

## Class 18: Clojure (1)

**Held:** Friday, 4 March 2011

**Summary:** We begin our exploration of Clojure, yet another dialect of Lisp.

**Related Pages:**

- EBoard.
- Lab: Clojure (1).
- Reading: Tate 7.1-7.2.

**Notes:**

- Assignment 7 will be distributed late Saturday night.
- No CS table or CS extras next week.

**Overview:**

- Why JVM?
- Clojure vs. Scheme.
- A Challenge.
- Lab.

## Why JVM?

- One of you asked “Why build Erlang and Clojure on top of the JVM?”
  - Implicit: And on top of the CLR.
- I thought the answer was obvious, but the question was asked (and not answered), so it seems useful to cover in class.
- How many reasons can you come up with?

## Clojure vs. Scheme

- What do you see as the big differences between Clojure and Scheme? (At least given what you’ve learned about Clojure.)
  - Syntax
  - Data types
  - Functions provided
  - Changes in function operation
  - Other
- Why do you think the designer of Clojure made those decisions?

## Challenge

- What's the most concise code you can write for `(in-order nums)`, that holds if `nums` is in order from smallest to largest.
- You may assume that `nums` is a list of numbers.

## Lab

- Do The lab.
- Be prepared to reflect.