edited several files

Online-Help
C-h  c  which command does this keystroke invoke
C-h  k  which command does this keystroke invoke and what does it do?
C-h  l  what were my last 100 typed keys
C-h  w  what key-combo does this command have?
C-h  f  what does this function do
C-h  v  what's this variable and what is it's value
C-h  b  show all keycommands for this buffer
C-h  t  start the emacs tutorial
C-h  i  start the info reader
C-h  C-k  start up info reader and go to a certain key-combo point
C-h  F  show the emacs FAQ
C-h  p  show infos about the Elisp package on this machine

Search/Replace
C-s  Search forward
C-r  search backward
C-q  return to where search started (if you are still in search mode)
M-@  query replace

  Space or y  replace this occurence
  Del or n  don't replace
  .  only replace this and exit (replace)
  ,  replace and pause (resume with Space or y)
  !  replace all following occurences
  ^  back to previous match
RETURN or q  quit replace

Search/Replace with regular expressions
Characters to use in regular expressions:
  ^  beginning of line
  $  end of line
  .  single char
  .*  group or null of chars
  \<  beginning of a word
  \\>  end of a word
every char inside the brackets (for example [a-z] means every small letter)

M C-s RETURN search for regular expression forward
M C-r RETURN search for regular expression backward
M C-s incremental search
C-s repeat incremental search
M C-r incremental search backwards
C-r repeat backwards
M-x query-replace-regexp search and replace

Window-Commands
C-x 2 split window vertically
C-x o change to other window
C-x 0 delete window
C-x 1 close all windows except the one the cursors in
C-x ^ enlarge window
M-x shrink-window command says it ;-)  
M C-v scroll other window
C-x 4 f find file in other window
C-x 4 o change to other window
C-x 4 0 kill buffer and window
C-x 5 2 make new frame
C-x 5 f find file in other frame
C-x 5 o change to other frame
C-x 5 0 close this frame

Bookmark commands
C-x r m set a bookmark at current cursor pos
C-x r b jump to bookmark
M-x bookmark-rename says it
M-x bookmark-delete " 
M-x bookmark-save "  
C-x r l list bookmarks

d mark bookmark for deletion
r rename bookmark
s save all listed bookmarks
f show bookmark the cursor is over
m mark bookmarks to be shown in multiple window
v show marked bookmarks (or the one the cursor is over)
t toggle listing of the corresponding paths
w " path to this file
x delete marked bookmarks
Del ?
q quit bookmark list

M-x bookmark-write write all bookmarks in given file
M-x bookmark-load load bookmark from given file
Shell
M-x shell starts shell modus
C-c C-c same as C-c under unix (stop running job)
C-d delete char forward
C-c C-d Send EOF
C-c C-z suspend job (C-z under unix)
M-p show previous commands

DIRectory EDitor (dired)
C-x d start up dired
C (large C) copy
d mark for erase
D delete right away
e or f open file or directory
g reread directory structure from file
G change group permissions (chgrp)
k delete line from listing on screen (don't actually delete)
m mark with *
N move to next line
O open file in other window and go there
C-o open file in other window but don't change there
P print file
q quit dired
Q do query-replace in marked files
R rename file
u remove mark
v view file content
x delete files marked with D
z compress file
M-DEL remove all marks (whatever kind)
~ mark backup files (name~ files) for deletion
# mark auto-save files (#name#) for deletion
*/ mark directory with * (C-u * removes that mark again)
= compare this file with marked file
M-= compare this file with it's backup file
! apply shell command to this file
M-} change to the next file marked with * od D
M-{ "previous"
% d mark files described through regular expression for deletion
% m " (with *)
+ create directory
> changed to next dir
< change to previous dir
s toggle between sorting by name or date

Maybe into this category also fits this command:
M-x speedbar starts up a separate window with a directory view

Telnet
M-x telnet starts up telnet-modus
C-d  either delete char or send EOF
C-c C-c  stop running job (similar to C-c under unix)
C-c C-d  send EOF
C-c C-o  clear output of last command
C-c C-z  suspend execution of command
C-c C-u  kill line backwards
M-p  recall previous command

Text
Works only in text mode
M-s  center line
M-S  center paragraph
M-x  center-region name says

Macro-commands
C-x  {  start macro definition
C-x }  end of macro definition
C-x e  execute last defined macro
M-n C-x e  execute last defined macro n times
M-x name-last-kbd-macro  give name to macro (for saving)
M-x insert-keyboard-macro  save named macro into file
M-x load-file  load macro
M-x macroame  execute macroname

Programming
M C-\  indent region between cursor and mark
M-m  move to first (non-space) char in this line
M-^  attach this line to previous
M-;  formatize and indent comment
C, C++ and Java Modes
M-a  beginning of statement
M-e  end of statement
M C-a  beginning of function
M C-e  end of function
C-c RETURN  Set cursor to beginning of function and mark at the end
C-c C-q  indent the whole function according to indentation style
C-c C-a  toggle modus in which after electric signs (like { }) ; ; /*) emacs does the indentation
C-c C-d  toggle auto hungry mode in which emacs deletes groups of spaces with one del-press
C-c C-u  go to beginning of this preprocessor statement
C-c C-c  comment out marked area
More general (I guess)
M-x outline-minor-mode  collapses function definitions in a file to a mere {...}
M-x show-subtree  If you are in one of the collapsed functions, this un-collapses it
In order to achive some of the feats coming up now you have to run etags *.c *.h *.cpp (or what ever ending you source files have) in the source directory
M-.  (Thats Meta dot) If you are in a function call, this will take you to it's definition
M-x tags-search ENTER  Searches through all you etaged
M-.  (Meta comma) jumps to the next occurence for tags-search
M-x tags-query-replace yum. This lets you replace some text in all the tagged files
GDB (Debugger)
M-x gdb starts up gdbm in an extra window

Version Control
C-x v d show all registered files in this dir
C-x v = show diff between versions
C-x v u remove all changes since last checkin
C-x v ~ show certain version in different window
C-x v l print log
C-x v i mark file for version control add
C-x v h insert version control header into file
C-x v r check out named snapshot
C-x v s create named snapshot
C-x v a create changelog file in gnu-style