

Intro to Functional Programming

Ray Shih

Hi

- My name is **Ray Shih**
- B.S., M.S. of NTU CSIE
- Worked in WOOMOO
- Work in Mobiusbobs
- A **Fullstack Software Engineer**, including
 - iOS, Android, Web backend/Frontend



Functional Programming is the future



TensorFlow



Redux

Today, I'll teach you
how to programming

What?

I already know how to
programming!

$$x = x + 1$$

WTF is this?

Well then....

Forget it.

Back to
math of high school

Problem 1

$$a = 1$$

$$b = 2$$

$$c = a + b$$

$$c = ?$$

$$b = 2$$

$$c = 1 + b$$

$$c = ?$$

$$c = 1 + 2$$

$$c = ?$$

$$c = 3$$

What are functions?

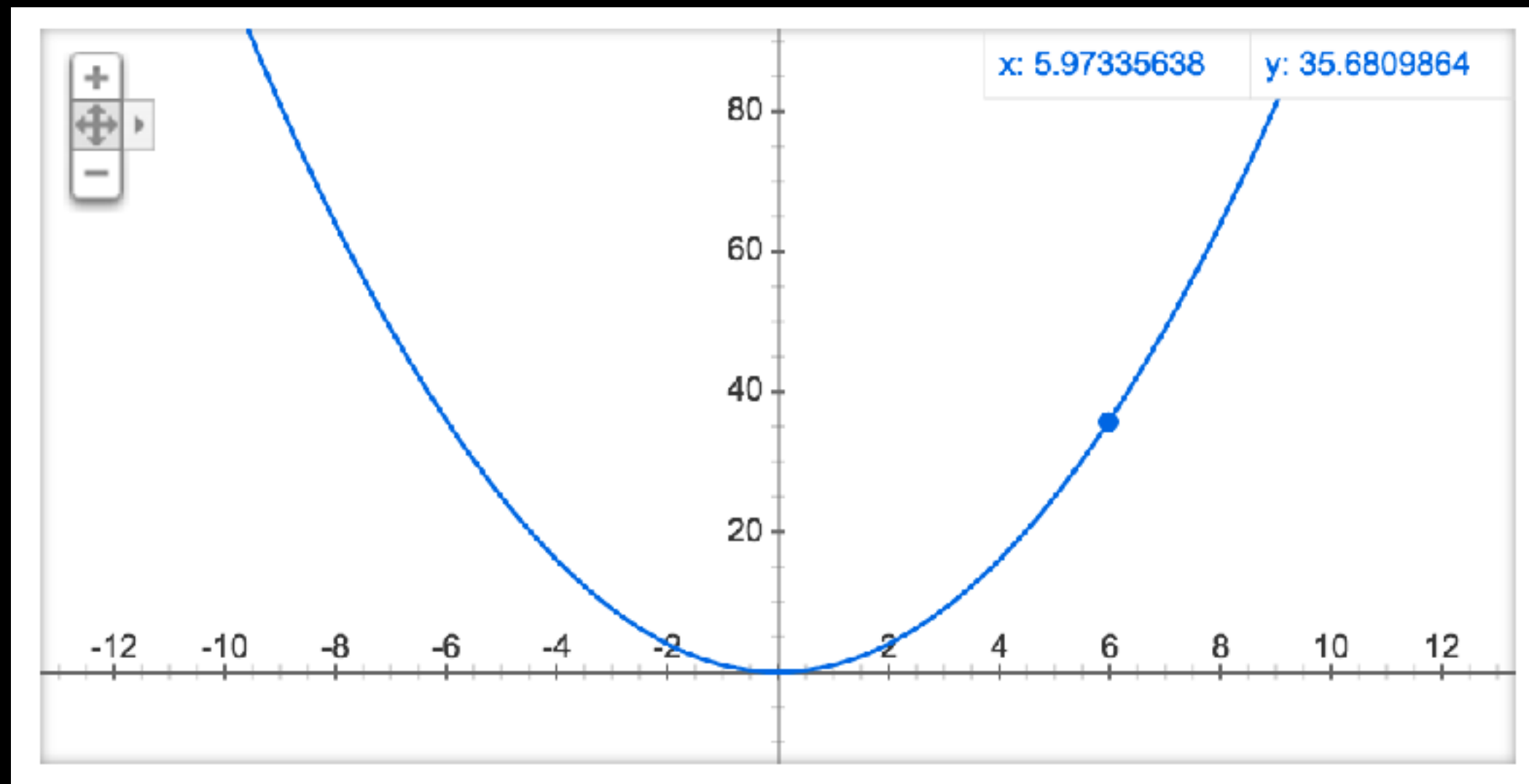
$$y = f(x)$$

$$f(x) = x^2$$

$$y = f(x)$$

$$y = x^2$$

$$y = x^2$$



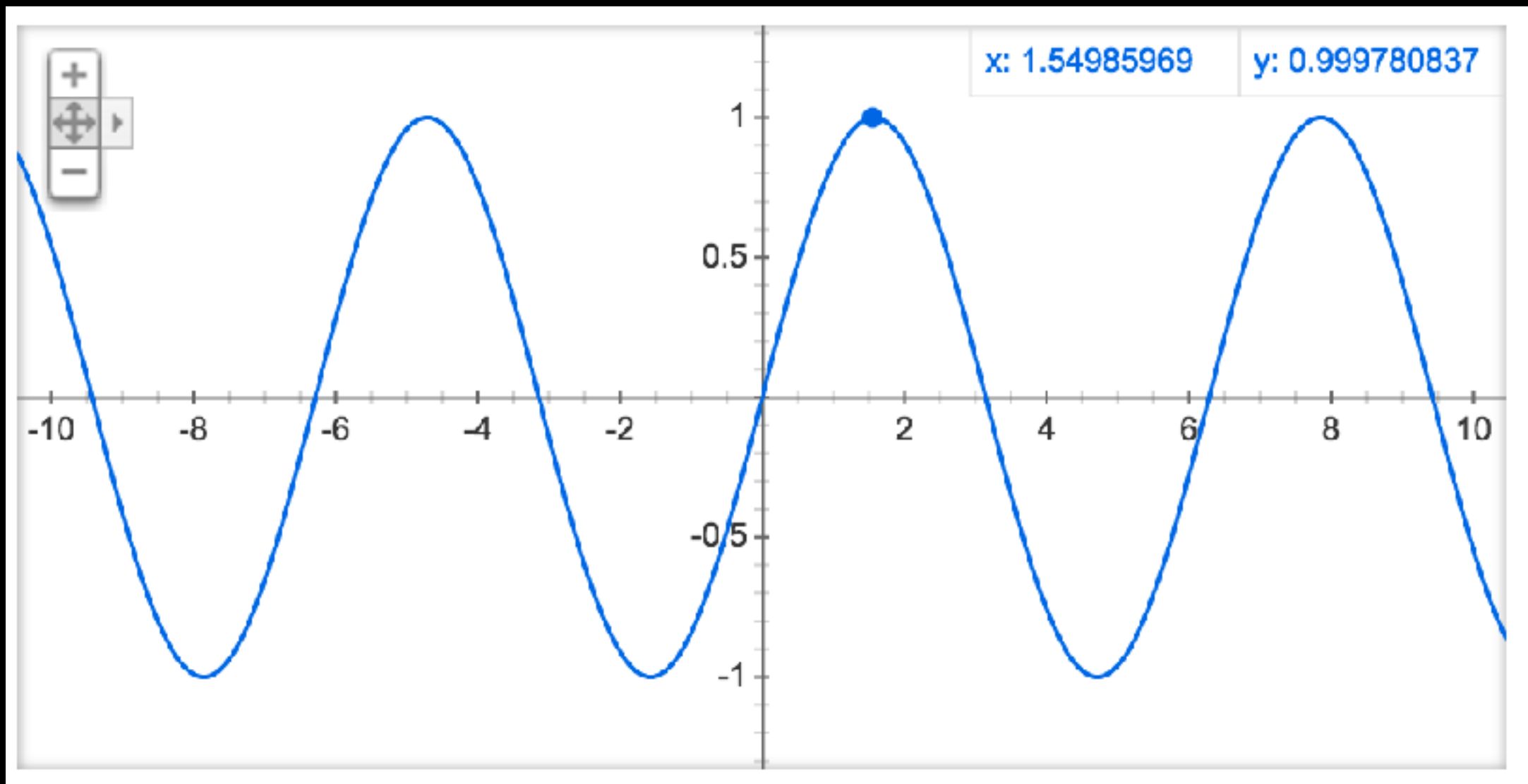
$$y = f(x)$$

$$f(x) = \sin(x)$$

$$y = f(x)$$

$$y = \sin(x)$$

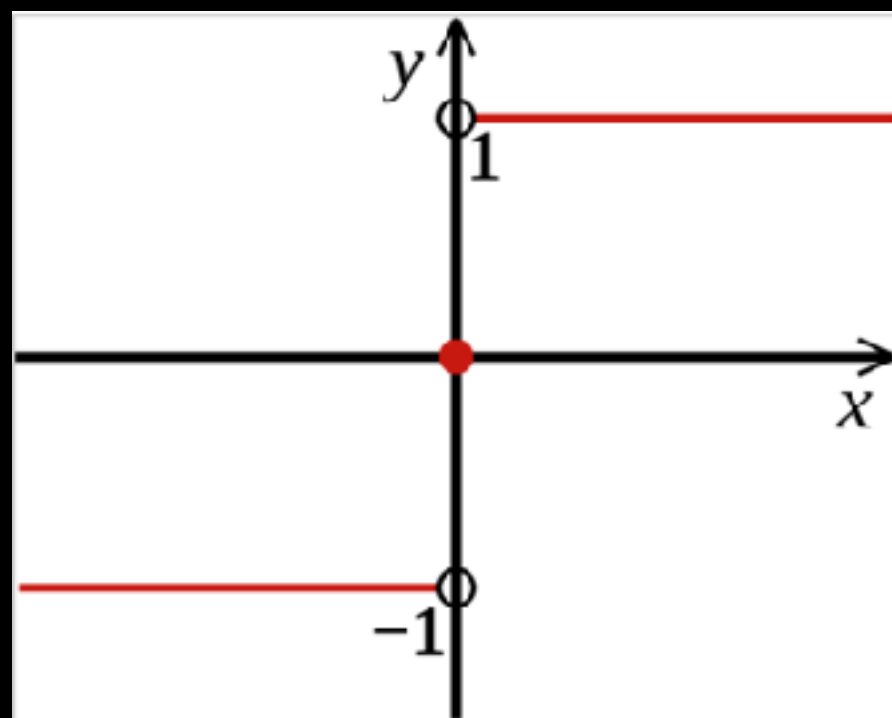
$$y = \sin(x)$$



Conditional function

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

$$y = f(x)$$



$+$ is also a function

+ is also a function

$$\text{add}(a, b) = a + b$$

$$c = \text{add}(1, 2)$$

$$c = a + b$$

$$c = 1 + 2$$

$$c = 3$$

Now you know
what are functions

So where is
programming?

`add(a, b) = a + b`

`c = add(1, 2)`

`c = 3`

add a b = a + b

c = add 1 2

Problem 2

$$1 + 2 + \dots + n$$

$$f(n) = 1 + 2 + \dots + n$$

$$f(4) = 1 + 2 + 3 + 4$$

$$f(4) = 1 + 2 + 3 + 4$$

$$f(3) = 1 + 2 + 3$$

$$f(4) = f(3) + 4$$

$$f(3) = 1 + 2 + 3$$

$$f(2) = 1 + 2$$

$$f(3) = f(2) + 3$$

$$f(n) = f(n-1) + n$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n - 1) + n, & \textit{otherwise} \end{cases}$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n-1) + n, & \textit{otherwise} \end{cases}$$

$$f(5) = f(4) + 5$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n-1) + n, & \textit{otherwise} \end{cases}$$

$$f(5) = f(4) + 5$$

$$f(5) = f(3) + 4 + 5$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n-1) + n, & \textit{otherwise} \end{cases}$$

$$f(5) = f(4) + 5$$

$$f(5) = f(3) + 4 + 5$$

$$f(5) = f(2) + 3 + 4 + 5$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n-1) + n, & \textit{otherwise} \end{cases}$$

$$f(5) = f(4) + 5$$

$$f(5) = f(3) + 4 + 5$$

$$f(5) = f(2) + 3 + 4 + 5$$

$$f(5) = f(1) + 2 + 3 + 4 + 5$$

$$f(n) = \begin{cases} 1, & n == 1 \\ f(n-1) + n, & \textit{otherwise} \end{cases}$$

$$f(5) = f(4) + 5$$

$$f(5) = f(3) + 4 + 5$$

$$f(5) = f(2) + 3 + 4 + 5$$

$$f(5) = f(1) + 2 + 3 + 4 + 5$$

$$f(5) = 1 + 2 + 3 + 4 + 5$$

$$f\ 1 = 1$$

$$f\ n = f\ (n - 1) + n$$

$$f\ 5 = f\ 4 + 5$$

$$f\ 5 = f\ 3 + 4 + 5$$

$$f\ 5 = f\ 1 + 2 + 3 + 4 + 5$$

$$f\ 5 = 1 + 2 + 3 + 4 + 5$$

Problem 3

Fibonacci number

Fibonacci numbers (from wiki)

In mathematical terms, the sequence

$$F_n = F_{n-1} + F_{n-2},$$

with seed values^{[1][2]}

$$F_1 = 1, F_2 = 1$$

or^[5]

$$F_0 = 0, F_1 = 1.$$

Fibonacci numbers

Written in conditional form

$$f(n) = \begin{cases} 0, & n == 0 \\ 1, & n == 1 \\ f(n-1) + f(n-2), & \text{otherwise} \end{cases}$$

$$f(n) = \begin{cases} 0, & n == 0 \\ 1, & n == 1 \\ f(n-1) + f(n-2), & \textit{otherwise} \end{cases}$$

$$f(0) = 0$$

$$f(1) = 1$$

$$f(n) = f(n-1) + f(n-2)$$

$$f(n) = \begin{cases} 0, & n == 0 \\ 1, & n == 1 \\ f(n-1) + f(n-2), & \textit{otherwise} \end{cases}$$

$$f\ 0 = 1$$

$$f\ 1 = 1$$

$$f\ n = f\ (n-1) + f\ (n-2)$$

Problem 4

sum

This is a list
[1, 2, 3, 4, 5]

They are same list

[1, 2, 3, 4, 5]

1:[2, 3, 4, 5]

1:2:3:[4, 5]

1:2:3:4:5:[]

```
x = [1, 2, 3, 4, 5]  
total = sum(x)
```

$$\mathit{sum}(x) = \begin{cases} 0, & x \text{ match } [] \\ n + \mathit{sum}(ns), & x \text{ match } (n : ns) \end{cases}$$

$$\text{sum}(x) = \begin{cases} 0, & x \text{ match } [] \\ n + \text{sum}(ns), & x \text{ match } (n : ns) \end{cases}$$

$$\begin{aligned} \text{sum}([1, 2, 3, 4, 5]) &= \\ 1 + \text{sum}([2, 3, 4, 5]) &= \\ 1 + 2 + \text{sum}([3, 4, 5]) &= \\ 3 + 3 + \text{sum}([4, 5]) &= \\ 6 + 4 + \text{sum}([5]) &= \\ 10 + 5 + \text{sum}([]) &= \\ 15 \end{aligned}$$

$$\textit{sum}(x) = \begin{cases} 0, & x \text{ match } [] \\ n + \textit{sum}(ns), & x \text{ match } (n : ns) \end{cases}$$

$$\textit{sum} \ [] = 0$$

$$\textit{sum} \ (n : ns) = n + \textit{sum} \ ns$$

Problem 5

map

$x = [1, 2, 3]$

$y = \text{doubleAll}(x)$

$$\textit{doubleAll}(x) = \begin{cases} [], & x \text{ match } [] \\ 2n : \textit{doubleAll}(ns), & x \text{ match } (n : ns) \end{cases}$$

`doubleAll [] = []`

`doubleAll (n:ns) =`

`(2 * n) : doubleAll ns`

$$\text{doubleAll}(x) = \begin{cases} [], & x \text{ match } [] \\ 2n : \text{doubleAll}(ns), & x \text{ match } (n : ns) \end{cases}$$

`doubleAll([1, 2, 3]) =`

`2:doubleAll([2, 3]) =`

`2:4:doubleAll([3]) =`

`2:4:6:doubleAll([]) =`

`2:4:6:[] =`

`[2, 4, 6]`

```
x = [1, 2, 3]  
y = incAll(x)
```

$$\mathit{incAll}(x) = \begin{cases} [], & x \text{ match } [] \\ (n + 1) : \mathit{incAll}(ns), & x \text{ match } (n : ns) \end{cases}$$

$$\begin{aligned} \mathit{incAll}([1, 2, 3]) &= \\ 2:\mathit{incAll}([2, 3]) &= \\ 2:3:\mathit{incAll}([3]) &= \\ 2:3:4:\mathit{incAll}([]) &= \\ 2:3:4:[] &= \\ [2, 3, 4] \end{aligned}$$

$$incAll(x) = \begin{cases} [], & x \text{ match } [] \\ (n + 1) : incAll(ns), & x \text{ match } (n : ns) \end{cases}$$

`incAll [] = []`

`incAll (n:ns) =
 (n + 1) : incAll ns`

```
doubleAll [] = []  
doubleAll (n:ns) =  
    (2 * n):doubleAll ns
```

```
incAll [] = []  
incAll (n:ns) =  
    (n + 1):incAll ns
```

`map f [] = []`

`map f (n:ns) =
 (f n) : map f ns`

`inc x = x + 1`

`incAll = map inc`

`map f [] = []`

`map f (n:ns) =
 (f n):map f ns`

`inc x = x + 1`

`incAll [] = map inc []`

`incAll (n:ns) = map inc (n:ns)`

`map f [] = []`

`map f (n:ns) =
 (f n):map f ns`

`inc x = x + 1`

`incAll [] = []`

`incAll (n:ns) =
 map inc (n:ns) =
 (f n):map f ns`

`map f [] = []`

`map f (n:ns) =
 (f n):map f ns`

`inc x = x + 1`

`incAll [] = []`

`incAll (n:ns) =
 map inc (n:ns) =
 (inc n):map inc ns`

`incAll x = map inc x`

`incAll [] = []`

`incAll (n:ns) =
 (n + 1):incAll ns`

`inc x = x + 1`

`incAll [] = []`

`incAll (n:ns) =
 (inc n):incAll ns`

`incAll [] = []`

`incAll (n:ns) =
 (n + 1):incAll ns`

`inc x = x + 1`

`incAll [] = []`

`incAll (n:ns) =
 (n + 1):incAll ns`


```
doubleAll [] = []  
doubleAll (n:ns) =  
  (2 * n):doubleAll ns
```

```
incAll [] = []  
incAll (n:ns) =  
  (n + 1):incAll ns
```

`double x = x * 2`

`doubleAll =
 map double`

`inc x = x + 1`

`incAll = map inc`

Now we have map function!
How about reduce function?

Now we have map function!
How about reduce function?
Exercise! Hint: sum

Problem 6

Sort

list concat

$[1, 2, 3] ++ [4, 5, 6]$

$[1, 2, 3, 4, 5, 6]$

`inc x = x + 1`

`incAll = map inc`

`inc x = x + 1`

`inc = (+ 1)`

`inc 3 =`

`(+ 1) 3 =`

`3 + 1 =`

`4`


```
incAll = map (+ 1)
```

where clause

$\text{inc } n = n + a$

where $a = 1$

sort [] = []

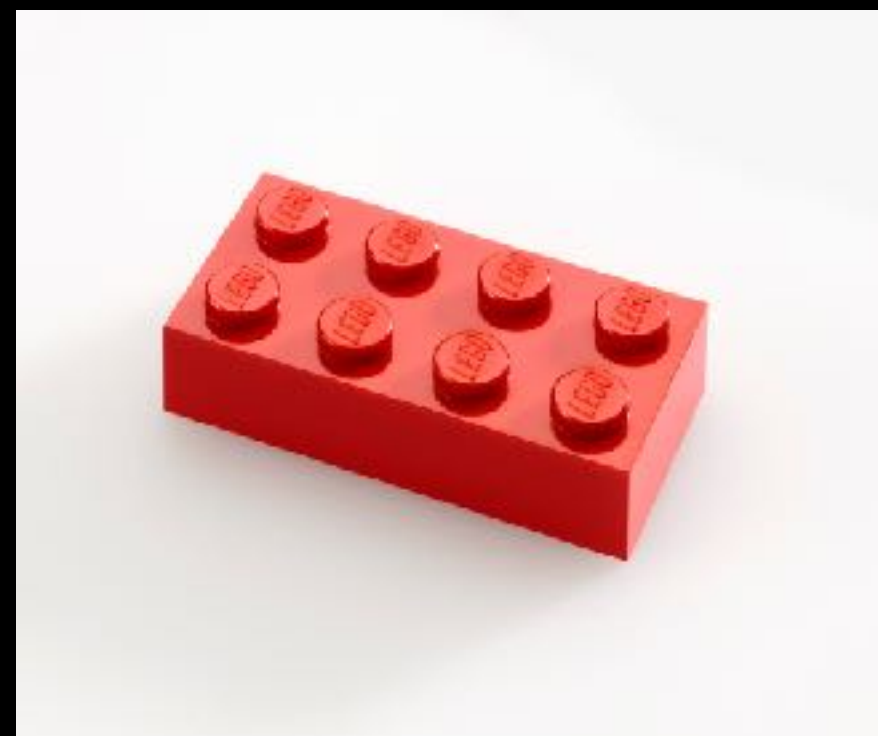
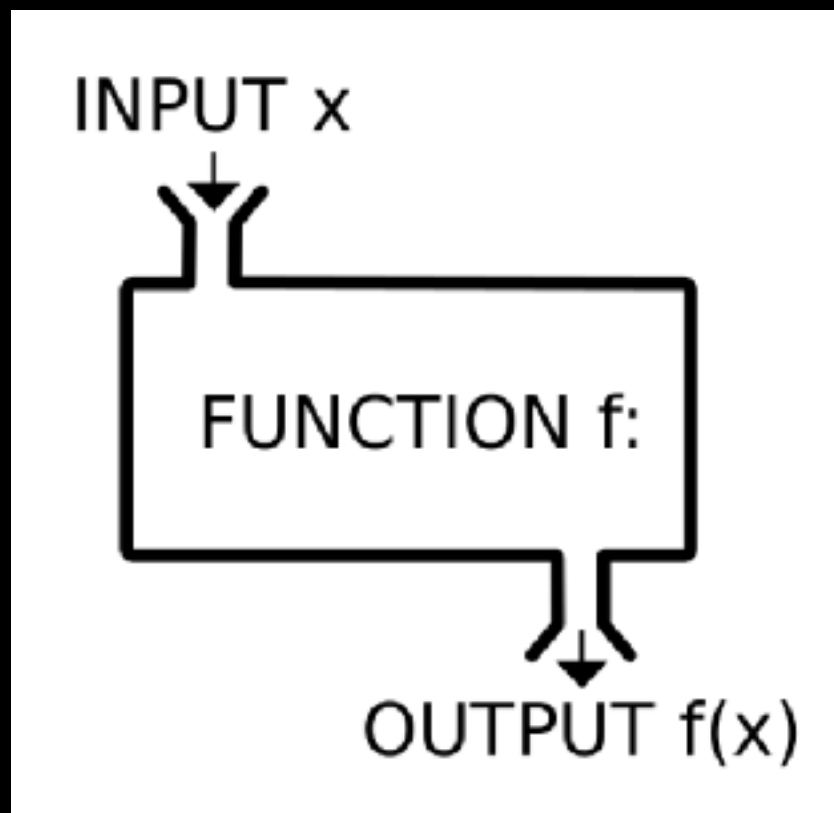
```
sort [] = []
sort (n:ns) =
    sort smaller ++ [n] ++ sort larger
where
    smaller = filter (< n) ns
    larger  = filter (=> n) ns
```

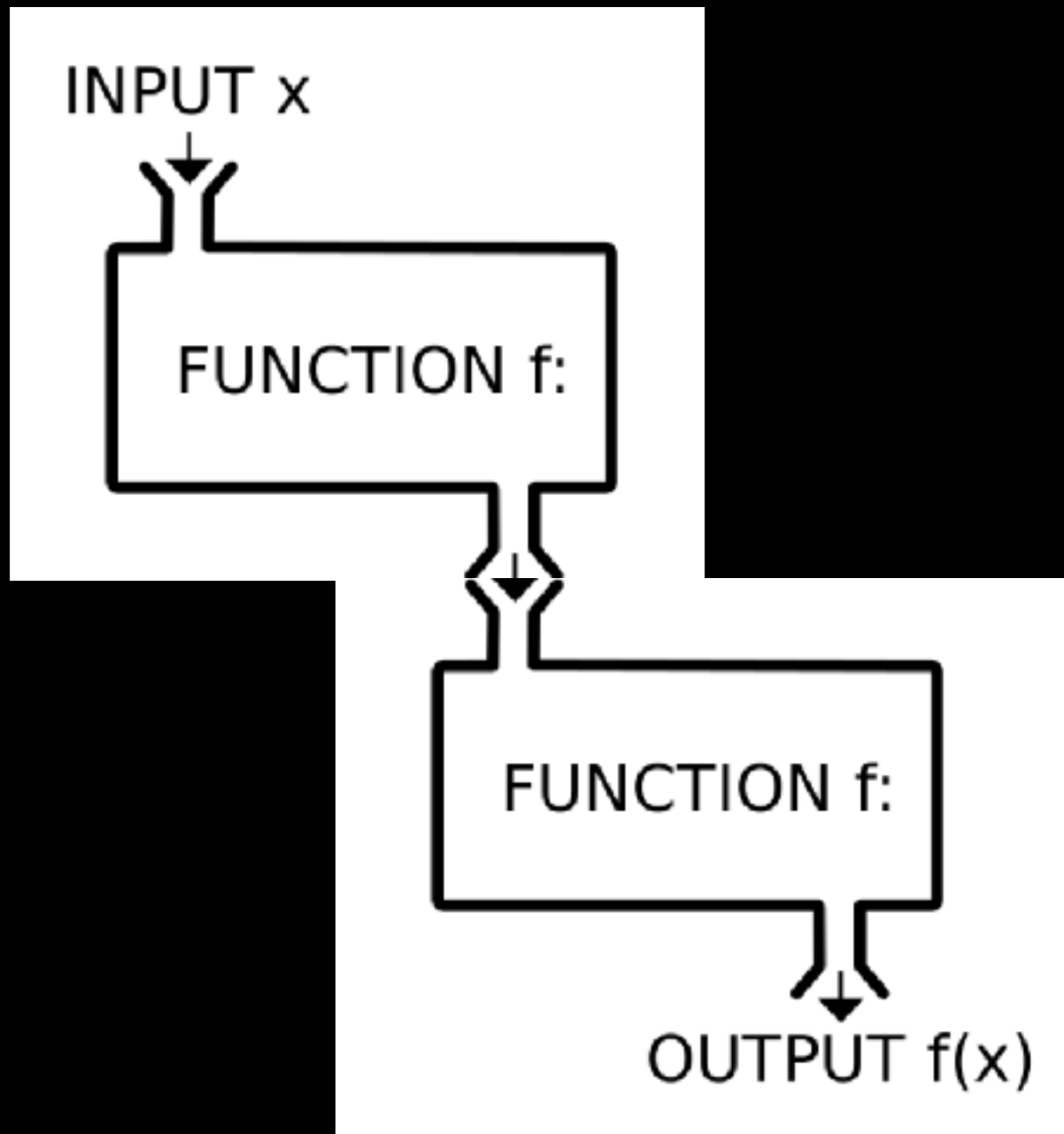
Great!

Now you know
functional programming

So why
functional programming?

Composability

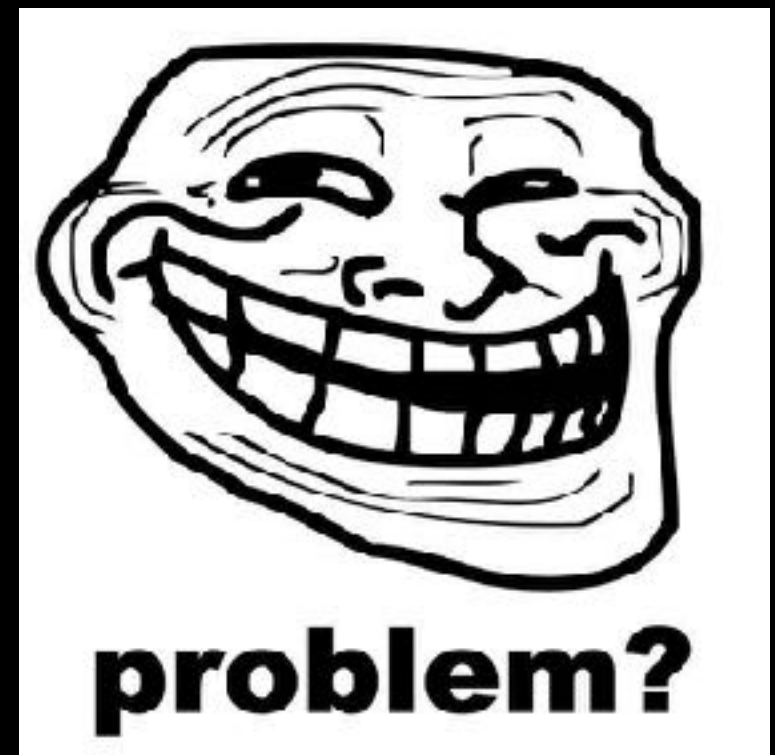




Easier to maintain

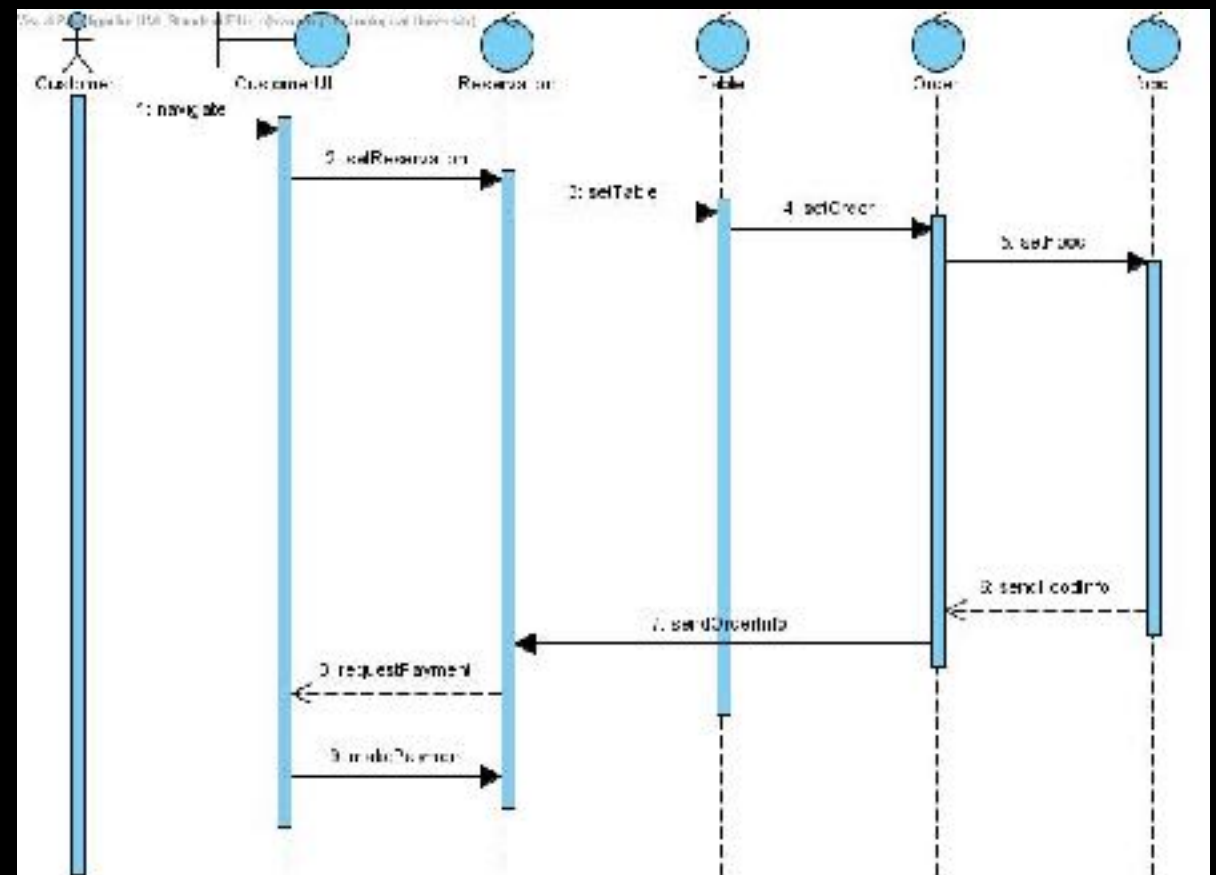
```
for (var i = 0; i < arr.length; i++) {  
    // do something  
    arr.push(i);  
    // do other things  
}
```

```
for (var i = 0; i < arr.length; i++) {  
    // do something  
    arr.push(i);  
    // do other things  
}
```



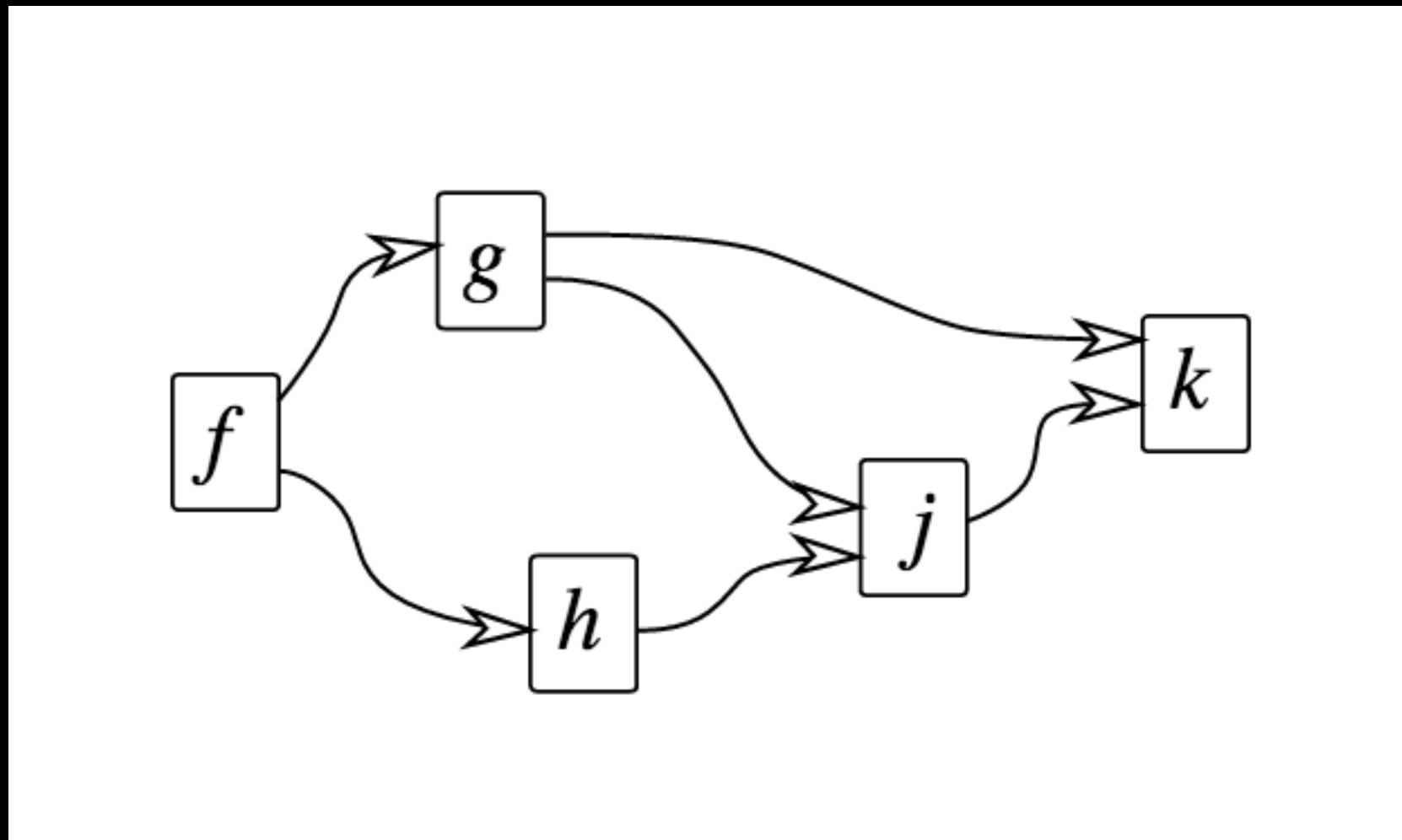
Easier to understand

No! you cannot



<https://erestaurant.wordpress.com/2009/09/14/class-diagrams-sequence-diagram/>

Data Flow Programming



<http://www.macs.hw.ac.uk/~rs46/posts/2015-09-07-distributed-functional-futures.html>

How about OO?

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

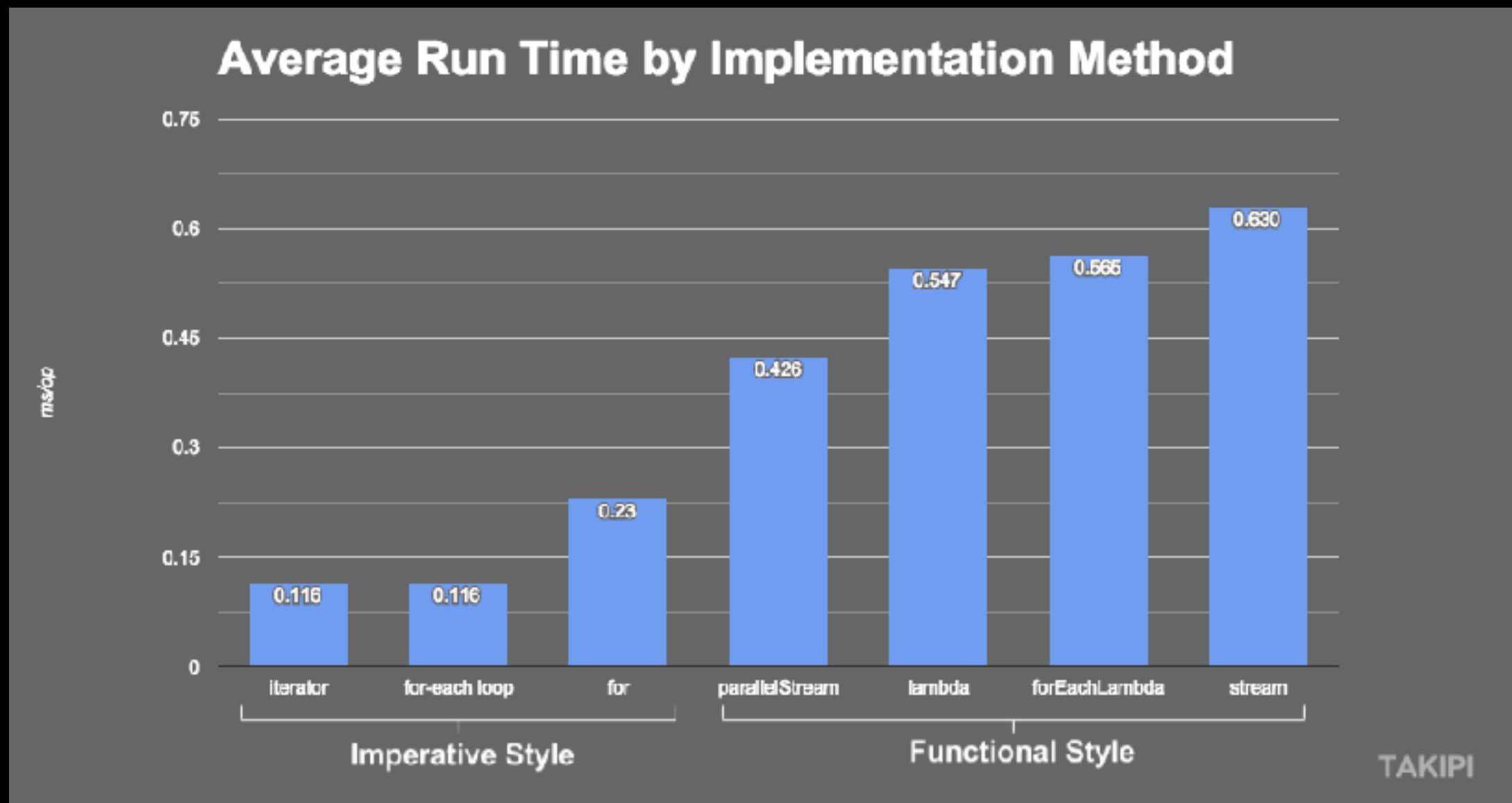
- Functions
- Functions
- Functions, also
- Functions
- Yes, functions
- Oh my, functions again!
- Functions
- Functions ☐

<https://vimeo.com/113588389>

<http://www.slideshare.net/ScottWlaschin/fp-patterns-buildstuffit>

Faster?

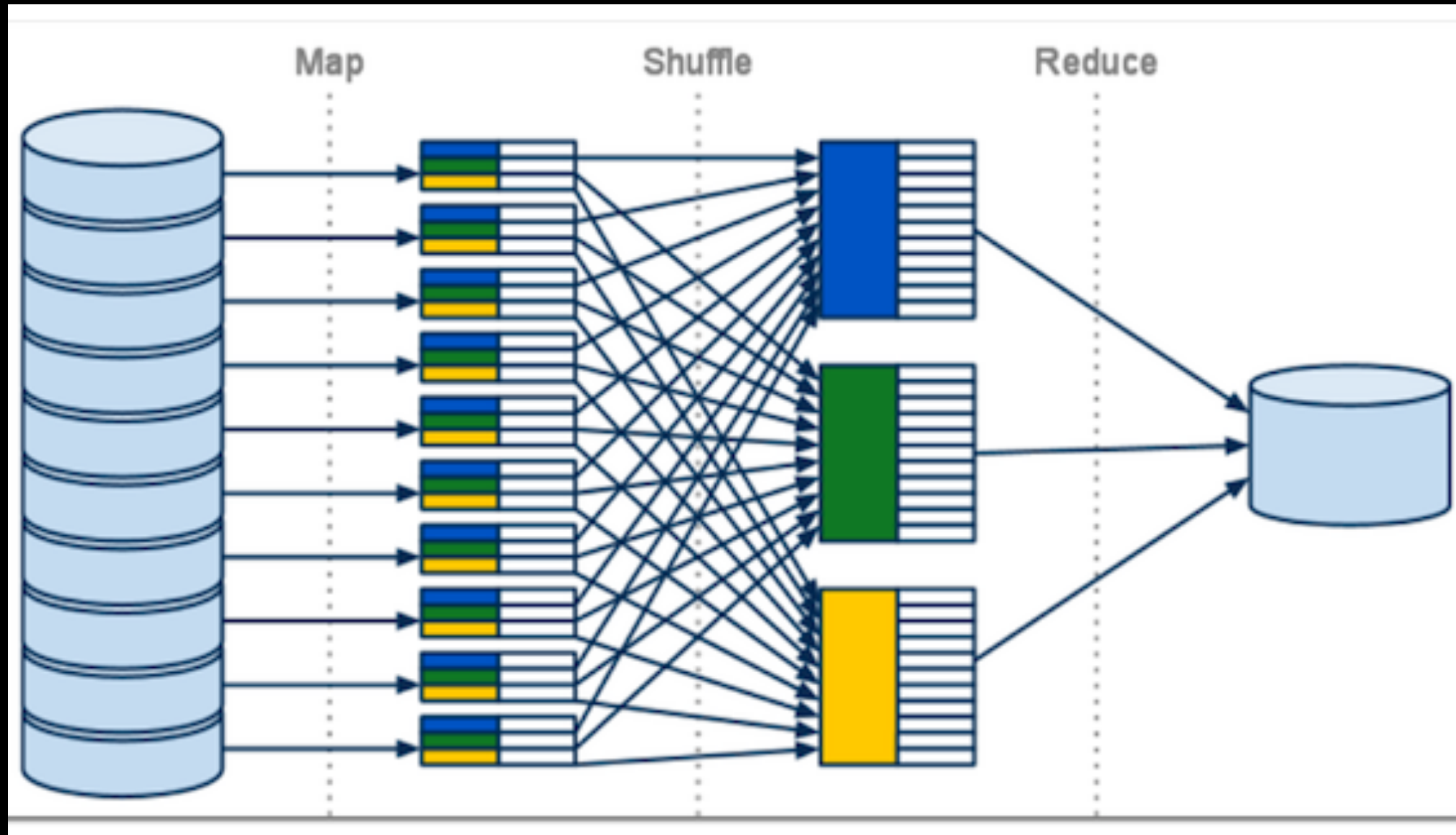
Faster? Nope



<http://blog.takipi.com/benchmark-how-java-8-lambdas-and-streams-can-make-your-code-5-times-slower/>

Slower?

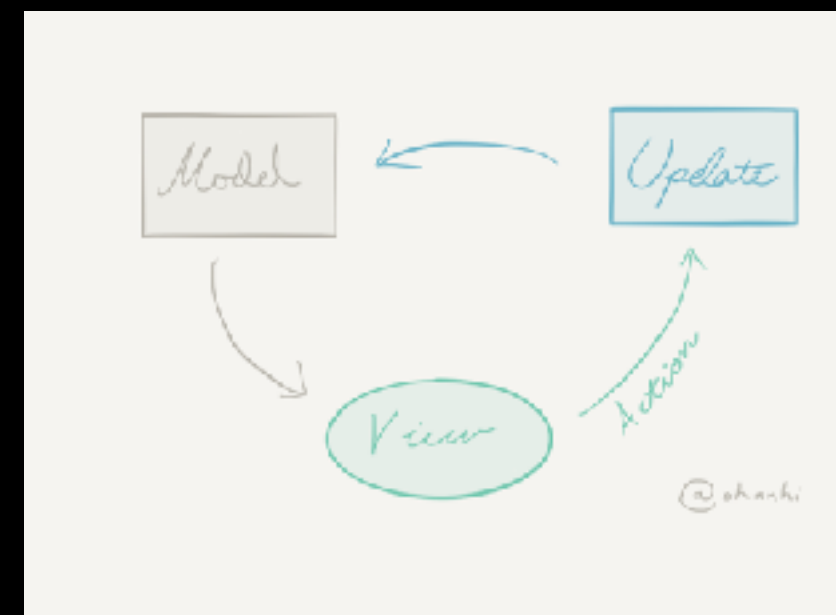
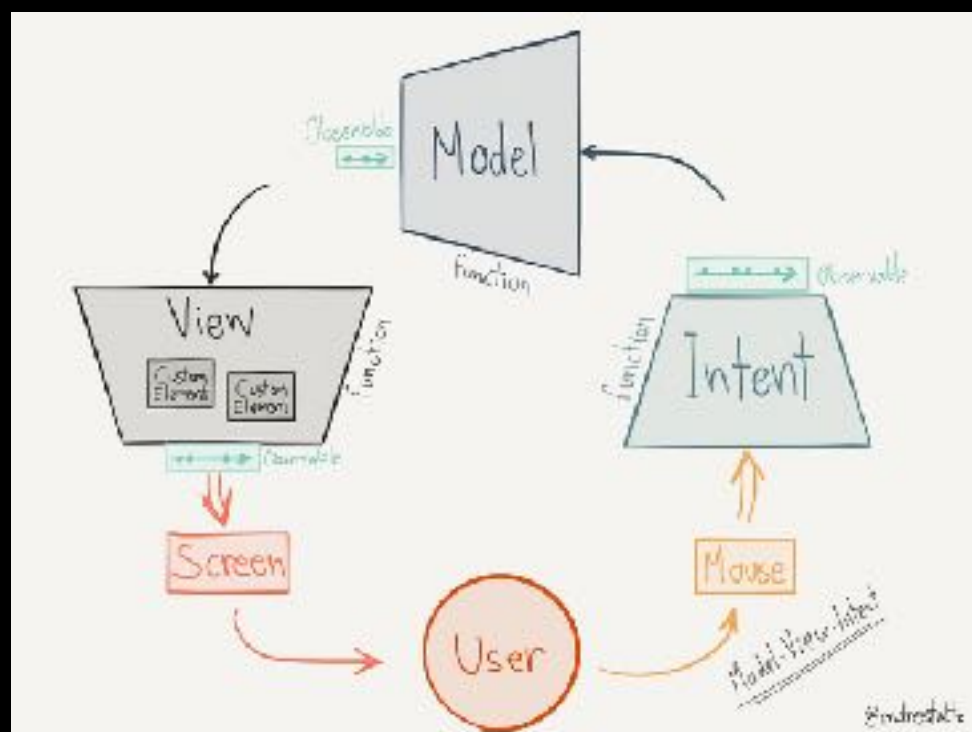
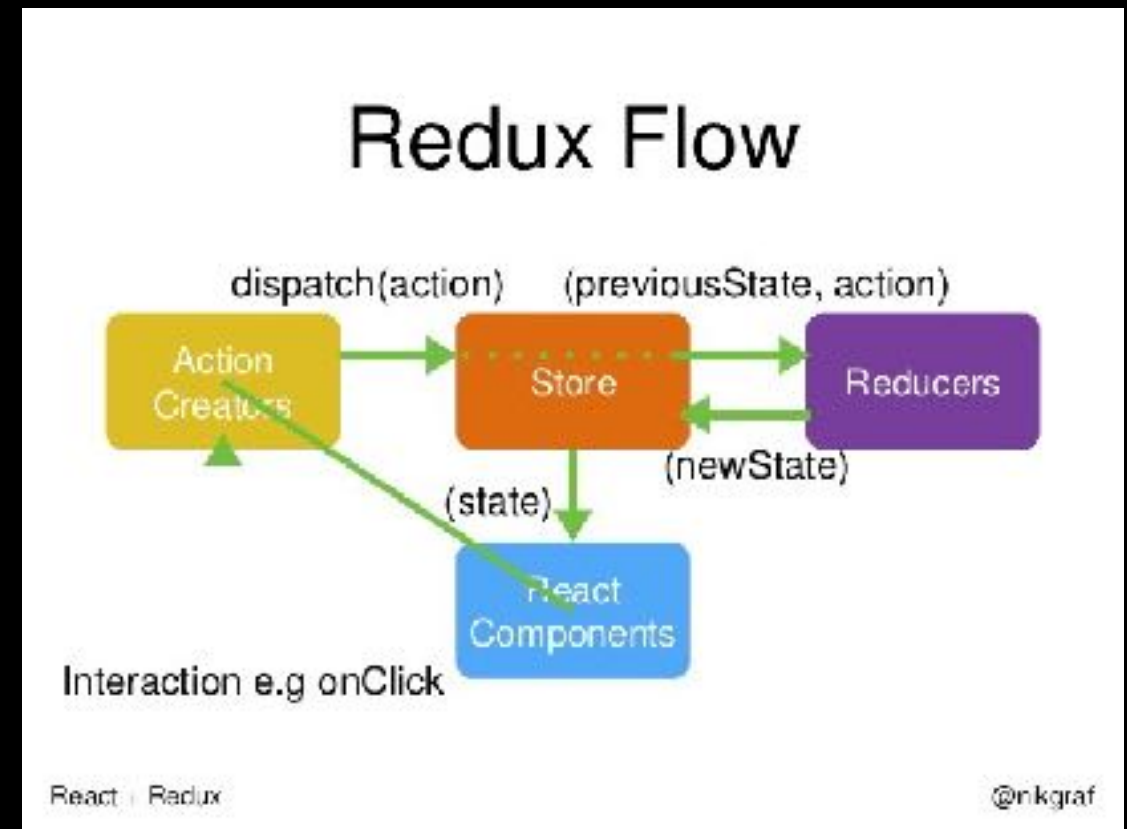
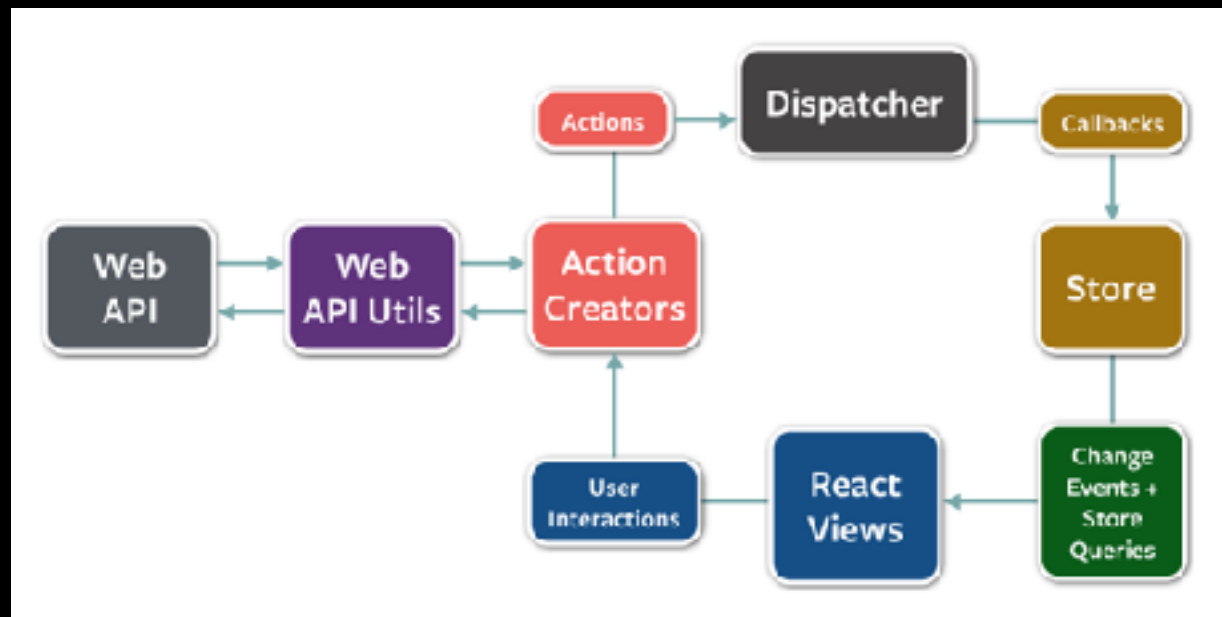
Slower? Nope



<https://www.linkedin.com/pulse/20140818193221-22744472-big-data-analytics-mapreduce>

Where to use?

Web frontend



Mobile?

Actually all client side

GitHub - trikita/jedux: Redux architecture for Android in good old java

<https://github.com/trikita/jedux> ▾ 翻譯這個網頁

Redux architecture for Android in good old java. Contribute to jedux development by creating an account on GitHub.

GitHub - glung/redux-java: The java version of Redux : a predictable ...

<https://github.com/glung/redux-java> ▾ 翻譯這個網頁

redux-java. The java version of Redux : a predictable state container for java apps. Redux-java has been designed Android in mind but is not constrained to it ...

GitHub - brianegan/bansa: A state container for Java & Kotlin, inspired ...

<https://github.com/brianegan/bansa> ▾ 翻譯這個網頁

bansa - A state container for Java & Kotlin, inspired by Redux & Elm. ... this little project: An easier way to write Android UIs & Apps. Perhaps an easy way to start ...

Writing a Todo app with Redux on Android – Medium

<https://medium.com/.../writing-a-todo-app-with-redux-on-android-5de3...> ▾ 翻譯這個網頁

Writing a Todo app with Redux on Android. Android community seems to be actively looking for the right architecture for their apps. We've passed through the ...

The evolution of Android architecture

zserge.com/blog/android-mvp-mvvm-redux-history.html ▾ 翻譯這個網頁

2016年3月28日 - Android history: from no architecture to MVP to MVVM to Redux.

Android架构移植“Redux” - 简书

www.jianshu.com/p/a5bccd56b71e ▾ 轉為繁體網頁

2016年6月22日 - 本篇是基于AndroidFlux背景写的关于Redux在Android上应用的一文，需要提前了解一些Flux的知识。Redux是Flux模式的一种实现，目前 ...

GitHub - ReduxKit/ReduxKit: Redux for Swift - a predictable state ...

<https://github.com/ReduxKit/ReduxKit> ▾ 翻譯這個網頁

ReduxKit - Redux for Swift - a predictable state container for Swift apps. ... source 'https://github.com/CocoaPods/Specs.git' platform :ios, '8.0' pod 'ReduxKit', ...

GitHub - ReSwift/ReSwift: Unidirectional Data Flow in Swift - Inspired ...

<https://github.com/ReSwift/ReSwift> ▾ 翻譯這個網頁

ReSwift - Unidirectional Data Flow in Swift - Inspired by Redux. ... source 'https://github.com/CocoaPods/Specs.git' platform :ios, '8.0' pod 'ReSwift'. And run pod ...

GitHub - oursky/Redux: Swift implementation of Redux

<https://github.com/oursky/Redux> ▾ 翻譯這個網頁

Contribute to Redux development by creating an account on GitHub. ... source 'https://github.com/CocoaPods/Specs.git' platform :ios, '8.0' pod 'Redux', ...

用Objective-C 實作Redux 架構 Nelson 寫些iOS 開發的東東

nelson.logdown.com/posts/2016/08/03/redux-in-objective-c ▾

2016年8月3日 - Flux / Redux 一開始提出是給網站使用的架構，後來有人把它套用到iOS 開發，不過我查到的資料都是使用Swift 實作，無可否認使用Swift 來實作這 ...

Brushes Redux on the App Store - iTunes - Apple

<https://itunes.apple.com/tw/app/brushes-redux/id932089074?mt=8> ▾

免費 - iOS

2016年4月7日 - Brushes Redux is a painting app designed exclusively for iOS. Rewritten from the ground up, Brushes Redux is universal — the same version ...

Redux for iOS - jtribe

blog.jtribe.com.au/redux-for-ios/ ▾ 翻譯這個網頁

2016年1月30日 - In the center of both Flux and Redux is the unidirectional data flow. This blog describes how Redux can be used as an iOS application ...

User Interfaces

- events (keydown, mouse move/click...etc)
- state
- update **function**: (event, state) -> state
- render **function**: state -> view

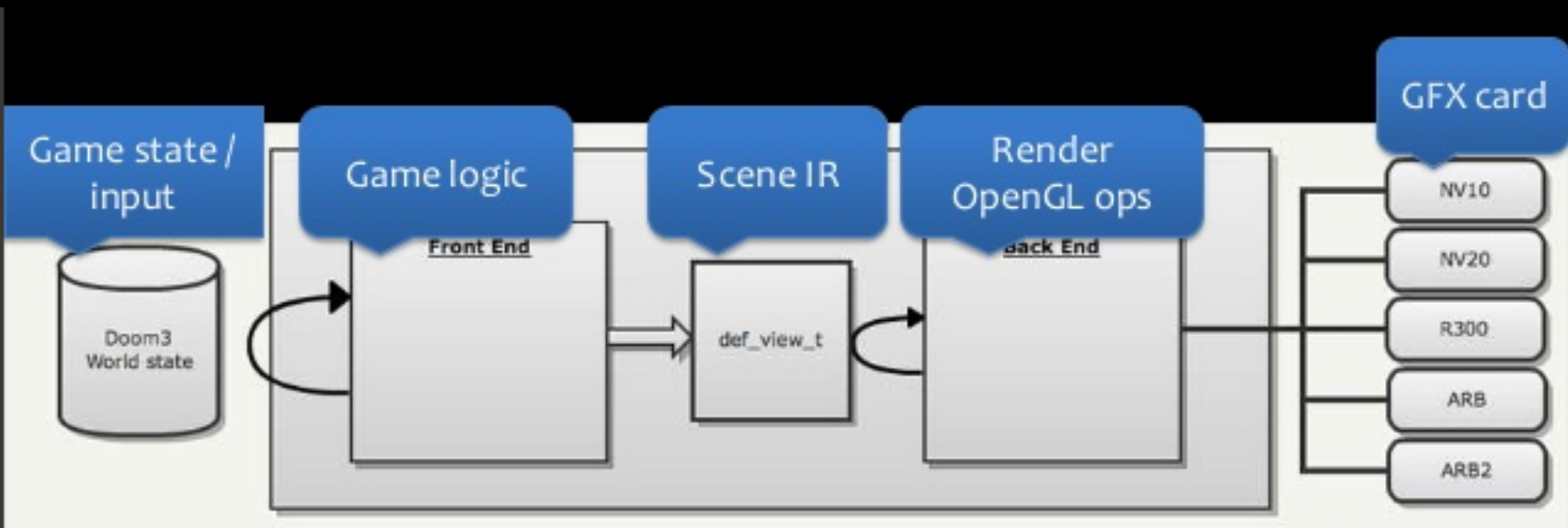
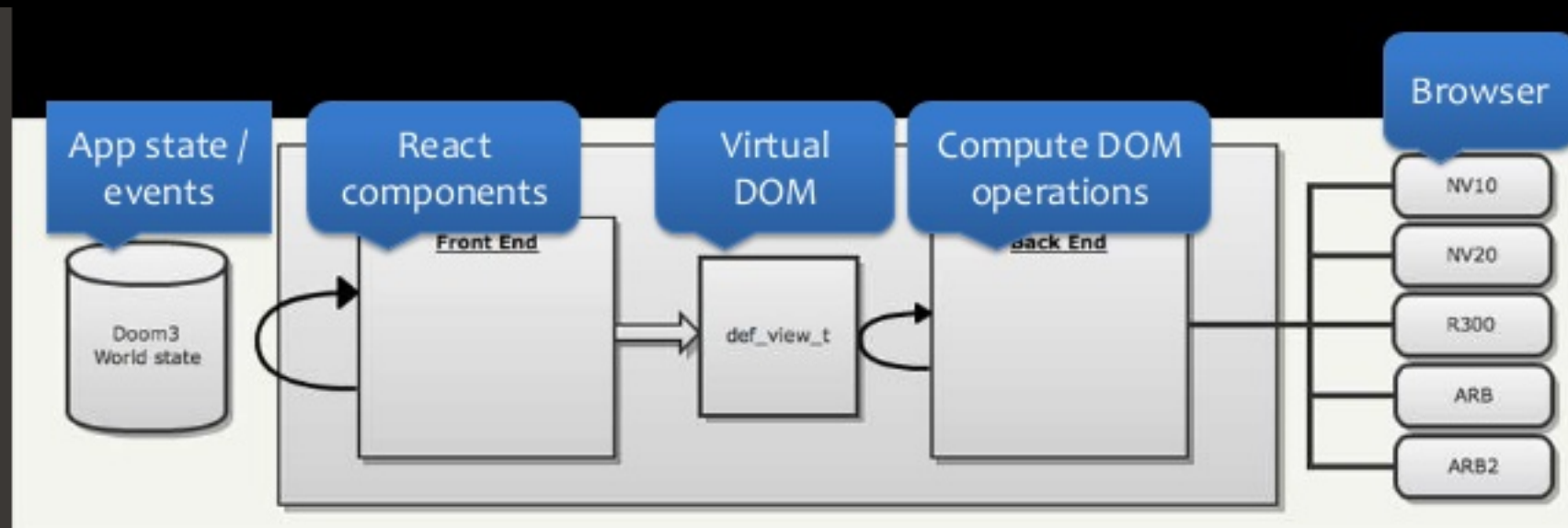
Web backend

Of course!

request -> response

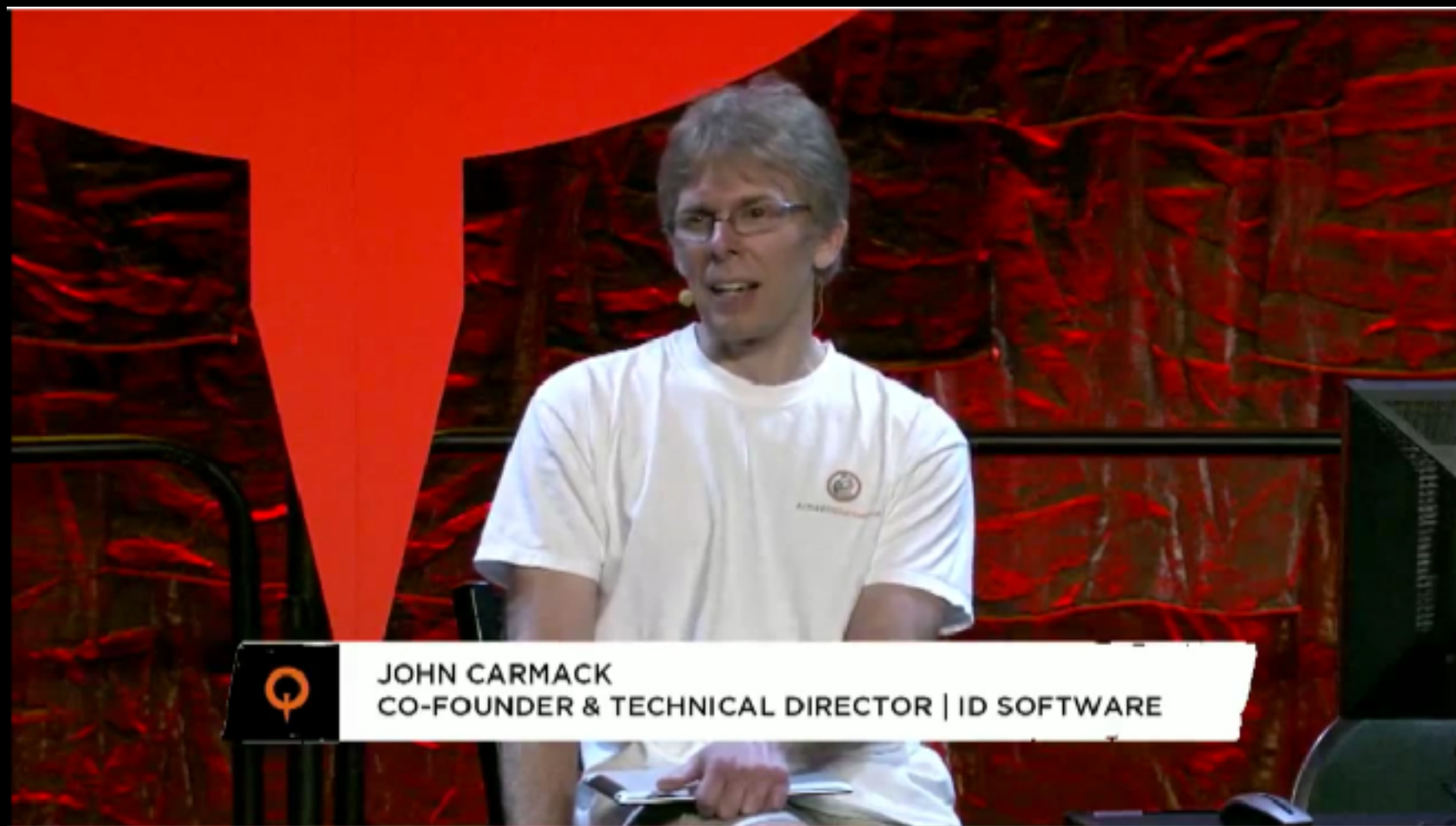
Game Programming!

data -> view



Game Programming

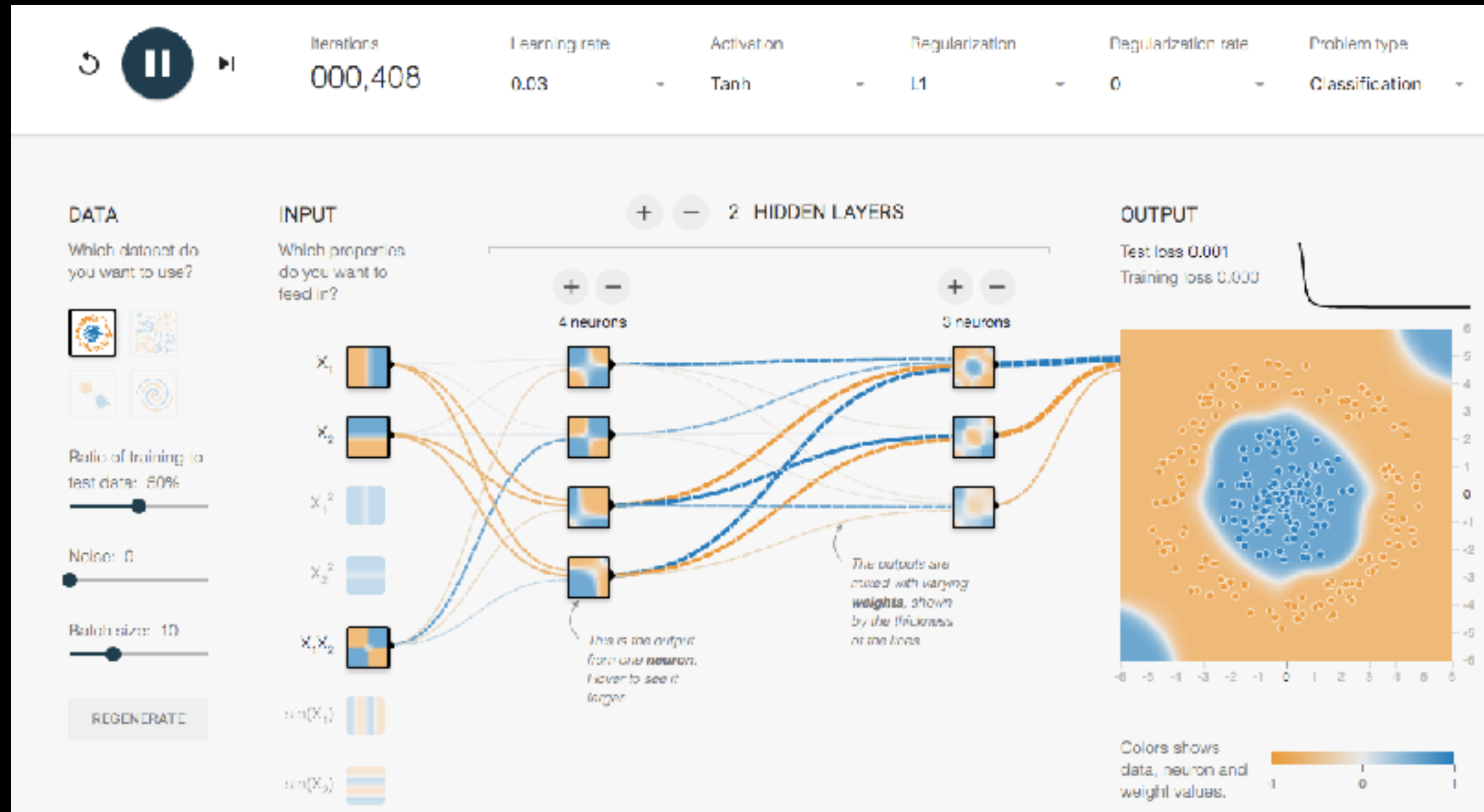
- <https://www.youtube.com/watch?v=eNWAceEu1jpU>



Machine learning!



TensorFlow



<https://techcrunch.com/2016/04/13/google-launches-distributed-version-of-its-tensorflow-machine-learning-library/>

Where to use?
Almost anywhere

How to learn?

- Think more when you're programming
- Learn Haskell! learnyouahaskell.com