

# BOOT.ORG

Chromebook

September 25, 2017

## Contents

<b>1</b>	<b>CUSTOM SETTINGS</b>	<b>1</b>
1.1	First of all . . . . .	1
1.2	Setting the Lisp environment . . . . .	2
1.3	Org and L <sup>A</sup> T <sub>E</sub> X specific settings . . . . .	2
1.4	Web . . . . .	2
1.5	Smex . . . . .	3
1.6	Ido . . . . .	3
1.7	Navigating . . . . .	3
1.8	Tools . . . . .	3

## 1 CUSTOM SETTINGS

### 1.1 First of all

```
(require 'org)
;; (if (window-system)
;;     (split-window-right (floor (* 0.85 (window-width)))))

(setq package-archives '(("gnu" . "https://elpa.gnu.org/packages/")
  ("marmalade" . "https://marmalade-repo.org/packages/")
  ("melpa" . "https://melpa.org/packages/")))

(keyboard-translate ?\ ( ?\[)
(keyboard-translate ?\[ ?\ ( )
(keyboard-translate ?\ ) ?\ ] )
(keyboard-translate ?\ ] ?\ ) )
```

```

(defun which-term ()
  (cond ((string-equal user-full-name "Thinkpad")
    (setq slime-lisp-implementations '((sbcl ("C:\\Program Files\\Steel Bank Common Lisp\\
    (setq Tex-auto-save t)
    (setq TeX-parse-self t)
    (require 'tex-mik)
    (load "auctex.el" nil t t)
    (setq org-export-with-LaTeX-fragments "dvipng")))
    ((string-equal user-full-name "iMac")
    (princ "hello"))
    ((string-equal user-full-name "Chromebook")
    (princ "world"))))

  (which-term)

  ;; (toggle-frame-fullscreen)

```

## 1.2 Setting the Lisp environment

```

(add-hook 'lisp-interaction-mode-hook 'slime-mode)
(add-hook 'lisp-mode-hook 'slime-mode)
(add-hook 'lisp-interaction-mode-hook 'electric-pair-mode)
(add-hook 'emacs-lisp-mode-hook 'electric-pair-mode)
(add-hook 'lisp-mode-hook 'electric-pair-mode)

```

## 1.3 Org and L<sup>A</sup>T<sub>E</sub>X specific settings

```

(setq Tex-auto-save t)
(setq TeX-parse-self t)
(setq org-export-with-LaTeX-fragments "dvipng")
(setq org-return-follows-link t)

```

## 1.4 Web

```

(add-hook 'sgml-mode-hook 'emmet-mode)
(add-hook 'css-mode-hook 'emmet-mode)

```

## 1.5 Smex

```
(global-set-key (kbd "M-x") 'smex)
(global-set-key (kbd "M-X") 'smex-major-mode-commands)

;; Old M-x.
(global-set-key (kbd "C-c C-c M-x") 'execute-extended-command)
```

## 1.6 Ido

```
(setq ido-enable-flex-matching t)
(setq ido-everywhere t)
(ido-mode 1)
```

## 1.7 Navigating

```
(defalias 'yes-or-no-p 'y-or-n-p)
(global-set-key (kbd "M-p") 'ace-window)
```

## 1.8 Tools

```
(defun halt ()
  (interactive)
  (save-some-buffers)
  (kill-emacs))

(defun lisp ()
  (interactive)
  (split-window-horizontally)
  (find-file "~/emacs.d/casio.cl")
  (sly))
```