```
# Emacs: Xah Fly Keys Tutorial
# By Cybertron – 20/11/2019 - based on:
# Xah Lee. Date: 2018-03-30. Last updated: 2018-11-13.

# Master Command/Insert Mode Switch

# command-mode and insert-mode

alt-space or home Command-mode
f     Insert-mode
space-space       Insert-mode

# Cursor Movement
# set querty-abnt for abnt2 keyboard

j l k i           Left, right, down and up
u o         Word-left, word-right
h ; (or ç)  Beginning/end of line/indent/paragraph
Space h      Beginning-of-buffer
Space n      End-of-buffer

# Text Editing

e r    Delete word left/right
d      To deletes the left char or bracket. The brackets can be of any type ()[]
{}
Ctrl-u d    Delete only the brackets
5     Deletes the right char or bracket
g     Deletes the current text block. Text blocks are separated by empty lines.
w     Shrink-whitespaces. Delete blank lines or white spaces.
Space g      Kill-line. Delete to end of line.

# undo

y      Undo
Ctrl+g y     Redo. To redo again, just press y one or more time. To reverse
direction, press Ctrl+g again.

# I recommend not to use any undo/redo package. If you have them installed
(e.g.undo tree), uninstall it.

# Cut Copy Paste

x      xah-cut-line-or-region to cut. It'll cut current line.
c      xah-copy-line-or-region
v      xah-paste-or-paste-previous. When pressed twice, it'll paste the previous
copied text (kill-ring). Repeat to swap to earlier copied text.
v Ctrl+g v  Paste twice
v Return v   Paste twice
v space v   Paste twice
To paste something 5 times    Ctrl+u 5 v.
Space c      xah-copy-all-or-region
Space x      xah-cut-all-or-region
Space t      xah-show-kill-ring. Show copy history. Useful when you want text
that you copied long ago. Just move cursor there and copy the part you want.

# Sometimes you want to copy something and use it for the next 10 minutes while,
use:

Space k 3    xah-copy-to-register-1
Space k 4    xah-paste-from-register-1
```

# Text Selection
# These are quick selections. 90% of the time, they are all you need.

6   Selects text block.
7   Selects line.
8   Selects current word. If cursor is before a bracket, select the bracket and inner text. The command name is xah-extend-selection, but currently best used for just selecting word or bracket text.
9   Selects inner text between brackets or quotes, e.g. "inside string".
Space a   Mark-whole-buffer
Ctrl-g   Cancel a selection

# Mark
# For more complex selection, use mark.

t   set-mark-command to mark. Then, move cursor will create selection. Press again to unmark.
Space s   exchange-point-and-mark This is sometimes useful. It puts cursor to the other side of selection.
0   Move cursor to previous marked position in current buffer. xah-pop-local-mark-ring. Try pressing it a few times.
Ctrl+d   Move cursor to previous marked position, of all buffers. pop-global-mark

M-x

a   To call a command by name.
Space Enter execute-extended-command

# iSearch

n   isearch-forward. Use this to move cursor too. You don't need other packages to jump to locations. Or space-u/F9-u. [see [5]Emacs: avy ace-jump vs isearch]
Space y   xah-search-current-word or selection.

# When in isearch:

Ctrl+y   To paste.
left/right arrow keys ← →   Do search backward/forward.
up/down arrow keys ↑ ↓   Do previous/next search item.
Return   Exit isearch and leave cursor at the current location.
Ctrl+g   Exit isearch and leave cursor at original location.

# There's no key to begin backward search. Just do isearch then press left arrow key.

# Find Replace

Space r   query-replace
Space k r   query-replace-regexp

# I recommend [6]Emacs: xah-find.el, Find Replace in Pure Elisp for find replace in a directory.

# Open Close Save buffer/file
# You can use the following standard keys for now.

Ctrl+n   New buffer/file
Ctrl+o   Open buffer/file

Ctrl+w	Close buffer/file
Ctrl+s	Save buffer/file
Ctrl+Shift+s	Save as buffer/file
Ctrl+Shift+t	Open last closed buffer/file

# When you are comfortable with xah-fly-keys, and feel like you want to reduce
# using Control, then learn the following.

Space i l	New buffer/file
Space i e	Open buffer/file
Space k j	Close buffer/file
Space ;	Save buffer/file
Space i ,	Save as buffer/file
Space i r	Open last closed buffer/file

# Open Files

Space i f	xah-open-file-at-cursor
Space i g	xah-copy-file-path
Space m	Go to dired and place cursor on file name. (dired-jump) Open in external apps (for image, pdf, vidio/audio files, etc.):
Space i w	xah-open-in-external-app
Space i s	xah-show-in-desktop

# Bookmark related for fast open files:

Space i p	bookmark-set
Space i i	bookmark-bmenu-list list bookmarks
Space i o	xah-open-file-fast

# Open recently opened/closed files:

Space i j	recentf-open-files
Space i r	xah-open-last-closed
Space i y	xah-open-recently-closed

# Press Space i Ctrl+h to see a full list.
# You don't need any extra packages to efficiently open files.
# You should never need to type file path to open file, unless it's a new file you rarely work with.
# All files or directories you work daily, should be bookmarked, or with open recent.

# Split Buffer

3	Un-split all. Delete-other-windows
4	Split top/bottom split-window-below
Ctr-space k ,	Switch cursor to next pane, and if no split pane, switch to next window if any. xah-next-window-or-frame
Space 3	Remove current pane. delete-window
Space 4	Split left/right. split-window-right

# If you are on desktop with large monitor, do use a workflow of 2 windows (emacs frame) side by side. Avoid workflow with 1 fullscreen.
# [see [7]Programer Workflow Efficiency]
# [see [8]Emacs: Effective Windows Management]

# switch buffer

Space f	switch-to-buffer
Space i d	(ibuffer) list buffers.

```
Ctrl+8       xah-next-user-buffer
Ctrl+7       xah-previous-user-buffer
Ctrl+6       xah-next-emacs-buffer
Ctrl+5       xah-previous-emacs-buffer

# Working with Quote and Brackets
# xah-fly-keys has a system to work with brackets. Brackets includes ()[]{}“”
and all Unicode brackets.
# These allows you to work efficiently in any programing language with lots
brackets, such as Java, JavaScript, especially lisp.
# You don't need other packages for working with brackets. If you code emacs
lisp, i also recommend [9]Emacs: Xah Emacs Lisp Mode

# Move to Brackets

m .    Move to left/right bracket. (xah-backward-left-bracket) (xah-forward-
right-bracket)
/      (xah-goto-matching-bracket). When cursor is on a bracket, jump to the
matching one. If cursor is not on a bracket, move it to parent left bracket.

# Select Brackets

8     Selects the whole content - including the bracket - when the cursor in
inside them (xah-extend-selection)

# Delete Brackets

d     To delete one char to the left. (xah-delete-backward-char-or-bracket-text)
If the character is a bracket character (any of ()[]{}), the whole bracketed
text will be deleted, including the bracket.
Ctrl+u d    Delete bracket on the left of cursor, and its matching pair.
5      To delete single bracket without pair, move cursor to the left of it,
then press 5.

# Change Brackets

a     xah-change-bracket-pairs
M+x    xah-change-bracket-pairs

# Insert Brackets
# Always insert left/right brackets together.

# Turn on electric-pair-mode on
# (electric-pair-mode 1)

# Alternatively,

Space d k    Insert parenthesis pair. (xah-insert-paren)
Space d l    (xah-insert-square-bracket)
Space d j    (xah-insert-brace)
Space d u    (xah-insert-ascii-double-quote)
Space d i    (xah-insert-ascii-single-quote)
Space d Ctrl+h     To see all other brackets and quotes you can insert.

# Font Size

Ctrl++       text-scale-increase
Ctrl+-       text-scale-decrease
Space l a    text-scale-increase
Space l a -  text-scale-decrease
b     Cycle through capitalized, lowercase and uppercase word
```

```
# More Advanced
# ok, the above are the basics.
# Now, there are about 200 commands with leader keys. These basically replace what emacs C-x does.
# You might want to learn them as you go.
# When in command mode, press Space Ctrl+h to list all leader key sequence commands.
# Remember to use Alt+x describe-function and Alt+x describe-key to find what's the key or command name.
# Misc Notes
# Command Mode Key Must be Most Easy Key
# It's critically important that the key to switch to command mode must be the most easy key on your keyboard.
# I suggest the CapsLock on typical PC keyboards or laptop keyboards.
# For how, see [10]Emacs: Xah Fly Keys Customization

# Insert Mode is Default GNU Emacs Key
# In plain GNU Emacs, in some modes, typing a letter key does not insert letter, but runs a special command. For example, in [11]dired, 【m】 for mark, 【u】 for unmark.

# Note, when in insert mode, it acts as of xah-fly-keys isn't on.

# So, whenever you think there is some problem, do switch to insert mode.

# Location: http://ergoemacs.org/misc/xah-fly-keys_tutorial.html
# [1]DA Emacs
```