

Emacs ハック入門 — 暗記支援ツールの作成 —

<http://howm.sourceforge.jp/a/elhack.pdf>

1 Lisp コードのその場実行

*scratch*バッファに以下を書いて行末で C-x C-e
(C-x は `Control` キーを押しながら `x` キーを押す)

```
(+ 3 4)
```

以下もそれぞれ同様に

```
(* 2 3 4)
(* 2 (+ 3 4))
(* (+ 2 3) 4)
(format "%s + %s = %s" 3 4 (+ 3 4))
(format-time-string "%Y-%m-%d %H:%M:%S")
(insert "abc")
```

- $2 * 3 + 4 * 5$ や $2 * (3 + 4) * 5$ も計算してみよう

2 関数を作る

*scratch*バッファに以下を書いて、さいごに C-x C-e

```
(defun my-insert-time ()
  (insert (format-time-string "%H%M")))

```

以下も書いて C-x C-e

```
(my-insert-time)
```

- [at 10:35] のような時刻表記に直してみよう
- my-insert-date も作ってみよう

3 関数をコマンドにしてキー割り当て

*scratch*バッファに以下を書いて、さいごに C-x C-e

```
(defun my-insert-time ()  
  (interactive)  
  (insert (format-time-string "%H%M")))
```

以下も書いて C-x C-e

```
(define-key global-map (kbd "C-c C-t") 'my-insert-time)
```

C-c C-t を入力

- 好みの時刻表記に直してみよう
- 日付入力コマンドも作って C-c C-d に割り当てよう

4 メジャーモードの作成

my.el というファイルに以下を書いて保存

```
(define-derived-mode my-mode nil "My")  
(define-key my-mode-map (kbd "RET") 'my-hide)  
(define-key my-mode-map (kbd "SPC") 'my-search)
```

```
(defun my-hide ()  
  (interactive)  
  ;; 先頭からカーソルまでだけを表示  
  (narrow-to-region (point-min) (point)))
```

```
(defun my-search ()  
  (interactive)  
  ;; narrow-to-region を取り消して全体を表示  
  (widen)  
  (search-forward "-"))
```

```
(defun my ()  
  (interactive)  
  (my-mode)  
  (goto-char (point-min)))
```

1. M-x load-file `Enter` my.el `Enter` と入力
(M-x は `Esc` キーを押してから `x` キーを押す)
2. *scratch*バッファで M-x my `Enter` と入力
ウィンドウ下部のモードラインに「My」と表示される
3. スペースキーを押すたびに次の「-」へカーソルが移る
4. `Enter` キーを押すとカーソル以降が表示から消える
(スペースキーを押すとまた表示される)
5. M-x lisp-interaction-mode `Enter` で元のモードに戻る

5 課題：暗記支援ツールの作成

スペースキーを押すたびに「,」までを順次表示する anki コマンドを作ろう

hat,	hat, 帽子 hit,	hat, 帽子 hit, 安打 hut,
------	-----------------	----------------------------

うまくできたら次の関数を使って改良しよう

```
(defun anki-dup () ; バッファを複製する関数
  (let ((orig (current-buffer)))
    (switch-to-buffer (generate-new-buffer "*anki*"))
    (insert-buffer-substring orig)))

(defun anki-shuffle () ; 行の順序をシャッフルする関数
  (goto-char (point-min))
  (sort-subr nil 'forward-line 'end-of-line 'random))
```

参考文献

- [1] 広瀬雄二：『やさしい Emacs-Lisp 講座』, カットシステム, 1999.
<http://www.gentei.org/~yuuji/elisp/elisplec.html>
- [2] 「プログラミングの楽しさ：ソフトウェアを作って公開しよう」, 宮崎大学 第19回先導的 IT セミナー, 2012.