

Summary

- Chapter 01: What Ever Are We Doing?
 - Introductions
 - A Brief Encounter
- Chapter 02: First Class Functions
 - A Quick Review
 - Why Favor First Class?
- Chapter 03: Pure Happiness with Pure Functions
 - Oh to Be Pure Again
 - Side Effects May Include...
 - 8th Grade Math
 - The Case for Purity
 - In Summary
- Chapter 04: Currying
 - Can't Live If Livin' Is without You
 - More Than a Pun / Special Sauce
 - In Summary
 - Exercises
- Chapter 05: Coding by Composing
 - Functional Husbandry
 - Pointfree
 - Debugging
 - Category Theory
 - In Summary
 - Exercises
- Chapter 06: Example Application
 - Declarative Coding
 - A Flickr of Functional Programming
 - A Principled Refactor
 - In Summary
- Chapter 07: Hindley-Milner and Me
 - What's Your Type?
 - Tales from the Cryptic
 - Narrowing the Possibility
 - Free as in Theorem
 - Constraints
 - In Summary
- Chapter 08: Tupperware
 - The Mighty Container
 - My First Functor

- Schrödinger's Maybe
- Use Cases
- Releasing the Value
- Pure Error Handling
- Old McDonald Had Effects...
- Asynchronous Tasks
- A Spot of Theory
- In Summary
- Exercises
- Chapter 09: Monadic Onions
 - Pointy Functor Factory
 - Mixing Metaphors
 - My Chain Hits My Chest
 - Power Trip
 - Theory
 - In Summary
 - Exercises
- Chapter 10: Applicative Functors
 - Applying Applicatives
 - Ships in Bottles
 - Coordination Motivation
 - Bro, Do You Even Lift?
 - Operators
 - Free Can Openers
 - Laws
 - In Summary
 - Exercises
- Chapter 11: Transform Again, Naturally
 - Curse This Nest
 - A Situational Comedy
 - All Natural
 - Principled Type Conversions
 - Feature Envy
 - Isomorphic JavaScript
 - A Broader Definition
 - One Nesting Solution
 - In Summary
 - Exercises
- Chapter 12: Traversing the Stone
 - Types n' Types
 - Type Feng Shui
 - Effect Assortment

- Waltz of the Types
- No Law and Order
- In Summary
- Exercises
- Appendix A: Essential Functions Support
 - always
 - compose
 - curry
 - either
 - identity
 - inspect
 - left
 - liftA*
 - maybe
 - nothing
 - reject
- Appendix B: Algebraic Structures Support
 - Compose
 - Either
 - Identity
 - IO
 - List
 - Map
 - Maybe
 - Task
- Appendix C: Pointfree Utilities
 - add
 - chain
 - concat
 - eq
 - filter
 - flip
 - forEach
 - head
 - intercalate
 - join
 - last
 - map
 - match
 - prop
 - reduce
 - replace

- `safeHead`
- `safeLast`
- `safeProp`
- `sequence`
- `sortBy`
- `split`
- `take`
- `toLowerCase`
- `toString`
- `toUpperCase`
- `traverse`
- `unsafePerformIO`