Using λSnap! to Teach the Big Idea of Data

2014-08-08@ 11am
Event Space
MIT Media Lab
Cambridge, MA

UC Berkeley
Sr Lecturer SOE
Dan Garcia
Broadening Participation in HS CS!

- **New Course: “Computer Science : Principles”**
  - Engaging, accessible, inspiring, rigorous
  - Focused on the fundamental concepts of computing (Computational Thinking)
  - An impetus for college curriculum reform
  - Available nationwide, Fall 2016-Spring 2017

- **SINGLE SOURCE OF NATIONAL LEVERAGE!**

CollegeBoard

[csprinciples.org](http://csprinciples.org)
what is CS Principles?

7 big ideas

- computing is a Creative activity.
- Abstraction reduces information and detail to facilitate focus on relevant concepts.
- Data and information facilitate the creation of knowledge.
- Algorithms are used to develop and express solutions to computational problems.
- Programming enables problem solving, human expression, and creation of knowledge.
- the Internet pervades modern computing.
- computing has global Impacts.

check out the complete curriculum framework at: csprinciples.org
CSP Curricular Framework: Data

3. Data and Information

1. People use computer programs to process information to gain insight and knowledge.
   1. Use computers to process information, find patterns, and test hypotheses about digitally processed information to gain insight and knowledge
   2. Collaborate when processing information to gain insight and knowledge
   3. Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notation, and precise language

2. Computing facilitates exploration and the discovery of connections in information.
   1. Extract information from data to discover and explain connections, patterns, or trends (Students are not expected to know specific formulas or options available in spreadsheet or database software packages)
   2. Use large data sets to explore and discover information and knowledge

3. There are trade-offs when representing information as digital data
   1. Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information
UC Berkeley’s BJC
The Beauty and Joy of Computing

Grant Winner

Pilot

Pilot x3

Award Winner

- 2009Fa: 16 students (pilot)
- 2010Fa: 90 students
- 2011Sp: 90 students
- 2011Su: ~25 HS teachers in BJC Family!
- 2011Fa: 250 Students
- 2012Sp: 250 Students
- 2012Su: ~100 HS teachers online!
- 2012Fa: 250 Students & 60 UCB online pilot
- 2013Sp: 250 Students
- 2013Su: ~175 HS teachers in BJC Family!
- 2013Fa: 360 Students
- 2014Sp: 250 Students
- 2014Su: ~250 HS teachers (~10 faculty) in BJC Family

bjc.berkeley.edu
Let's double all of your salaries!

1
2
3
length: 3

SALARIES

Using Snap! to Teach the Big Idea of Data
First try using repeat block

```
DoubleSalaries : salaries using repeat

script variables

set index to 1
set doubled_salaries to list
repeat length of salaries
add Double item index of salaries to doubled_salaries
change index by 1
report doubled_salaries
```
First try using repeat block ... yay!

Double Salaries | SALARIES | using repeat

SALARIES

length: 3

1 1
2 7
3 12

length: 3

1 2
2 14
3 24

Using Snap! to Teach the Big Idea of Data
First try using **for** block

```
Double Salaries
script variables doubled salaries
set doubled salaries to list
for index = 1 to length of salaries
  add Double item index of salaries to doubled salaries
report doubled salaries
```
First try using for block ... yay!

**SALARIES**

1
2
3

**Double Salaries**

1
2
3

**SALARIES**

1
2
3

**Length: 3**
First try using **for each block**

- Double salaries
- using `foreach`

- script variables `doubled salaries`

- `set doubled salaries` to `list`

- for each `salary` of `salaries`

- `add Double salary` to `doubled salaries`

- report `doubled salaries`
First try using **for each block** ... yay!

**SALARIES**

Double salaries **SALARIES** using **foreach**
But can we do better?
One line!!

Using Snap! to Teach the Big Idea of Data
Higher-Order Functions (HOFs)

- Useful HOFs (can also 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”
### Big Data: 1880 baby names, let’s do it

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>F</td>
<td>7065</td>
</tr>
<tr>
<td>Anna</td>
<td>F</td>
<td>2604</td>
</tr>
<tr>
<td>John</td>
<td>M</td>
<td>9655</td>
</tr>
<tr>
<td>William</td>
<td>M</td>
<td>9532</td>
</tr>
</tbody>
</table>

Using Snap! to Teach the Big Idea of Data