

클로저 실행환경 설정하기

사실 설치 환경은 계속해서 변하기 때문에, 여기서 소개하는 방법도 특정 시점에는 유효하지 않거나, 최선이 아닐 수 있다. 따라서 이 문서의 내용을 참조는 하되, 클로저 위키의 Getting Started 페이지를 직접 보는 것이 가장 좋다. 웹 페이지 주소는 http://www.assembla.com/wiki/show/clojure/Getting_Started나 http://en.wikibooks.org/wiki/Clojure_Programming/Getting_Started다.

설치가 귀찮다면

설치가 귀찮고 바로 REPL을 띄워 책의 예제를 따라해 보고 싶다면 TryClojure(<http://www.try-clojure.org/>)를 이용하면 된다. 웹 브라우저를 통해 REPL을 띄울 수 있다. 간편한 방법이긴 하지만, 줄 바꿈이나 괄호 매칭 등이 되지 않기 때문에, 그다지 권하고 싶지는 않다.

윈도에서 설정하기

윈도 유저라면 아마도 윈클릭 인스톨러를 찾고 있을 것이다. 이 경우에 Clojure Box(<http://clojure.bighugh.com/>)를 설치하는 것이 가장 빠르다. 자바 1.5 이상이 설치되어 있다면, Clojure Box를 내려받아 더블클릭하면 끝이다. 설치가 끝난 후 실행하면 바로 REPL을 띄울 수 있고, Clojure Contrib 등의 주요 라이브러리도 모두 패키지 안에 포함되어 있다. emacs 편집기에 익숙하지 않다면, Ctrl-h t를 입력해서 튜토리얼을 따라해 보도록 하자(Ctrl을 누른 채로 h를 누르고 emacs 창 아래에 C-h 표시가 뜨면 그 후에 t를 누르라는 뜻이다).

맥과 리눅스에서 설정하기

빌드 환경은 leiningen(<http://github.com/technomancy/leiningen>)을, 편집기는 emacs(<http://github.com/technomancy/emacs-starter-kit>)를 쓰는 것이 좋다고 본다. 일단 leiningen 페이지의 installation 절차를 따라 leiningen을 설치한 후, 터미널에서 lein repl 을 쳐 보면 바로 repl을 띄울 수 있다. 이제 좀더 편하게 작업하려면 emacs와 repl을 연결시킬 필요가 있는데 그 절차는 다음과 같다(http://www.assembla.com/wiki/show/clojure/Getting_Started_with_Emacs 참고).

1. emacs 설치하기

emacs-starter-kit 페이지(<http://github.com/technomancy/emacs-starter-kit>)의 installation 절차를 따라 emacs를 설치한다. emacs 편집기에 익숙하지 않다면, Ctrl-h t를 입력해서 튜토리얼을 따라해 보자(Ctrl을 누른 채로 h를 누르고 emacs 창 아래에 C-h 표시가 뜨면 그 후에 t를 누르라는 뜻이다).

2. emacs lisp package archive 설치하기

<http://tromeey.com/elpa/install.html>에서 설명하는 절차를 따라서 emacs lisp package archive(ELPA)를 설치한다.

3. clojure-mode와 swank-clojure 설치하기

emacs에서 M-x package-list-packages라고 입력하면(M은 meta 키로 보통 alt 키를 말한다. M-x package-list-packages는 alt를 누른 채로 x를 누르고, emacs 창 아래에 M-x 표시가 뜨면 그 후에 package-list-packages를 입력하라는 뜻이다), 패키지들의 리스트가 뜬다. 방향키로 커서를 움직인 후 i 키를 눌러 clojure-mode와 slime-repl 패키지를 선택하자. 그리고 x를 누르면 선택한 패키지가 설치된다.

4. inferior-lisp-program variable 설정하기

emacs에서 M-x describe-variable 입력 후, 다시 inferior-lisp-program 이라고 치면, customize를 클릭해서 이 variable을 수정할 수 있다고 나온다. customize를 클릭 후, Inferior Lisp Program의 값을 lein repl로 바꾼 후 Save for future sessions를 눌러 저장하자.

5. emacs에서 repl 띄우기

emacs에서 M-x inferior-lisp이라고 입력하면 repl이 뜨는 것을 볼 수 있다. leiningen은 관련 라이브러리 설치, 클래스 패스 문제, jar 파일 생성 등을 한꺼번에 해결해 주는 굉장히 편리한 도구다. 튜토리얼 페이지(<http://github.com/technomancy/leiningen/blob/master/TUTORIAL.md>)를 꼭 읽어 보기 바란다. 그리고 slime은 lisp과 emacs를 이어주는 도구라고 할 수 있는데, 역시 관련 명령어들을 알아두면 굉장히 편해진다. slime 홈페이지(<http://common-lisp.net/project/slime/>)의 매뉴얼이나 스크린 캐스트 등을 살펴보자.

위키 페이지(http://www.assembla.com/wiki/show/clojure/Getting_Started)를 보면 알겠지만 Eclipse나 Vim을 이용하는 것 역시 가능하다. 다만 전통적으로 lisp 커뮤니티가 emacs를 선호하는 경향이 있기 때문에 여기에서는 emacs 위주로 소개하였다. 하지만 편집기나 개발 환경은 취향의 문제이기 때문에 위키 페이지를 보고 자신에게 친숙한 환경을 선택하면 된다. 실제로 유명한 lisp 해커인 폴 그레이엄은 vi를 사용한다.