Battle Plan

- Lexical Scope
- Closures
- Unnecessary Wrapping
- Currying
- Partial Application
- References

Lexical Scope

No time travel => predictable behavior.

Variables in function body bound at definition.

What's the alternative? Why would you use it?

How do we get lexical scope to work?
Closures

• How we get lexical scope to work
• When you return a function, really return closure
• Closure has 2 parts:
  • (1) a pointer to a function
  • (2) a pointer to the env where function defined

Unnecessary Wrapping

You wouldn’t write:

```plaintext
if x then true else false
```

So don’t write:

```plaintext
fn x => foo x
```

Currying: Delicious Functions

Currying

• If functions are values, we can return them!
• Functions that return functions that return functions
• Who needs tuples?
Partial Application

• No need to provide all fund args at once
• More flexible, mix and match, function factory
• Closures make it all fall out naturally

References

• Make it easier to copy algorithms from textbooks
• Note we can still provide functional interface
• Break substitution!!! BE CAREFUL