BROADENING PARTICIPATION IN COMPUTING

CS FOR ALL

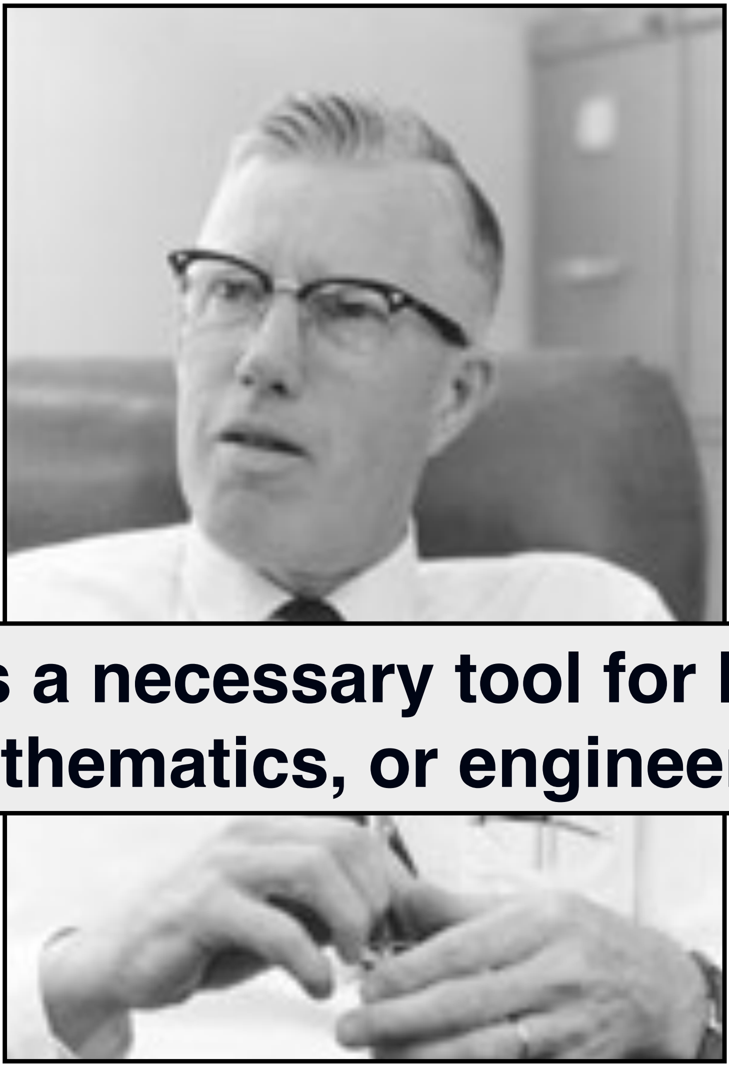
Story

- Computer science was originally invented to be taught to everyone
  - To help them learn other subjects
  - To support democratic ideals

- Our definition of computer science has become more narrow.
  - Computing education is now an *alternative endpoint*.

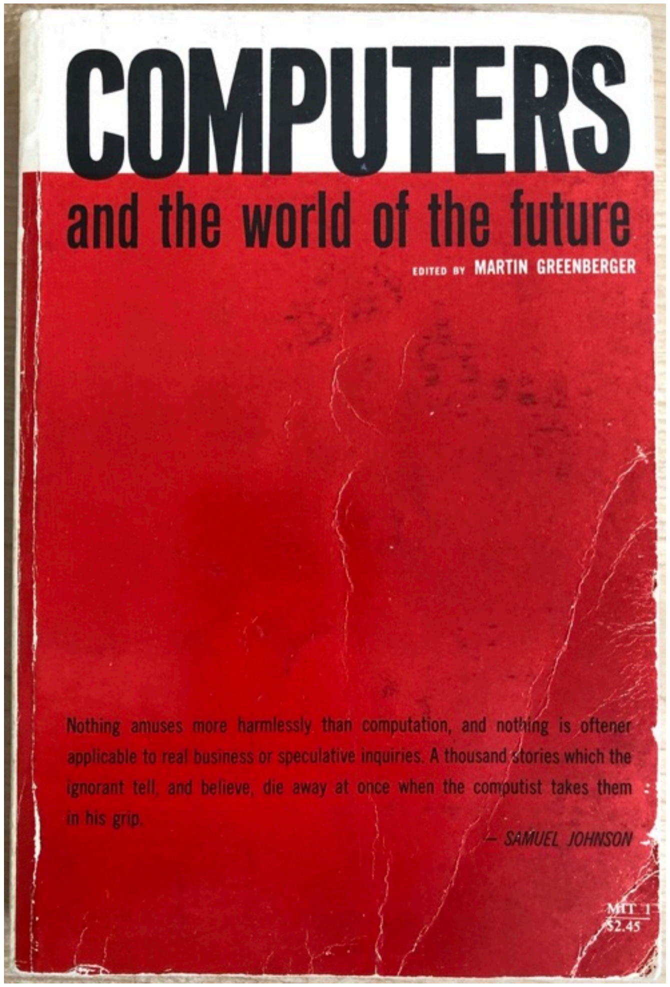
- Meeting the original goals of CS addresses both BPC and CSforAll

1961



The computer is a necessary tool for learning science, mathematics, or engineering

1968



COMPUTERS

and the world of the future

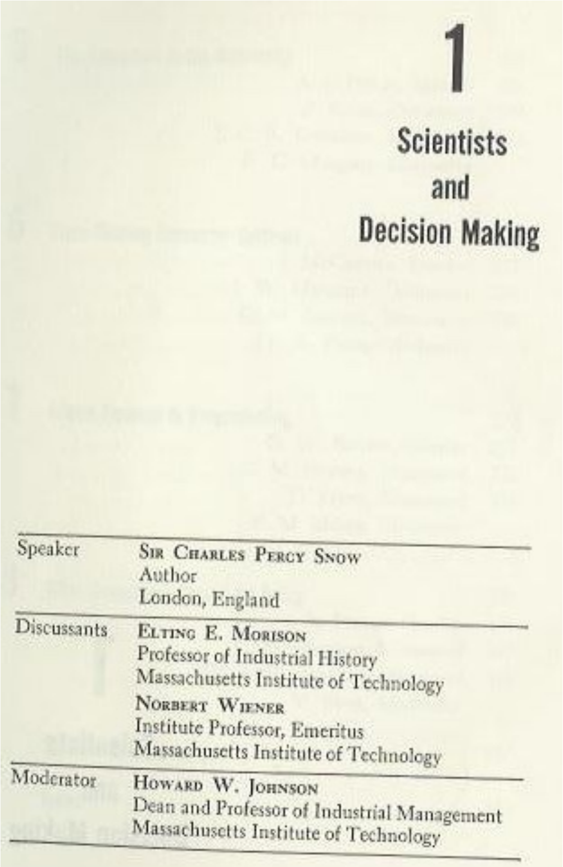
EDITED BY MARTIN GREENBERGER

Nothing amuses more harmlessly than computation, and nothing is oftener applicable to real business or speculative inquiries. A thousand stories which the ignorant tell, and believe, die away at once when the computist takes them in his grip.

— SAMUEL JOHNSON

MIT 1
$2.45

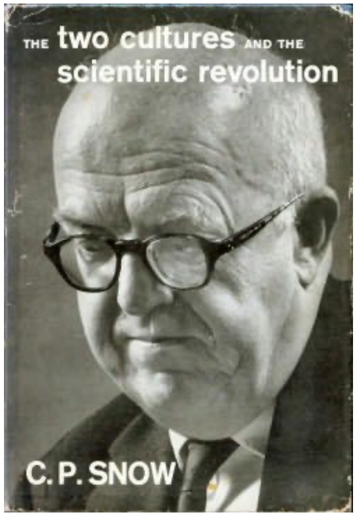
1961



1

Scientists and Decision Making

Speaker	Sir Charles Percy Snow Author London, England
Discussants	Elting E. Morison Professor of Industrial History Massachusetts Institute of Technology Norbert Wiener Institute Professor, Emeritus Massachusetts Institute of Technology
Moderator	Howard W. Johnson Dean and Professor of Industrial Management Massachusetts Institute of Technology



“A handful of people, having no relation to the will of society, having no communication with the rest of society, will be taking decisions in secret which are going to affect our lives in the deepest sense."

Peter Naur (1928-2016)

Turing Laureate (2005)

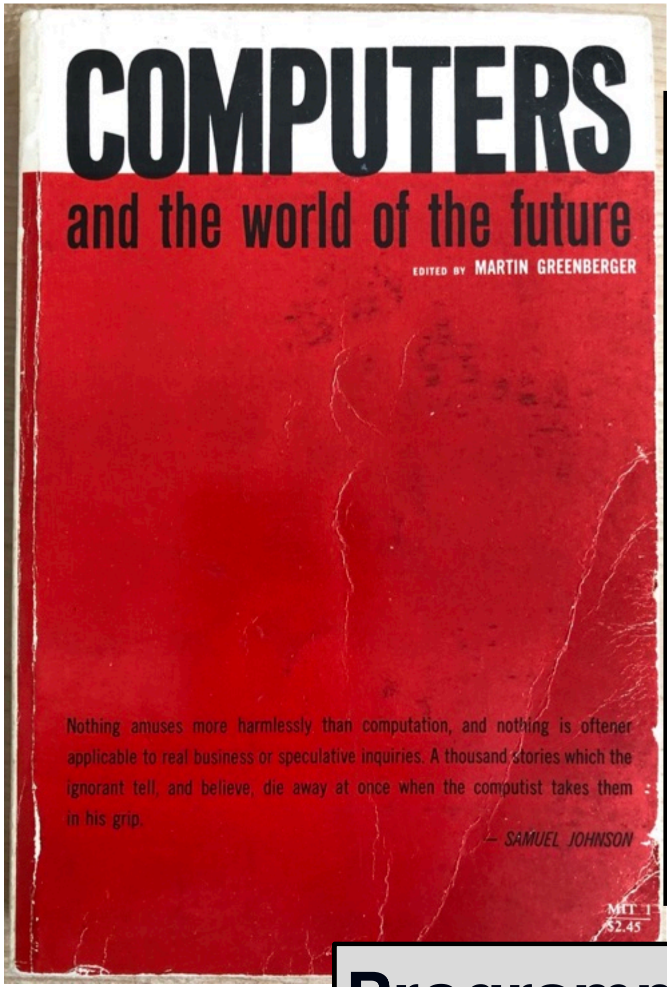
1966: Danmarks Radios Rosenkjærforelæsninger

Datalogi – læren om data

“Once informatics has become well established in general education, the mystery surrounding computers in many people’s perceptions will vanish. This must be regarded as perhaps the most important reason for promoting the understanding of informatics. This is a necessary condition for humankind’s supremacy over computers and for ensuring that their use do not become a matter for a small group of experts, but become a usual democratic matter, and thus through the democratic system will lie where it should, with all of us.”

fag.

Thanks to Michael Caspersen



COMPUTERS

and the world of the future

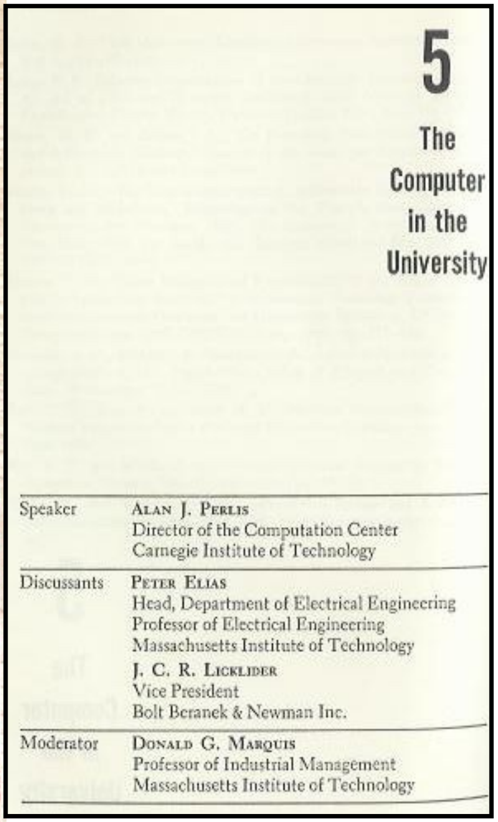
EDITED BY MARTIN GREENBERGER

Nothing amuses more harmlessly than computation, and nothing is oftener applicable to real business or speculative inquiries. A thousand stories which the ignorant tell, and believe, die away at once when the computist takes them in his grip.

— SAMUEL JOHNSON

MIT 1
$2.45

1961



5

The Computer in the University

Speaker	Alan J. Perlis Director of the Computation Center Carnegie Institute of Technology
Discussants	Peter Elias Head, Department of Electrical Engineering Professor of Electrical Engineering Massachusetts Institute of Technology J. C. R. Licklider Vice President Bolt Beranek & Newman Inc.
Moderator	Donald G. Marquis Professor of Industrial Management Massachusetts Institute of Technology

Alan Perlis

Photo: CMU

Programming changes how we understand

First published definition of Computer Science

“The study of computers and all the phenomena surrounding them.”

***Science*, 1967**

Alan Perlis

Herb Simon

Alan Newell

This is broader than how most people define computer science today.
Let’s call this *Computing*

Definitions of Computer Science

- Computer Science is the study of computers and computational systems. (Encyclopedia Brittanica)
- Computer science is the study of computers and algorithmic processes, including their principles, design, implementation, and impact on society. (Tucker, 2006 - K-12 CS Framework)
- Computer science is the foundational discipline with an emphasis on discovery related to programming, algorithms, and data structures. (ACM/IEEE Computing Curriculum 2021)

Definitions of Computer Science

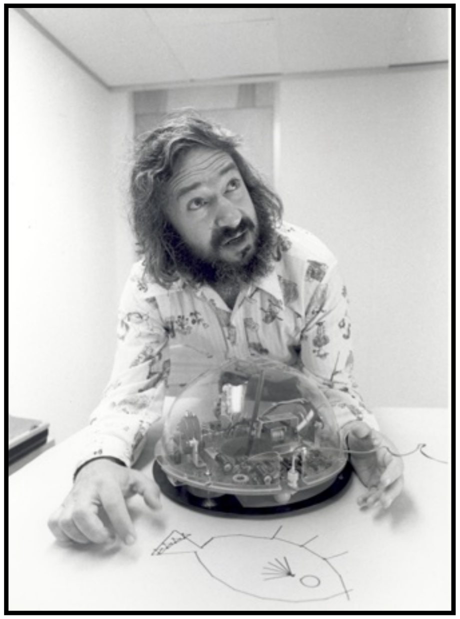
“The study of computers and all the phenomena surrounding them.” (Perlis, Newell, & Simon, 1967)

- Computer Science is the study of computers and computational systems. (Encyclopedia Brittanica)

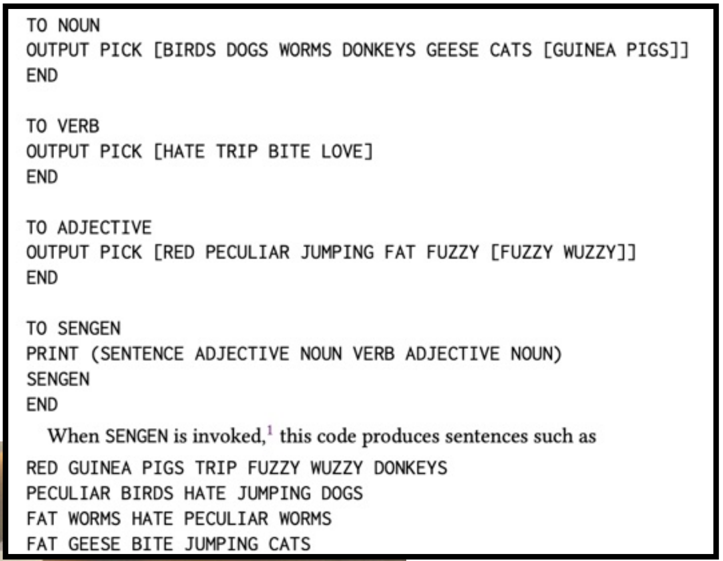
- Computer science is the study of computers and algorithmic processes, including their principles, design, implementation, and impact on society. (Tucker, 2006 - K-12 CS Framework)

- Computer science is the foundational discipline with an emphasis on discovery related to programming, algorithms, and data structures. (ACM/IEEE Computing Curriculum 2021)

Seymour Papert



Seymour Papert claimed “that children can learn to program and learning to program can affect the way that they learn everything else.”



```
TO NOUN
OUTPUT PICK [BIRDS DOGS WORMS DONKEYS GEESE CATS [GUINEA PIGS]]
END

TO VERB
OUTPUT PICK [HATE TRIP BITE LOVE]
END

TO ADJECTIVE
OUTPUT PICK [RED PECULIAR JUMPING FAT FUZZY [FUZZY WUZZY]]
END

TO SENGEN
PRINT (SENTENCE ADJECTIVE NOUN VERB ADJECTIVE NOUN)
SENGEN
END
```

When SENGEN is invoked,[1] this code produces sentences such as

```
RED GUINEA PIGS TRIP FUZZY WUZZY DONKEYS
PECULIAR BIRDS HATE JUMPING DOGS
FAT WORMS HATE PECULIAR WORMS
FAT GEESE BITE JUMPING CATS
```

1968

Thanks to Gary Stager

President Obama “CS for All”

2016

Computer science (CS) is a “new basic” skill necessary for economic opportunity and social mobility.

Photo: ObamaWhitehouse.archives.gov

When did this become about “economic opportunity”?

- Forsythe, Perlis, Snow, Naur, Simon, Papert, and Newell were all arguing for computing education ***decades*** before Silicon Valley.
  - They weren’t about preparing students for software development jobs.

- **“The Case for Alternative Endpoints”** (BJET, 2021)
  Mike Tissenbaum, David Weintrop, Nathan Holbert, Tamara Clegg

- **What are the goals of Computing Education, if not job skills?**

Who gets access to these powerful ideas?

WHO GETS COMPUTING EDUCATION TODAY

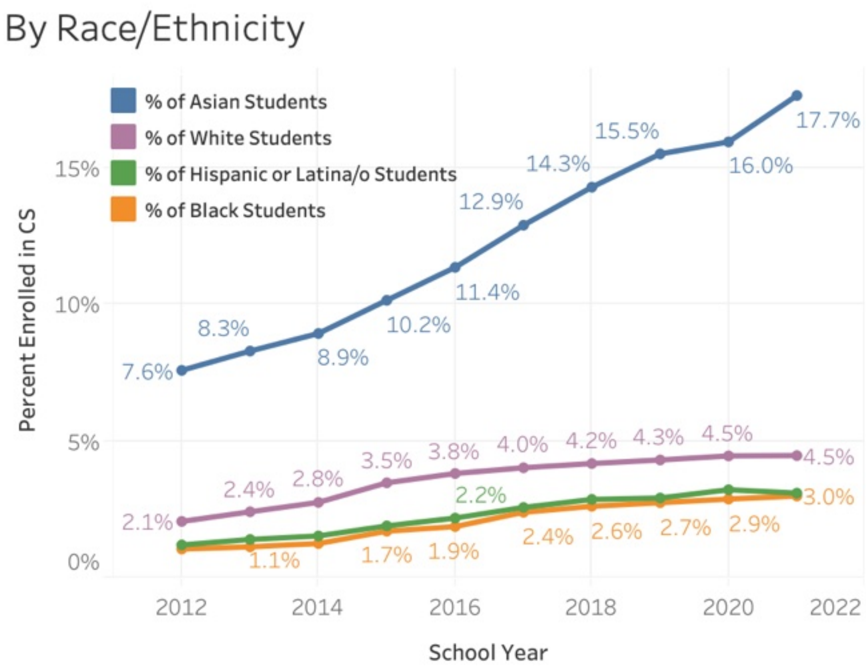
5.6%

https://advocacy.code.org/stateofcs

5.6%

Texas dashboard accessed through the ECEP State Dashboards page

By Race/Ethnicity



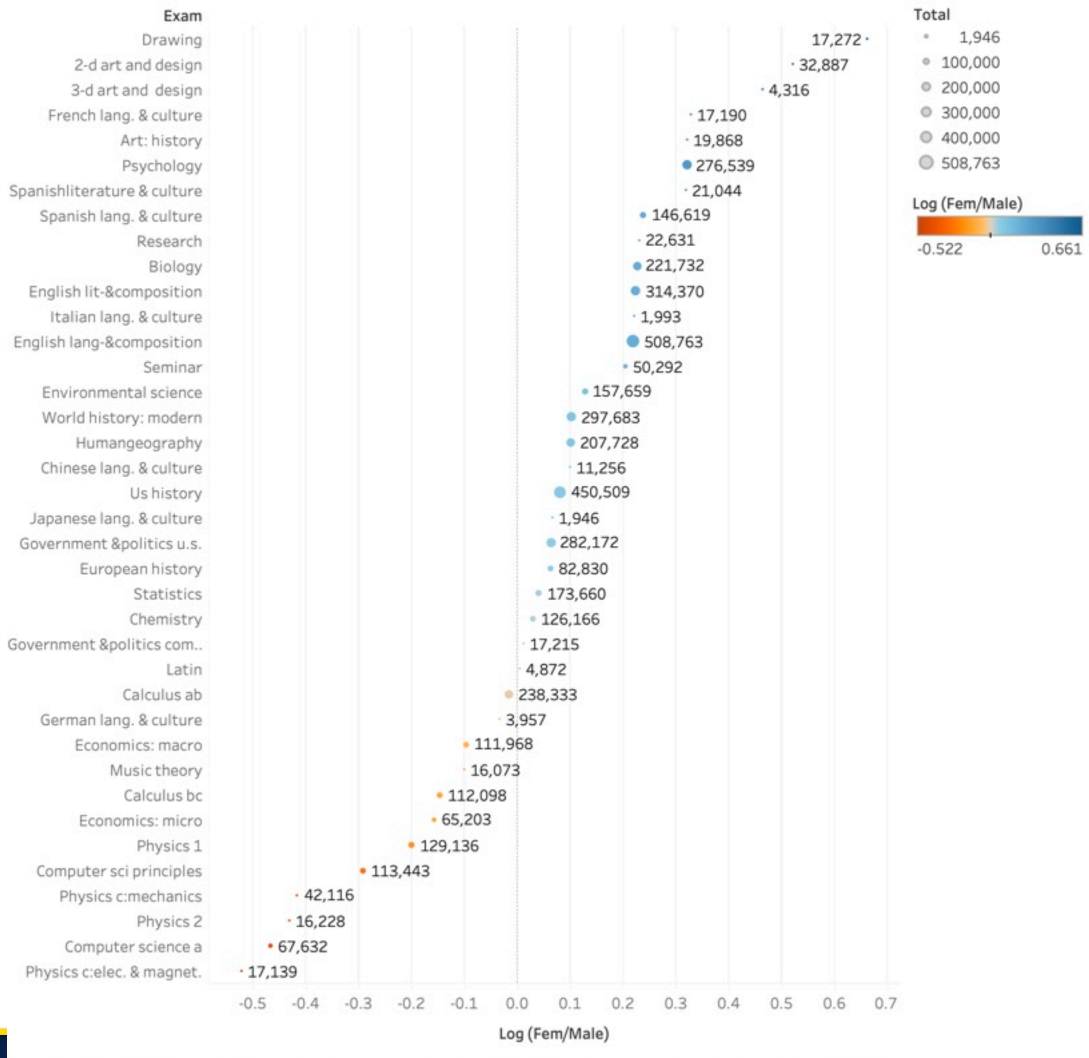
IN THE 2021-22 SCHOOL YEAR:

4.4% OF ALL HIGH SCHOOL STUDENTS TOOK A CS COURSE

5.6% OF STUDENTS AT SCHOOLS OFFERING CS TOOK A CS COURSE

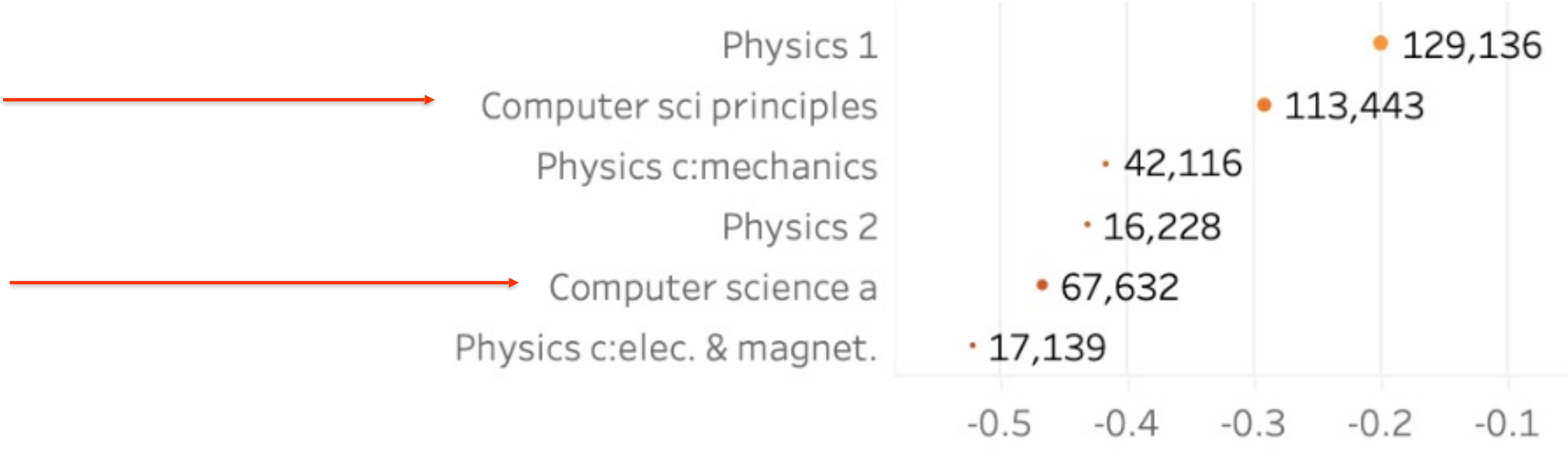
https://advocacy.code.org/stateofcs

Log (# Female / # Male) for Advanced Placement Exams in 2021



Sum of Log (Fem/Male) for each Exam. Color shows sum of Log (Fem/Male). Size shows sum of Total. The marks are labeled by sum of Total.

Data and Visualization from Barbara Ericson and Willa Hua



If we want to teach **Computer Science for All**, we have to teach where "All" are.

And that's not CS classes.

Data and Visualization from Barbara Ericson and Willa Hua

Broadening access and participation in computing

WHAT WE’RE TRYING AT MICHIGAN

M | LSA COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
UNIVERSITY OF MICHIGAN

Program in Computing for the Arts and Sciences

Mark Guzdial, mjguz@umich.edu

What does LSA need in Computing Education?

Dean Anne Curzan and Associate Dean for Undergrad Ed Tim McKay charged the Computing Education Task Force 2020-2021

- What do LSA students need to know about computing?
- What classes and programs already exist?
- Where should we be going?

- Conducted dozens of interviews, reviewed hundreds of courses, surveyed over 100 LSA faculty.

- Final report is available:

3 Themes for Computing Education in LSA

- ***Computing for Discovery***: Computational science enables new discoveries across natural and physical sciences.
- ***Computing for Expression***: Computing has changed how we communicate and engage with others, from social media to Pixar to AR/VR.
- ***Critical Computing,* or *Computing for Justice***: Computers and applications are pervasive in our daily lives, and thus have immense cultural, social, and political influence. Who is supported by computing, who is oppressed, and how can we create better models?

Program in Computing for the Arts and Sciences

Launched Summer 2022 - me and Gus Evrard, a first-generation computational cosmologist.

Our lecturer, Brian Miller, has a PhD in Music and worked as a data scientist.
Academic Program Manger, Tyrone Stewart, has a PhD in American Culture.
Faculty teaching in PCAS come from Linguistics and Anthropology.

Goals:

- To **meet the needs of all LSA students** to learn about computing, especially programming.
- To create **new computing courses** around the themes of justice, expression, and discovery.
- To **create new credentials** to enhance majors and provide computing-centric minors in all divisions

Developing Courses and Minors

Course code for PCAS: **COMPFOR** – COMPuting FOR…

First courses were taught last year:

- **COMPFOR 111 “Computing’s Impact on Justice: From Text to the Web”**
- **COMPFOR 121 “Computing for Creative Expression”**

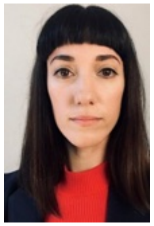
Courses introduced in Fall 2023:

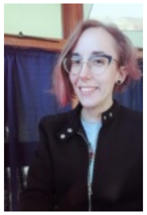
- COMPFOR 131 “Introduction to Python for the Sciences.” Worked with faculty across the Natural Sciences to develop the new course.
- COMPFOR 101 “The Transistor Disruption: How a Tiny Tool Transforms Society and Science”

- Winter 2024: Introduction to R for Scientists, Python Programming for Digital Media, “**A**l**l**en Anatomy: How ChatGPT works”
  Developing minors in **Expression** and **Discovery**.

What should be in these classes?

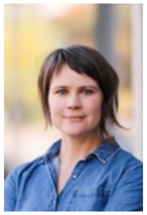
Formed advisory groups of faculty who self-identified as being about Computing for Expression or Computing for Justice.



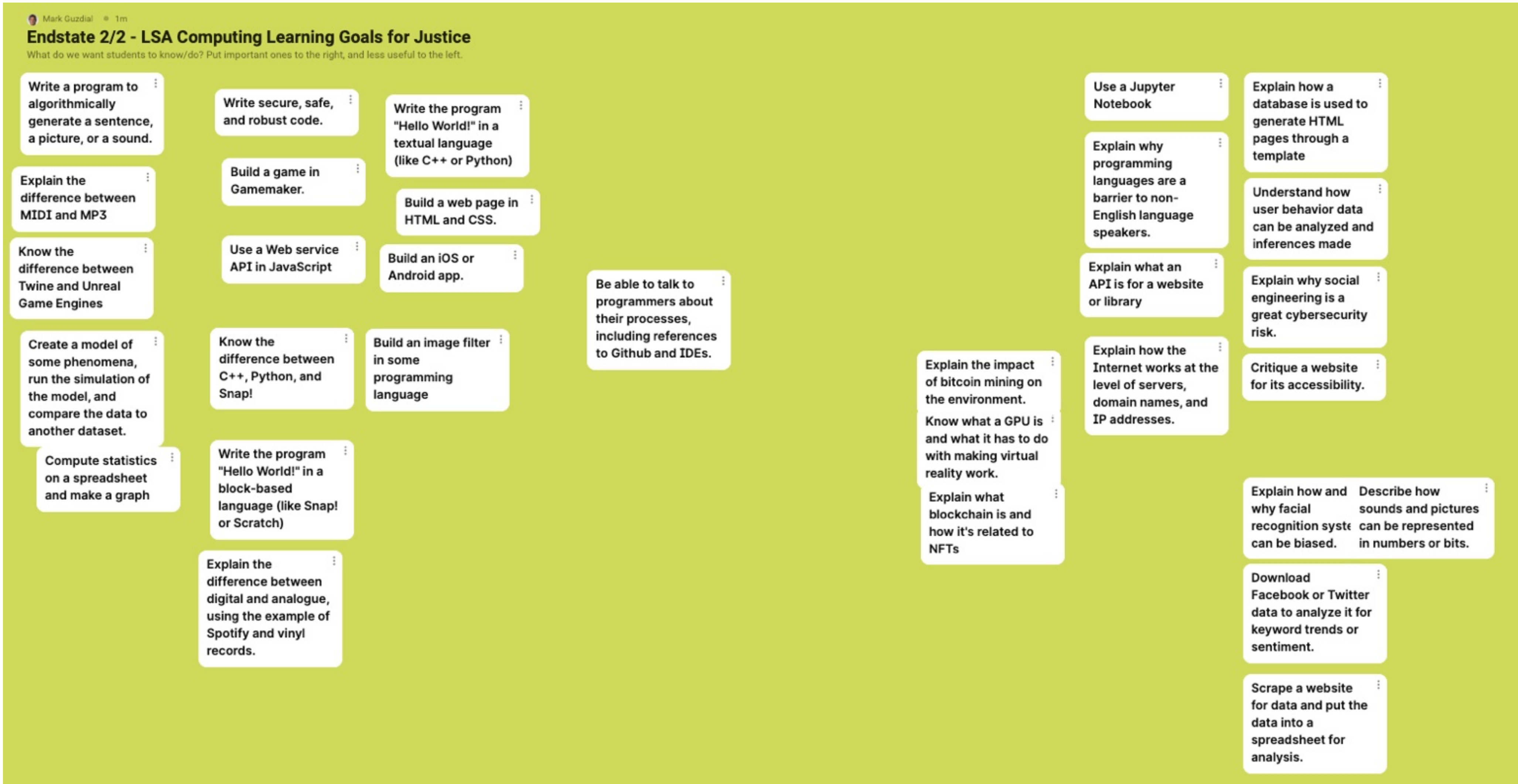


Created two sets of shared whiteboards of:

1. Learning objectives that were identified by the computing education task force
2. Student activities to support reaching those learning objectives.



- Instructions: “Please move to the right those that you think are important for the course, and move the left those that you think are less or unimportant for the course.”



Mark Guzdial • 1m

Endstate 2/2 - LSA Computing Learning Goals for Justice

What do we want students to know/do? Put important ones to the right, and less useful to the left.

Write a program to algorithmically generate a sentence, a picture, or a sound.

Explain the difference between MIDI and MP3

Know the difference between Twine and Unreal Game Engines

Create a model of some phenomena, run the simulation of the model, and compare the data to another dataset.

Compute statistics on a spreadsheet and make a graph

Write secure, safe, and robust code.

Build a game in Gamemaker.

Use a Web service API in JavaScript

Know the difference between C++, Python, and Snap!

Write the program "Hello World!" in a block-based language (like Snap! or Scratch)

Explain the difference between digital and analogue, using the example of Spotify and vinyl records.

Write the program "Hello World!" in a textual language (like C++ or Python)

Build a web page in HTML and CSS.

Build an iOS or Android app.

Build an image filter in some programming language

Be able to talk to programmers about their processes, including references to Github and IDEs.

Explain the impact of bitcoin mining on the environment.

Know what a GPU is and what it has to do with making virtual reality work.

Explain what blockchain is and how it's related to NFTs

Use a Jupyter Notebook

Explain why programming languages are a barrier to non-English language speakers.

Explain what an API is for a website or library

Explain how the Internet works at the level of servers, domain names, and IP addresses.

Explain how a database is used to generate HTML pages through a template

Understand how user behavior data can be analyzed and inferences made

Explain why social engineering is a great cybersecurity risk.

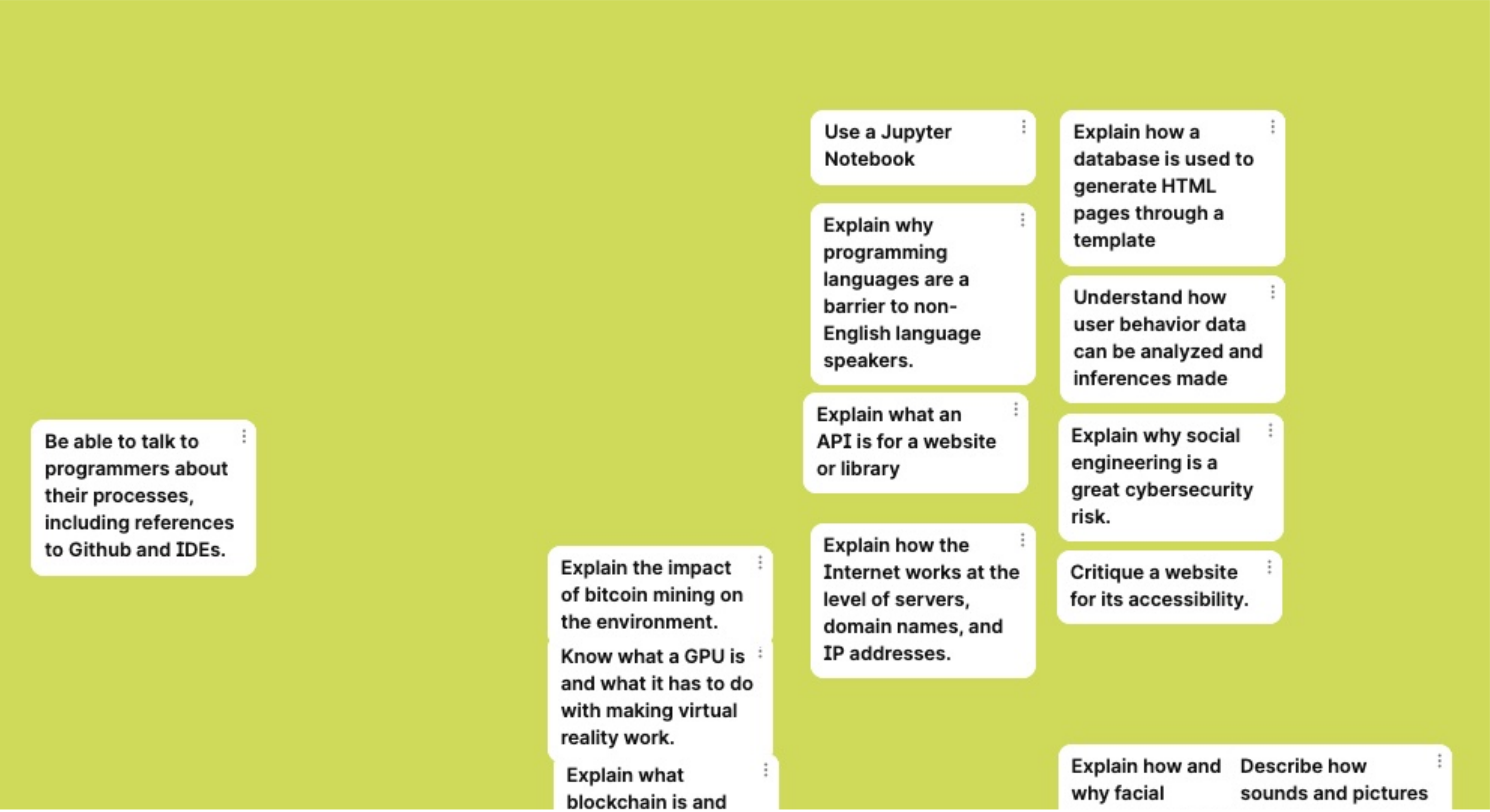
Critique a website for its accessibility.

Explain how and why facial recognition syste can be biased.

Describe how sounds and pictures can be represented in numbers or bits.

Download Facebook or Twitter data to analyze it for keyword trends or sentiment.

Scrape a website for data and put the data into a spreadsheet for analysis.



Be able to talk to programmers about their processes, including references to Github and IDEs.

Explain the impact of bitcoin mining on the environment.

Know what a GPU is and what it has to do with making virtual reality work.

Explain what blockchain is and

Use a Jupyter Notebook

Explain why programming languages are a barrier to non-English language speakers.

Explain what an API is for a website or library

Explain how the Internet works at the level of servers, domain names, and IP addresses.

Explain how a database is used to generate HTML pages through a template

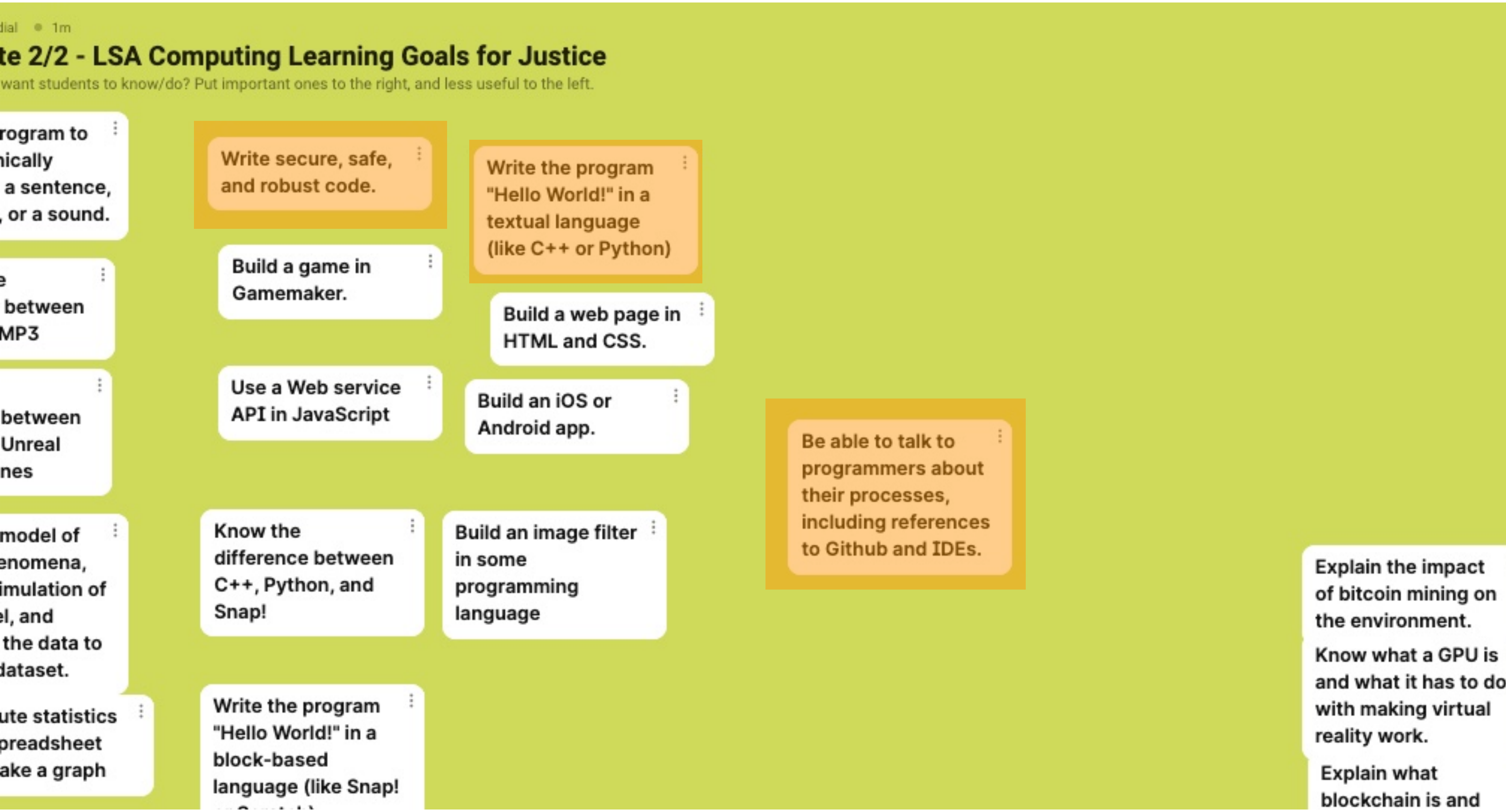
Understand how user behavior data can be analyzed and inferences made

Explain why social engineering is a great cybersecurity risk.

Critique a website for its accessibility.

Explain how and why facial

Describe how sounds and pictures



dial • 1m

te 2/2 - LSA Computing Learning Goals for Justice

want students to know/do? Put important ones to the right, and less useful to the left.

rogram to
nically
a sentence,
, or a sound.

e
between
MP3

between
Unreal
nes

model of
enomena,
imulation of
el, and
the data to
dataset.

ute statistics
preadsheet
ake a graph

Write secure, safe, and robust code.

Build a game in Gamemaker.

Use a Web service API in JavaScript

Know the difference between C++, Python, and Snap!

Write the program "Hello World!" in a block-based language (like Snap!

Write the program "Hello World!" in a textual language (like C++ or Python)

Build a web page in HTML and CSS.

Build an iOS or Android app.

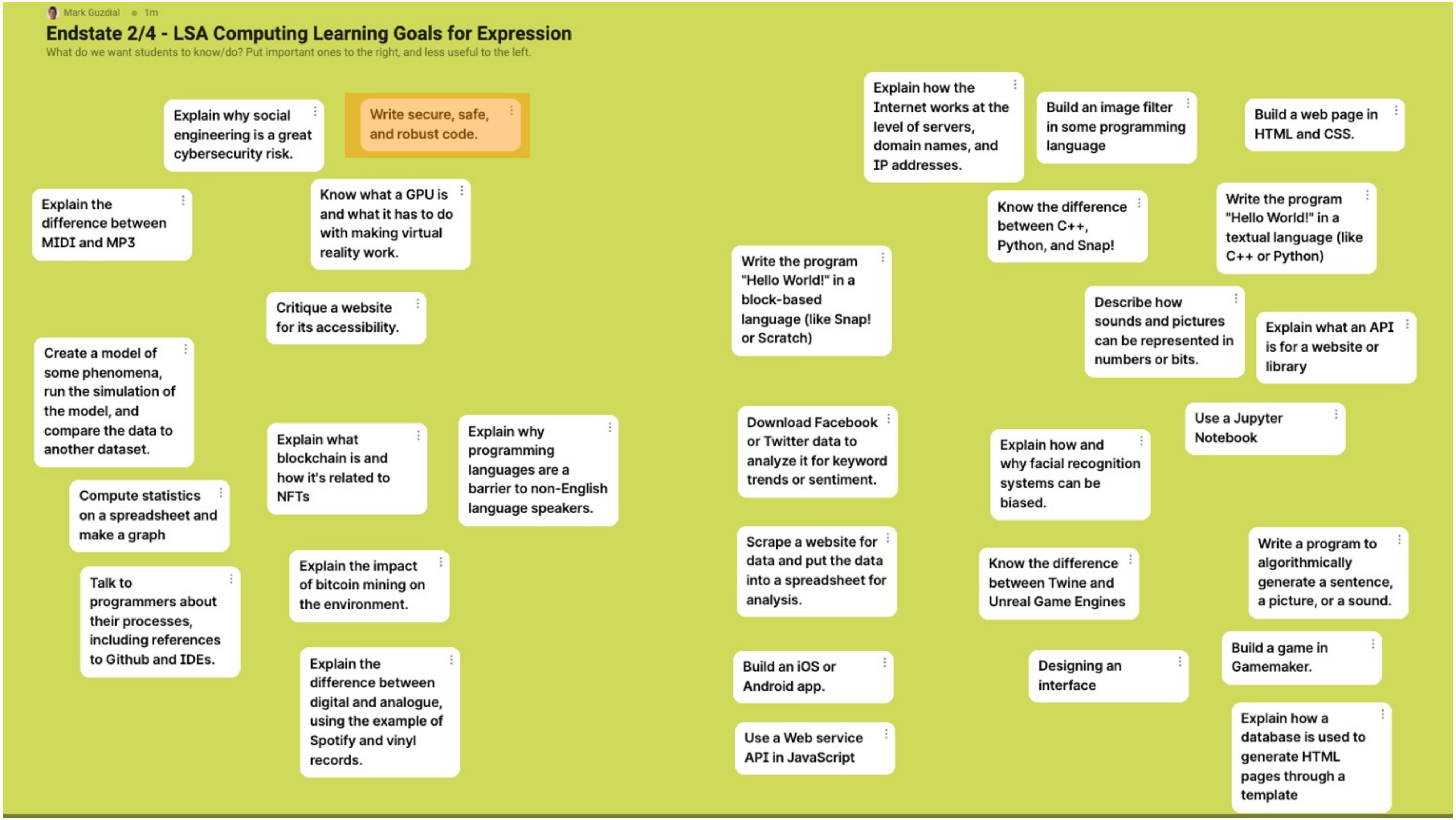
Build an image filter in some programming language

Be able to talk to programmers about their processes, including references to Github and IDEs.

Explain the impact of bitcoin mining on the environment.

Know what a GPU is and what it has to do with making virtual reality work.

Explain what blockchain is and



Mark Guzdial • 1m

Endstate 2/4 - LSA Computing Learning Goals for Expression

What do we want students to know/do? Put important ones to the right, and less useful to the left.

Explain the difference between MIDI and MP3

Create a model of some phenomena, run the simulation of the model, and compare the data to another dataset.

Compute statistics on a spreadsheet and make a graph

Talk to programmers about their processes, including references to Github and IDEs.

Explain why social engineering is a great cybersecurity risk.

Know what a GPU is and what it has to do with making virtual reality work.

Critique a website for its accessibility.

Explain what blockchain is and how it's related to NFTs

Explain the impact of bitcoin mining on the environment.

Explain the difference between digital and analogue, using the example of Spotify and vinyl records.

Write secure, safe, and robust code.

Explain why programming languages are a barrier to non-English language speakers.

Write the program "Hello World!" in a block-based language (like Snap! or Scratch)

Download Facebook or Twitter data to analyze it for keyword trends or sentiment.

Scrape a website for data and put the data into a spreadsheet for analysis.

Build an iOS or Android app.

Use a Web service API in JavaScript

Explain how the Internet works at the level of servers, domain names, and IP addresses.

Know the difference between C++, Python, and Snap!

Explain how and why facial recognition systems can be biased.

Know the difference between Twine and Unreal Game Engines

Designing an interface

Build an image filter in some programming language

Describe how sounds and pictures can be represented in numbers or bits.

Use a Jupyter Notebook

Build a web page in HTML and CSS.

Write the program "Hello World!" in a textual language (like C++ or Python)

Explain what an API is for a website or library

Write a program to algorithmically generate a sentence, a picture, or a sound.

Build a game in Gamemaker.

Explain how a database is used to generate HTML pages through a template

An Alternative Path

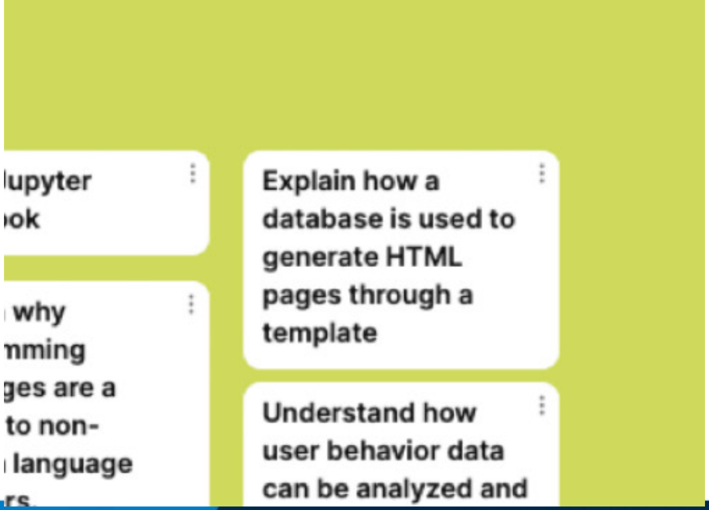
What do Humanities Scholars want Students to Know about the Internet?

History Professor, LaKisha Simmons gave me a list:

1. There are things called databases.

2. That databases, if they are designed well, are easy to index and to find information in.

3. Databases could be used to automatically generate Web pages.



upyter
ok

why
nming
ges are a
to non-
language
rs.

Explain how a database is used to generate HTML pages through a template

Understand how user behavior data can be analyzed and

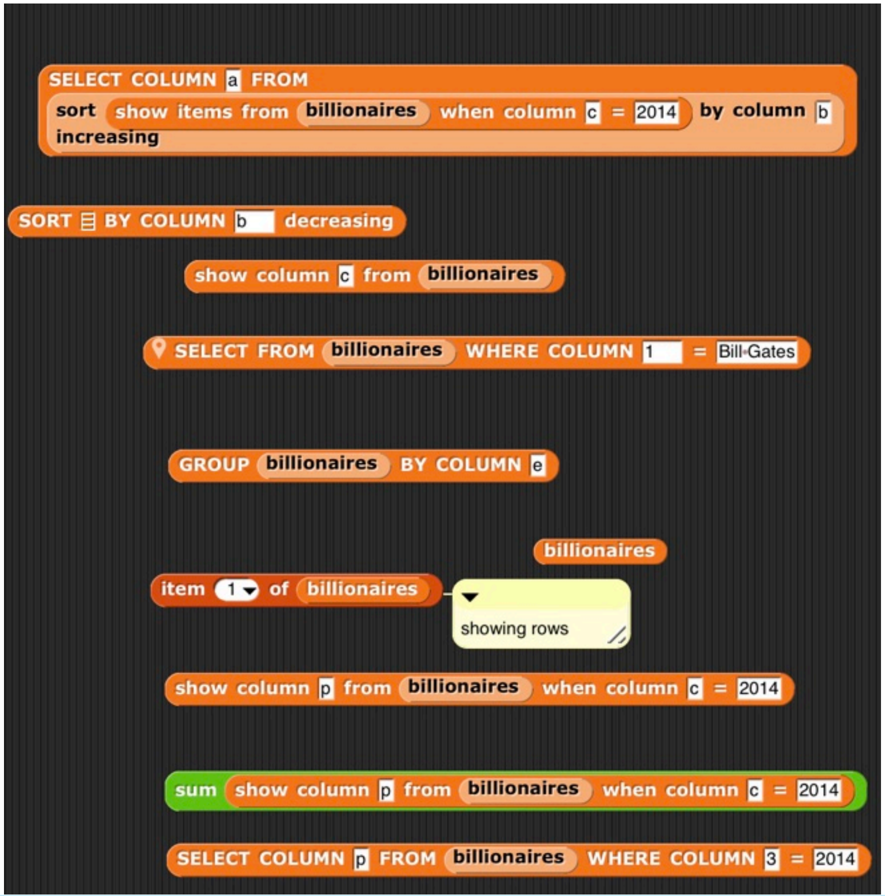
These are “advanced” topics in undergraduate CS

- No CS program ***starts*** there.
  - Everyone starts with introductory programming,
    then data structures and algorithms,
    then…

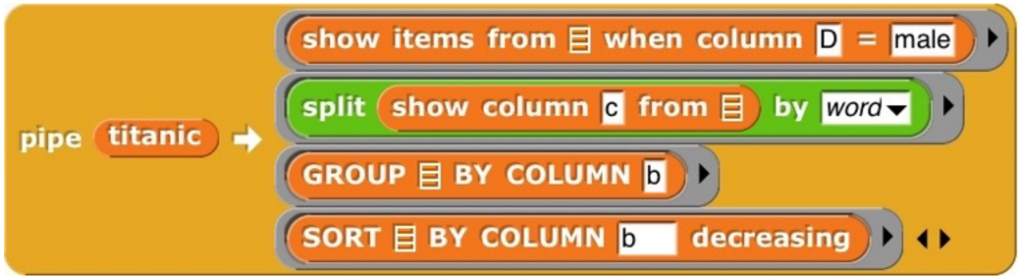
- Do we have to?

- Alternative Paths for Alternative Endpoints

Supporting CSV files as databases



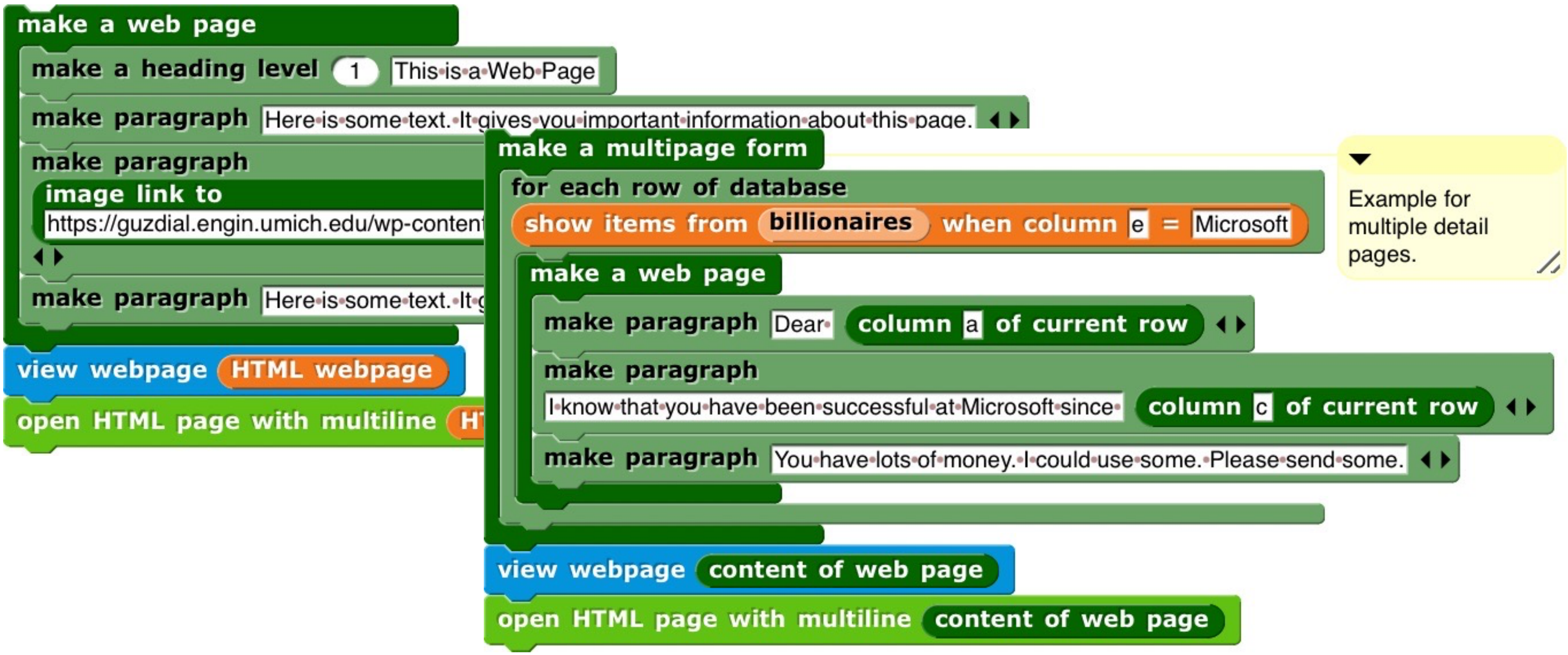
```
SELECT COLUMN a FROM
sort show items from billionaires when column c = 2014 by column b
increasing

SORT ⌸ BY COLUMN b decreasing

show column c from billionaires

SELECT FROM billionaires WHERE COLUMN 1 = Bill Gates

GROUP billionaires BY COLUMN e

billionaires

item 1 of billionaires
showing rows

show column p from billionaires when column c = 2014

sum show column p from billionaires when column c = 2014

SELECT COLUMN p FROM billionaires WHERE COLUMN 3 = 2014
```



```
pipe titanic ➜ show items from ⌸ when column D = male
               split show column c from ⌸ by word
               GROUP ⌸ BY COLUMN b
               SORT ⌸ BY COLUMN b decreasing
```

Thanks to Fuchun Wang

But how are we going to do Web pages?   Custom Snap Blocks!

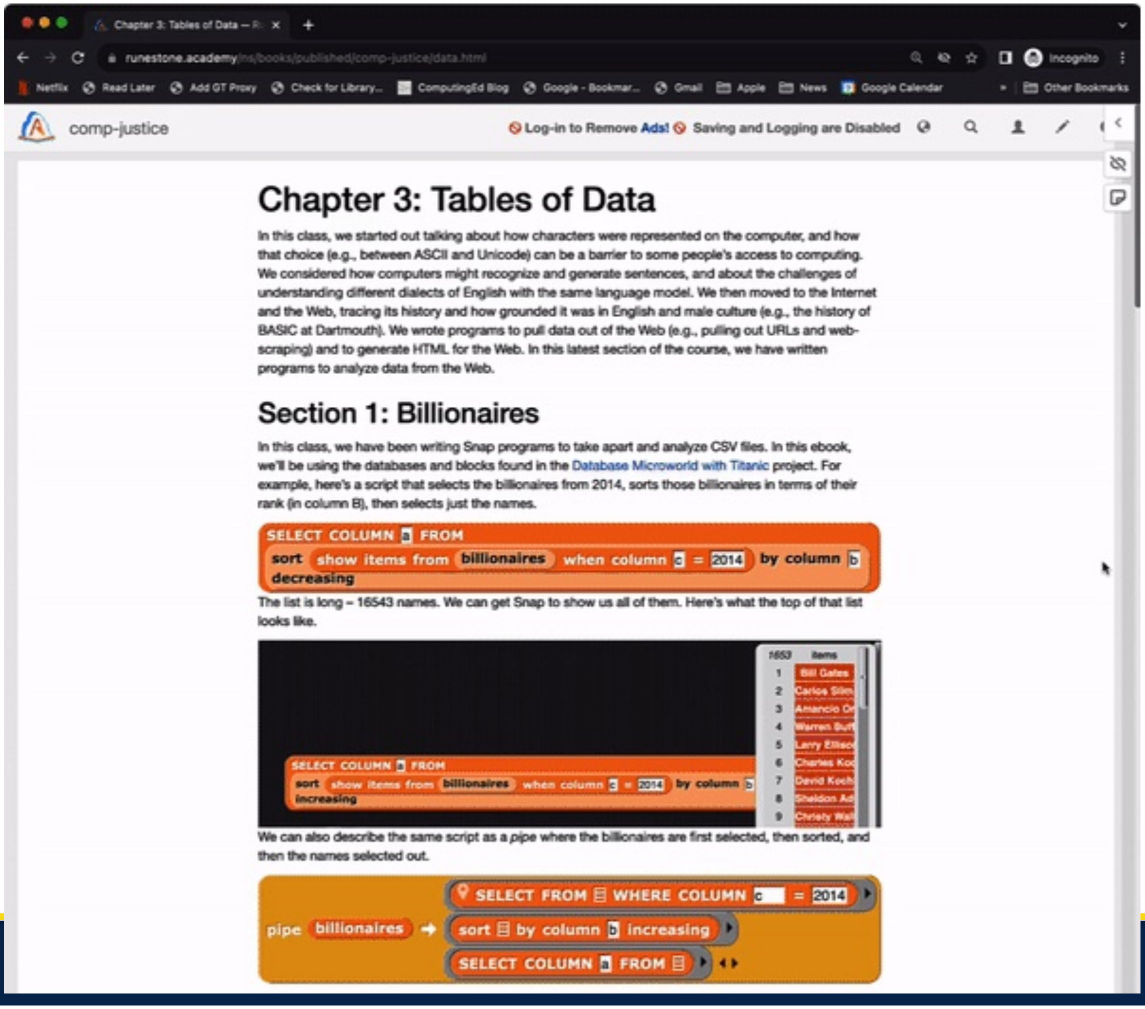


```
make a web page
  make a heading level (1) This is a Web Page
  make paragraph Here is some text. It gives you important information about this page. ◂▸
  make paragraph
    image link to
    https://guzdial.engin.umich.edu/wp-conten
    ◂▸
  make paragraph Here is some text. It g
view webpage (HTML webpage)
open HTML page with multiline (H
```

```
make a multipage form
  for each row of database
    show items from (billionaires) when column e = Microsoft
    make a web page
      make paragraph Dear (column a of current row) ◂▸
      make paragraph
      I know that you have been successful at Microsoft since (column c of current row) ◂▸
      make paragraph You have lots of money. I could use some. Please send some. ◂▸
view webpage (content of web page)
open HTML page with multiline (content of web page)
```

Example for multiple detail pages.

Connecting to SQL in EBook



Chapter 3: Tables of Data

In this class, we started out talking about how characters were represented on the computer, and how that choice (e.g., between ASCII and Unicode) can be a barrier to some people's access to computing. We considered how computers might recognize and generate sentences, and about the challenges of understanding different dialects of English with the same language model. We then moved to the Internet and the Web, tracing its history and how grounded it was in English and male culture (e.g., the history of BASIC at Dartmouth). We wrote programs to pull data out of the Web (e.g., pulling out URLs and web-scraping) and to generate HTML for the Web. In this latest section of the course, we have written programs to analyze data from the Web.

Section 1: Billionaires

In this class, we have been writing Snap programs to take apart and analyze CSV files. In this ebook, we'll be using the databases and blocks found in the Database Microworld with Titanic project. For example, here's a script that selects the billionaires from 2014, sorts those billionaires in terms of their rank (in column B), then selects just the names.

```
SELECT COLUMN a FROM
  sort (show items from (billionaires) when column c = 2014) by column b
  decreasing
```

The list is long – 16543 names. We can get Snap to show us all of them. Here's what the top of that list looks like.

```
SELECT COLUMN a FROM
  sort (show items from (billionaires) when column c = 2014) by column b
  increasing
```

1653	items
1	Bill Gates
2	Carlos Slim
3	Amancio Or
4	Warren Buf
5	Larry Elliso
6	Charles Ko
7	David Koch
8	Sheldon Ad
9	Christy Wal

We can also describe the same script as a *pipe* where the billionaires are first selected, then sorted, and then the names selected out.

```
pipe (billionaires) → SELECT FROM ☰ WHERE COLUMN c = 2014
                      sort ☰ by column b increasing
                      SELECT COLUMN a FROM ☰
```

Open Research Questions

Research Questions We’re Exploring

- How can we support making the notional machines of arts, humanities, and sciences faculty real?
- Are arts and humanities students and faculty getting what they need from these classes?
- What is the process that students follow when programming in these classes, and how does that interact with the unusual structure (e.g., multiple languages, worked examples)?

Research Questions We’re NOT Exploring

- Do these students major in computer science or information?
- How difficult is it for these students to learn C++?
- Are they taking jobs in the computing industry?
- Are they learning how to write safe, secure, and robust code?

Summary: Who should teach Computing?

- Computing Education is different than CS Education.
- Computing Education for everyone is why CS was invented.
- If you want to reach everyone, they won’t be in CS classes.
- Maybe CS departments shouldn’t teach everyone computing. But if not, then who?

Programming can be a Tool for Learning Anything

WE NEED TO MAKE COMPUTING ACCESSIBLE TO EVERYONE

Some of the Collaborators on This Work

- Barbara Ericson, Gus Evrard, Kelly Campbell, Miranda Parker, Kathryn Cunningham, Amber Solomon, Bahare Naimipour, Tamara Nelson-Fromm, Emma Dodoo, Tammy Shreiner, Elise Lockwood, Adaline de Chenne.
- Undergraduate researchers: Aryan Bannerjee, Alexandra Rostkowycz, Erin Shi, Brandon Geng, Jessica Zhang, Ben Steinig, Fuchun Yang, Aoife Harte, Chloe Nguyen, Kashmira Reddy, Kristen Taurence, Angela Li, Derrick White, Jessie Houghton.

- https://lsa.umich.edu/computingfor
- http://computinged.wordpress.com
- http://guzdial.engin.umich.edu

Thank you!