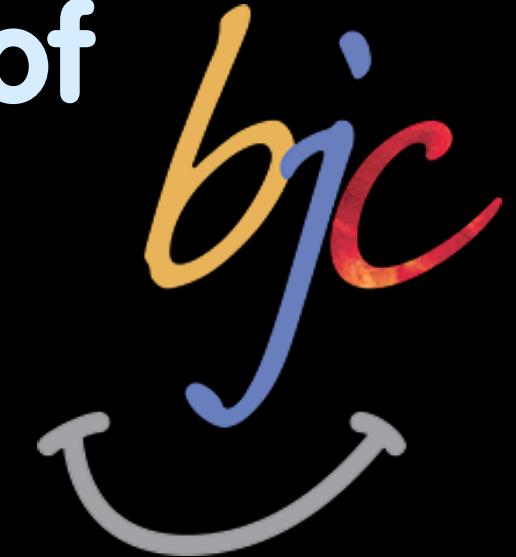


The Beauty and Joy of Computing

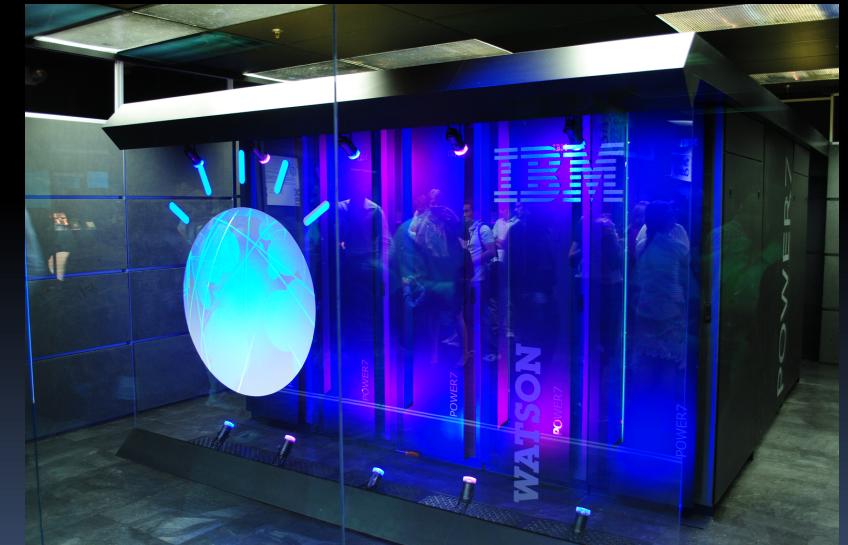


Lecture #17 Higher Order Functions



Putting Watson to use for City Apps

The City of Surrey in British Columbia, Canada is the first city to use IBM's Watson technology as the basis for a new app, "My Surrey". The app has been described as 'Siri for local government' because of the ability for users to use natural language to find answers.



**Higher Order
Functions...**

Why? Basics...



Why Use Functions? (review)

```
pen down  
repeat (4)  
  move (25) steps  
  turn (90) degrees  
pen up
```

```
pen down  
repeat (4)  
  move (100) steps  
  turn (90) degrees  
pen up
```

```
pen down  
repeat (4)  
  move (395) steps  
  turn (90) degrees  
pen up
```

Draw Square of Side **length**

```
pen down  
repeat (4)  
  move (length) steps  
  turn (90) degrees  
pen up
```

Calling it...

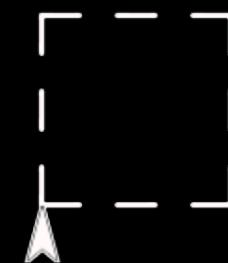
Draw Square of Side **100**



Why Use Higher-Order Functions?



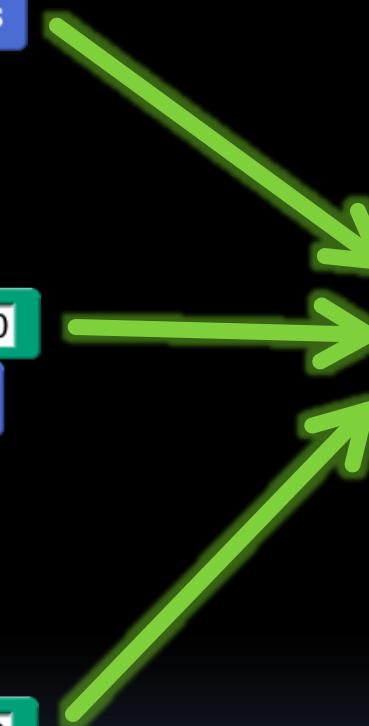
```
repeat (4)
  Draw Line [100]
  turn [90] degrees
```



```
repeat (4)
  Draw Dashed Line [100]
  turn [90] degrees
```



```
repeat (4)
  Draw Wiggly Line [100]
  turn [90] degrees
```



```
Draw [LineDrawer λ 100 Square]
repeat (4)
  run [LineDrawer] with inputs [100 ←→]
  turn [90] degrees
```

Calling it...

```
Draw Wiggly Line [100 Square]
```



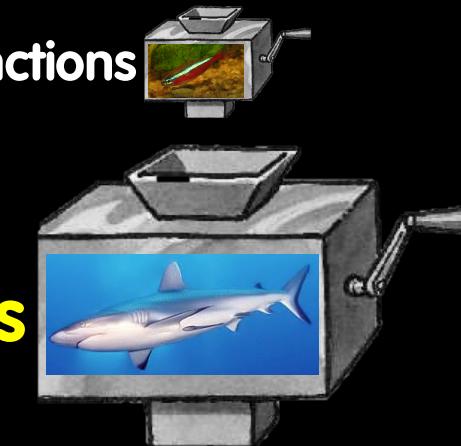
Why HOFs are like Pregnant Fish, Sharks

Data (e.g., Sentences, Words, Booleans, Lists)

Normal
Fish



Functions



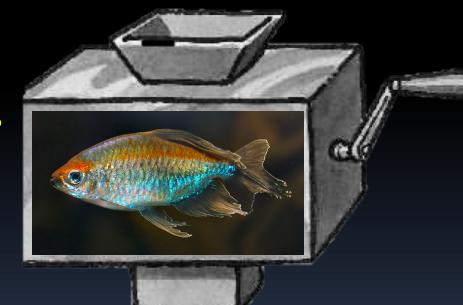
Data (e.g., Sentences, Words, Booleans, Lists)

Sharks

Data (e.g., Sentences, Words, Booleans, Lists)

Data (e.g., Sentences, Words, Booleans, Lists)

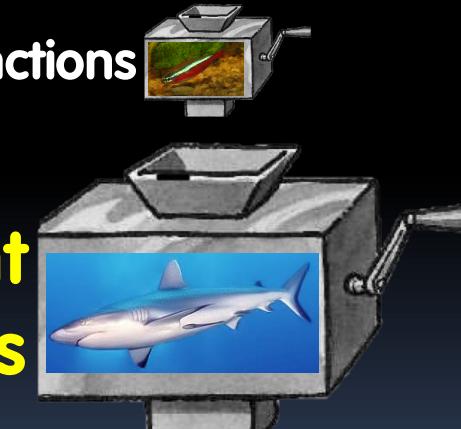
Pregnant
Fish



Functions

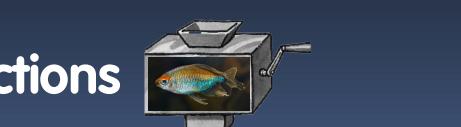


Functions



Pregnant
Sharks

Functions



Source: Brian Harvey, Wikipedia (Fbattail, Aka, Evdaimon)

UC Berkeley "The Beauty and Joy of Computing": Higher-Order Functions (5)

HOFs you have seen before

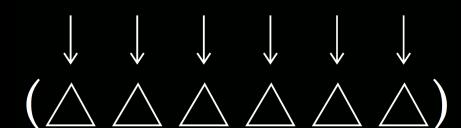
Useful HOFs (you can build your own!)

- **map Reporter over List**
 - Report a new list, every element E of List becoming Reporter(E)

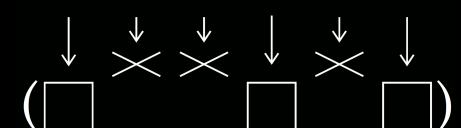
- **keep items s.t. Predicate from List**
 - Report a new list, keeping only elements E of List if Predicate(E)

- **combine with Reporter over List**
 - Combine all the elements of List with Reporter(E)
 - This is also known as “reduce”

(□ □ □ □ □ □)



(□ □ □ □ □ □)



(□ □ □ □ □ □)

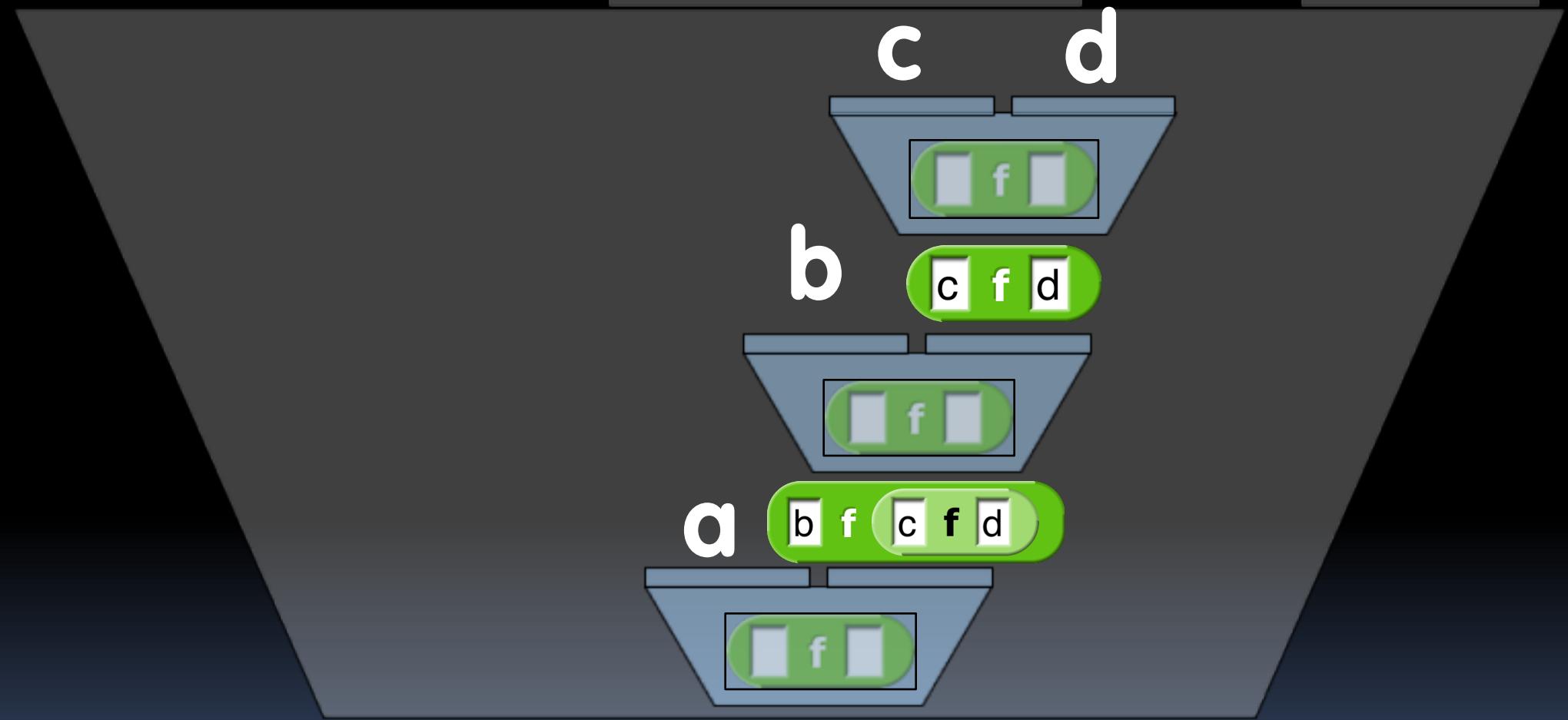




How combine works:

List
1 a
2 b
3 c
4 d
+ length: 4

combine with Reporter over List



combine above the abstraction line



(a b c d)

combine using **f**

a **f** b **f** c **f** d

Your **f** should be **associative**, and work if it's



((a **f** b) **f** (c **f** d))

or...



(((a **f** b) **f** c) **f** d)

etc...



Garcia



What is reported here?

- a) abcd
- b) acdb
- c) bdac
- d) dcba
- e) (nothing)



combine with **joinswap** items of **list [a b c d ↔]**



Acronym

Acronym Algorithm

(the Beauty and Joy of Computing)



keep only
uppercase words

(Beauty Joy Computing)

map first letter only

(B J C)

combine into one
word (join)

BJC





Acronym (uses map, keep, combine)



Acronym sentence → list the•Beauty•and•Joy•of•Computing

BJC

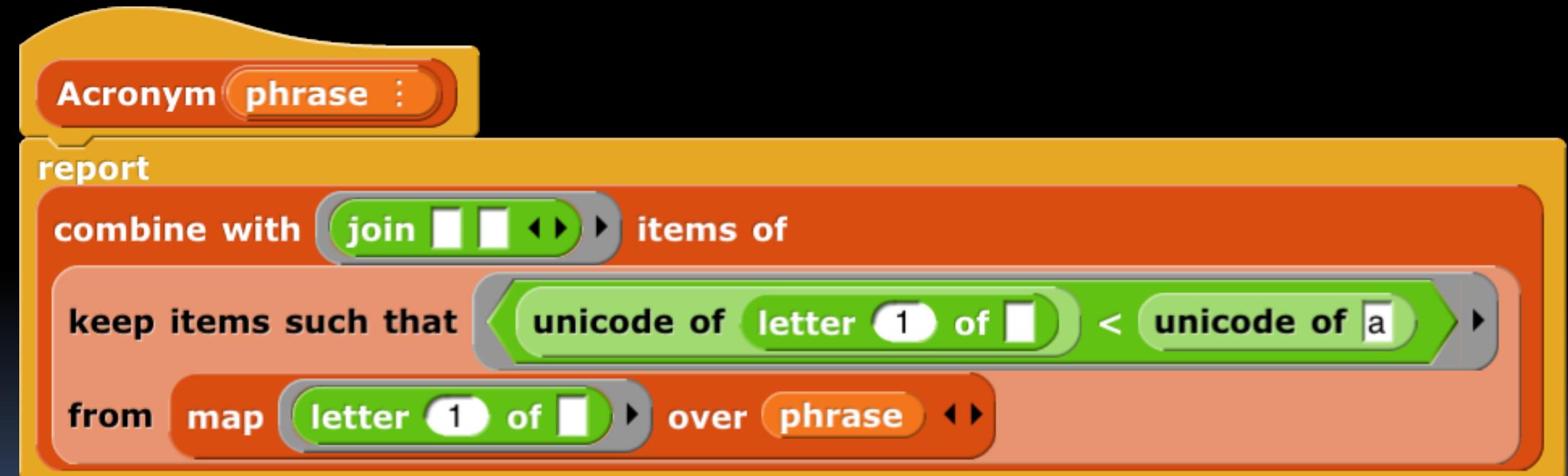


(Cal) Clicker Question



What is reported if I swap **keep** and **map** in Acronym and call it on (the Beauty and Joy of Computing)

- a) bjc
- b) BJC
- c) CJB
- d) (nothing)
- e) Error



HOF Tools & Demo



HOF tools, sharks, pregnant fish, mymap

