jupytercon2018

Flipped Learning with Jupyter:

Experiences, Good Practices, Supporting Research

@LorenaABarba @RobertTalbert

Overview

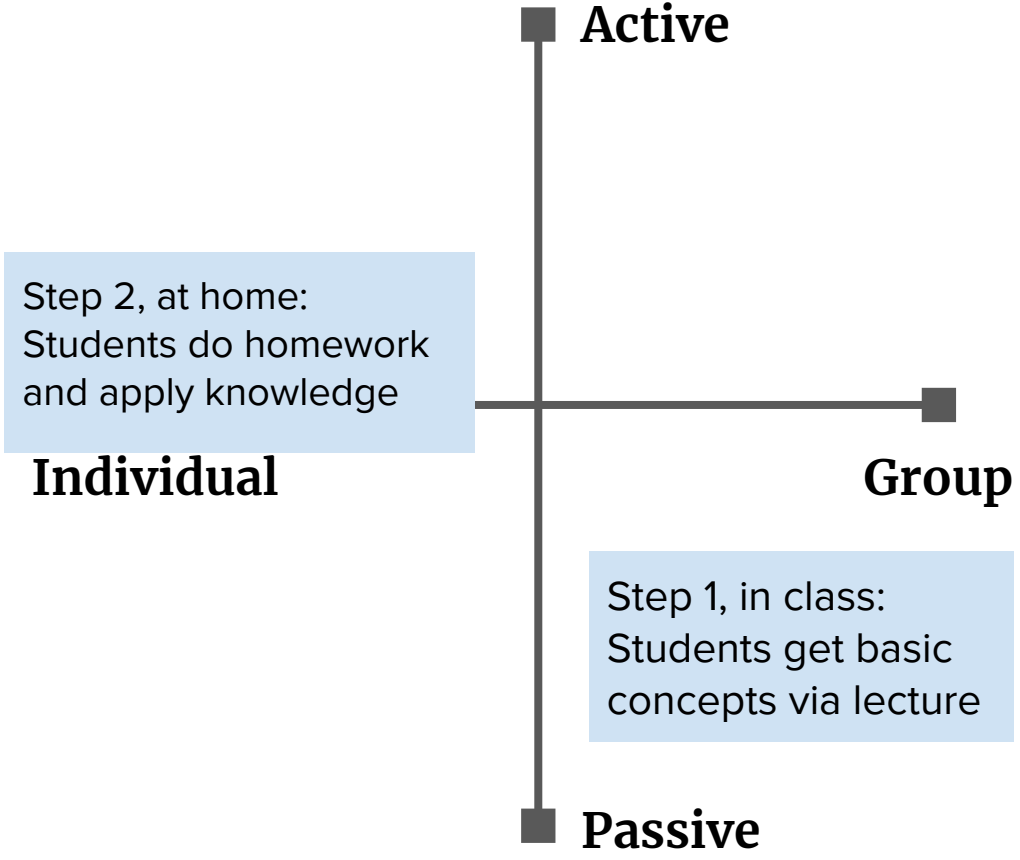
1. Introductions & Flipped Learning overview
2. Jupyter in pre-class work
3. Jupyter with in-class work
4. Jupyter with post-class work & wrap-up

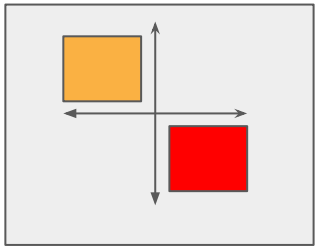
The traditional model of instruction...

The traditional model

Problem #1: The traditional model tends toward passivity.

Problem #2: The active part of the traditional model is left to the student.

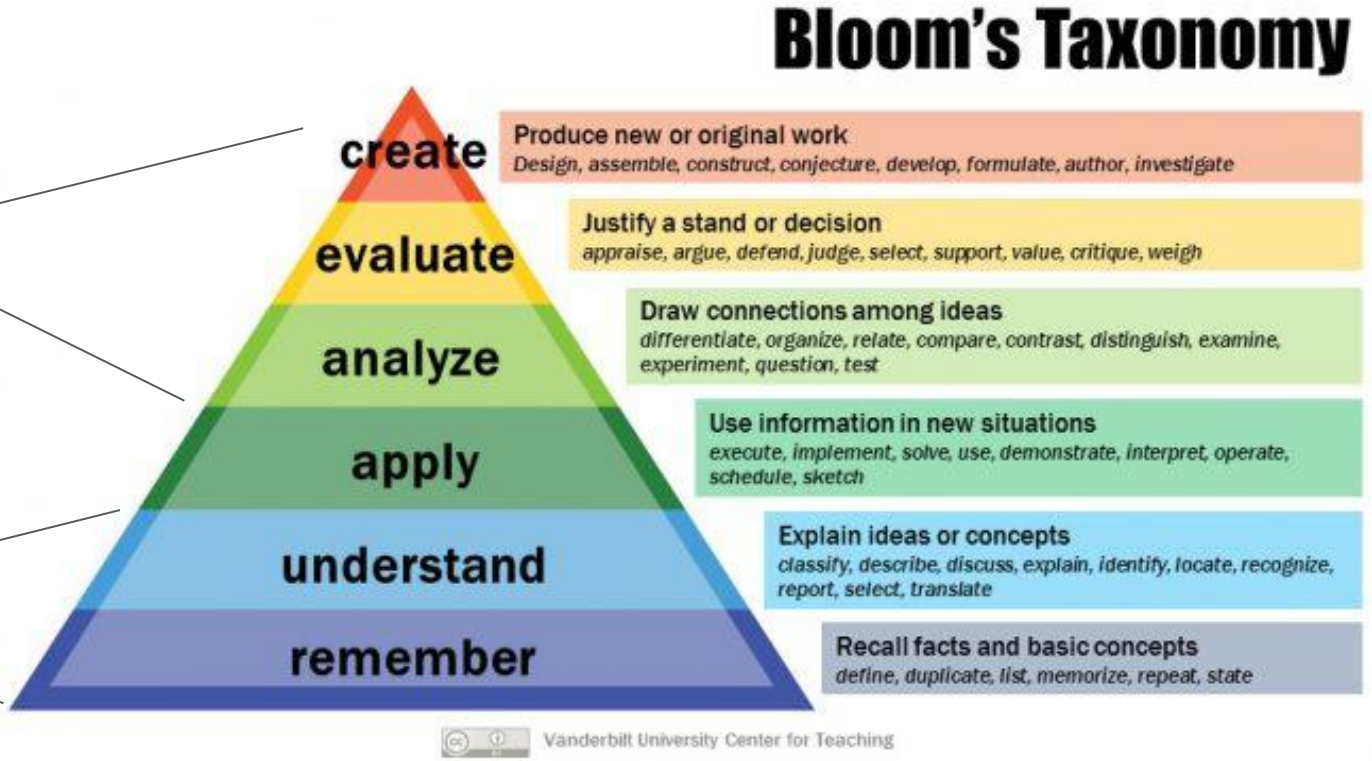




Problem #3: The traditional model misaligns task complexity with access to support.

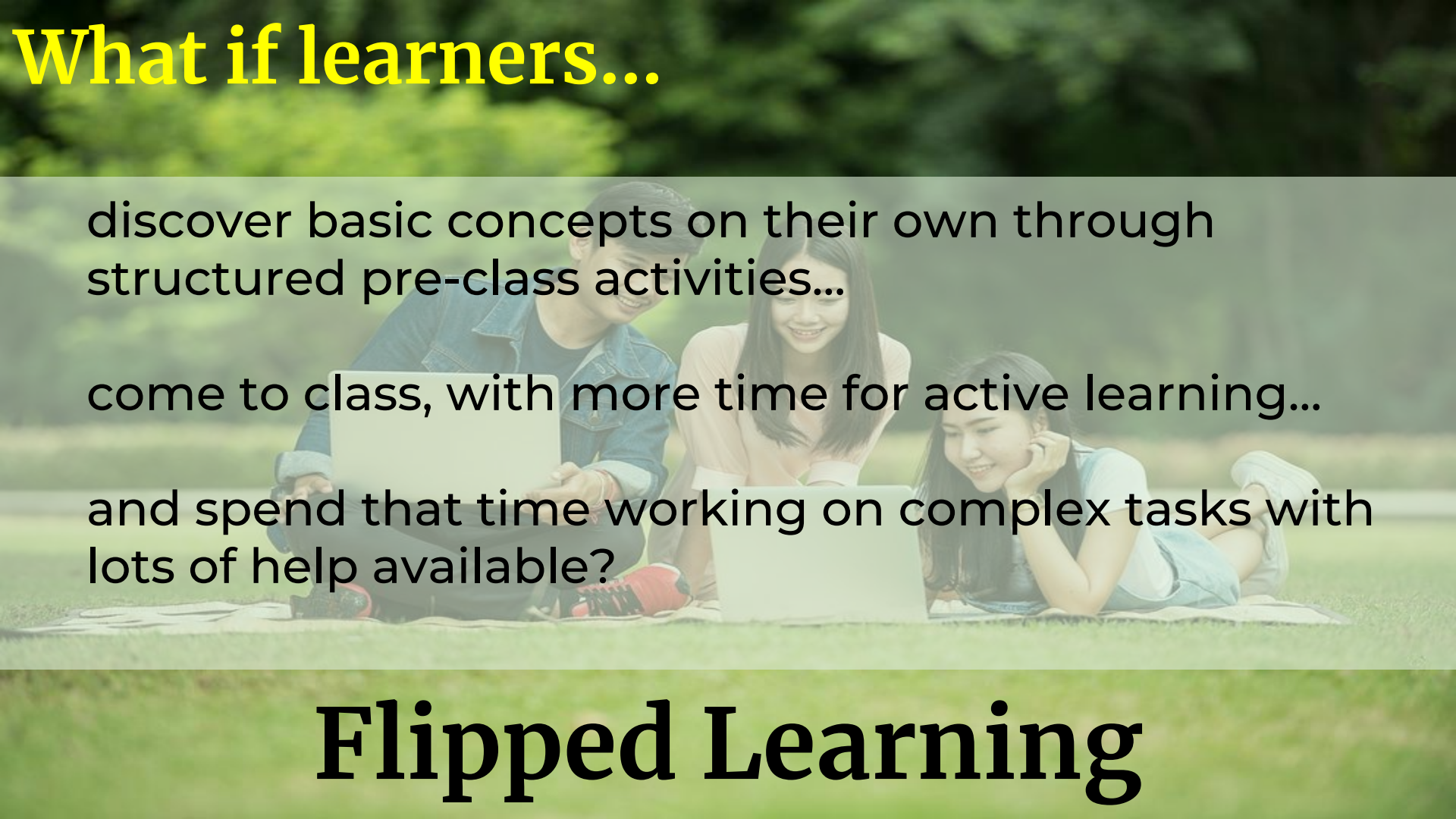
Individual + active (?)

Group + passive




The traditional model is virtually opposite what helps learners the most.

What if learners...



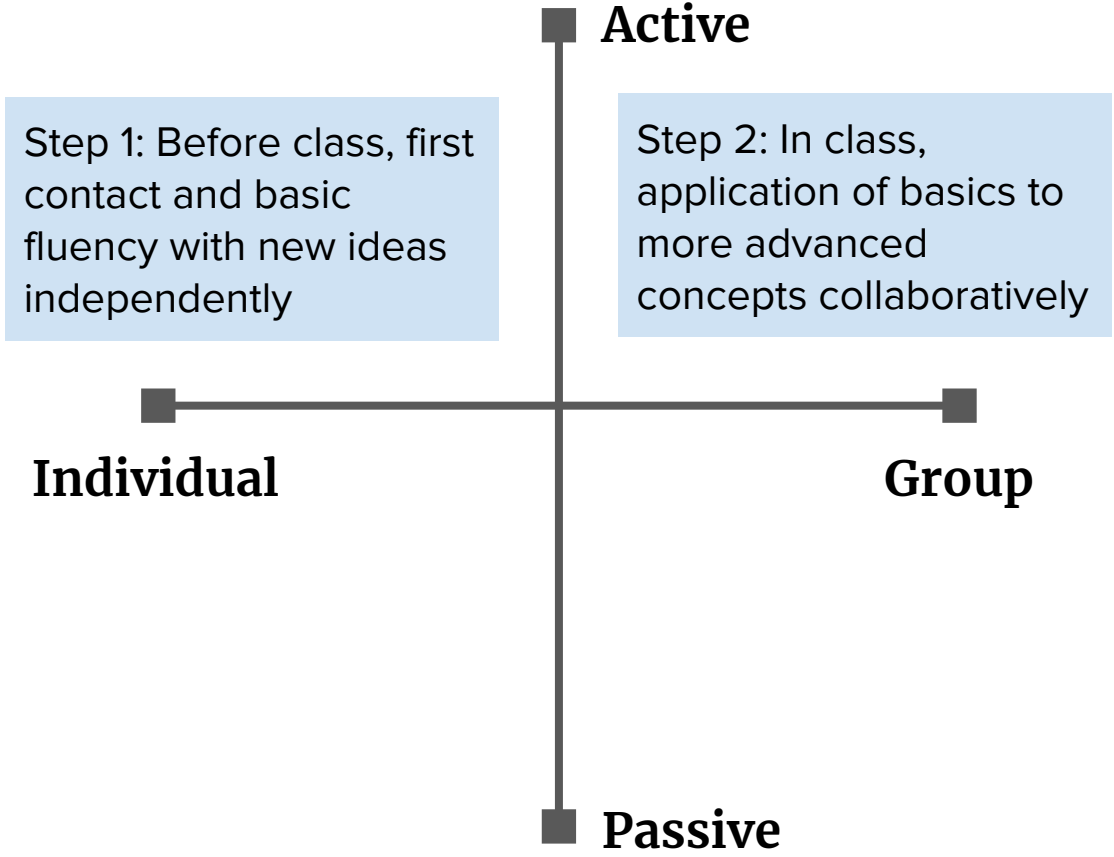
discover basic concepts on their own through structured pre-class activities...

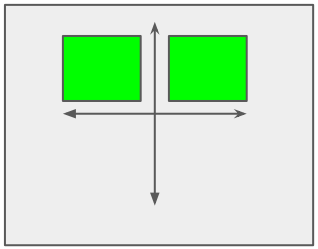
come to class, with more time for active learning...

and spend that time working on complex tasks with lots of help available?

Flipped Learning

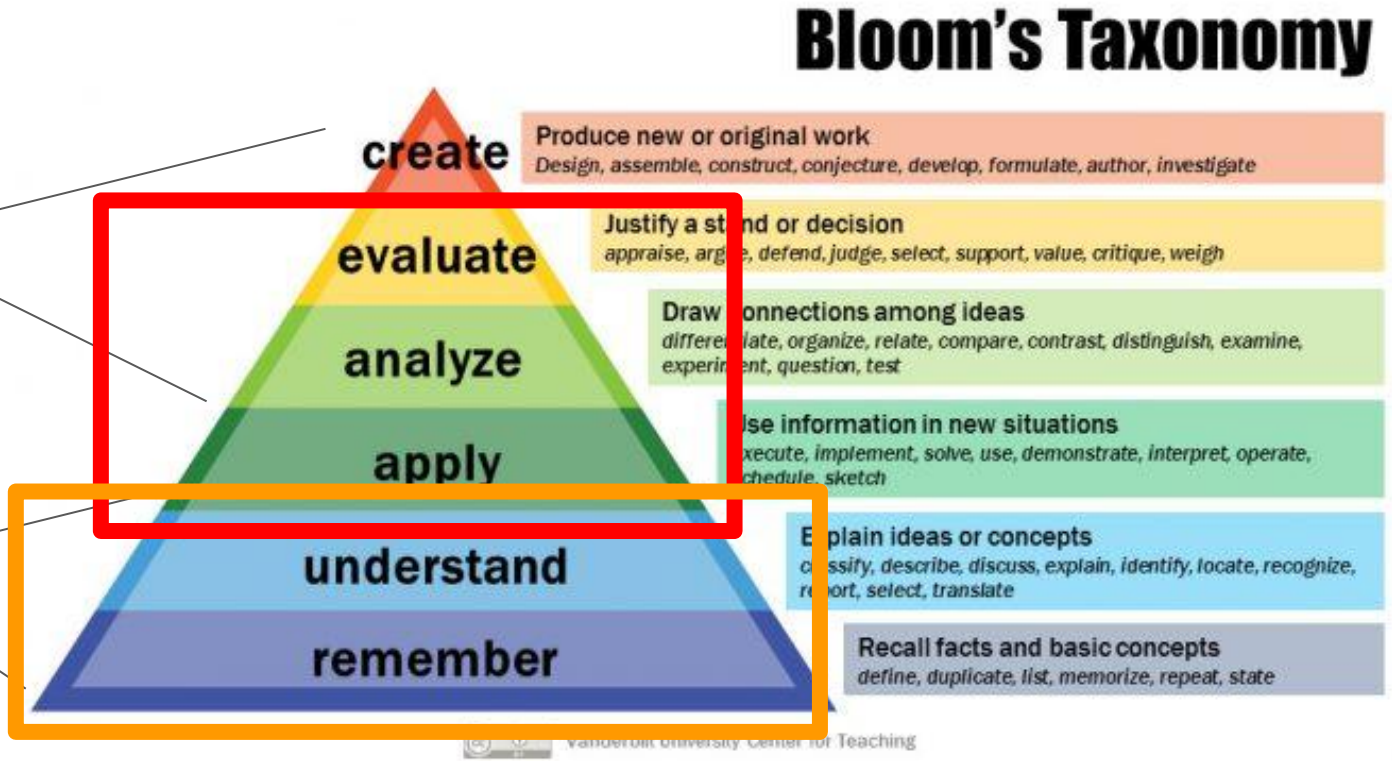
Flipped learning model

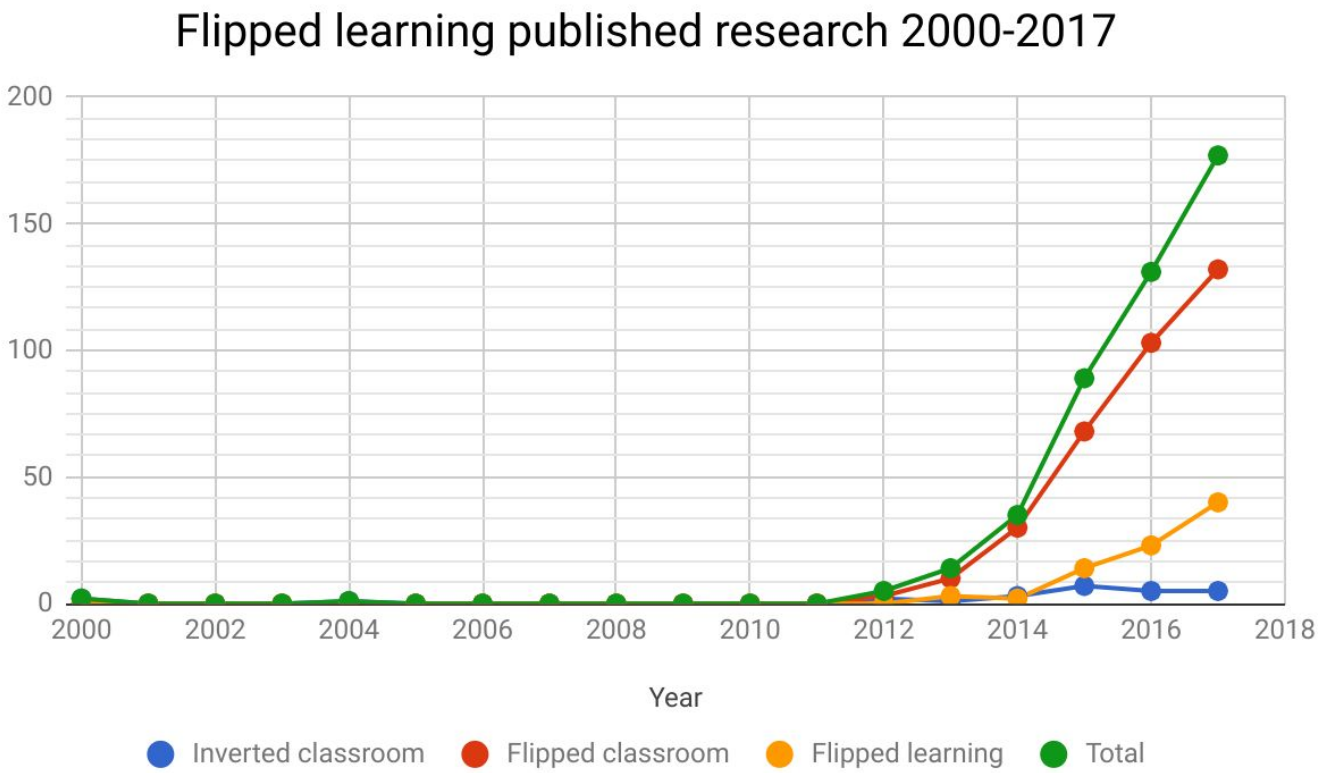




Group + active

Individual + active





https://rtalbert.org/how-much-research-update-2018/

jupyter

PRE-CLASS WORK

Structured inquiry to learn new material independently and prepare for further learning

IN-CLASS WORK

Active, group-focused work on application and analysis

POST-CLASS WORK

Deep work on high-level tasks focusing on creativity and synthesis

Pre-class work

1. Jupyter is a new genre of OER
2. Discovery activities by working through structured computational narrative
3. Worked-example effect

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

A gallery of interesting Jupyter Notebooks

Edit New Page

Andres Soto Villaverde edited this page 15 days ago · 42 revisions

Table of Contents

Recurring topics in OER

1. reducing cost of textbooks for students
2. increasing access (for worldwide learners)
3. copyright and licenses
4. altruism & public good

What did OER miss from FOSS?

1. developing in the open
2. collaborating/contributing
3. community around OS projects
4. culture & value-based framework

Flipped-learning format

1. **Typical**—direct instruction moves from the group space to the individual space (e.g., videos)
2. **Improved**—use *discovery based learning* instead, by working through a sequence of structured computational notebooks.

Flipping the Flipped Classroom: A Study of the Effectiveness of Video Lectures Versus Constructivist Exploration Using Tangible User Interfaces

Bertrand Schneider, Stanford University, 485 Lasuen Mall
Paulo Blikstein, Stanford University, 485 Lasuen Mall

Pages: 5–17

https://doi.org/10.1109/TLT.2015.2448093

With Jupyter

1. Notebooks as a replacement of the textbook
2. Worked-example effect

Worked-example effect

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

Learning and Instruction
Volume 16, Issue 2, April 2006, Pages 87-91

Guest editorial

Optimising worked example instruction: Different ways to increase germane cognitive load ☆

Fred Paas [a, b] ✉, Tamara van Gog [a]

Educational Psychology
Vol. 30, No. 3, May 2010, 349–367

Routledge
Taylor & Francis Group

Worked example effects in individual and group work settings

Endah Retnowati, Paul Ayres* and John Sweller

School of Education, University of New South Wales, Sydney, New South Wales, Australia

Journal of Educational Psychology
2015, Vol. 107, No. 3, 689–704

0022-0663/15/$12.00 http://dx.doi.org/10.1037/edu0000018

The Worked Example Effect, the Generation Effect, and Element Interactivity

Ouhao Chen, Slava Kalyuga, and John Sweller
University of New South Wales

In-class work

1. Active learning with live coding
2. Jupyter demo...

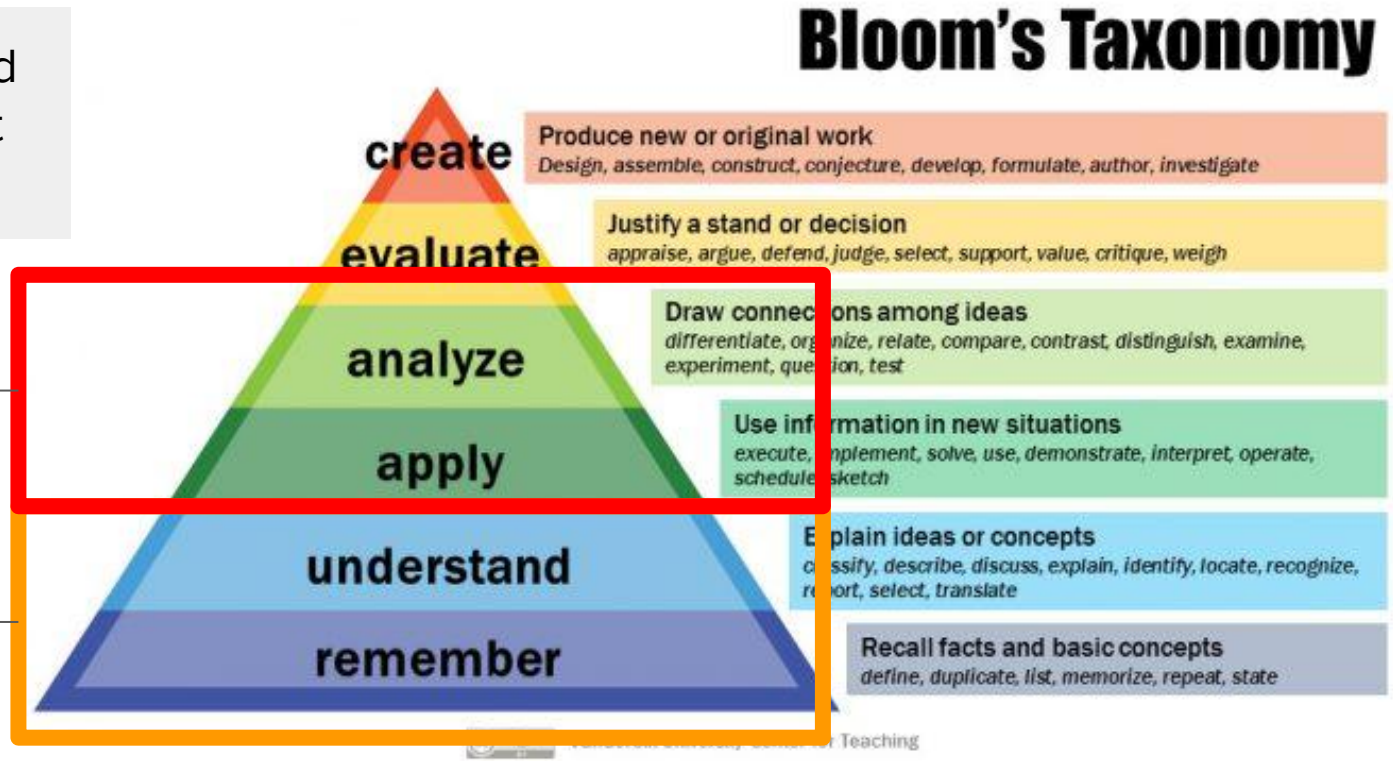
Active learning increases student performance in science, engineering, and mathematics

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)

Class time in a flipped learning environment focuses on this...



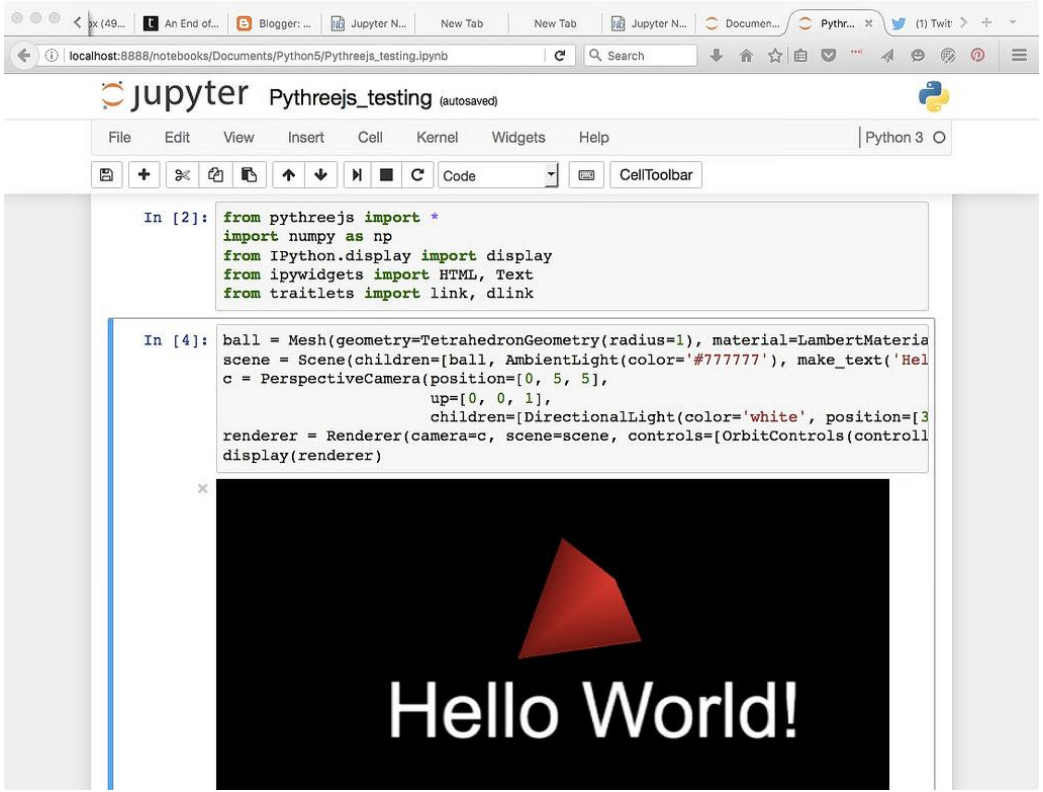
...because this has already been done

What makes in-class work effective?

Challenging but accessible (Zone of Proximal Development)

Connects concepts

Socially constructivist



What can Jupyter during class look like in practice?

http://bit.ly/TalbertJupyterConExample

Post-class work

1. Assignments and projects
2. Students creating their own content

Student quote on Jupyter:

“Made class way more interactive, because you could go back and edit the notebook. If there’s something students didn’t understand or wasn’t clear enough, she could easily go back and add more explanation for a chapter in the notebook”

“Almost like a textbook I could continuously edit”

Computable content

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

Assessment 1: strings

Bookmark this page

Graded Jupyter Notebook

Instructions

This is the first assessment for the section "Play with data in Jupyter," dealing with string manipulations.

Download the notebook, work on it in your local Jupyter install or in a cloud service, then upload your solved notebook here to get it graded automatically. The notebook includes three exercises.

Be sure to follow the instructions to the letter!

You have unlimited submissions—just upload a new notebook and it will get auto-graded and your score updated.

Download Student Notebook

Student Upload

Notebook Name: engcomp1hw1.ipynb

Choose File no file selected

Upload

Graded Jupyter Notebook XBlock

1. Write assignment using nbgrader
2. Upload requirements.txt with dependencies: XBlock builds course image
3. Upload instructor notebook, enter settings
4. Students download assignment & solve
5. Uploaded assignment:
    a. launches Docker container with requirement
    b. auto-grading gives student a score report, writes into gradebook

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
4. Learning through creating own content