Jupyter-first
Hybrid and online teaching

@LorenaABarba

Professor of Mechanical and Aerospace Engineering
The George Washington University, Washington DC

Efforts in open education

Sharing OER since 2009 via

— iTunes U
— YouTube
— TED-Ed
— GitHub
— self-hosted Open edX site

Disseminating via

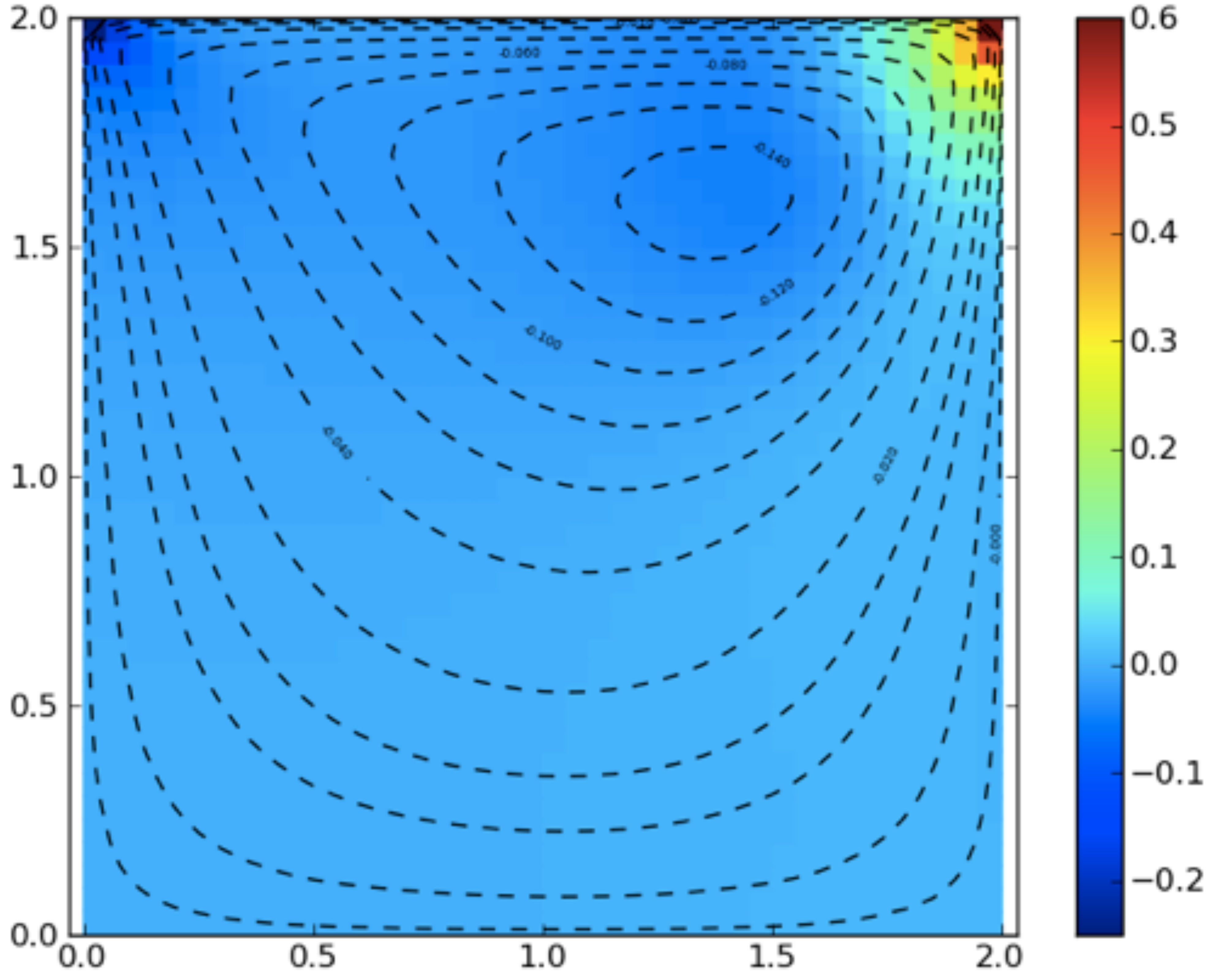
— Twitter & self-hosted blog at lorenabarba.com

Added views ~1,079,010

checked 24 Jan. 2023

Lorena A. **Barba group**

CFD Python: 12 steps to Navier-Stokes



Cavity flow solution at Reynolds number of 200 with a 41x41 mesh.

Posted on 07.22.2013

Lessons

- Quick Python Intro
- Step 1
- Step 2
- CFL Condition
- Step 3
- Step 4
- Array Operations with NumPy
- Step 5
- Step 6
- Step 7
- Step 8
- Defining Function in Python
- Step 9
- Step 10
- Optimizing Loops with Numba
- Step 11
- Step 12

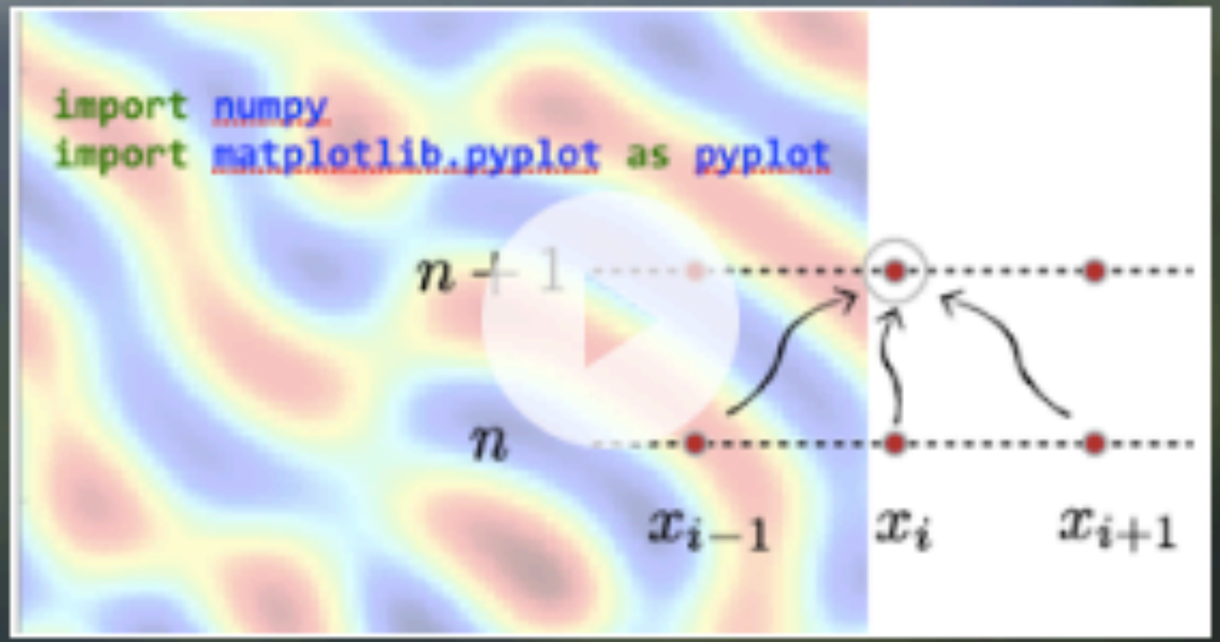
https://github.com/barbagroup/CFDPython

GW | Online

Courses Sign in Create Account

Practical Numerical Methods with Python

MAE 6286



Start Date: Sep 1, 2017

Duration: 15 weeks

Price: Free

Enroll Now

Organization:	GW
Enrollment End:	Dec 31, 2017
Effort:	15 weeks / 6 hours per week
Subject:	Numerical Methods

`https://openedx.seas.gwu.edu`

Lorena Barba @LorenaABarba · Nov 3, 2018

What would happen if we taught calculus to first-year engineering students, and then never again used calculus in engineering courses? Well, they'd forget it, of course.

8 15 48

Lorena Barba
@LorenaABarba

Why teach computer programming as an isolated course, then not _use_ programming in other courses? —Instead of learning to code, we should focus on coding to learn. #engineerscode

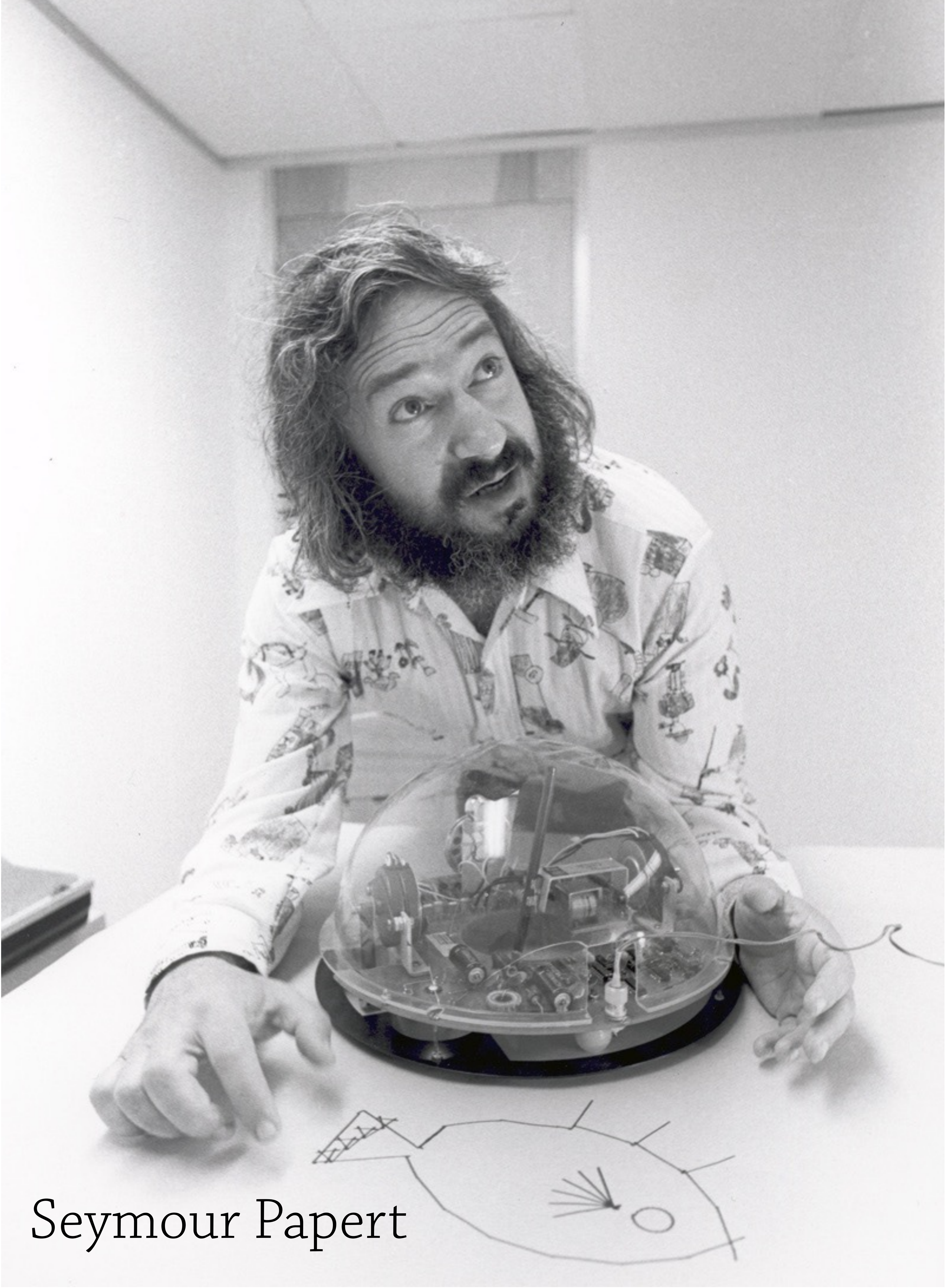
1:25 PM · Nov 3, 2018 · Twitter Web Client

View Tweet activity

86 Retweets **255** Likes

Rather than learn to code—Code to learn

- calls for “learning to code” — well-intentioned and worthwhile, but they miss the point
- Narratives centered on jobs, producing skilled workers…



Seymour Papert

Computational Thinking:

—children can learn to program and it can affect the way they learn everything else

The Journal of Mathematical Behavior

Volume 14, Issue 2, June 1995, Pages 253-280

Paradox, programming, and learning probability: A case study in a connected mathematics framework ☆

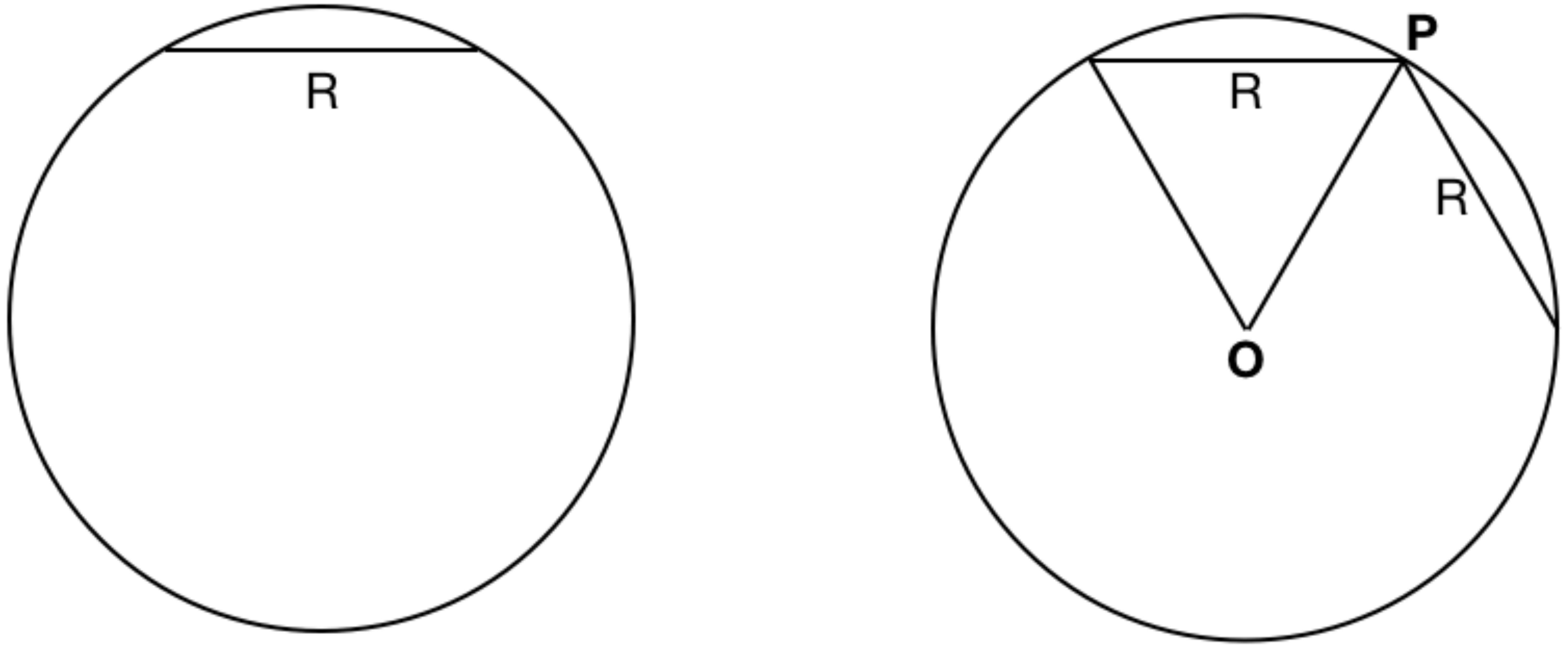
Uri Wilensky

Show more

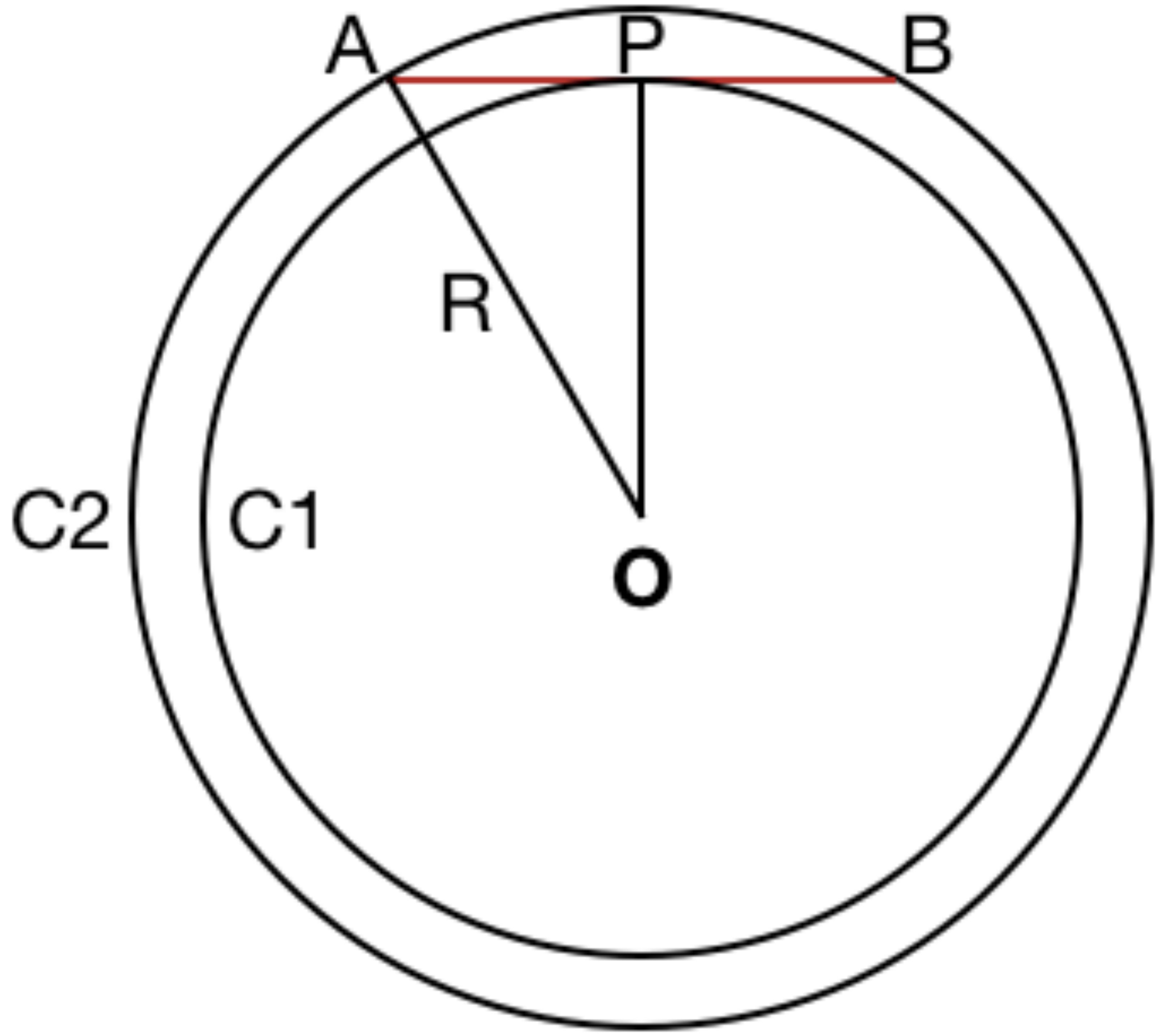
https://doi.org/10.1016/0732-3123(95)90010-1

Question:

If you choose a random chord on a given circle, what is the probability that the chord is longer than the circle's radius?



A learner's diagram to answer the probability question



AB = R
OA = R
AP = R/2

Area C2: πR^2
Area C1: $\frac{3}{4} \pi R^2$

A diagram of the interviewer's challenge to Ellie's answer

Faced with the paradox, Ellie starts writing a program...

DISK INCLUDED

ConStatS: Software for Conceptualizing Statistics

A USER'S MANUAL

Tufts University
Curricular Software Studio

“ This trialogue between Ellie's mental model, the expression of her mental model in encapsulated code and the running of that code, allowed Ellie to successively refine the creative structure of her thought.”

Teacher's challenge:

Presenting a topic to learners through a computational approach, how to structure an activity that affords learners the opportunity to engage with complex ideas?

It does not happen by accident...

Klahr, D. and Carver, S.M., 1988. Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. Cognitive Psychology, 20(3), pp.362-404.

How to eachive learning through programming?

1. Develop a model for what you want students to learn through computing. Use only the programming needed for that learning to happen.

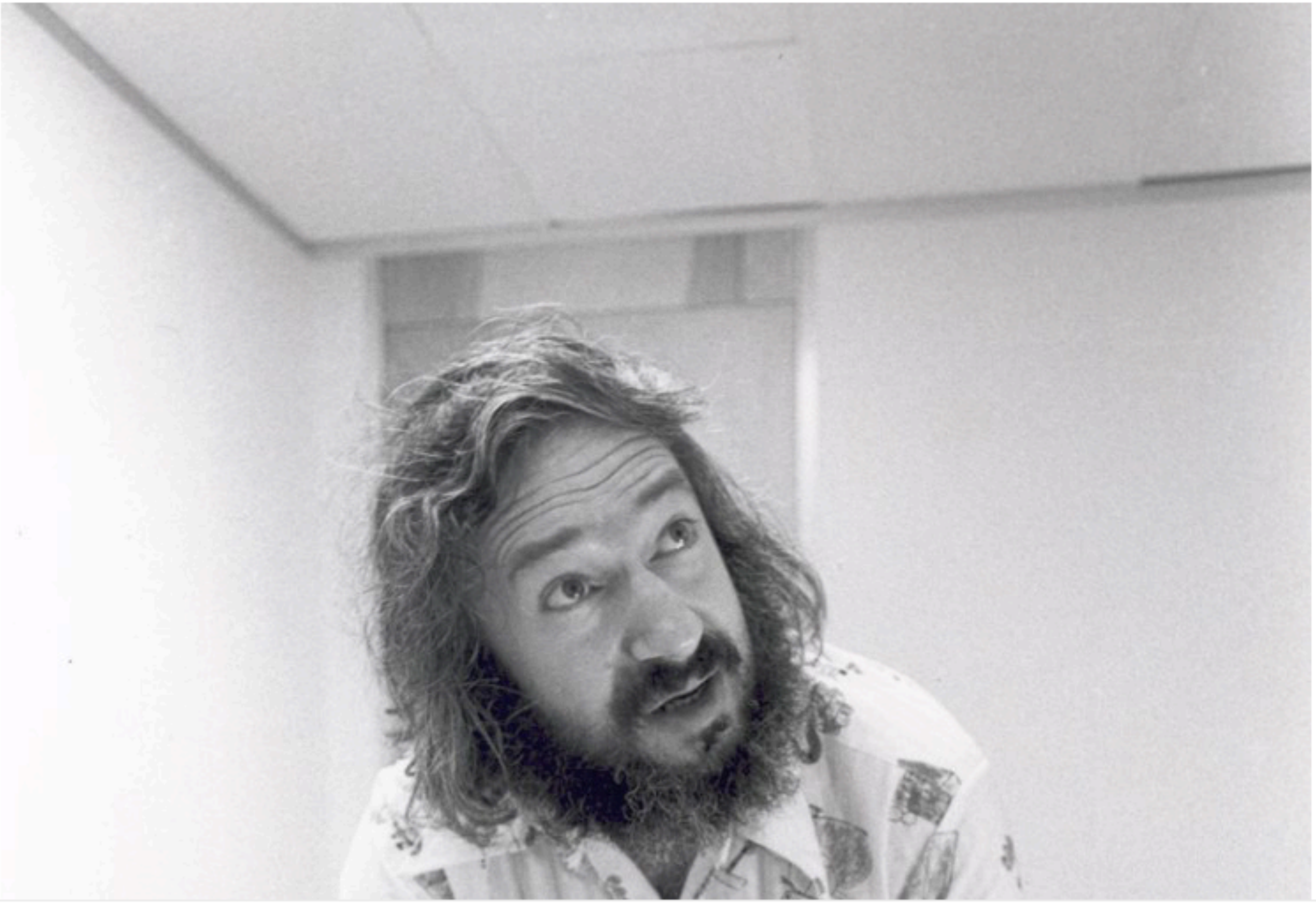
2. Use scaffolding and programming environments that support students in learning the programming needed for the learning objectives.

3. Learning both programming and the target subject (mathematics, physics, probability) may be synergistic: it will take less time than learning each oen separately (but more time than learning just programming, or just the isolated target subject). Pace will be paramount to success.

Guzdial, M., 2015. Learner-centered design of computing education: Research on computing for everyone. Synthesis Lectures on Human-Centered Informatics, 8(6), pp.1-165.

What is computational thinking?

Computational Thinking: I do not think it means what you think it means

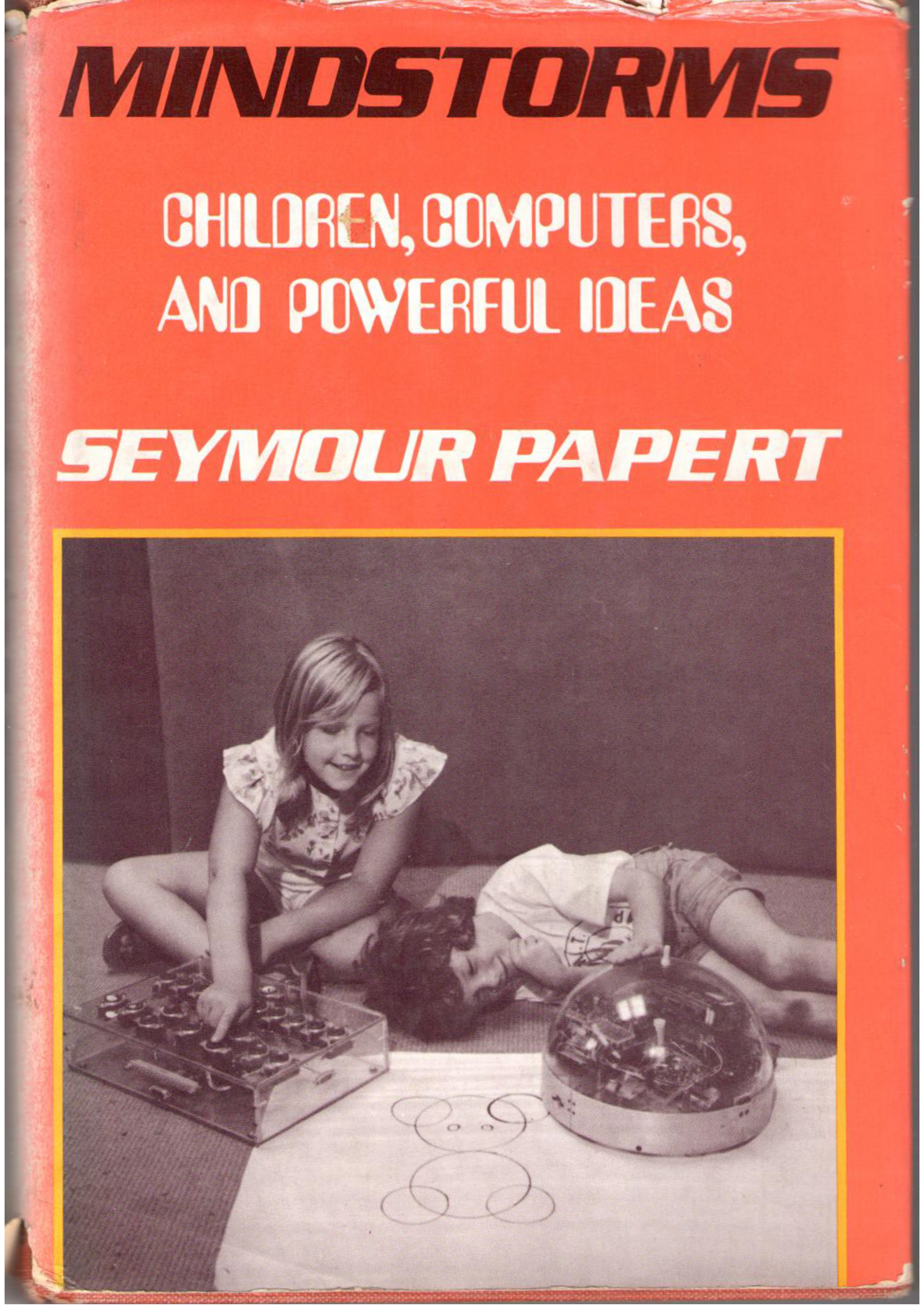


Tags: Coding, Computational Thinking, Education

Category: Blog

To me…

… as a computational scientist, the essence is what we can *do* while interacting with computers, as extensions of our mind, to create and discover. That's not the popular message today.



In most cases, although the experiments have been interesting and exciting, they have failed to make it because they were too primitive. Their computers simply did not have the power needed for the most engaging and shareable kinds of activities. Their visions of how to integrate computational thinking into everyday life was insufficiently developed. But there will be more tries, and more

"Mindstorms" (1980), p. 182

A few talked about the computer as a teaching machine.

This book too poses the question of what will be done with personal computers, but in a very different way. I shall be talking about how computers may affect the way people think and learn. I begin to characterize my perspective by noting a distinction between two ways computers might enhance thinking and change patterns of access to knowledge.

— Seymour Papert, "*Mindstorms*" (1980)

The Power Principle

What comes first, 'using' or 'understanding'? The natural mode of learning is to first use, leading slowly to understanding. New ideas are a source of power to do something.

Project Before Problemm

Projects are primary. Problems come up in the course of projects and are sometimes 'solved' and sometimes 'dissolved.' (The student using random-color effects on screen: had you asked here what she was doing, she would not have said 'problem-solving.')

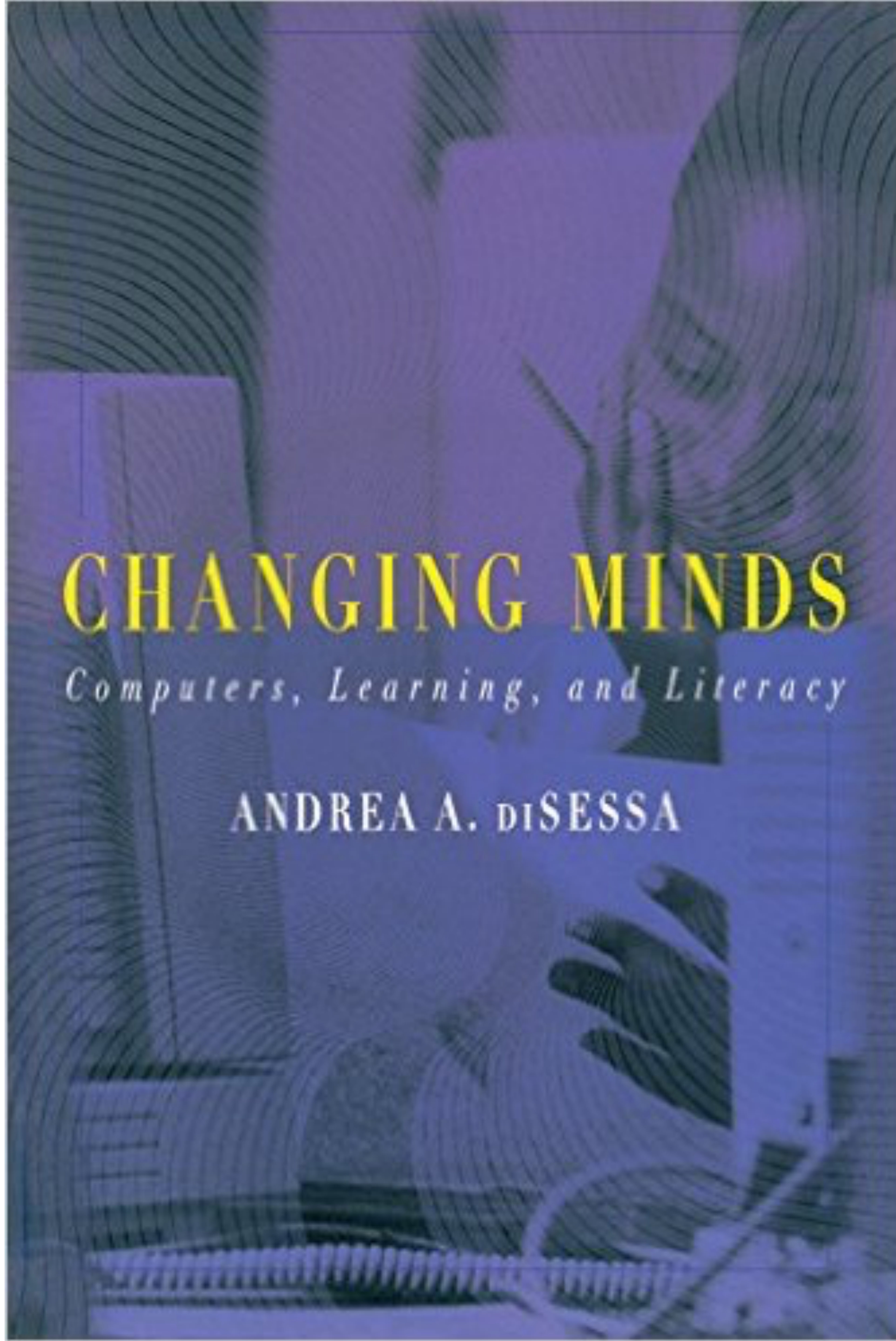
Media Defines Content

Old-school activities involve making inscriptions on paper, while in Papert's alternative involve 'manipulating a computer-based microworld.' New media open the door to new content.

The goal is to use computational thinking to forge ideas.

“Computers can be the foundation of a new and dramatically enhanced literacy.”

Andrea diSessa, 2000.



The defining feature of a literacy is that it's infrastructural

Literacy is a socially widespread deployment of skills and capabilities in a context of material support to achieve valued intellectual ends —diSessa, 2001

The killer app: Jupyter

A new genre of open educational resources (OER).

The course of the future – and the technology behind it

June 1, 2017

Jupyter Notebooks powering Berkeley's data science curriculum

The role of Jupyter is to give students, researchers, journalists or industry engineers tools that give them a coherent handle on the entire process of computational exploration and discovery. We have built it so the same tools are used for individual data analysis or to create a published article, course or book. — Fernando Perez

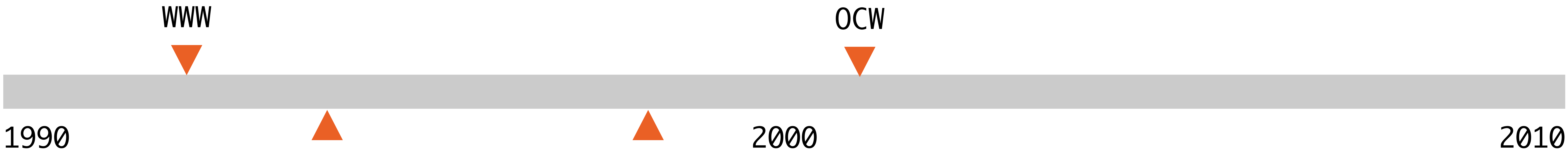
Open Education

Open education

- Open Ed movement was inspired by free & open source software (FOSS).
- F*eatures missed*: open development, networked collaboration, community, value-based framework…
- **OS ethics and practices: put computing at the center of engineering education**

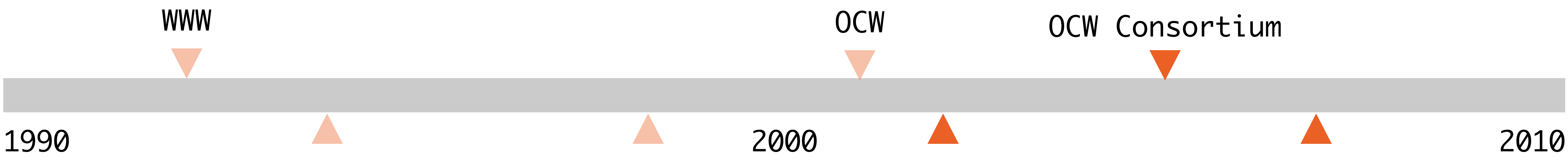
History of OER

- 1994: “learning object” —idea that digital materials can be made to be *reused*.
- 1998: “open content” —idea that principles of FOSS could be applied to content.
- 2001 —founding of Creative Commons —MIT OpenCourseWare launched.



History of OER

- 2002: “open educational resources” coined — UNESCO Forum.
- Others join the OCW movement: Rice, JHU, Tufts, CMU, USU…
- 2005: The OpenCourseWare Consortium
- 2007: OECD “Giving Knowledge for Free…”



Recurring topics in OER

- reducing cost of textbooks for students
- increasing access (for worldwide learners)
- copyright and licenses
- altruism & public good

What did OER miss from FOSS?

- developing in the open
- collaborating/contributing
- community around OS projects
- culture & value-based framework

FOSS: developing in the open

- The OER narrative is often about: creation vs. adoption, author vs. user
- MIT OCW was never open for contributions.
- Rice's Connexions *intended* to be open for contributions, but this feature faded…

We create huge amounts of OER, but there is very little reuse...

— Stephen Downes,
VI International Seminar of the UNESCO chair in e-Learning (June2010)

https://youtu.be/AQCvj6m4obM

Openness is about the possibilities of communicating with other people. It’s not about *stuff*, what you do with stuff. It’s about what you do with each other

— Stephen Downes, 2017

https://youtu.be/FPHYAFcUziA

Teaching in the open

- Open development, on GitHub
- Jupyter for teaching: go.gwu.edu/jupyter4edu
- Publish learning objects—digital materials can be made to be *reused*.

Why Open Education?

Pedagogy of openness—open teaching & learning practices actively promote rich networks, lively communities, and fertile connections.

Openness

…serves a pedagogical purpose: learning is richer by open sharing.

Coordination

…in the model of open-source culture, to create value together, fostering innovation & leadership.

Instructional design

Key concepts and design principles

1. idea of "computable content"
2. open pedagogy
3. modularization
4. harnessing "worked-example effect"
5. f2f active learning with live coding
6. learners documenting their work

Computable content

Educational content made powerfully interactive via compute engines in the learning platform.

Engineers Code: re-usable computing modules for undergraduate engineering

Example:

http://go.gwu.edu/engcomp3lesson1

jupyter
nbviewer

JUPYTER FAQ

EngCom3_flyatchange / notebooks_en

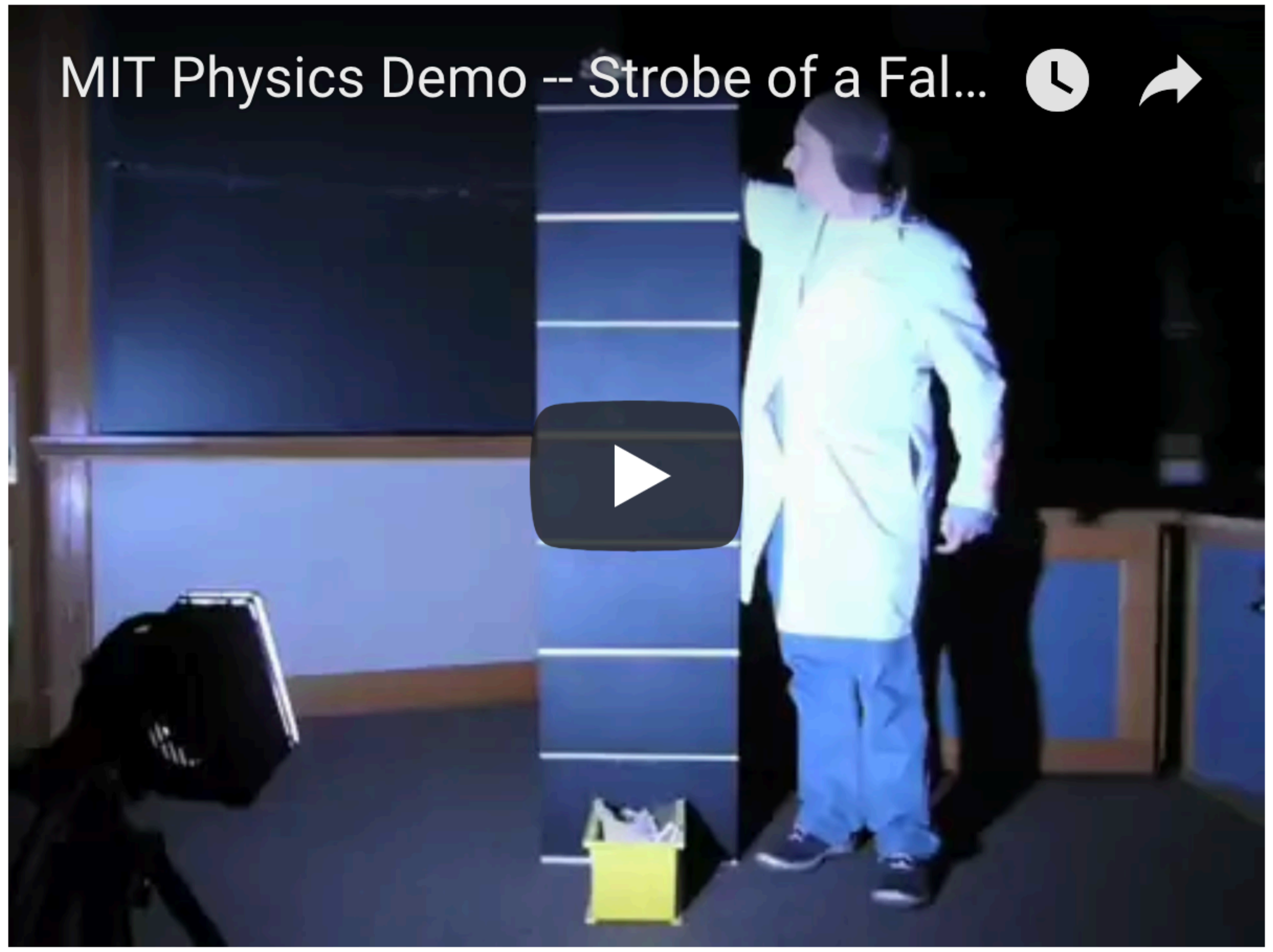
Catch things in motion

Acceleration of a falling ball

Let's start at the beginning. Suppose you want to use video capture of a falling ball to *compute* the acceleration of gravity. Could you do it? With Python, of course you can!

Here is a neat video we found online, produced over at MIT several years ago [1]. It shows a ball being dropped in front of a metered panel, while lit by a stroboscopic light. Watch the video!

In [1]:
```
from IPython.display import YouTubeVideo
vid = YouTubeVideo("xQ4znShlK5A")
display(vid)
```

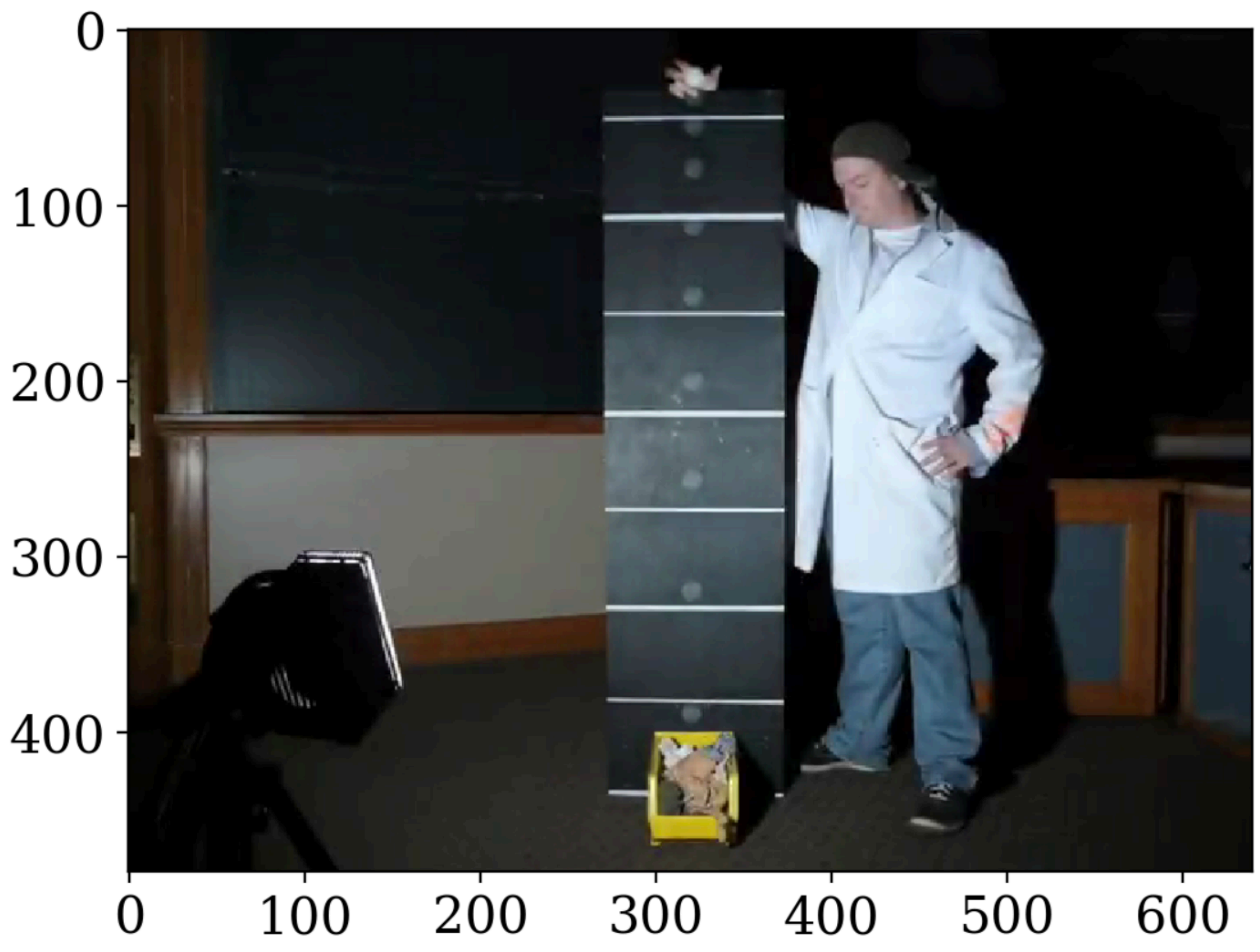


In [11]:
```python
fig = pyplot.figure()

pyplot.imshow(image, interpolation='nearest')

coords = []
def onclick(event):
    '''Capture the x,y coordinates of a mouse click on the image'''
    ix, iy = event.xdata, event.ydata
    coords.append([ix, iy])

connectId = fig.canvas.mpl_connect('button_press_event', onclick)
```



In [12]:
```python
coords
```

Out[12]:
```
[[270.77840909090912, 53.306818181818073],
 [270.77840909090912, 107.85227272727263],
 [272.07711038961043, 163.6964285714285],
 [272.07711038961043, 219.54058441558436],
 [272.07711038961043, 274.08603896103892],
 [272.07711038961043, 328.63149350649348],
 [273.37581168831173, 383.17694805194799],
 [274.67451298701303, 435.125]]
```

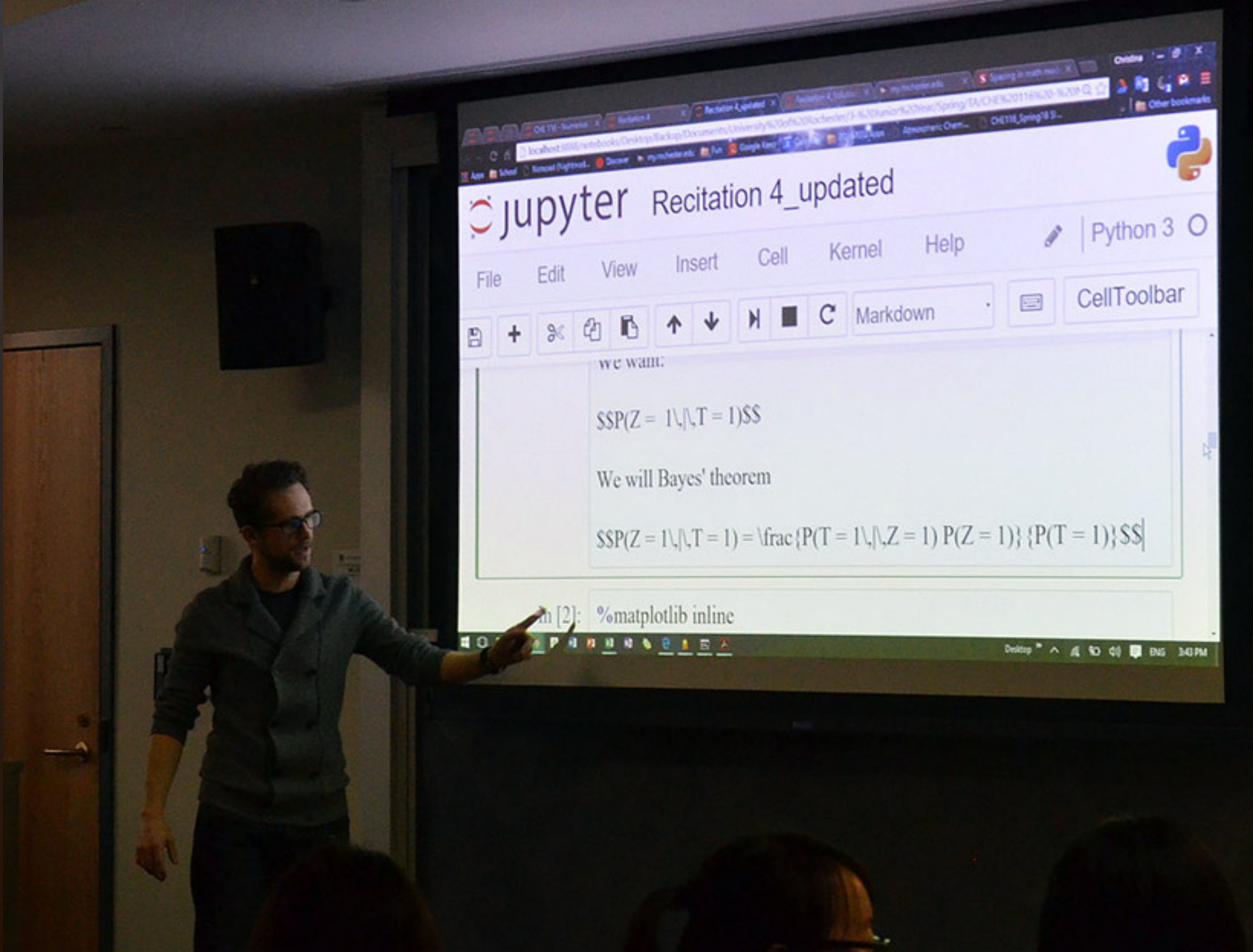
How to develop lessons:

1. Break it down into small steps
2. Chunk small steps into bigger steps
3. Add narrative and connect
4. Link out to documentation
5. Interleave easy exercises
6. Spice with challenge questions/tasks
7. Publish openly online!

Flipped learning with Jupyter:

1. Interactive via computation
2. Guided exploration before a normative explanation, exploiting worked-example effect
3. Active learning (in class), e.g. live coding

In class…


jupyter
Recitation 4_updated
File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Markdown
CellToolbar
$$P(Z = 1\,|\,T = 1)$$
We will Bayes' theorem
$$P(Z = 1\,|\,T = 1) = \frac{P(T = 1\,|\,Z = 1) P(Z = 1)}{P(T = 1)}$$
%matplotlib inline

Active learning increases student performance in science, engineering, and mathematics

Scott Freeman[a,1], Sarah L. Eddy[a], Miles McDonough[a], Michelle K. Smith[b], Nnadozie Okoroafor[a], Hannah Jordt[a], and Mary Pat Wenderoth[a]

[a]Department of Biology, University of Washington, Seattle, WA 98195; and [b]School of Biology and Ecology, University of Maine, Orono, ME 04469

Edited* by Bruce Alberts, University of California, San Francisco, CA, and approved April 15, 2014 (received for review October 8, 2013)

– meta-study of 225 prior studies on active learning
– students in lecture-based courses were 55% more likely to fail than those in active learning classes

Worked-example effect:

- when providing full guidance on how to solve a problem results in better student performance than problem-solving conditions with no guidance (it is a cognitive-load effect)

A technology creates its own pedagogy.