

---

# CLOJURE & 3 DAVES

Jay Fields - DRW Trading





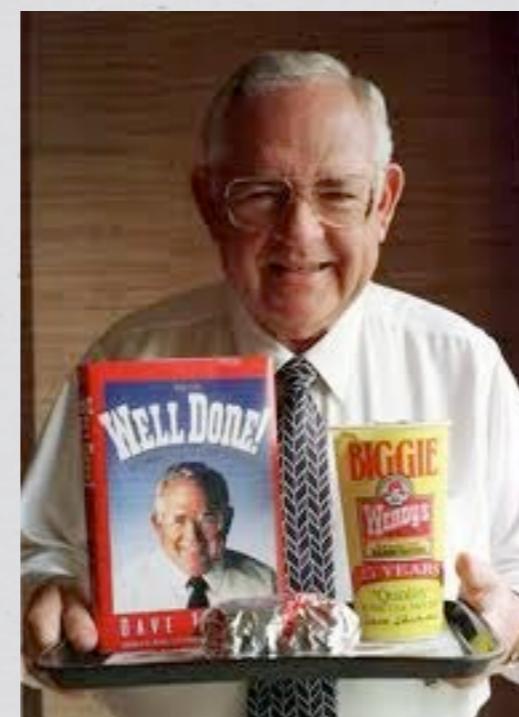
**“KLOCS kill”**

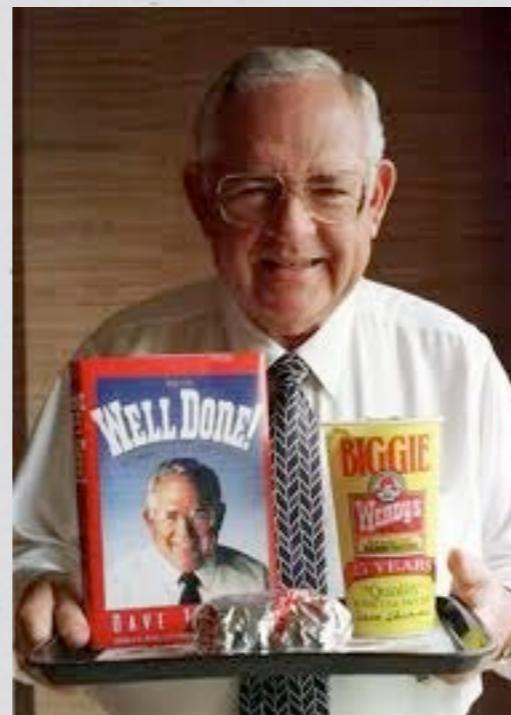
**-- Dave Thomas**



**“KLOCS kill”**

**-- Dave Thomas**





*“It was the closest to purgatory that I've ever experienced while I've been living.”*

-- Dave Thomas



**“KLOCS kill”**

**-- Dave Thomas**

```
new File("/tmp/file.txt").exists()
```

```
// File.exists("/tmp/file.txt") is impossible to define
```

# **Data & Behavior are Separated**

```
new File("/tmp/file.txt").exists()
```

```
// File.exists("/tmp/file.txt") is impossible to define
```

```
// Returns the sum of the elements of a
int sum(int[] a) {
    int result = 0;
    for (int i : a)
        result += i;
    return result;
}
// no closures, thus no collection
closure methods
```

# (reduce + 0 ints)

```
// Returns the sum of the elements of a
int sum(int[] a) {
    int result = 0;
    for (int i : a)
        result += i;
    return result;
}
// no closures, thus no collection
closure methods
```

```
HashMap<String, Double> hm =  
    new HashMap<String, Double>();  
  
hm.put( "A" , new Double( 3434.34 ) );  
hm.put( "B" , new Double( 123.22 ) );  
hm.put( "C" , new Double( 1378.00 ) );  
hm.put( "D" , new Double( 99.22 ) );  
hm.put( "E" , new Double( -19.08 ) );
```

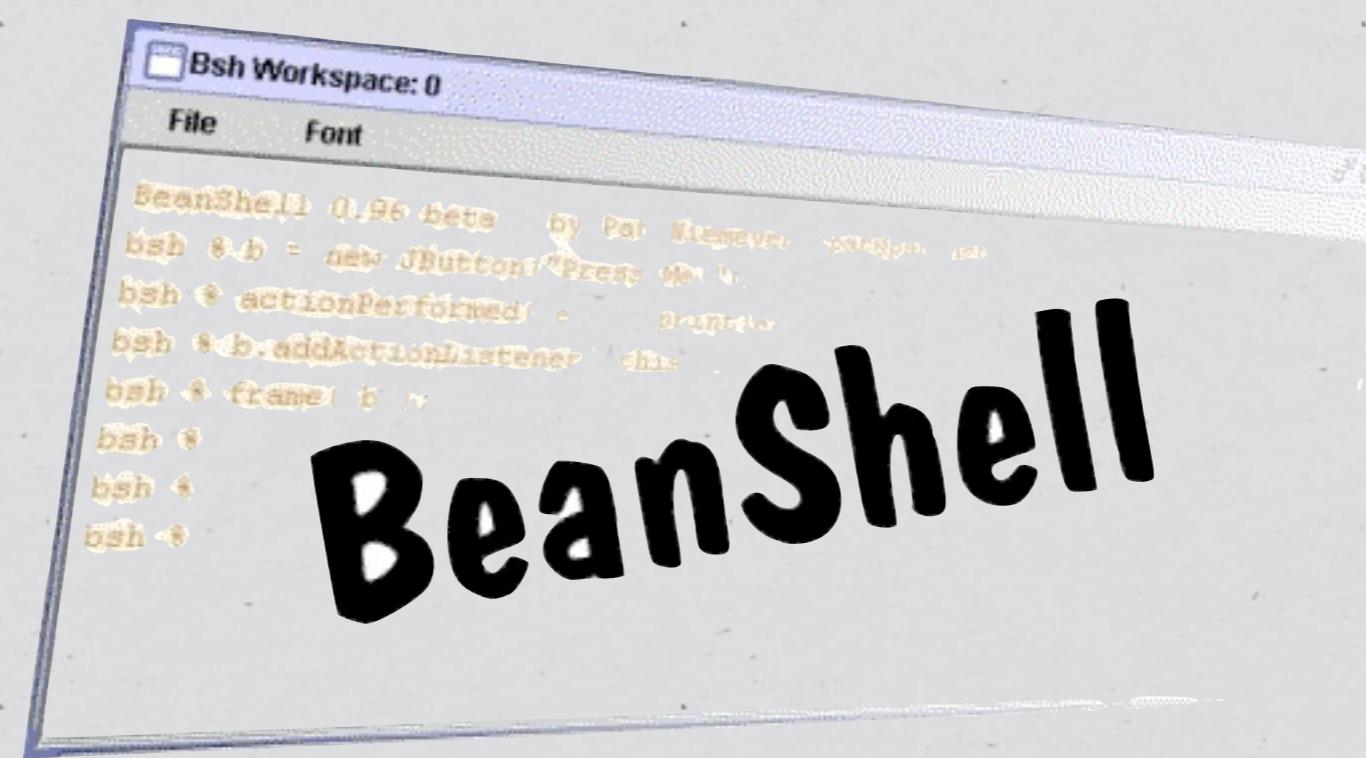
**{"A" 3434.34 "B" 123.22  
"C" 1378 "D" 99.22 "E" -19.08}**

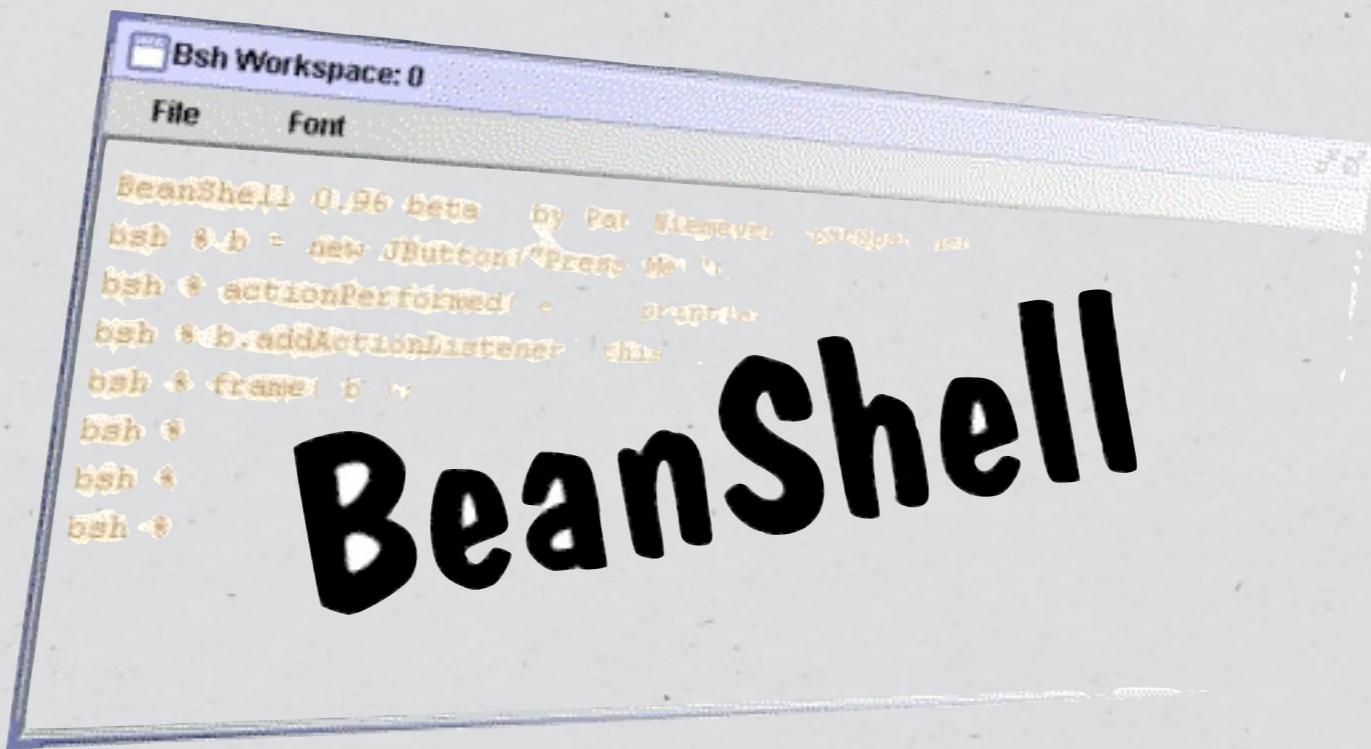
```
HashMap<String, Double> hm =  
    new HashMap<String, Double>();  
  
hm.put("A", new Double(3434.34));  
hm.put("B", new Double(123.22));  
hm.put("C", new Double(1378.00));  
hm.put("D", new Double(99.22));  
hm.put("E", new Double(-19.08));
```

```
package some.klass.some.where; public class ItsJustSomeData extends  
SomeOtherData { private final String name; private final String age; private  
final String birthPlace; public ItsJustSomeData(String name, String age,  
String birthPlace) { this.name = name; this.age = age; this.birthPlace =  
birthPlace; } public String getName() { return name; } public String  
getAge() { return age; } public String getBirthPlace() { return birthPlace; }  
public void setName(String name) { this.name = name; } public void setAge  
(String age) { this.age = age; } public void setBirthPlace(String birthPlace) {  
this.birthPlace = birthPlace; } }
```

**(defrecord Person [name age birth-place])**

```
package some.klass.some.where; public class ItsJustSomeData extends  
SomeOtherData { private final String name; private final String age; private  
final String birthPlace; public ItsJustSomeData(String name, String age,  
String birthPlace) { this.name = name; this.age = age; this.birthPlace =  
birthPlace; } public String getName() { return name; } public String  
getAge() { return age; } public String getBirthPlace() { return birthPlace; }  
public void setName(String name) { this.name = name; } public void setAge  
(String age) { this.age = age; } public void setBirthPlace(String birthPlace) {  
this.birthPlace = birthPlace; } }
```

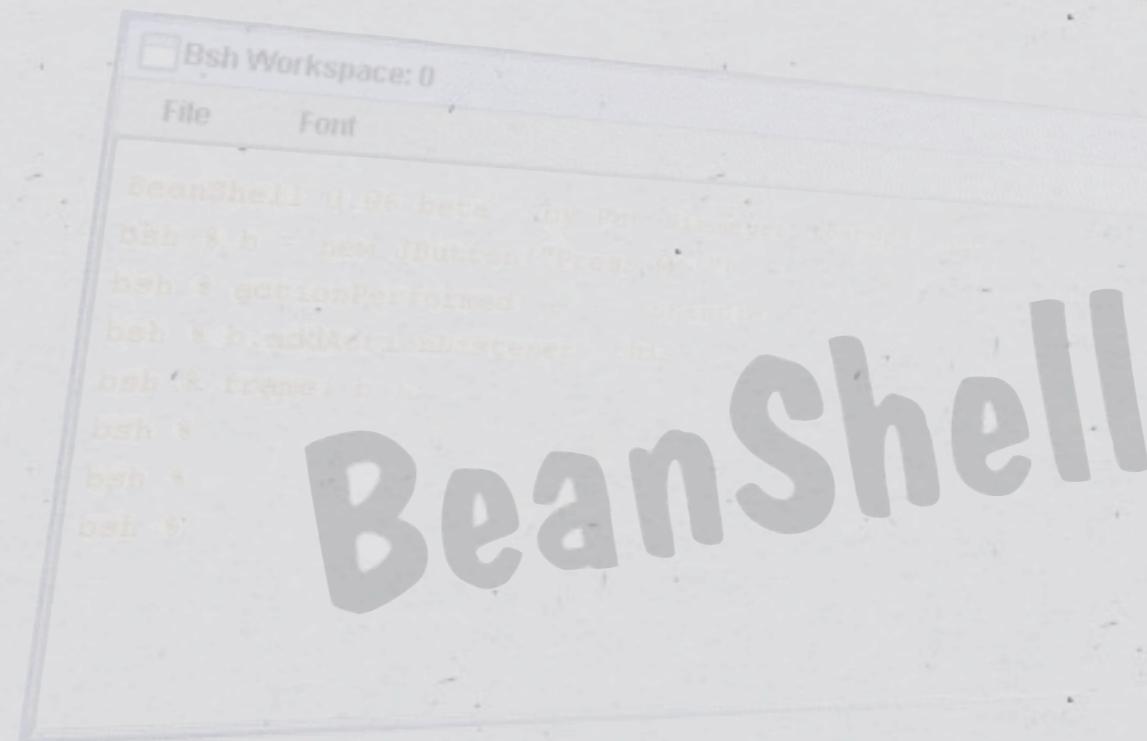




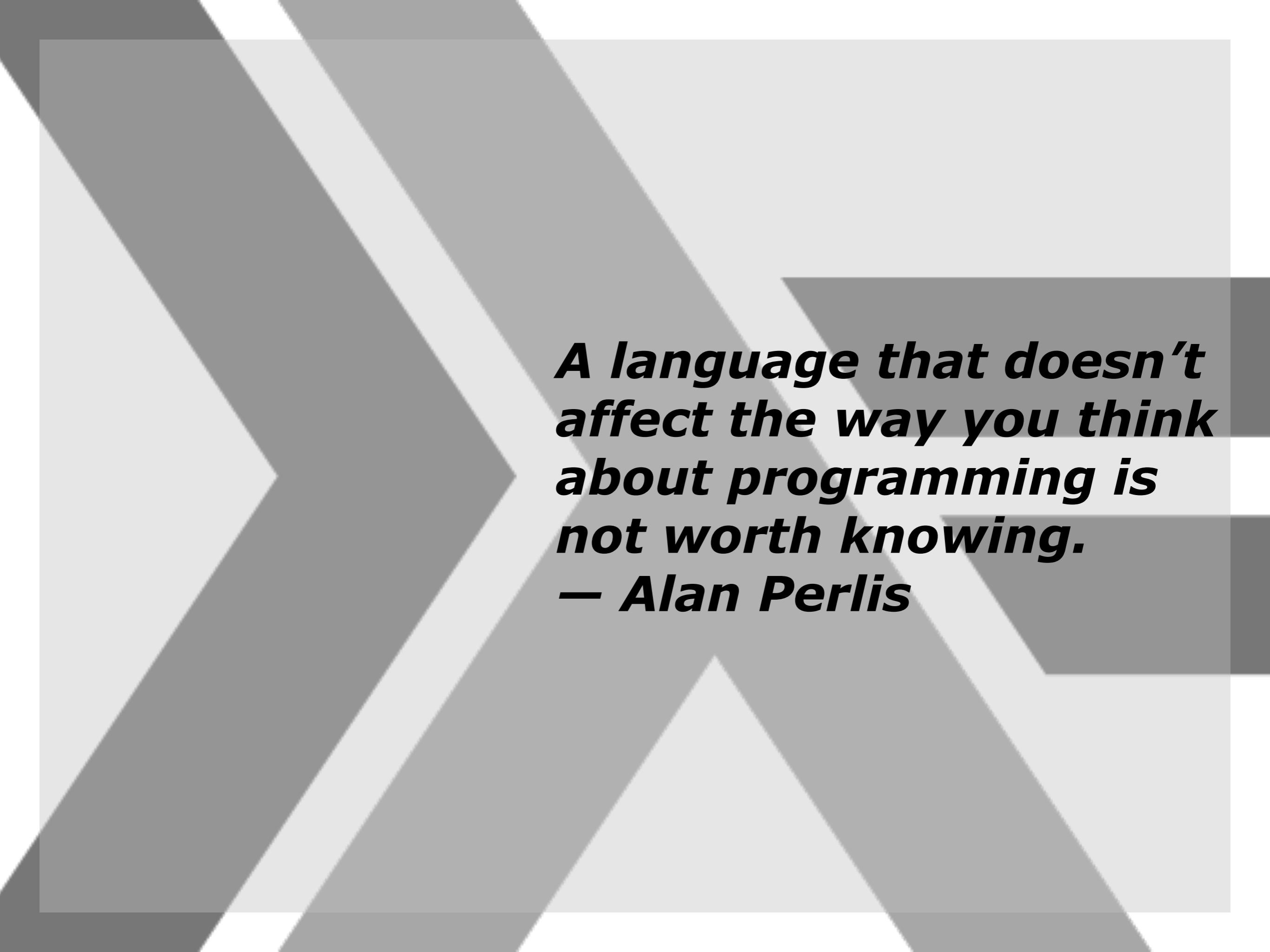
**“Lightweight Scripting For Java”**

```
java -cp clojure.jar clojure.main
```

```
user=> (+ 1 2 3)  
6
```



***A language that doesn't  
affect the way you think  
about programming is  
not worth knowing.  
— Alan Perlis***



***A language that doesn't  
affect the way you think  
about programming is  
not worth knowing.  
— Alan Perlis***





# data & behavior separation

It is better to have 100 functions  
operate on one data structure than 10  
functions on 10 data structures  
- Perlis (again)

## (anecdote)

requirement: When a user sets a default execution type, remove all existing saved execution types

# (general application)

Clojure's fluent interfaces: -> & ->>

**code is data / data is code**

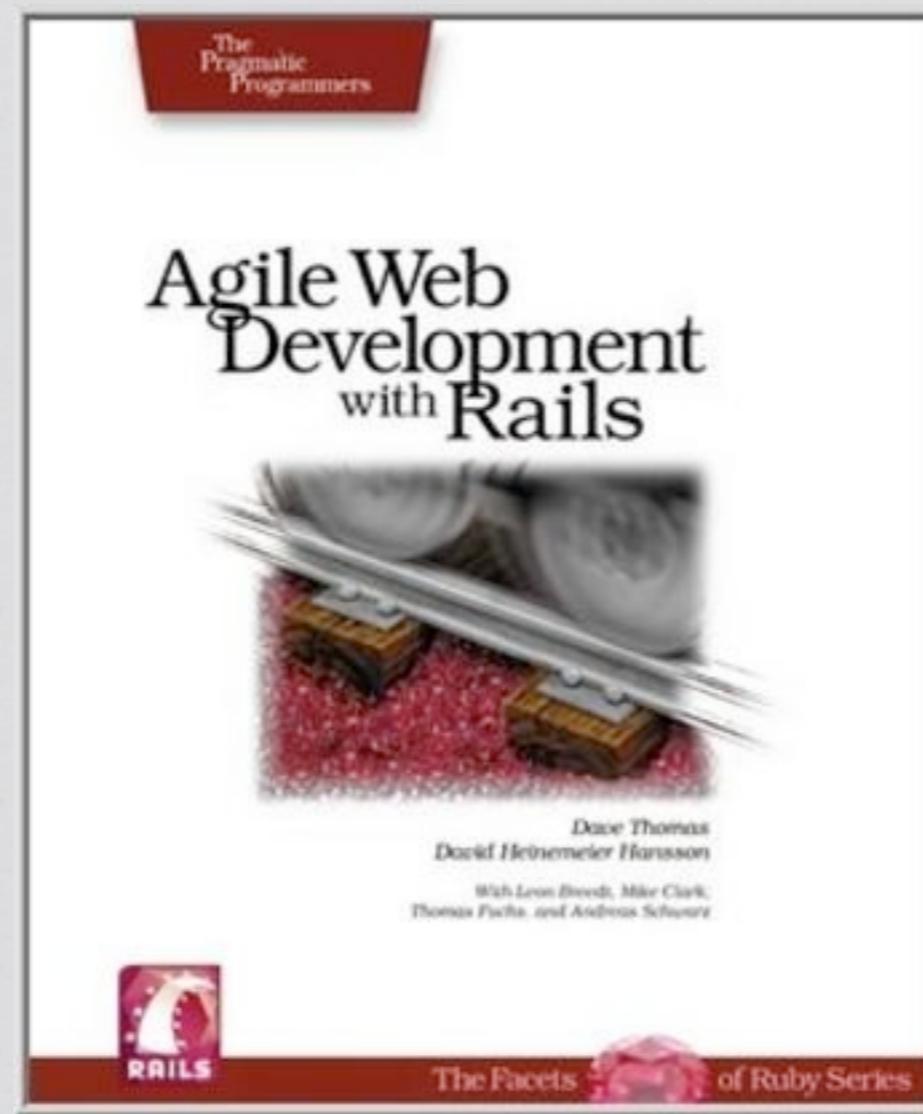
# code is data / data is code

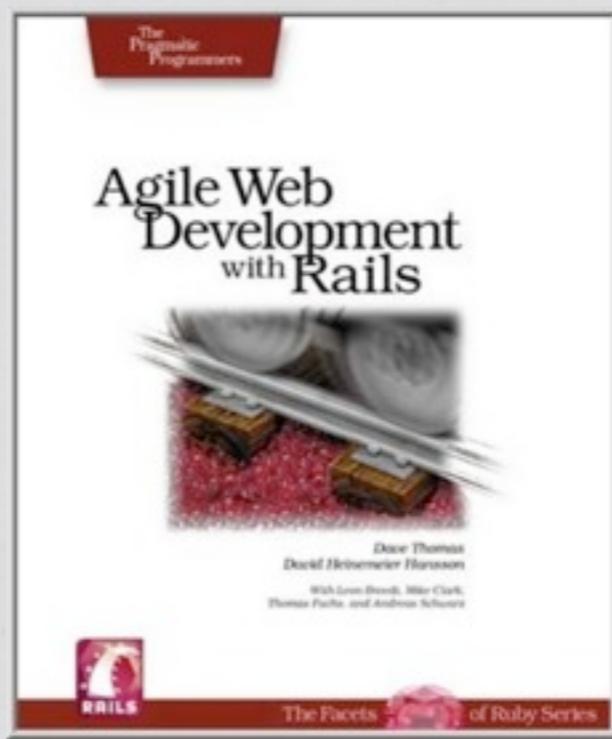
```
(scenario
  (foo 4 4)
  (expect (interaction (foo 1 4)))))

failure in
(scenario_failure_examples.clj:64)
  expected: (foo 1 4) once
  got: (foo 4 4)
```

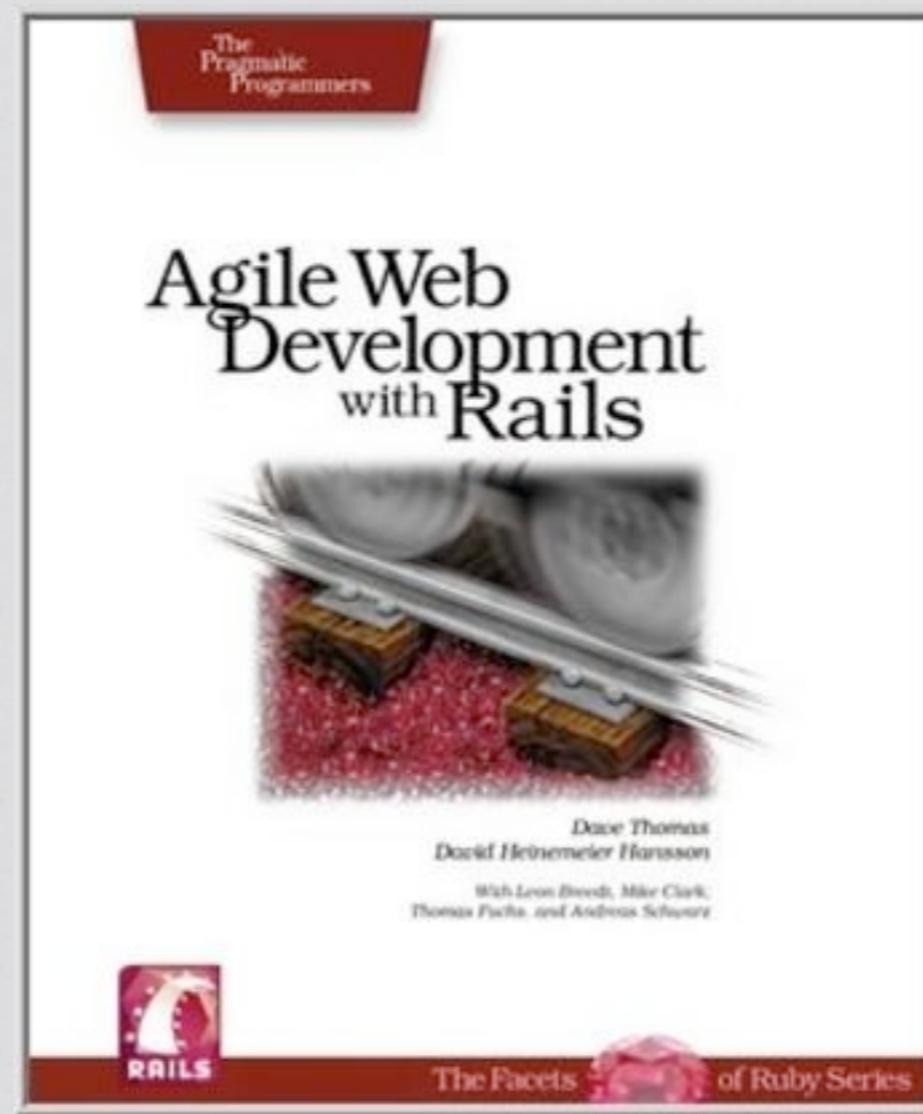
# reload & redefine

# reload & redefine

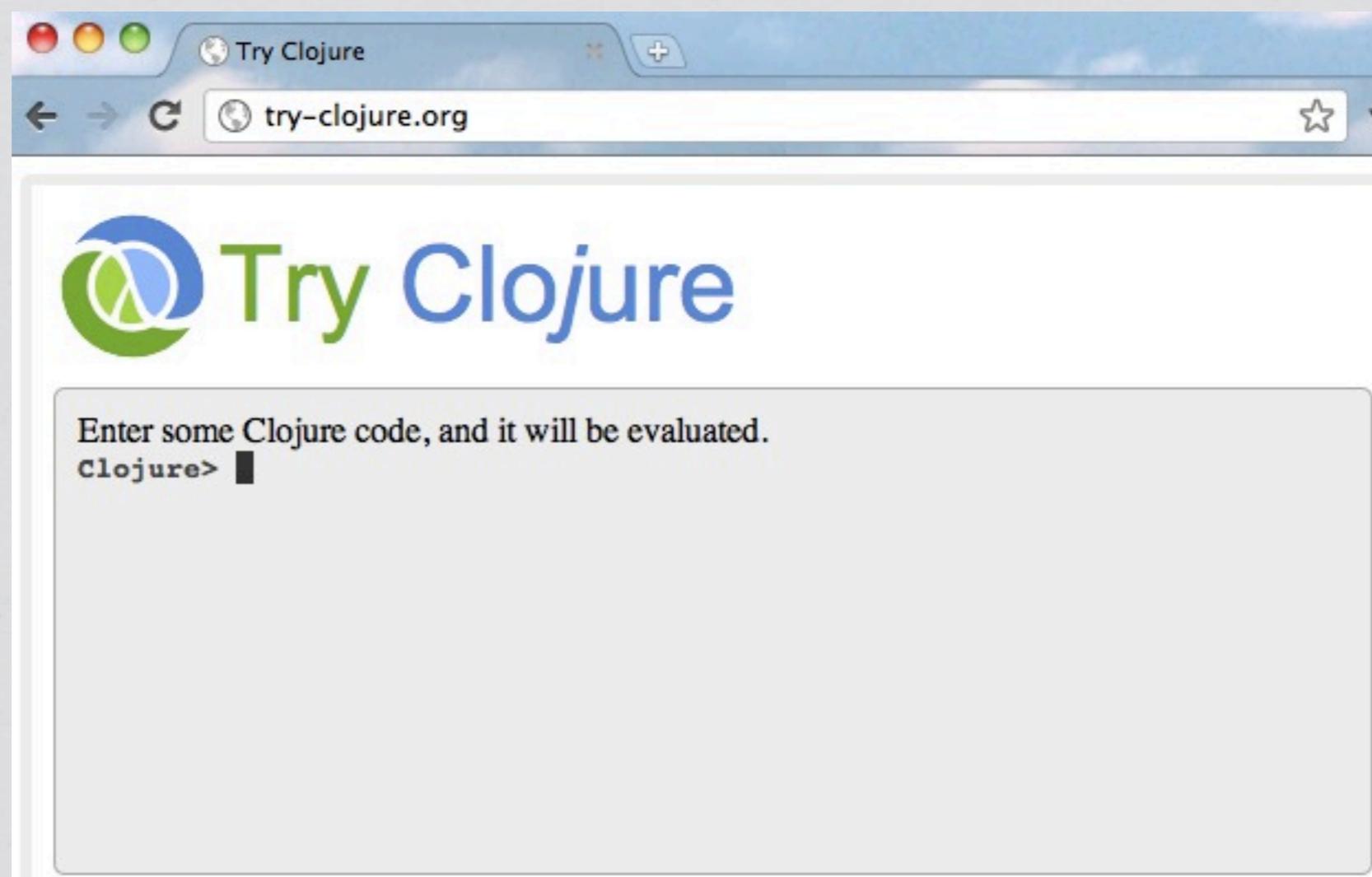




# reload & redefine



# reload & redefine



A nighttime aerial photograph of a city skyline, likely Hong Kong, featuring numerous skyscrapers with illuminated windows. In the foreground, a multi-lane highway or bridge is visible, with the motion of traffic appearing as streaks of light due to a long exposure. The scene is dominated by warm yellow and orange tones from the city lights.

concurrency

# (and)

- ❖ \* collection closure functions
- ❖ \* literal map, set, and list syntax
- ❖ \* low ceremony (bonus: homoiconic)
- ❖ \* REPL
- ❖ \* functional
- ❖ \* fast
- ❖ \* **Rich Hickey**

---

**FINITO**

---