Maybe's and Either's for worry-free dot-chaining

UtahJS SLC 2017-04

by Seth House @whiteinge

https://github.com/whiteinge/presentations

Maybe and Either?

Dot-chaining (Lodash)

```
var minion_table = _.chain(minion_grains)
    .values()
    .map('grains')
    .filter(x => x.cpuarch != 'sparc')
    .slice(curPageStart, curPageOffset)
    .sortBy('num_cpus', 'num_gpus')
    .map(x => _.pick(x, visibleColumns))
    .groupBy('os')
    .value()
```

Dot-chaining (RxJS)

```
var konamiCode = $(document).keyupAsObservable()
    .map(ev => ev.keyCode)
    .windowWithCount(10, 1)
    .selectMany(x => x.sequenceEqual(konami))
    .filter(equal => equal)
    .subscribe(() => console.log('KONAMI!'));
```

Maybe

- Represents a value.
- ...or not a value.
- Pass it around as though it was a value.
- Resolve it when you care; fall back to a default.

Either

- Represents a success or failure condition.
- Pass it around as though it was a value.
- Resolve it when you care; handle both options.

Other Common Interfaces

- Accumulate validation errors – Validation.
- Abstract away asynchronous tasks – Task.
- …or build your own.

Resources

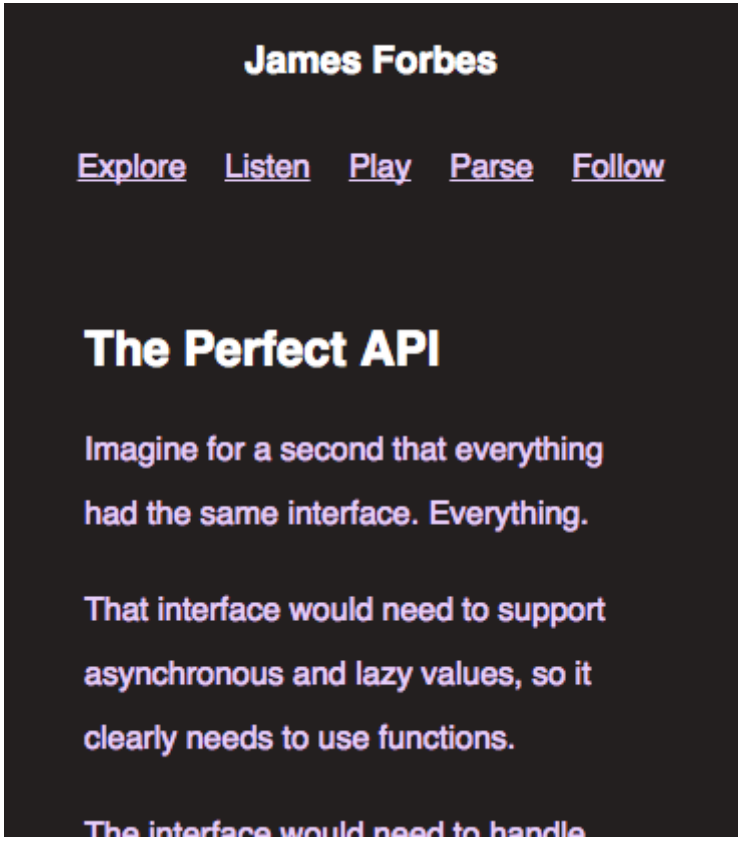
Folktale

http://folktalejs.org/

Professor Frisby Introduces Composable Functional JavaScript

110 minutes

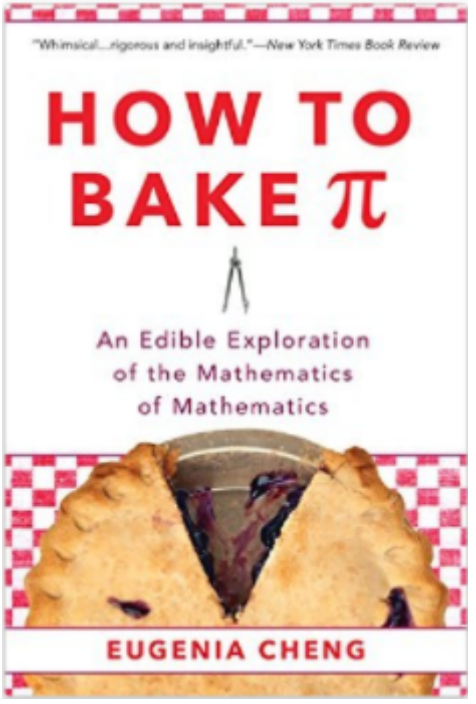
https://egghead.io/courses/professor-frisby-introduces-composable-functional-javascript



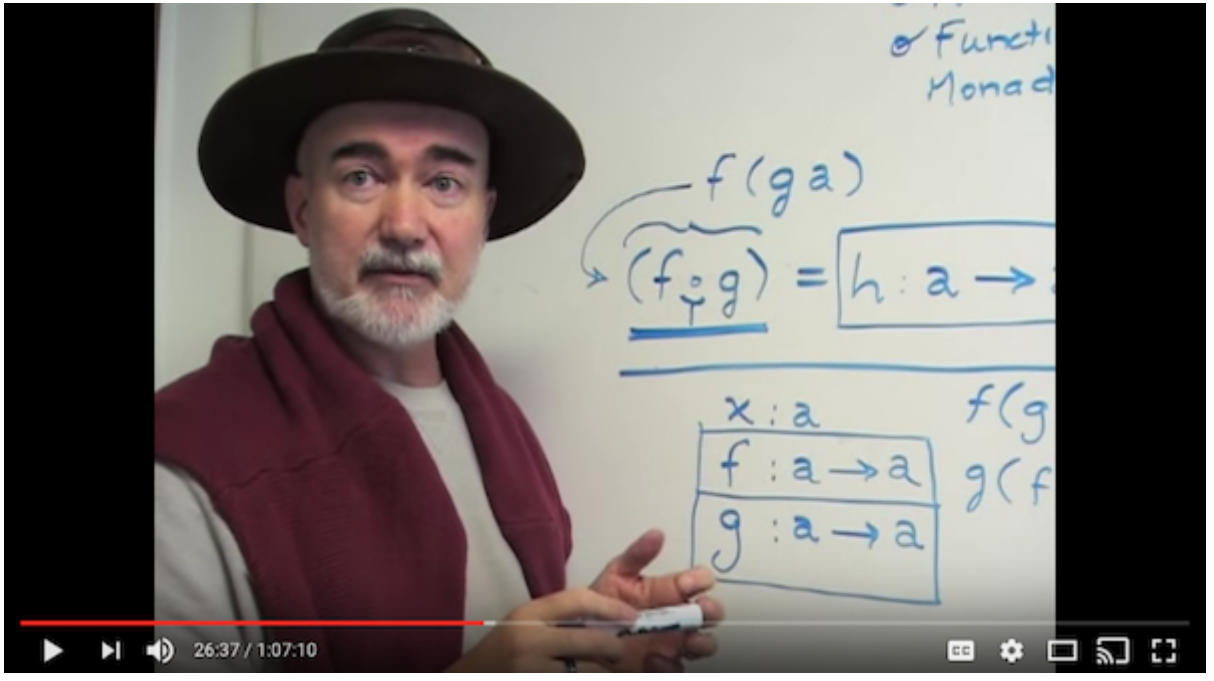
https://james-forbes.com/?/posts/the-perfect-api

Erik Meijer – The essence of interface-based design

https://www.youtube.com/watch?v=JMP6gI5mLHc



http://www.goodreads.com/book/show/23360039-how-to-bake-pi



Brian Beckman: Don't fear the Monad

https://www.youtube.com/watch?v=ZhuHCtR3xq8