



The Beauty and Joy of Computing

Higher Order Functions I



UC Berkeley EECS
Sr Lecturer SOE
Dan Garcia

CODING IS COOL AGAIN!

The market for classes in coding (esp focused on the Internet) is booming, so says the NY Times. Codecademy is one of the biggest sites; CS10 is a great first step!!

(Image Credit: New York Times)



<http://www.nytimes.com/2012/03/28/technology/for-an-edge-on-the-internet-computer-code-gains-a-following.html>



Midterm Results (paper only)





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 396 steps
  turn 90 degrees
pen up
```



```
Draw Square of Side length
pen down
repeat 4
  move length steps
  turn 90 degrees
pen up
```

The power of **generalization!**





But how general can we be?

```
Min of list
script variables best so far
set best so far to item 1 of list
# foreach item of list
if item < best so far
  set best so far to item
report best so far
```

```
Max of list
script variables best so far
set best so far to item 1 of list
# foreach item of list
if item > best so far
  set best so far to item
report best so far
```

```
Closest to 6 list
script variables best so far
set best so far to item 1 of list
# foreach item of list
if item closer to 6 than best so far
  set best so far to item
report best so far
```

```
find best element using better from list
script variables best so far
set best so far to item 1 of list
# foreach item of list
if call better with inputs item best so far
  set best so far to item
report best so far
```

```
find best element using closer to 6 than
from list 2 5 1 9 4
```

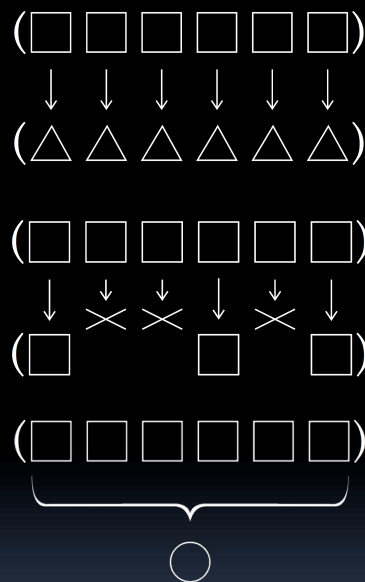
The power of **generalization!**





Today

- Functions as Data
- Higher-Order Functions
- 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 such that** 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”
- Acronym example
 - **keep** → **map** → **combine**



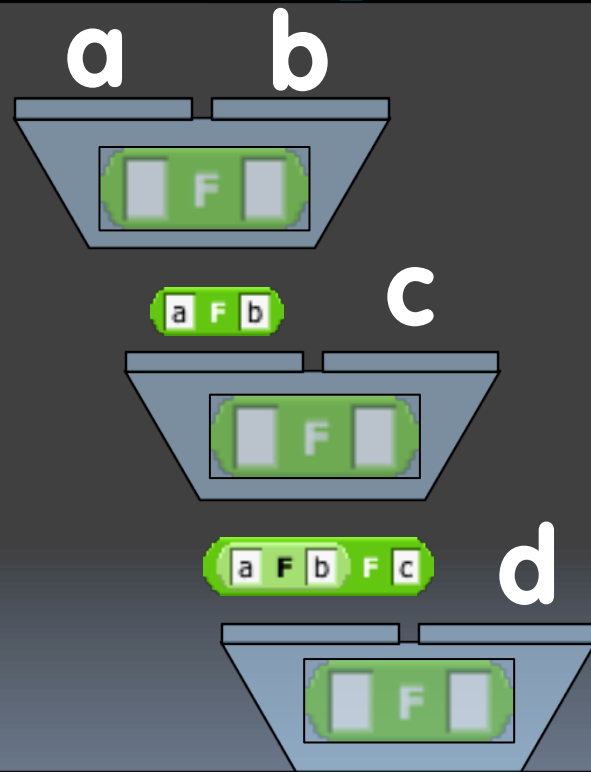


List

| | |
|---|---|
| 1 | a |
| 2 | b |
| 3 | c |
| 4 | d |

+ length: 4

combine with Reporter over List





Peer Instruction



I understand higher-order functions.

- a) Strongly disagree
- b) Disagree
- c) Neutral
- d) Agree
- e) Strongly agree



Summary

- Functions as data is **one of the two (programming) big ideas** in this course
- It's a beautiful example of the **abstraction of the list iteration details**
- Google (and other companies) use this!
 - They use "map-reduce"

(Image Credit: *Simply Scheme* by Brian Harvey & Matt Wright)

