

# Wealthsimple

## Functional Programming With Ruby

Pei Li

[pei@wealthsimple.com](mailto:pei@wealthsimple.com)

[wealthsimple.com](https://wealthsimple.com)  
[@wealthsimple](#)



# Why Functional Programming?

- Concise.

Achieve more with fewer lines of code.

- Predictable.

Easily understand and predict what the code does.

- Testable.

Greater confidence in the correctness of your code.



# Functional Programming...

- Avoids shared states.
- Avoids mutable data.
- Avoids side effects.



# The Fundamentals

- **map**

Applies an operation to every element in a list.

- **select**

Filters a list based on the evaluation of an expression.

- **reduce**

Applies an operation that aggregates every element in a list.



# Map

```
list.map { |element| operation(element) }
```



# Select

```
list.select { |element| expression(element) }
```



# Reduce

```
list.reduce { |aggregation, element| operation(aggregation, element) }
```



# Putting It All Together

## Live Demo





# Resources

Intro to Ruby: <https://www.codecademy.com/tracks/ruby/resume>

Ruby On Rails: [http://guides.rubyonrails.org/getting\\_started.html](http://guides.rubyonrails.org/getting_started.html)

Functional Programming: <http://joelmccracken.github.io/functional-programming-in-ruby>



# Wealthsimple

Questions?

Pei Li

[pei@wealthsimple.com](mailto:pei@wealthsimple.com)

[wealthsimple.com](https://wealthsimple.com)  
[@wealthsimple](#)

