

CL Portacle :

Common Lispのオールインワン開発環境

Satoshi Imai

Twitter: [@masatoi0](#)

Github: [masatoi](#)

Blog: [masatoi.github.io](#)



Common Lispの開発環境

- Lisp系言語はエディタのサポートが必須
 - 括弧の対応、自動インデント、REPLとの連携、補完、etc...
- OSSでの選択肢は多くない
 - Emacs + SLIME
 - Vim + slimv
 - Atom + atom-slime
 - lem

Lisp入門者に何を薦めたらいいのか

- 大事だと思われること
 - インストールが簡単
 - マルチプラットフォームで動作する (Windows対応)
 - 追加の設定を必要とせず、最初からすぐに使える状態になっている
 - マウスで操作できる
 - 入門用だけでなく実用的なプログラムも開発できないといけない

CL> Portacle

- Common Lisp開発に必要なソフトウェアをまとめてアーカイブしたもの

-  Emacs (エディタ)
-  SBCL (Lisp処理系)
-  Quicklisp (パッケージ管理システム)
-  Git (バージョン管理システム)

Portacleの構成要素 (1)

- Emacs (エディタ)
 - 古くからあるソースコードエディタ
 - 拡張言語がLispの一方言であり (Emacs Lisp)、他のLisp系言語でもよく使われる
 - Common Lisp処理系との連携を取るための**SLIME**というパッケージが非常に優秀
 - Portacleに入っているEmacsにはCommon Lispで開発するための便利なパッケージ群が既にインストールされている

Portacleの構成要素 (2)

- SBCL (Common Lisp処理系)
 - Steel Bank Common Lispの略。CMUCLから派生した。カーネギーメロン大の創立者が鉄鋼業と銀行業の人だったことから名前がついた
 - Common Lispとは言語仕様のことであり、処理系は色々ある
 - SBCLは最も人気があるオープンソースの処理系の一つ
 - ネイティブコードまでコンパイルするので速い。適切な最適化をほどこせばC言語並に速くなる

Portacleの構成要素 (3)

- Quicklisp (パッケージ管理システム)
 - ネットワークからライブラリをダウンロードした後、コンパイル、ロードして利用可能な状態にする
 - 依存ライブラリもインストールされる
 - ライブラリの探索、一括アップデートなどもREPLからできる

Portacleの構成要素 (4)

- Git (バージョン管理システム)
 - 独立したGitのバイナリが同梱してある
 - magit: EmacsのGitフロントエンドのパッケージ
 - gitコマンドを直接操作しなくてもほとんどの操作をEmacsからできる

Portacleのインストール

- Windows版、Mac版、Linux版が用意されている
 - <https://portacle.github.io/>
- インストールはアーカイブをダウンロードして展開するだけ。管理者権限はいらない
 - 単なるフォルダなのでレジストリなどはいじらない
 - そのフォルダ内で完結しているので、USBメモリなどに入れば開発環境ごと持ち運べる

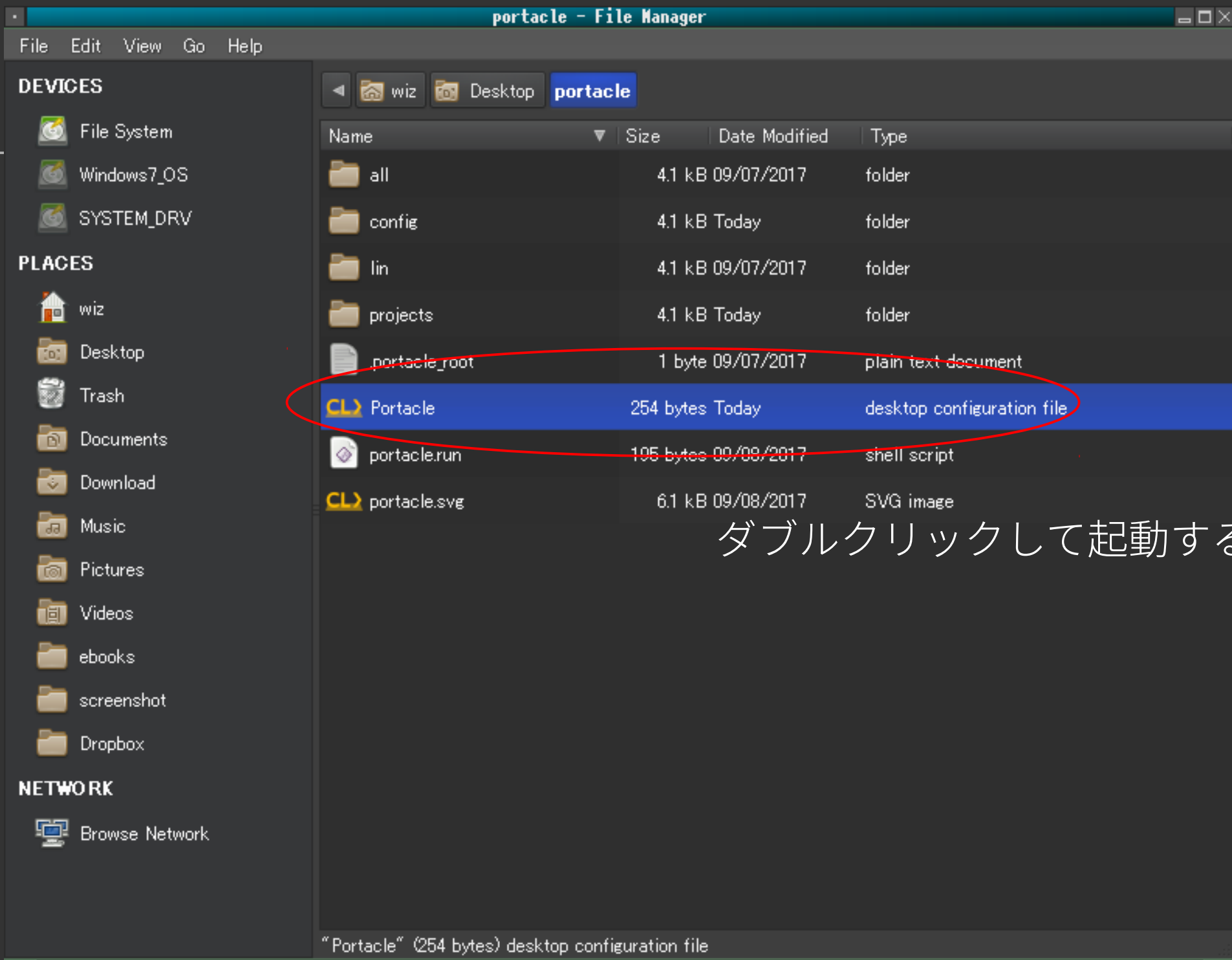


What This Is



Portacle is a complete IDE for Common Lisp that you can take with you on a USB stick. It is multi-platform and can be run on Windows, OS X, and Linux. Since it does not require any complicated installation process, it is set up and running in no time.

It lends itself very well both to absolute beginners of Lisp that just need a good starting point, as well as



File Edit Options Buffers Tools Development SLIME REPL Presentations Lisp Trace Help

```
1 ;;;; Welcome to Portacle, the Portable Common Lisp Environment.  
2 ;; For information on Portacle and how to use it, please read the website at  
3 ;; https://github.com/Shirane/portacle  
4 ;; or see the *portacle-help* buffer. You can switch to it by pressing this:  
5 ;; Ctrl+h  
6 ;; or by clicking on the *scratch* field below until it changes to read  
7 ;; *portacle-help*.  
8 ;;  
9 ;; You can use this buffer for notes and tinkering with small pieces of code.  
10  
□
```

```
- 448 U: *scratch* Lisp Interaction company Portacle 11 : 0 All  
1 ; SLIME 2.20  
2 CL-USER> █
```

```
* 22 U: *slime-repl sbcl* REPL company Paredit Portacle adoc 2 : 9 All
```

```
1 ;;;; Welcome to Portacle, the Portable Common Lisp Environment.
2 ;; For information on Portacle and how to use it, please read the website at
3 ;; https://github.com/ShirleyD/Portacle
4 ;; or see the *portacle-help* buffer. You can switch to it by pressing this:
5 ;; Ctrl+h
6 ;; or by clicking on the *scratch* field below until it changes to read
7 ;; *portacle-help*.
8 ;;
9 ;; You can use this buffer for notes and tinkering with small pieces of code.
10
```

起動した状態

*scratch*バッファ:

Common Lispの小さなコードの実験に使う

```
- 448 U: *scratch* Lisp Interaction company Portacle
```

```
11 : 0 All
```

```
1 ; SLIME 2.20
2 CL-USER> █
```

REPLバッファ:

対話的に式を評価できる

バッファとウィンドウ:

バッファは仮想的な作業領域のこと、メニューバーのBuffersから選択できる
ウィンドウは画面上の表示領域のこと。今は上下に2分割されている

```
* 22 U: *slime-repl sbcl* REPL company Paredit Portacle adoc
```

```
2 : 9 All
```

Emacsの基本操作 (1)

- カーソル移動は矢印キーかキーバインドで
 - C-n、C-p、C-f、C-b (↓ ↑ → ←)
- 表記の読み方
 - CはControl、MはMetaキー (大抵Altキー)
 - 複数のキーバインドを続けて打つキーストロークもある
 - C-x C-s (ファイル保存)
 - Controlを押しっぱなしでxとfを続けて打つ
 - C-x s (全てのバッファのファイルを保存する)
 - Controlとxを同時押しした後、Controlを離してsを打つ

Emacsの基本操作 (2)

- ファイル関係

- **C-x C-f** (ファイルを開く。またはファイルマネージャからマウスでファイルをドラッグ&ドロップする)
- **C-x k** (バッファを保存せず閉じる)
- **C-x C-w** (ファイルを別名保存)

- 編集関係

- **C-w** (カット)、**M-w** (コピー)、**C-y** (ペースト)
- マウスで領域選択した後に中ボタンクリックで領域を貼り付けられる

Emacsの基本操作 (3)

- 画面関係
 - **C-x 1** (現在のウィンドウを画面全体に広げる)
 - **C-x 2** (現在のウィンドウを縦に2分割する)
 - **C-x 3** (現在のウィンドウを横に2分割する)
 - **C-x o** (次のウィンドウに移動する。単にマウスでクリックしてもよい)
 - File→New Frameでフレーム (普通の意味でのウィンドウ) を作れる
- バッファの切り替え
 - **C-x b** の後にバッファの名前の一部を入力してエンター
 - またはメニューバーのBuffersの中から選ぶ

Emacsの基本操作 (4)

- **C-g** (操作をキャンセルするコマンド (重要!))
 - 操作の途中で分からなくなったらとにかくC-gを打つべし
- Emacsが出してくるポップアップ画面は大体qキーで消せる
- その他はEmacsの日本語チュートリアル参照
 - メニューバーで Help→Emacs tutorial (choose language)
 - ポップアップ画面が出るのでJapaneseをクリックする

Emacs 入門ガイド. 本ファイルの著作権については最後に御覧下さい。

NOTICE: The main purpose of the Emacs tutorial is to teach you the most important standard Emacs commands (key bindings). However, your Emacs has been customized by changing some of these basic editing commands, so it doesn't correspond to the tutorial. We have inserted colored notices where the altered commands have been introduced. [\[More\]](#)

Emacs のコマンドを入力するには、一般にコントロールキー（キートップに CONTROL とか CTRL とか CTL と書いてある）やメタキー（キートップに META とか ALT とか EDIT と書いてある）を使います。そこで、CONTROL とか META とかを書く代わりに、次のような記法を使うことにします。

- C-<文字> コントロールキーを押したまま、<文字>キーを押します。例えば、C-f はコントロールキーを押しながら f のキーを押すことです。
- M-<文字> メタキーを押したまま、<文字>キーを押します。もしメタキーがない場合は、エスケープキーを押してから離し、それから<文字>キーを押します。以降エスケープキーのことを <ESC> と書きます。

！重要！： Emacsを終了するには、C-x C-c をタイプします。
">" で始まる行は、その時何をすべきかを指示しています。例えば、

[このページの途中の空白行はわざと入れてあります。ガイドは下に続きます。]

>> では C-v（次の画面を見る）をタイプして次の画面に進んで下さい。

現在いるモードでのキーバインドを調べる

- メニューバーでHelp→Describe→List Key Bindings
- リンクになっているところをクリックすると説明文が出る
- qキーで抜ける
- メニューバーの項目の横に書いてあるキーバインドも参考に

File Edit Options Buffers Tools Help-Mode Development Help

```

1 ; SLIME 2.20
2 CL-USER> (/ 1 0)
3 ; Evaluation aborted on #<DIVISION-BY-ZERO {10037687C3}>.
4 CL-USER> (/ 1 0)
5 ; Evaluation aborted on #<DIVISION-BY-ZERO {1003B33A83}>.
6 CL-USER> []

```

* 172 U: *slime-repl sbcl* REPL company Paredit Portacle adoc

6 : 9 All

```

1248 C-x o      other-window
1249 C-x q      kbd-macro-query
1250 C-x r      Prefix Command
1251 C-x s      save-some-buffers
1252 C-x u      undo
1253 C-x v      vc-prefix-map
1254 C-x z      repeat
1255 C-x {      shrink-window-horizontally
1256 C-x }      enlarge-window-horizontally
1257 C-x DEL    backward-kill-sentence
1258 C-x C-SPC  pop-global-mark
1259 C-x C-+    text-scale-adjust
1260 C-x C--    text-scale-adjust
1261 C-x C-0    text-scale-adjust
1262 C-x C-;    comment-line
1263 C-x C-=    text-scale-adjust
1264 C-x <C-left> previous-buffer
1265 C-x <C-right> next-buffer
1266 C-x <left>  previous-buffer

```

% 53k U: *Help* Help company Portacle

1257 : 29 59%

Lispモード (1)

- .lisp拡張子のファイルを開くと自動的にLispモードになる
- 自動インデント
 - **Enter**を押すと改行と同時に自動的にインデントされる
 - インデントが崩れたときは**Tab**を押す
- ParEdit: 括弧の対応を自動で取ったりするEmacsの拡張 (多少クセがある)
 - 参考: [ParEditチュートリアル](#)
- キーワード補完、引数補完

Lispモード (2)

- .lispファイルを編集集中に式を評価できる
 - 評価したい式の閉じ括弧をハイライトさせた状態で **C-x C-e**
 - 評価結果が画面一番下のモードラインに表示される
 - または領域を選択した状態で **C-c C-r**
 - 領域内の式をまとめて評価できる
- REPLでは **M-p** または **C-↑** で過去の入力を再び出せる

Lispモード、REPLでの補完

- キーワード補完

- キーワードの途中まで入力して一瞬待つと候補ウィンドウが出る
- 候補が一つだと薄く表示されるので **Tab** で確定する

- 引数補完

- 括弧の先頭に関数を書くと、その関数を取るべき引数のリストが表示される

```
232 ; No value  
233 CL-USER> (print)
```

```
print  
print-backtrace  
print-not-readable  
print-not-readable-object  
print-object  
print-output-to-string  
print-symbol-with-prefix  
print-type  
print-type-specifier  
print-unreadable-object
```

printまで打ち込むと
補完ウィンドウが出る

print関数を取るべき引数のリスト

ParEdit: Lispモードでの編集補助

- 開き括弧を入力すると自動で閉じ括弧も入力される。常に括弧の数がバランスされるようになっている
- 括弧の前で **C-k** : 括弧単位でのカット。 **C-y** でペーストできる
- **C-→**、 **C-←** : 飲み込みと吐き出し
- **M-s**、 **M-↑** : 外のレベルの括弧を削除、自分を除く外のレベルの内容も削除

デバッグからの抜け方

- ゼロ除算を評価してみてもわざとエラーを出す
- エラーの内容、リスタート方法の選択、バックトレースの情報が出る
- デバッグを抜けるにはqキー

```

1 arithmetic error DIVISION-BY-ZERO signalled
2 Operation was (/ 1 0).
3 [Condition of type DIVISION-BY-ZERO]
4

```

エラーメッセージ

```
5 Restarts:
```

```

6 0: [RETRY] Retry SLIME REPL evaluation request.
7 1: [*ABORT] Return to SLIME's top level.
8 2: [ABORT] abort thread (#<THREAD "repl-thread" RUNNING {100240FFA3}>)
9

```

リスタート方法の選択

```
10 Backtrace:
```

```

11 0: (SB-KERNEL::INTEGER-/-INTEGER 1 0)
12 1: (/ 1 0)
13 2: (SB-INT:SIMPLE-EVAL-IN-LEXENV (/ 1 0) #<NULL-LEXENV>)
14 3: (EVAL (/ 1 0))
15 --more--

```

バックトレース:
moreをクリックすると
より以前まで遡れる

```
% 435 U: *sldb sbcl/1* sldb[1] company Portacle adoc
```

```
11 : 0 All
```

```

1 ; SLIME 2.20
2 CL-USER> (/ 1 0)
3 ; Evaluation aborted on #<DIVISION-BY-ZERO {10037687C3}>.
4 CL-USER> (/ 1 0)
5

```

ゼロ除算を起こしてエラーを起こすと、
上の画面が出てくる

```
* 105 U: *slime-repl sbcl* REPL company Paredit Portacle adoc
```

```
5 : 0 All
```

インスペクタの使い方

- REPLでの返り値の上で右クリック→inspectを選ぶとインスペクタが起動する
- マウスで要素をクリックして要素の詳細を見ることがができる
- lキーで前の要素に戻り、nキーで次の要素に移動
- qキーでインスペクタを閉じる

File Edit Options Buffers Tools Presentations Development Help

```

1 #<(SIMPLE-ARRAY CHARACTER (13)) {1003CCC45F}>
2 -----
3 Dimensions: (13)
4 Element type: CHARACTER
5 Total size: 13
6 Adjustable: NIL
7 Fill pointer: NIL
8 Contents:
9 0: #\H
10 1: #\e
11 2: #\l
12 3: #\l
13 4: #\o
14 5: #\,
15 6: #\
16 7: #\w
17 8: #\o
18 9: #\r
19 10: #\l

```

オブジェクトの情報が出る

文字列は文字の配列でしかないことが分かる

配列の要素が列挙されている

これもクリックできる

```
% 261 U: *slime-inspector* Slime-Inspector company {COMMON-LISP-USER sbcl} Portacle 3 : 0 Top
```

```

1 ; SLIME 2.20
2 CL-USER> (/ 1 0)
3 ; Evaluation aborted on #<DIVISION-BY-ZERO {10037687C3}>.
4 CL-USER> (/ 1 0)
5 ; Evaluation aborted on #<DIVISION-BY-ZERO {1003B33A83}>.
6 CL-USER> "Hello, world!"
7 "Hello, world!"
8 CL-USER> 

```

文字列”Hello, world!”を評価した結果を右クリックして inspectを選ぶ

```
* 213 U: *slime-repl sbcl* REPL company Paredit Portacle adoc 8 : 9 All
```