Functors in the JavaScript Space

Anto Aravinth 24th April, 2018
About Me!

1. Software Engineer @ PayPal

2. Open source contributor to React, Groovy, Learn Anything

3. Loves JS, Java, Golang, Postgresql

4. Functional Programming In JavaScript
Let’s Begin...
"There is no such thing as a new idea. It is impossible. We simply take a lot of old ideas and put them into a sort of mental kaleidoscope."
Couple Of Questions?

1. How many of you know FP?

2. How many of you know/heard about Functors?
Bit Of Theory On FP

1. FP is all about functions receiving inputs and working on it only

2. FP is about receiving functions and composing them to form abstractions. Ex. map, filter
So, Functors are part of functional programming.

Functor(s) is a datatype that implements map function.
Let’s write a simple Functor.
Implementing Map Function

Map function takes value out of Container, runs the given function and put the result back into Container
Contd. . .

Now, we have our Functor in place

Well, are there any Real world functors?

Array / []
Contd. . .

What if error occurs? Functor(s) do catch them?

Welcome to *Either* Functor

Code *Either*
Wishful Thinking

Most of our lives happens asynchronously.

Functor’s can solve them?
Contd. . .

AsyncBox(data)
  .map(/* some operation */)
  .map(/* some operation 2 */)  
  Should also catch errors like Either

Promise(data)
  .then(/* some operation */)
  .then(/* some operation 2 */)
  .catch(/* do error logic */)
What if...?

Promise.prototype.map = Promise.prototype.then

```javascript
Promise.resolve('Hello')
  .map(str => str + ' GIDS!')
  .then(console.log)
```
Contd . . .

Promise uses then instead of map

Automatically unwraps nested Promises

Is Promise a Functor? Well, yes . . .
Gotchas...

Functor obeys these two laws

Identity, [1, 2, 3].map(x => x)

Composable

```
const f = x => x + 1
const g = x => x * 2

[1, 2, 3].map(x => f(g(x))) // = [3, 5, 7]
[1, 2, 3].map(g).map(f) // = [3, 5, 7]
```
More examples . . .

1. Folktale - `Task`

2. Lodash - `_get`

3. Java – Optional<T>  //yeah JAVA!!
Let’s finish ...
Question & Quote

1. How many of you *used* Functors?

"There is no such thing as a new idea. It is impossible. We simply take a lot of old ideas and put them into a sort of mental kaleidoscope."

**Functors are everywhere around us, we need to change our mental kaleidoscope to see them**
Thank You!