Implementing an Open-access, Data-Science Programming Environment for Everyone

Austin Cory Bart, Javier Tibau, Eli Tilevich, Clifford A. Shaffer, Dennis Kafura

COMPSAC 2016 – CELT

June 13, 2016

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship, Grant No. DGE 0822220

Who am I?

- **Austin Cory Bart**
- PhD in Computer Science
- Certification in Learning Sciences
- (Software + Instructional) Design

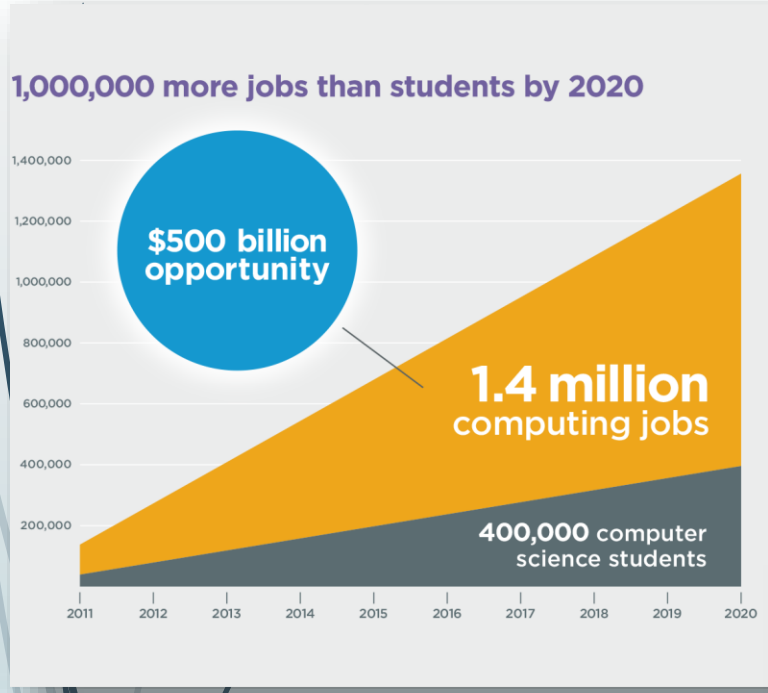
- Graduating Spring 2017!

Virginia Tech

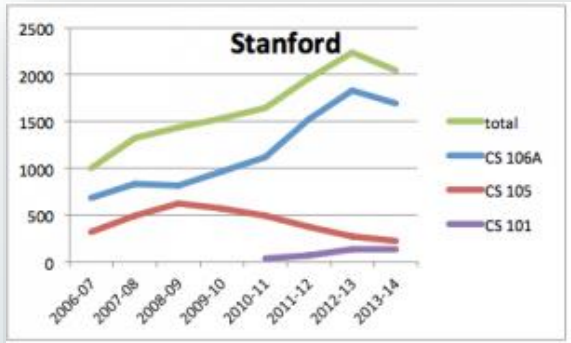
Overview

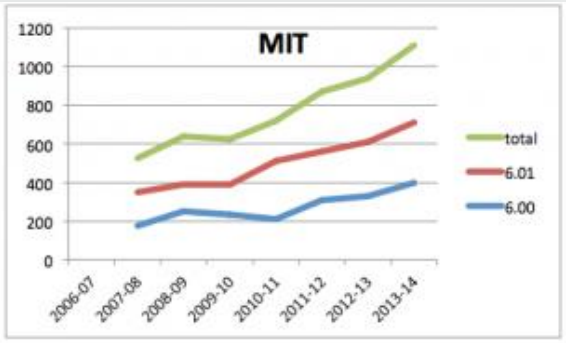
- Motivation
- Demo
- Features
- Evaluations
- Future Work

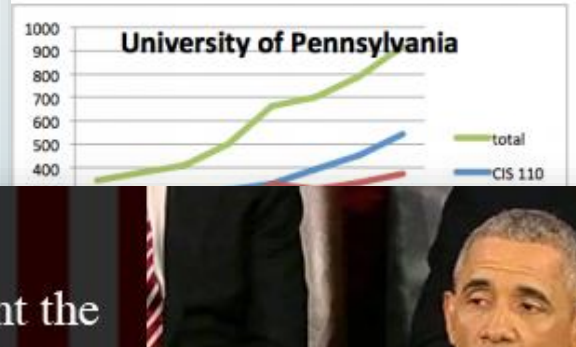
Computing For Everyone

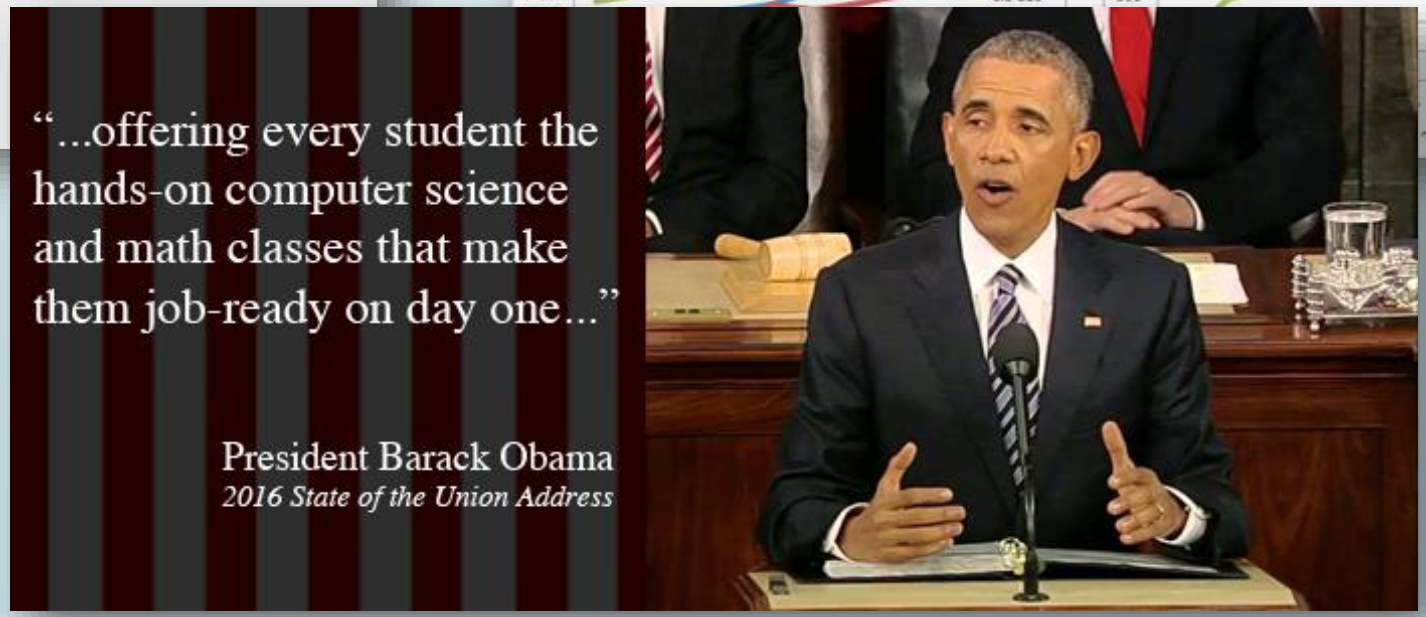


Computer science **is a top paying college degree** and computer programming **jobs are growing at 2X the national average.**









The Local Interest

"To this end, ***Virginia Tech will*** *comprehensively evaluate and modify the current Curriculum for Liberal Education to …* ***incorporate computational thinking*** *and informatics/digital fluency as basic skills* ***for all students****, thereby enabling our students to be engaged citizens and life-long learners."*

Contexts: Math and Business

Pure Math (e.g., Fibonacci)

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ \ldots$$

4. CONCLUSION

In this exposition, I showed how to infuse some algorithmic and mathematical aspects to guide the programming experience. The main theme is Fibonacci (and the golden ratio), which is a pleasant topic for many students. The typical paradigm that I support here is to first start with a warm up question (one that is not too trivial), then to

Saad Mneimneh. 2015. Fibonacci in The Curriculum: Not Just a Bad Recurrence. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 253-258.

Fun to the rescue!

… But authentic?

*Mark Guzdial and Allison Elliott Tew. 2006. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In Proceedings of the second international workshop on Computing education research (ICER '06). ACM, New York, NY, USA, 51-58. DOI=10.1145/1151588.1151597 http://doi.acm.org/10.1145/1151588.1151597

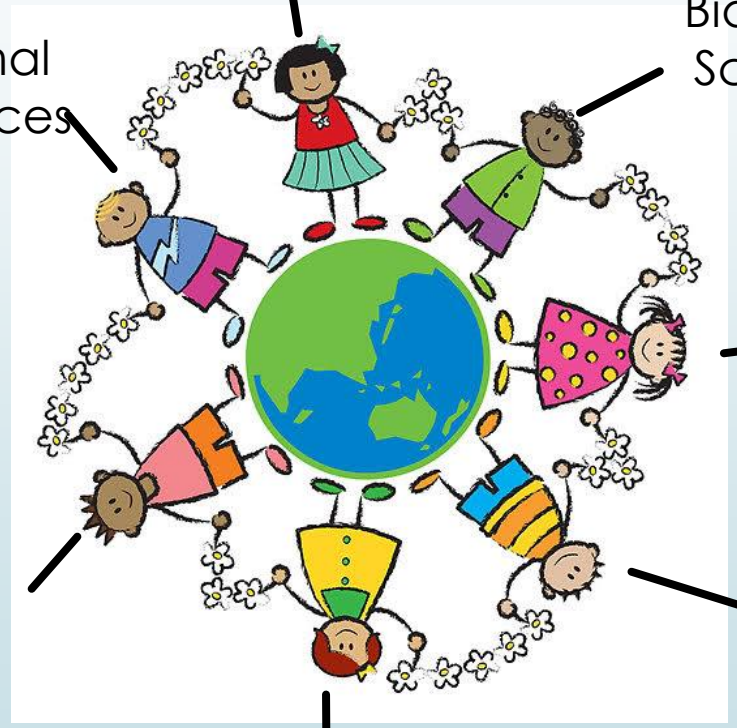
Diverse Majors

… with Rich Knowledge

English

Biological Sciences

Animal Sciences



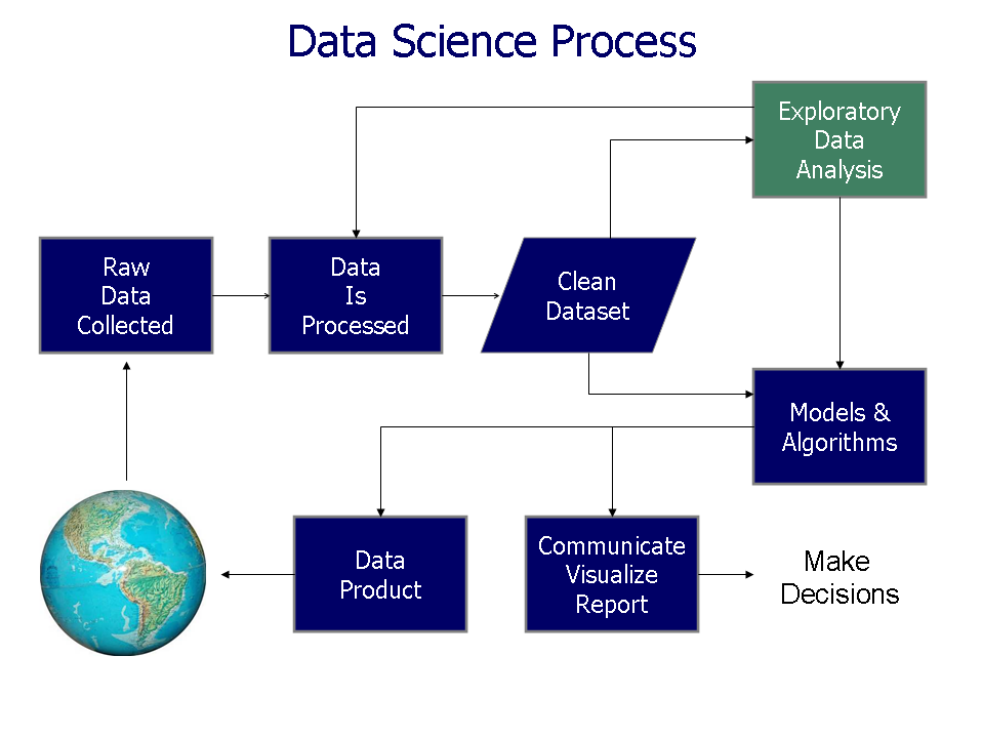
Education

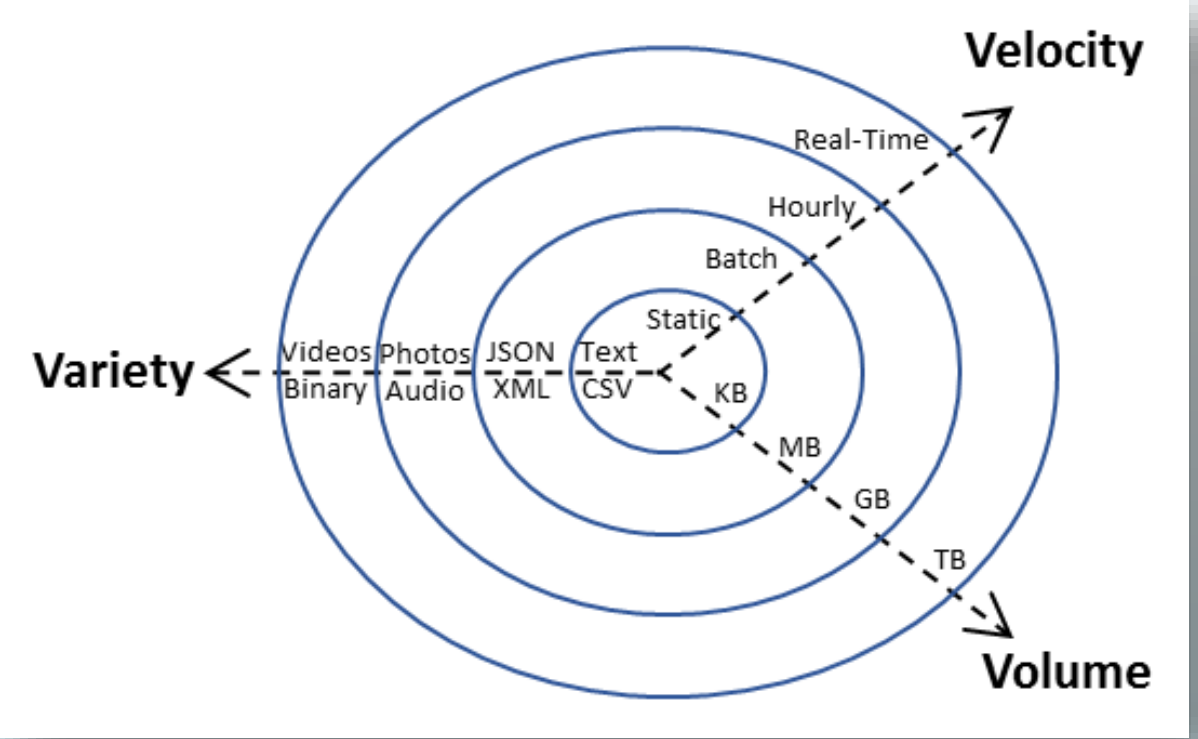
Theater Arts

History

Building Construction

Big Idea: Real-World Data





Computational Thinking via Data Science

Kafura, Dennis, Austin Cory Bart, and Bushra Chowdhury. "Design and Preliminary Results From a Computational Thinking Course." *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2015.

Design Goals

- Diverse learners
- And many of them
- With low self-efficacy

Design Goals

- Diverse learners
- And many of them
- With low self-efficacy

1. Give authentic problems
2. Minimize human effort
3. Reduce barriers
4. Fade scaffolds

Python

- Popular intro language
- Popular data science language
- Many great libraries

TIOBE Index

Jun 2016	Jun 2015	Change	Programming Language	Ratings	Change
1	1		Java	20.794%	+2.97%
2	2		C	12.376%	-4.41%
3	3		C++	6.199%	-1.56%
4	6	^	Python	3.900%	-0.10%
5	4	v	C#	3.786%	-1.27%

Existing Solutions?

Block-based Environments

- Blockly
- Scratch
- Snap
- Greenfoot

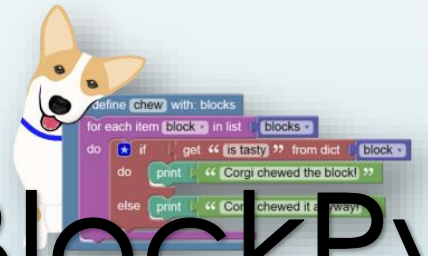
Not Python

Python in the Browser

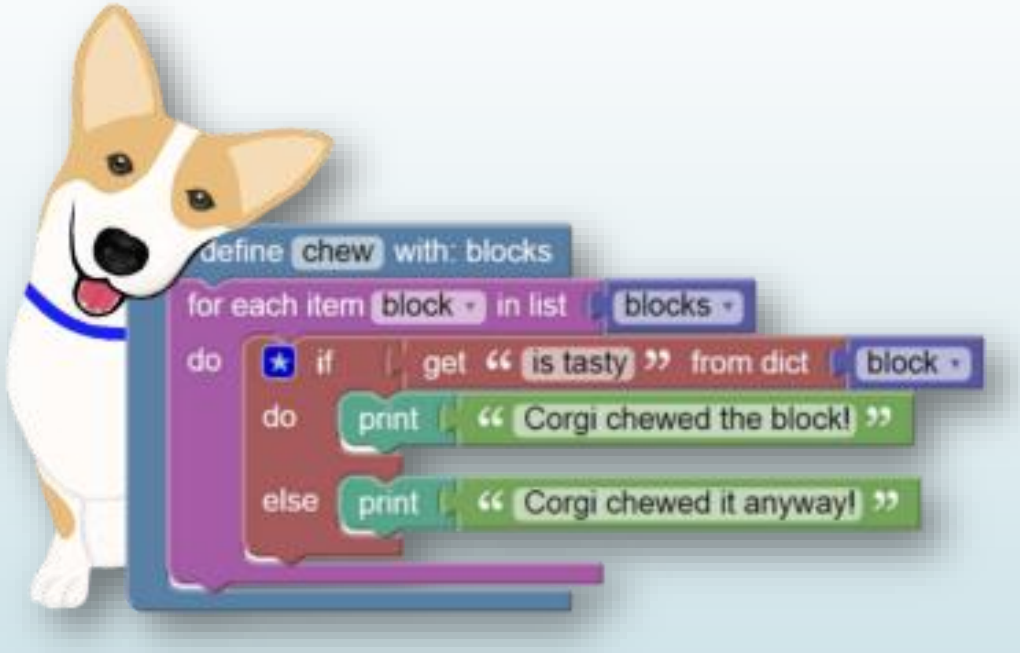
- CodeSkulptor
- Pythy
- Online Python Tutor

No blocks

Skulpt + blockly = BlockPy



BlockPy



www.blockpy.com

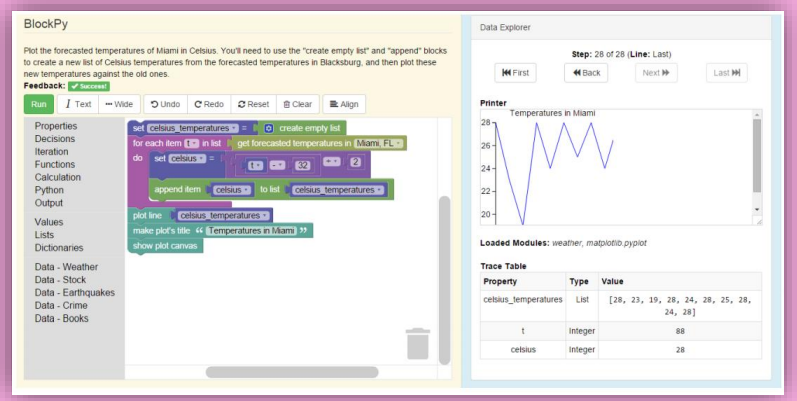
github.com/RealTimeWeb/blockpy/

BlockPy “Secret Sauce”

- Dual text/block
- Data Science
- Auto Feedback

Architecture

Client (Student Browser)



Blockly
(Visual Block Interface)

Skulpt
(Python Execution Engine)

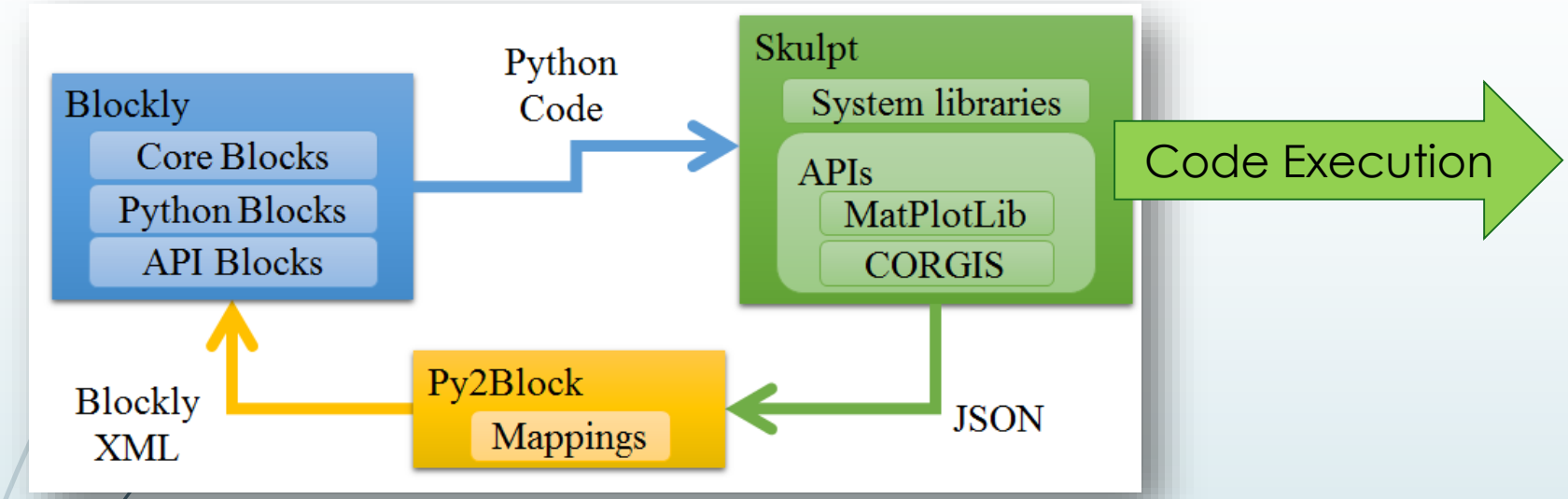
BlockPy Server

Assignment/Submission
Persistence

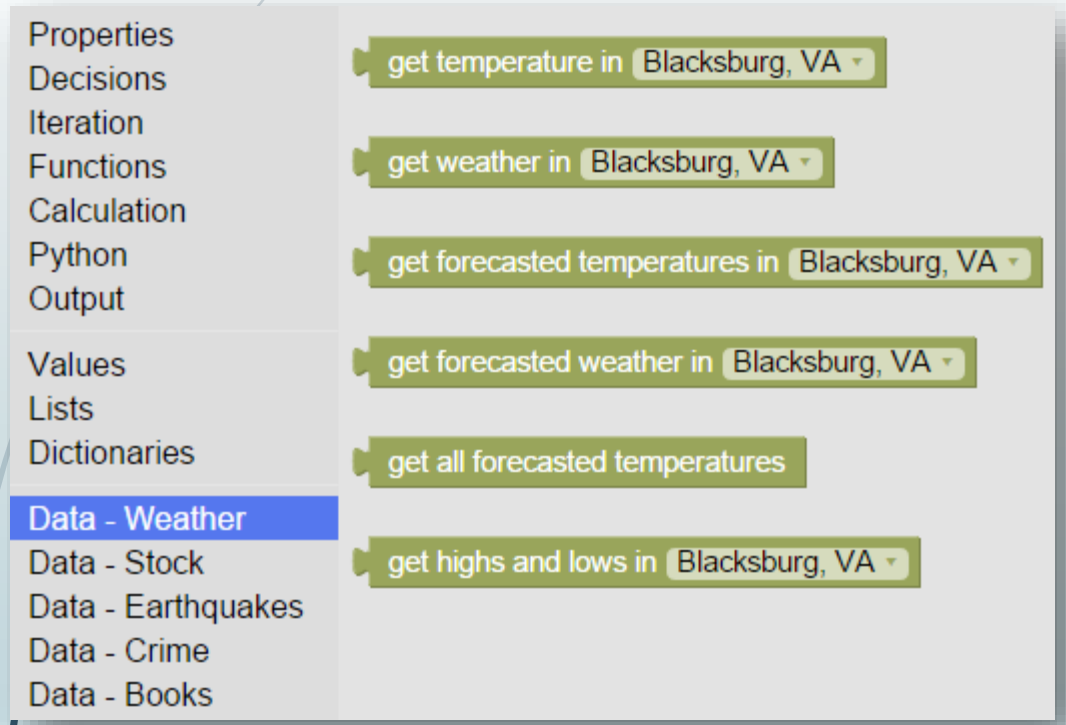
Learning Management System

Grade Tracking

Mutual Language Translation



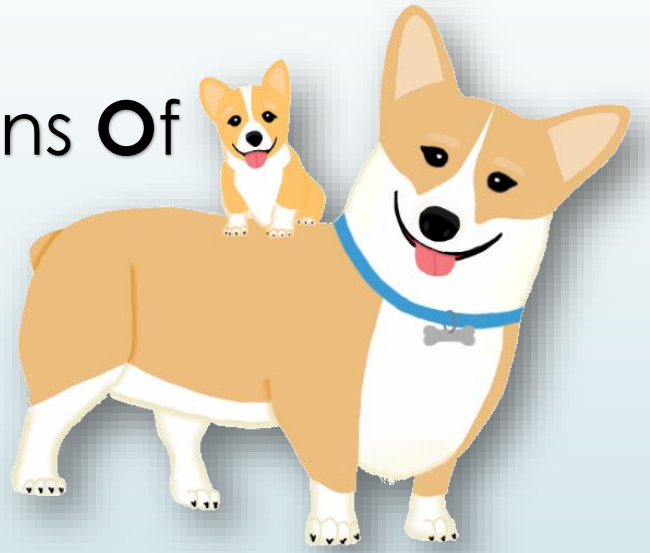
Data Blocks



Collections **O**f

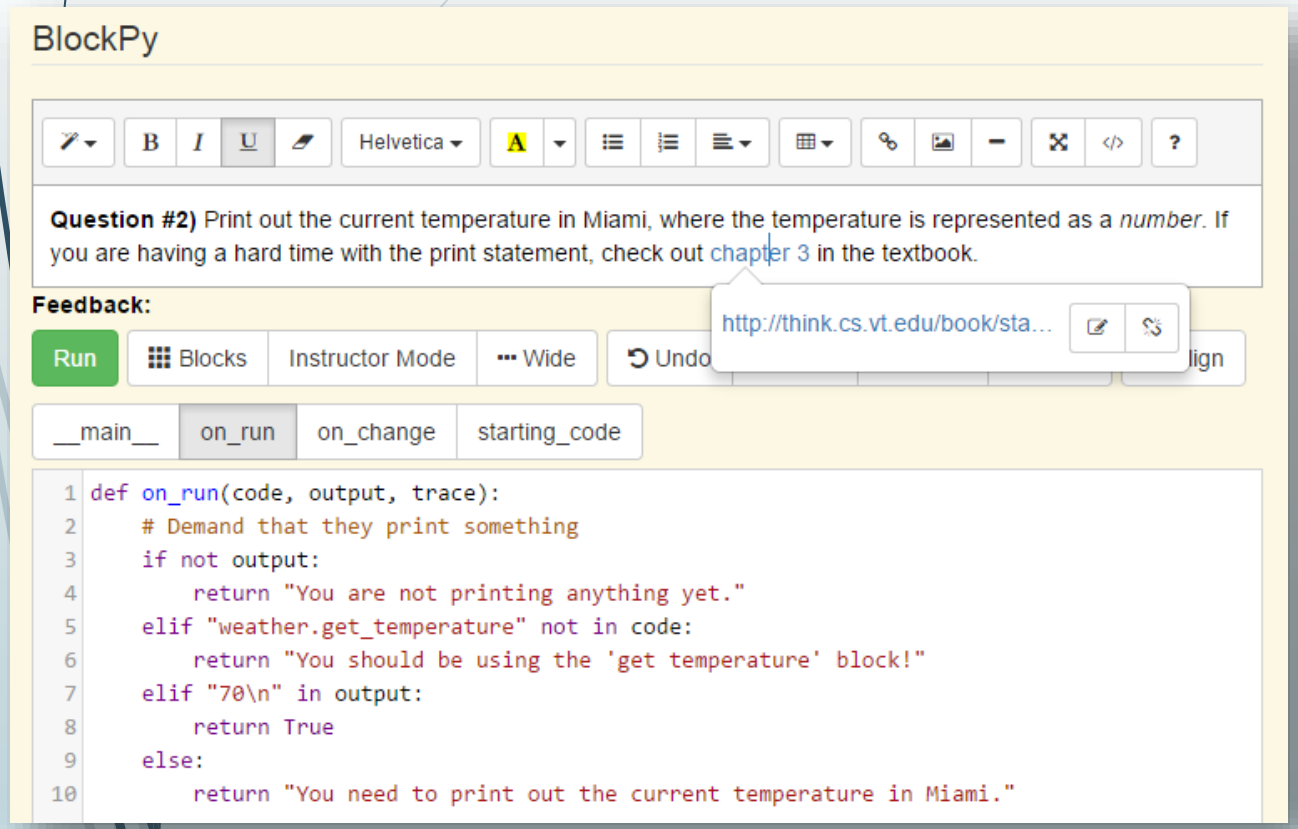
- **R**eal-time
- **G**iant
- **I**nteresting
- **S**ituated

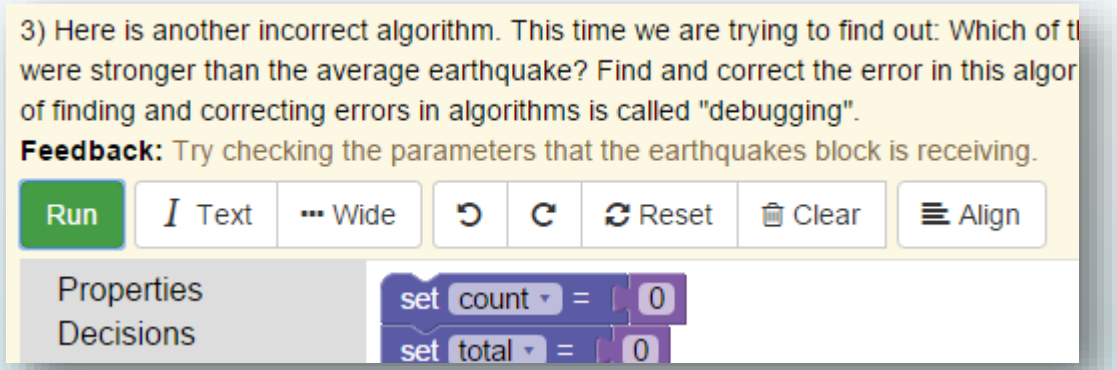
Datasets

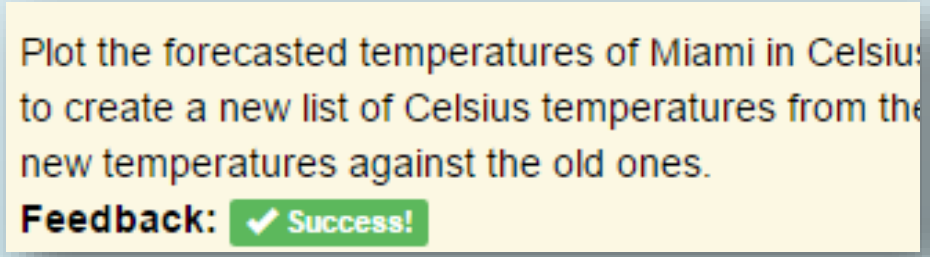


Tilevich, Eli, Clifford A. Shaffer, and Austin Cory Bart. "Creating stimulating, relevant, and manageable introductory computer science projects that utilize real-time, large, web-based datasets." *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, 2015.

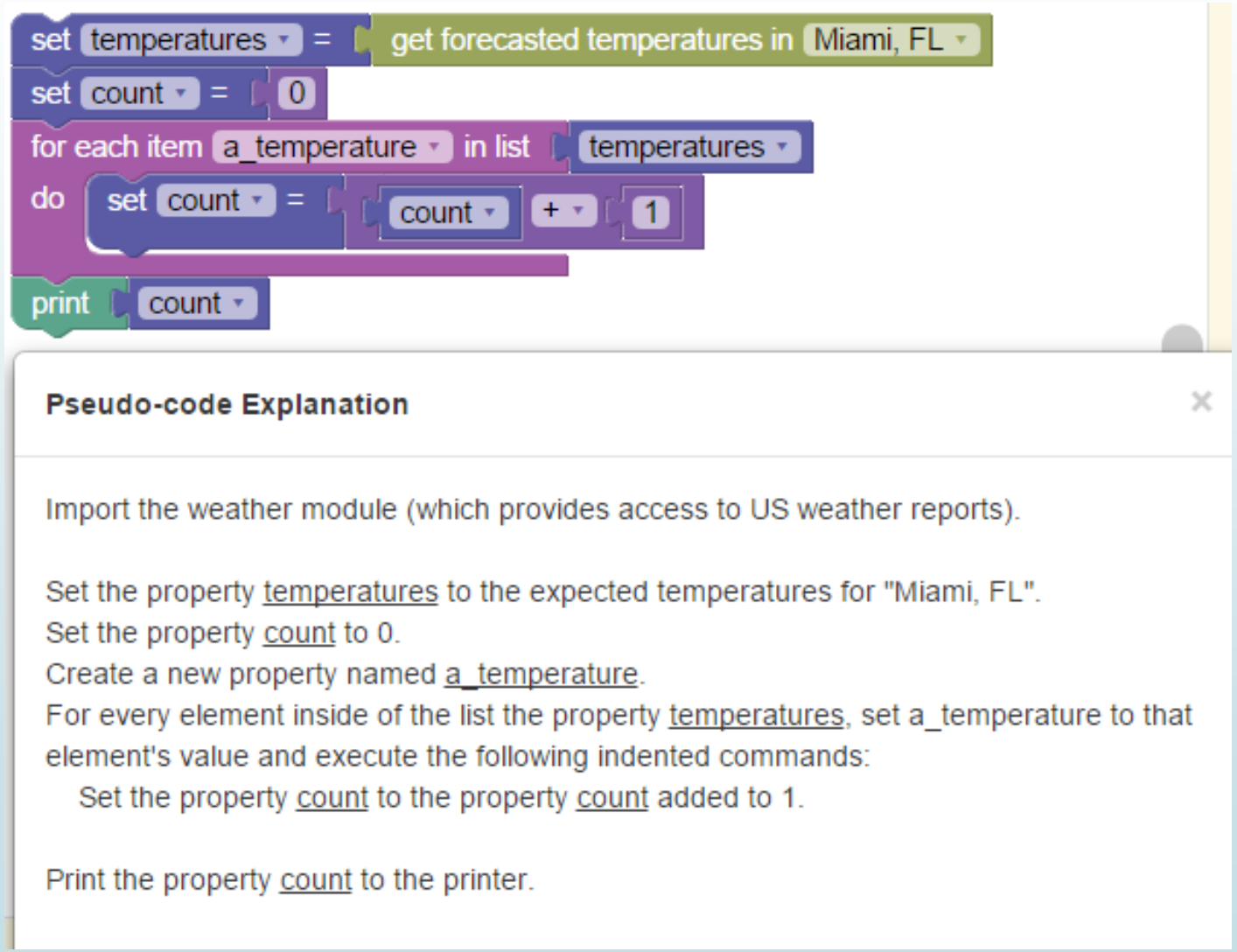
Guided Practice Problems



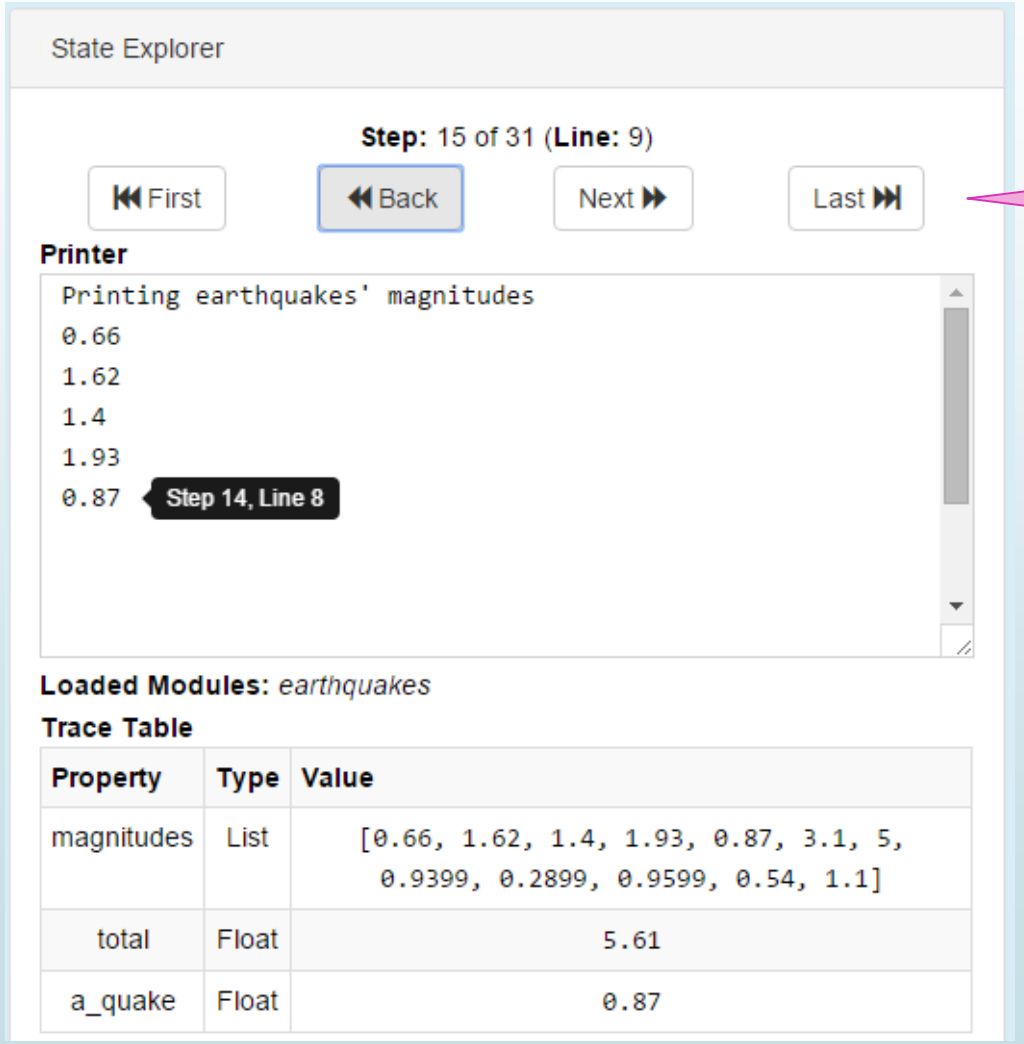




Pseudo-Code


set temperatures = get forecasted temperatures in Miami, FL
set count = 0
for each item a_temperature in list temperatures
do set count = count + 1
print count
Pseudo-code Explanation
Import the weather module (which provides access to US weather reports).
Set the property temperatures to the expected temperatures for "Miami, FL".
Set the property count to 0.
Create a new property named a_temperature.
For every element inside of the list the property temperatures, set a_temperature to that element's value and execute the following indented commands:
Set the property count to the property count added to 1.
Print the property count to the printer.

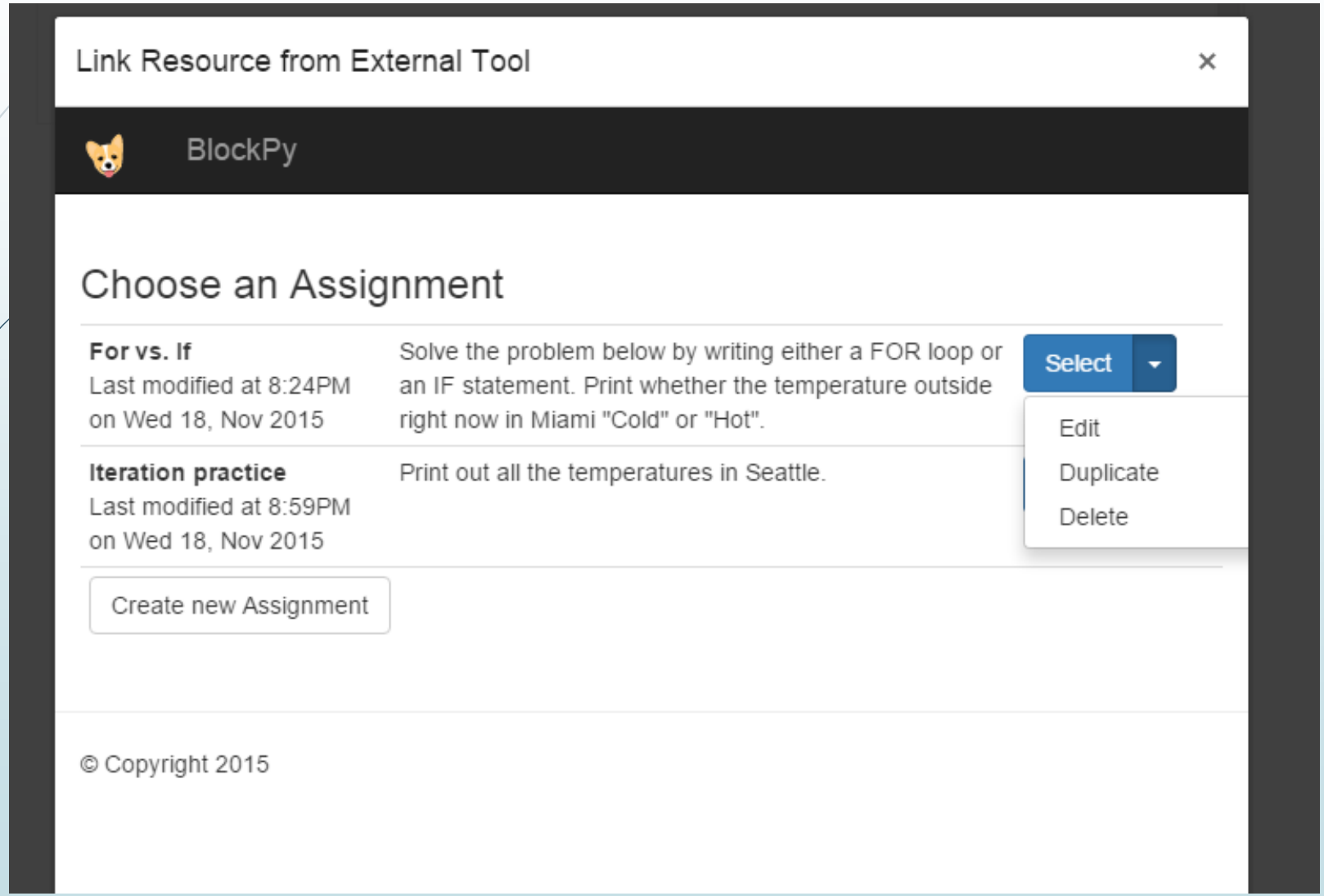
State Explorer



Trace step and line

Rewind variables AND console

LTI Interface



Evaluation

Spring 2015

- 35 students
- 4 days
- 57% female
- Version 1

Spring 2016

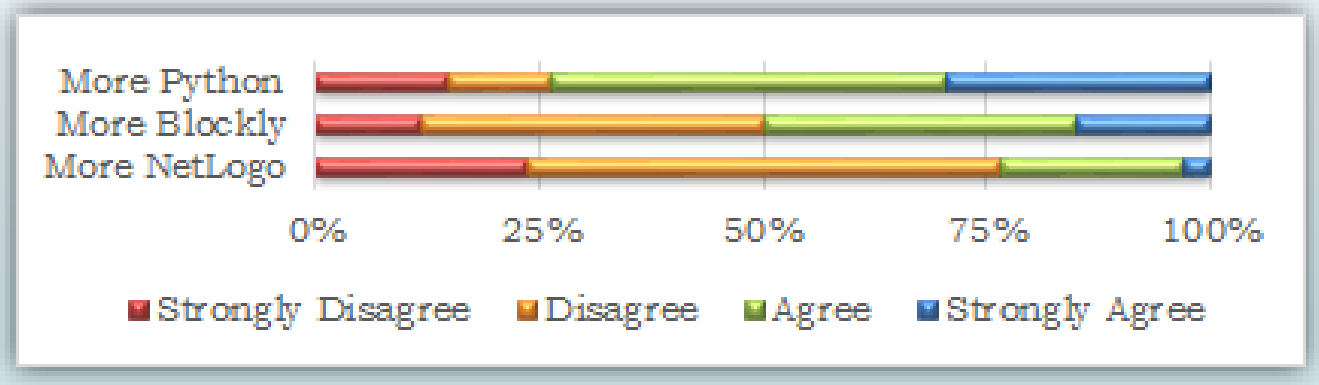
- 50 students
- 5 days
- 50% female
- Version 2

Two different post-surveys

Transition into Spyder

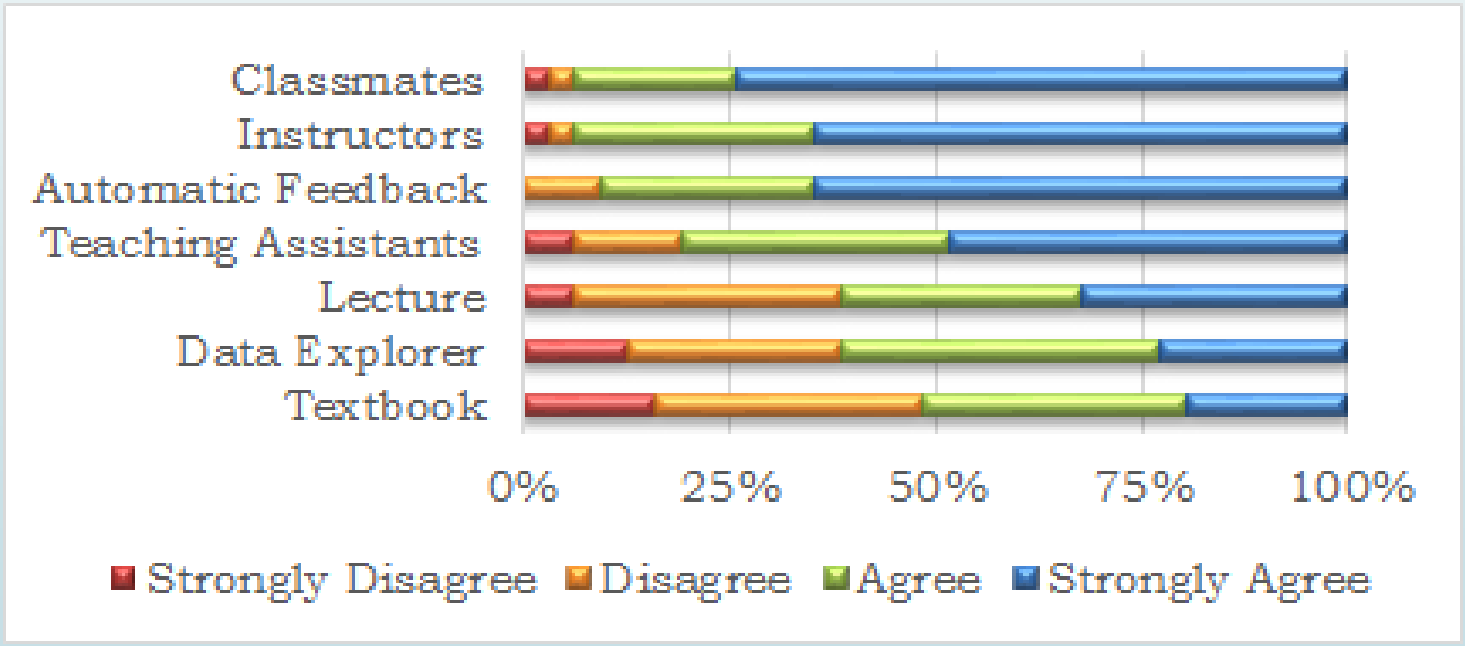
Spring 2015 - Perceptions

Do students want more time with NetLogo, BlockPy (Blockly), or Spyder (Python)?



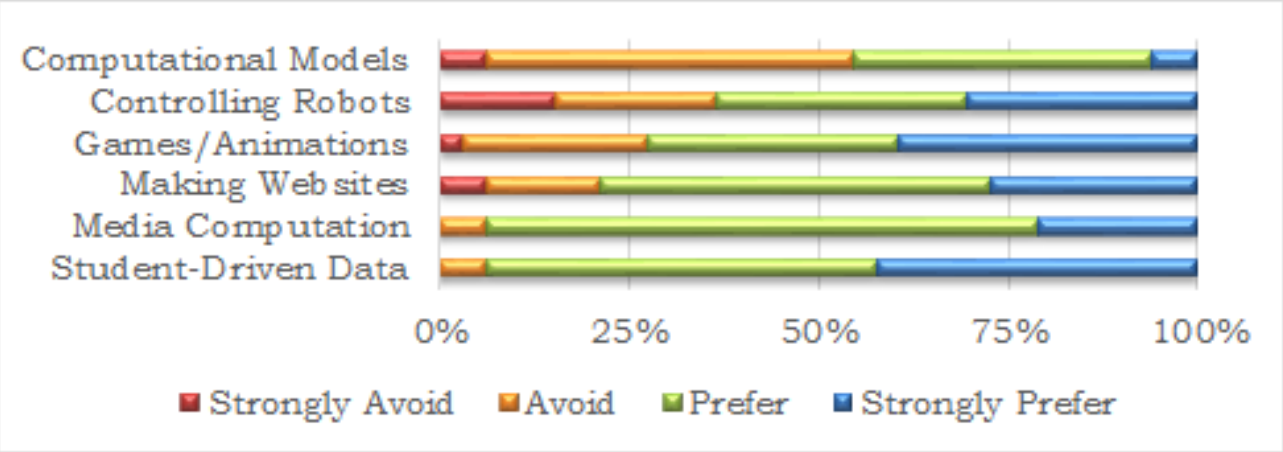
Spring 2015 – Post-Survey

How helpful to students' learning were these resources?

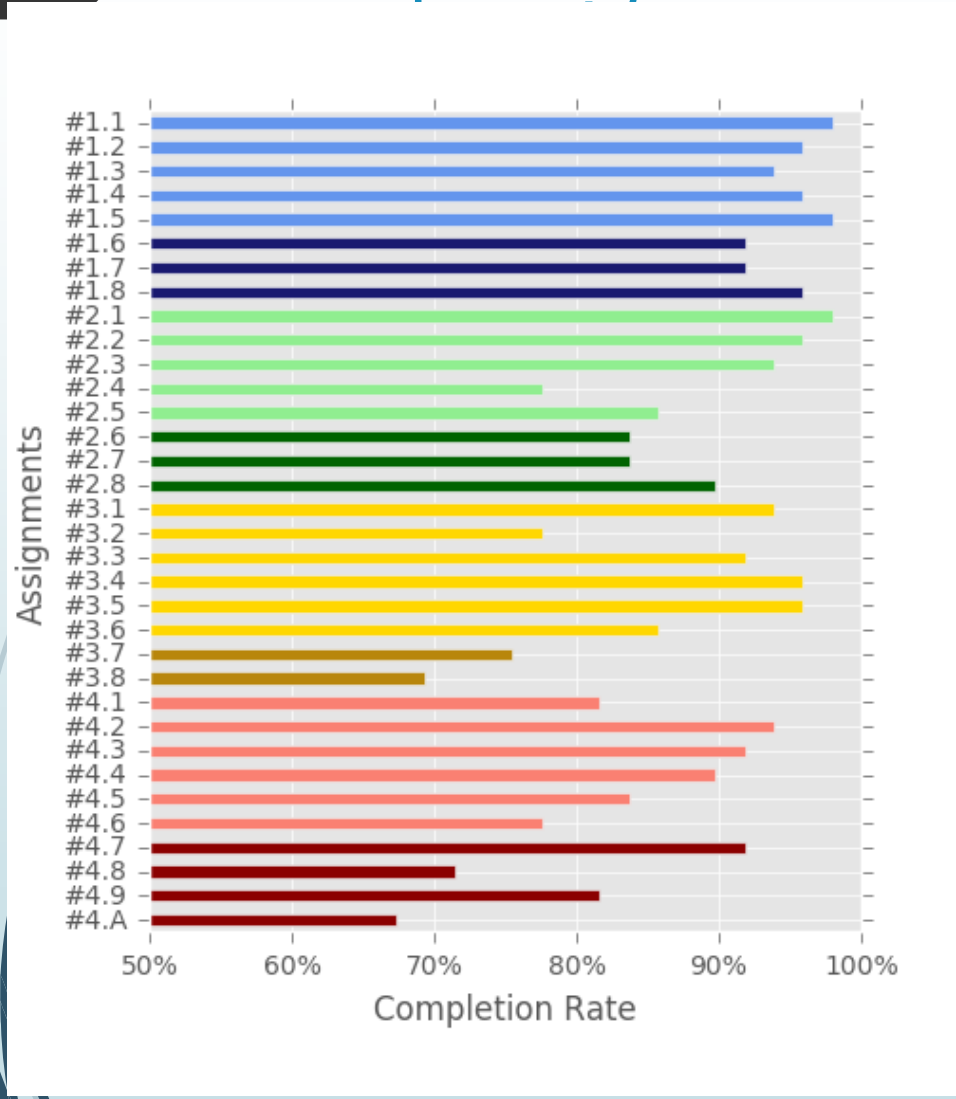


Spring 2015 – Contexts

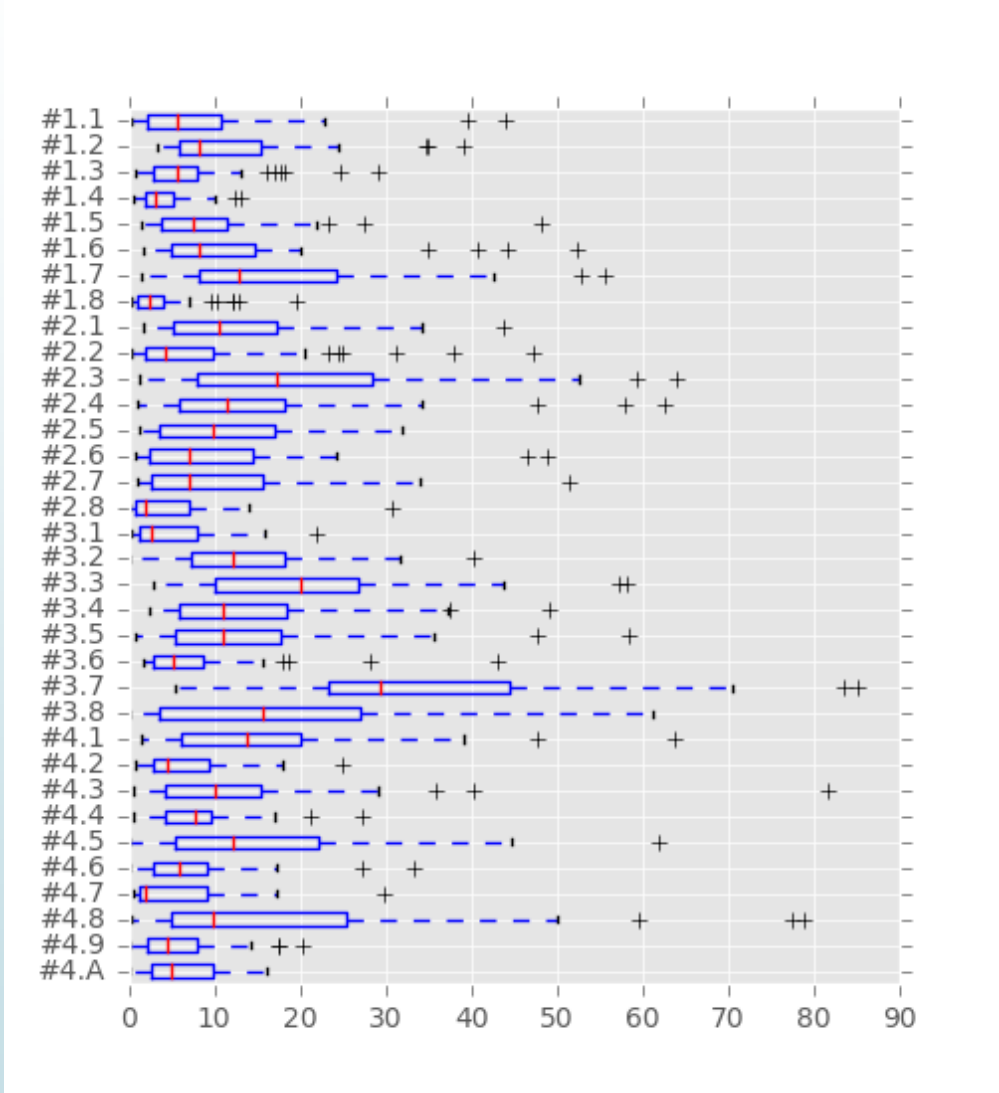
Which context seems most preferable to students?



Spring 2016 – Curriculum progress

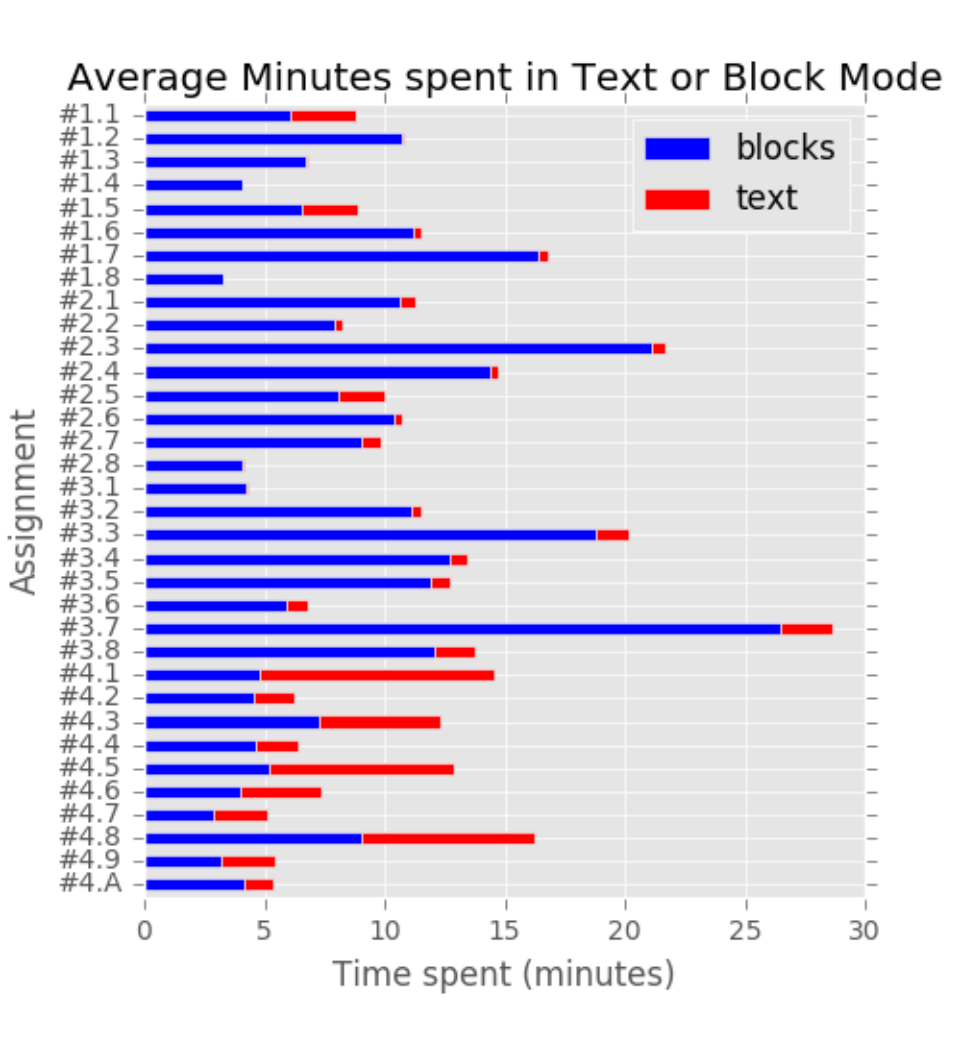


Most students completed most problems

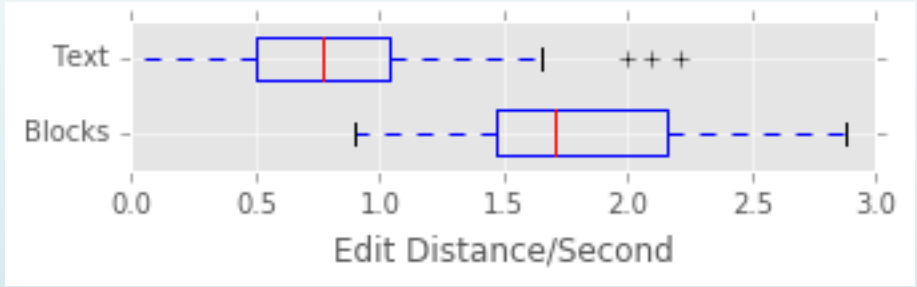


Most problems took about 15 minutes

Spring 2016 – Dual block/text

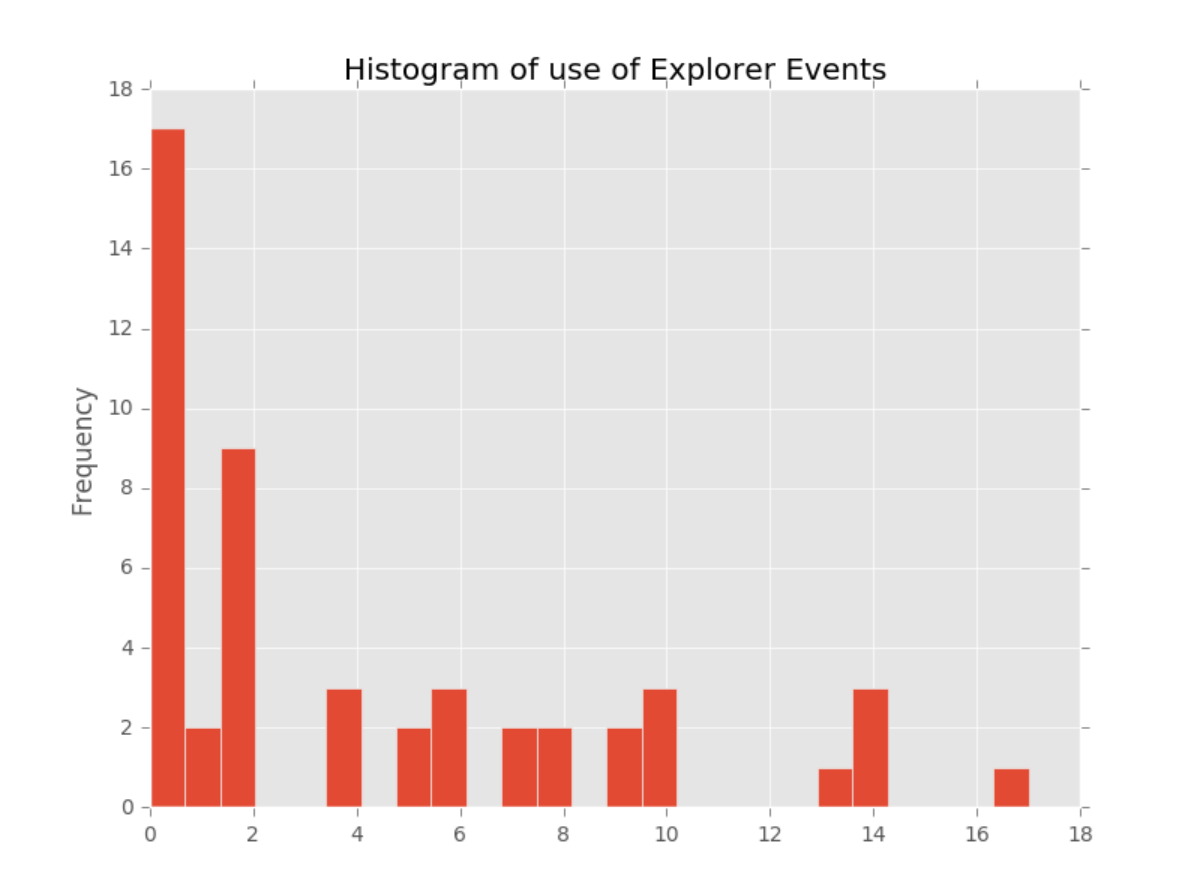


Students did transition to text



Blocks generate more code/sec

Spring 2016 – Tool events



Students use tools sparingly

Spring 2016 – Qualitative Results

- BlockPy => Spyder transition point was good
- Error reporting was better in Spyder
- 2/3 of students singled out blocks as helpful (only 12% singled out auto feedback)

Semantic Errors

- Semantic errors are slipping through

```
import stocks


stocks = stocks.get_past("FB")
new = []
if new:
    new = new + stocks
for stocks in stocks:
    print(stocks)
```

Dead codepath

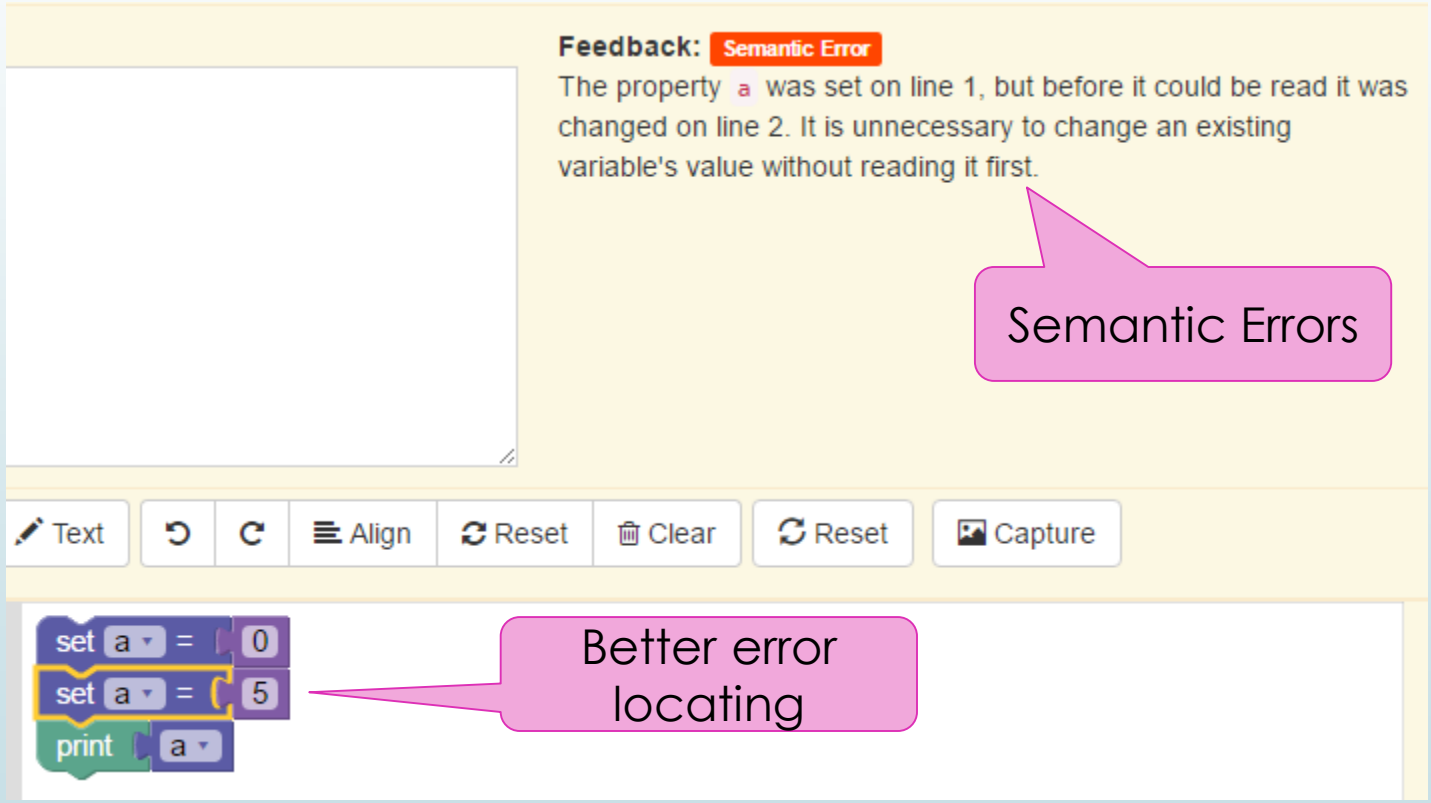
Incorrect appending

Iteration list is the same as the iteration variable

Looking forward

- How do we perfect the blocks?
- How can we give better feedback?
- How can we let students do better data science?

The Next Iteration



Conclusions

- BlockPy
  - Dual text/block
  - Auto feedback
  - Data Science
- For experimental curriculum

Thanks!

- Questions?