

Emacs Modesキーバインディング備忘録

Emacs の c-mode/c++-mode, shell-script-mode, makefile-mode, latex-mode, html-mode, python, ruby, perl, awk, VC パッケージのキーバインディングについてまとめます。

c-mode/c++-mode主要キーバインディング一覧

c-mode/c++-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動

M-a	c-beginning-of-statement	式の先頭へ移動
M-e	c-end-of-statement	式の末尾へ移動
C-M-a	c-beginning-of-defun	関数定義の先頭へ移動
C-M-e	c-end-of-defun	関数定義の末尾へ移動
C-c C-p	c-backward-conditional	CPP条件分岐の前の先頭へ移動
C-c C-n	c-forward-conditional	CPP条件分岐の次の末尾へ移動
C-c C-u	c-up-conditional	CPP条件分岐の先頭へ移動

カーソル位置の編集

C-c C-d	c-hungry-delete-forward	先の空白をすべて消去
C-c DELまたはC-c C-DEL	c-hungry-delete-backwards	前の空白をすべて消去
C-c :	c-scope-operator	「::」の挿入(c++-modeのみ)
M-jまたはC-M-j	c-indent-new-comment-line	コメント行内の改行(RET TAB と等価)
M-q	c-fill-paragraph	カーソル位置の式やコメント行の整形
C-c C-q	c-indent-defun	カーソル位置の関数定義のインデントを整形
C-M-q	c-indent-exp	カーソル位置の式のインデントを整形
C-M-h	c-mark-function	カーソル位置の関数定義の末尾にマークし先頭へ移動

リージョンの操作

C-c C-c	comment-region	リージョンをコメント行へ変換(C-u に続くと逆の動作)
C-c C-\	c-backslash-region	リージョンの行末すべてに「\」を付加(C-u に続くと逆の動作)

その他の操作

C-c .	c-set-style	Cコーディングスタイルを選択
-------	-------------	----------------

c-mode/c++-mode主要キーバインディング解説

前述の c-mode/c++-mode のキーバインディングのなかで、特に解説しておきたいものを記します。

C-M-h **c-mark-function** カーソル位置の関数定義の末尾にマークし先頭へ移動
この後に C-x C-x とすれば末尾に移動し確認でき、M-w とすれば関数定義全体がキルリングに保存されます。

C-c . **c-set-style** Cコーディングスタイルを選択

gnu, k&r, BSD, stroustrup, linux, python モジュールなどから選びます。他者のコードを編集する時にそれに合ったスタイルを選ぶとよいでしょう。その際は、以下の字下げ幅を目安にします。

gnu: 2, k&r: 5, BSD: 8, stroustrup: 4, linux: 8, python: 8=TAB

shell-script-mode主要キーバインディング一覧

shell-script-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

SHELLの設定

C-c : sh-set-shell /bin/sh, /bin/ksh, /bin/bash, /bin/zshなどを設定

カーソル位置の編集

C-c C-c	sh-case	case文の挿入
C-c C-f	sh-for	for文の挿入
C-c TAB	sh-if	if文の挿入
C-c C-r	sh-repeat	repeat文の挿入(/bin/zshのみ有効)
C-c C-s	sh-select	select文の挿入(/bin/shでは無効)
C-c C-u	sh-until	until文の挿入
C-c C-w	sh-while	while文の挿入
C-c C-l	sh-indexed-loop	増分による繰り返し文の挿入
C-c C-o	sh-while-getopts	getoptsによるオプション解析ループの挿入
C-c (sh-function	関数の挿入
C-c +	sh-add	増分を挿入「`expr \$i + 1`」や「`\${i+1}`」など
C-c C-t	sh-tmp-file	一時ファイル生成手続きを挿入

リージョンの操作

C-c C-\ sh-backslash-region リージョンの行末すべてに「\」を付加(C-u に続く逆の動作)

バッファの一行目に書いた「#!/bin/ksh」や C-c : で設定したSHELLによって、挿入される内容が変化しますので、用途に応じて Bourne Shell, Korn Shell, Bourne-Again Shell, Z Shell 向けなのかを考慮しましょう。

makefile-mode主要キーバインディング一覧

makefile-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動、その他

M-p	makefile-previous-dependency	前の依存関係へ移動
M-n	makefile-next-dependency	次の依存関係へ移動
C-c C-c	comment-region	リージョンをコメント行へ変換(C-u に続く逆の動作)
C-c C-\	makefile-backslash-region	リージョンの行末すべてに「\」を付加(C-u に続く逆の動作)
C-c RET RET	makefile-mode	一般 make ファイルモードへ移行
C-c RET C-b	makefile-bsdmake-mode	BSD make ファイルモードへ移行
C-c RET C-g	makefile-gmake-mode	GNU make ファイルモードへ移行
C-c RET C-p	makefile-makepp-mode	GNU makepp ファイルモードへ移行
C-c RET C-a	makefile-automake-mode	automake ファイルモードへ移行
M-TAB	completion-at-point	カーソル位置の補完

*よりすすんだ使い方についてはmakefile-modeのヘルプをご覧ください。

latex-mode主要キーバインディング一覧

latex-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動・編集、その他

C-c C-u	tex-goto-last-unclosed-latex-block	閉じていない\begin{...}へ移動
C-j	tex-terminate-paragraph	改段落の挿入 (RET RET と等価)
M-RET	latex-insert-item	\item の挿入
C-c {	tex-insert-braces	{ } の挿入
C-c C-s	latex-split-block	\end{...}\begin{...}の挿入、つまりブロックを分割

カーソル位置の対話的な編集、その他

C-c C-o	latex-insert-block	\begin{...}の挿入
C-c C-t	latex-insert-block	\begin{...}の挿入
C-c C-e	latex-close-block	\end{...}の挿入
C-c /	latex-close-block	\end{...}の挿入
C-c]	latex-close-block	\end{...}の挿入
M-TAB	ispell-complete-word	カーソル位置の単語をスペルチェック

*この Emacs 標準の latex-mode は単純で使い易いと思います。

コマンド支援機能

C-c C-b	tex-buffer
C-c C-c	tex-compile
C-c C-f	tex-file
C-c C-r	tex-region
C-c TAB	tex-bibtex-file
C-c C-v	tex-view
C-c C-p	tex-print
C-c C-q	tex-show-print-queue
C-c RET	tex-feed-input
C-c C-k	tex-kill-job
C-c C-l	tex-recenter-output-buffer

*筆者は makefile で latex のコンパイルを行うので、上記コマンド支援機能はまず使うことはありません。

html-mode主要キーバインディング一覧

html-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動・編集、その他

C-c C-b	sgml-skip-tag-backward	後方のタグへ移動
C-c C-f	sgml-skip-tag-forward	前方のタグへ移動
C-c C-dまたはC-c DEL	sgml-delete-tag	タグを除去
C-c C-n	sgml-name-char	&name;を挿入
C-c 8	sgml-name-8bit-mode	&#xxxx;を挿入

M-: (setq sgml-xml-mode t) sgml-xml-mode に設定(通常は自動判定でnilかt)

C-c C-j	html-line	 の挿入(sgml-xml-mode の場合は)
C-c RET	html-paragraph	<p>の挿入(sgml-xml-mode の場合は<p></p>)
C-c 1	html-headline-1	<h1></h1>の挿入
C-c 2	html-headline-2	<h2></h2>の挿入
C-c 3	html-headline-3	<h3></h3>の挿入
C-c 4	html-headline-4	<h4></h4>の挿入
C-c 5	html-headline-5	<h5></h5>の挿入
C-c 6	html-headline-6	<h6></h6>の挿入
C-c C-c -	html-horizontal-rule	<hr>の挿入(sgml-xml-mode の場合は<hr />)
C-c C-c l	html-list-item	の挿入(sgml-xml-mode の場合は)
C-c C-c o	html-ordered-list	...の挿入(sgml-xml-mode の場合は同上)
C-c C-c u	html-unordered-list	...の挿入(sgml-xml-mode の場合は同上)

カーソル位置の対話的な編集、その他

C-c C-c c	html-checkboxes	<input type="checkbox"...>...の挿入
C-c C-c h	html-href-anchor	...の挿入
C-c C-c i	html-image	の挿入
C-c C-c n	html-name-anchor	...の挿入
C-c C-c r	html-radio-buttons	<input type="radio"...>...の挿入
C-c C-o	sgml-tag	タグを挿入
C-c C-t	sgml-tag	タグを挿入
C-c C-e	sgml-close-tag	タグを閉じる
C-c /	sgml-close-tag	タグを閉じる
C-c]	sgml-close-tag	タグを閉じる
C-c C-a	sgml-attributes	タグの属性を対話的に挿入
C-c TAB	sgml-tags-invisible	タグを不可視・可視
M-TAB	ispell-complete-word	カーソル位置の単語をスペルチェック

*html-mode も単純で使い易いです。加えて c-mode, latex-mode に似たキーバインディングがあることが分かるでしょう。よって、Emacs 標準のモードであればいくつか覚えてしまえば、他のモードも勘で使えてしまうことが多いです。

web-mode主要キーバインディング一覧

web-mode において、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

全般

M-;	web-mode-comment-or-uncomment	コメント挿入またはアンコメント
M-q	fill-paragraph	カーソル位置の段落を整形
C-c C-f	web-mode-fold-or-unfold	タグ/ブロックの折り畳みのオンオフ
C-c TAB	web-mode-buffer-indent	インデント実行
C-c C-l	web-mode-file-link	ハイパーリンクの挿入
C-c RET	web-mode-mark-and-expand	カーソル位置のブロック末尾をマークし先頭に移動
C-c C-n	web-mode-navigate	タグの先頭か末尾から対応する末尾か先頭に移動
C-c C-s	web-mode-snippet-insert	HTML5, table, ul等の断片の挿入
C-c C-w	web-mode-whitespaces-show	空白等の不可視文字の表示・非表示

要素関連

C-c C-e /	web-mode-element-close	終了タグを挿入して要素を閉じる 但し「</」と書けば、自動的に終了タグを補完
C-c C-e a	web-mode-element-content-select	カーソル位置の要素の内容の先頭に移動
C-c C-e b	web-mode-element-beginning	カーソル位置の要素の先頭に移動
C-c C-e c	web-mode-element-clone	カーソル位置の要素を複製
C-c C-e d	web-mode-element-child	カーソル位置の要素の子要素の先頭に移動
C-c C-e e	web-mode-element-end	カーソル位置の要素の末尾に移動
C-c C-e f	web-mode-element-children-fold-or-unfold	カーソル位置の要素の全ての子要素の折り畳みのオンオフ
C-c C-e i	web-mode-element-insert	カーソル位置に要素を挿入（不適切な位置には挿入不可）
C-c C-e k	web-mode-element-kill	カーソル位置の要素を削除
C-c C-e m	web-mode-element-mute-blanks	カーソル位置の要素間にコメントを挿入
C-c C-e n	web-mode-element-next	カーソル位置の次の要素に移動
C-c C-e p	web-mode-element-previous	カーソル位置の前の要素に移動
C-c C-e r	web-mode-element-rename	カーソル位置の要素のタグ名を変更
C-c C-e s	web-mode-element-select	カーソル位置の要素末尾をマークし先頭に移動
C-c C-e t	web-mode-element-transpose	カーソル位置の要素と次の要素を交換
C-c C-e u	web-mode-element-parent	カーソル位置の要素の親要素に移動
C-c C-e v	web-mode-element-vanish	カーソル位置の要素のタグを削除
C-c C-e w	web-mode-element-wrap	カーソル位置の要素を要素で囲む

タグ関連

C-c C-t a	web-mode-tag-attributes-sort	カーソル位置のタグの属性を並び替える
C-c C-t b	web-mode-tag-beginning	カーソル位置のタグの先頭に移動
C-c C-t e	web-mode-tag-end	カーソル位置のタグの末尾に移動
C-c C-t m	web-mode-tag-match	カーソル位置のタグの末尾か先頭に移動
C-c C-t n	web-mode-tag-next	カーソル位置の次のタグに移動
C-c C-t p	web-mode-tag-previous	カーソル位置の前のタグに移動
C-c C-t s	web-mode-tag-select	カーソル位置のタグ末尾をマークし先頭に移動

属性関連

C-c C-a b	web-mode-attribute-beginning	カーソル位置の属性の先頭に移動
-----------	------------------------------	-----------------

C-c C-a e	web-mode-attribute-end	カーソル位置の属性の末尾に移動
C-c C-a i	web-mode-attribute-insert	カーソル位置の属性を挿入
C-c C-a k	web-mode-attribute-kill	カーソル位置の属性を削除
C-c C-a n	web-mode-attribute-next	カーソル位置の次の属性に移動
C-c C-a p	web-mode-attribute-previous	カーソル位置の前の属性に移動
C-c C-a s	web-mode-attribute-select	カーソル位置の属性末尾をマークし先頭に移動
C-c C-a t	web-mode-attribute-transpose	カーソル位置の属性と次の属性を交換

ブロック関連

C-c C-b b	web-mode-block-beginning	カーソル位置のコードブロックの先頭に移動
C-c C-b c	web-mode-block-close	カーソル位置のコードブロックを閉じる
C-c C-b e	web-mode-block-end	カーソル位置のコードブロックの末尾に移動
C-c C-b k	web-mode-block-kill	カーソル位置のコードブロックを削除
C-c C-b n	web-mode-block-next	カーソル位置の次のコードブロックに移動
C-c C-b p	web-mode-block-previous	カーソル位置の前のコードブロックに移動
C-c C-b s	web-mode-block-select	カーソル位置のコードブロック末尾をマークし先頭に移動

DOM関連

C-c C-d a	web-mode-dom-apostrophes-replace	バッファの内容のシングルクォート「'」を「'」に置換
C-c C-d d	web-mode-dom-errors-show	バッファのタグの不一致を表示
C-c C-d e	web-mode-dom-entities-replace	カーソル位置のエンティティ文字を元の文字に戻す
C-c C-d n	web-mode-dom-normalize	バッファをHTMLとして正規化
C-c C-d q	web-mode-dom-quotes-replace	バッファの内容のダブルクォート「"」を置換
C-c C-d t	web-mode-dom-traverse	バッファをHTMLとしてDOMツリーを辿る
C-c C-d x	web-mode-dom-xpath	カーソル位置のXPATHをミニバッファに表示

ヘルパー関数

web-mode-apostrophes-replace	バッファの内容のシングルクォート「'」を「'」に置換
web-mode-entities-replace	カーソル位置のエンティティ文字を元の文字に戻す
web-mode-quotes-replace	バッファの内容のダブルクォート「"」を置換

*web-mode は執筆時点でまだ Emacs 標準ではありませんが、標準の html-mode では Javascript や PHP 対応において力不足になりつつあると思います。そういったときは、web-mode.el をインストールしておいて、M-x web-mode で切り替えるとよいでしょう。

python-mode主要キーバインディング一覧

python-modeにおいて、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置もしくはリージョンのインデントレベルの編集

C-c <	python-indent-shift-left	カーソル位置かリージョンのインデントレベルを下げる
C-c >	python-indent-shift-right	カーソル位置かリージョンのインデントレベルを上げる

カーソル位置の雛形の挿入

C-c C-t c	python-skeleton-class	クラス定義を始める
C-c C-t d	python-skeleton-def	関数定義を始める
C-c C-t f	python-skeleton-for	for 制御構文を始める
C-c C-t i	python-skeleton-if	if 制御構文を始める
C-c C-t t	python-skeleton-try	try 制御構文を始める
C-c C-t w	python-skeleton-while	while 制御構文を始める

ruby-mode主要キーバインディング一覧

ruby-modeにおいて、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動

C-M-b	ruby-backward-sexp	後方の式へ移動
C-M-f	ruby-forward-sexp	前方の式へ移動
C-M-p	ruby-beginning-of-block	ブロックの先頭へ移動
C-M-n	ruby-end-of-block	ブロックの末尾へ移動

カーソル位置の編集

C-M-q	ruby-indent-exp	カーソル以降のインデントを整形
C-c {	ruby-toggle-block	'do ... end' と '{ ... }' を書き換える

perl-mode主要キーバインディング一覧

perl-modeにおいて、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

カーソル位置の移動

C-M-a	perl-beginning-of-function	関数の先頭へ移動
C-M-e	perl-end-of-function	関数の末尾へ移動
C-M-h	perl-mark-function	カーソル位置の関数定義の末尾にマークし先頭へ移動

カーソル位置の編集

C-M-q	perl-indent-exp	カーソル以降のインデントを整形
-------	-----------------	-----------------

awk-mode主要キーバインディング一覧

awk-modeにおいて、使用頻度の高いキーバインディングをまとめておきます。

カーソル位置の移動

C-M-a	c-awk-beginning-of-defun	式の前頭へ移動
C-M-e	c-awk-end-of-defun	式の末尾へ移動

awk-modeはc-modeから作られており、その他はc-modeと同様です。

VC主要キーバインディング一覧

Emacs の VC(version control) パッケージにおいて、使用頻度の高いキーバインディングをグループ分けしてまとめておきます。

リポジトリの編集

C-x v d	vc-dir	git bzd svn cvs status の実行
C-x v +	vc-update	git pull bzd merge svn cvs update の実行
C-x v l	vc-print-log	git bzd svn cvs log を実行
C-x v L	vc-print-root-log	リポジトリのトップでのログを表示
C-x v I	vc-log-incoming	更新によるログを表示
C-x v O	vc-log-outgoing	変更によるログを表示
C-x v =	vc-diff	git bzd svn cvs diff の実行
C-x v D	vc-root-diff	リポジトリのトップでの diff を表示
C-x v g	vc-annotate	git bzd svn blame cvs annotate の実行

リポジトリの対話的な編集

C-x v u	vc-revert	バッファの編集を対話的に破棄
C-x v c	vc-rollback	リポジトリの編集を対話的に破棄
C-x v v	vc-next-action	git bzd svn cvs commit の対話的な実行
C-x v i	vc-register	git bzd svn cvs add の対話的な実行
M-x vc-delete-file		git bzd svn cvs rm の対話的な実行
M-x vc-rename-file		git bzd svn mv cvs rm && cvs add の対話的な実行
C-x v m	vc-merge	git bzd merge svn merge -r cvs update -j の対話的な実行
C-x v ~	vc-revision-other-window	別のリビジョンを対話的に表示
C-x v r	vc-retrieve-tag	タグやブランチのリビジョンを対話的に取得
C-x v s	vc-create-tag	リビジョンにタグやブランチを対話的に設定

*VCパッケージのバックエンドとして、上記では、git, bzd, svn, cvs で対応するコマンドの雰囲気を書きましたが、他に対応しているバックエンドには RCS, CSSC(GNU SCCS), Mercurial(hg), monotone(mtn), GNU Arch(tla) があります。

(2014/7/16, 2018/3/25 山田 泰司)

参考文献

1. Free Software Foundation, Inc., ``[GNU Emacs Manual](#),'' 2012.