JOURNAL OF CHEMICAL EDUCATION

pubs.acs.org/jchemeduc Technology Report

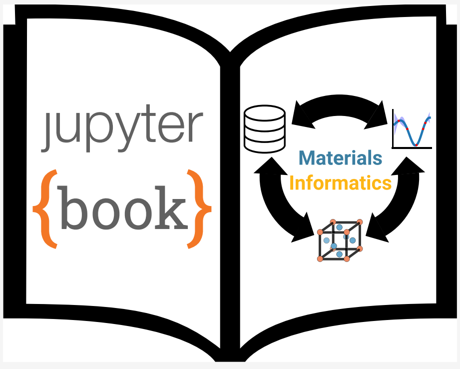
Using Jupyter Tools to Design an Interactive Textbook to Guide Undergraduate Research in Materials Informatics

Enze Chen* and Mark Asta

Cite This: J. Chem. Educ. 2022, 99, 3601−3606 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: With the growing desire to incorporate data science and informatics into STEM curricula, there is an opportunity to integrate research-based software and tools (e.g., Python) within existing pedagogical methods to craft new, accessible learning experiences. We show how the open-source Jupyter Book software can achieve this goal by creating a digital, interactive textbook compiled from Jupyter notebooks, which are already commonplace in research. Using Jupyter Book, we design an open-source, introductory materials informatics research curriculum where the Python programming exercises are supplemented with prose, graphics, slides, and discussion questions, all of which are embedded into a uniform web interface for streamlined access. Interactive programming capabilities, enabled through the JupyterHub cloud infrastructure, provide opportunities in these digital spaces for students to interrogate the code, test their own hypotheses, and deepen their comprehension. These authentic learning experiences demonstrate the broad utility of the Jupyter ecosystem in sustaining the growth of materials informatics education.



KEYWORDS: *Chemoinformatics, Computer-Based Learning, Inquiry-Based/Discovery Learning, Interdisciplinary/Multidisciplinary, Materials Science, Upper-Division Undergraduate*

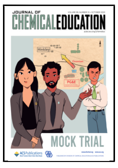
■ INTRODUCTION

In recent years, the infusion of machine learning (ML) and artificial intelligence (AI) into the materials and chemical sciences[1] has led to significant advancements in modeling structure−property relationships,[2] synthesis planning,[3] molecular design,[4] and drug discovery.[5] Scientists have leveraged ML/AI technologies to design new materials with the ultimate goal to achieve lower cost, accelerated time to market, and greater sustainability. Now, the maturation of these technologies in research presents exciting opportunities to transfer them into the classroom, where educators can design new informatics curricula to train the next generation of informatics-skilled scientists.[6−9]

With such pedagogical aspirations serving as crucial motivators, it is also important to identify the key challenges that could impede a successful implementation. The multidisciplinary nature of chemoinformatics calls for a robust curriculum that combines scientific domain knowledge, data science, and computing skills, which is difficult for students to fit into an already strained course load and for an instructor to gain mastery in all components.[7] Due to the comprehensive scope of materials informatics (MI), it has been advised that MI curricula be structured as small modules to facilitate its integration with existing courses and to help alleviate some of the aforementioned problems;[8,9] however, this integration presents its own challenges that must be carefully considered during instructional design. For example, it may not be obvious how a research-based tool can fit within the scope of the existing curriculum and whether additional software-specific training is necessary.

In particular, when designing any informatics curriculum, it is appealing to include programming exercises to give students hands-on experience with authentic problem solving tasks.[10] Given the popularity of the beginner-friendly Python programming language in scientific research,[11] many introductory-level programming courses in the physical sciences have chosen to use Python as well.[12−14] To improve code readability and reproducibility, there is a growing trend to write and execute Python code inside Jupyter notebooks,[15] which are now commonplace in chemoinformatics/MI courses[14,16,17] and workshops.[18−20] These digital notebooks merge prose, Python code, and additional multimedia elements into rich computational narratives that provide a gentle introduction for students. They are often provided as standalone files (`.ipynb` extension) for users to run locally, but this requires installing additional Python packages and managing software configurations on personal devices,[21] which

Received: July 3, 2022
Revised: September 8, 2022
Published: September 19, 2022



can be daunting for beginners. To circumvent these technological challenges, another popular option is to execute notebooks in the cloud,[19,22] where popular services such as Binder[23] and Google Colaboratory[24] (Colab) can provide a more consistent and equitable user experience. Several of these options have been compared in the literature,[25,26] which underscores the impact of cloud-based computing on education.

Herein, we report our experience using a related suite of Jupyter authoring tools, including the recently developed Jupyter Book environment,[27] to design an integrated, interactive online textbook to guide undergraduate research projects in the data-driven design of dielectric materials.[28] This work was inspired by existing textbooks published using Jupyter Book,[29,30] which gave us confidence to reenvision the remote research experience in light of the COVID-19 pandemic and utilize an interactive text to boost student engagement and performance.[31,32] Given the multitude of benefits of undergraduate research experiences,[33,34] the curriculum served as an experimental model to expand access to such opportunities while easing students into the research process. This Technology Report discusses the details of the implementation that uses open-source tools familiar to those working in the computational and informatics sciences, which lowers the barrier for researchers to design new curricula. The Jupyter Book interface was easy to navigate, and the ability to use JupyterHub[35] to run Python notebooks in the cloud enabled students to interact with the content entirely through a web browser without having to install anything locally. While the COVID-19 pandemic forced everyone to transition online, it also presented an opportunity to explore how the digital nature of informatics and rapid advancements in computing infrastructure can improve the scalability and accessibility of undergraduate research experiences.

■ IMPLEMENTATION

The primary software used in designing this curriculum is Jupyter Book, developed by the Executable Books Project as "an open source project for building beautiful, publication-quality books and documents from computational material."[27] It can be installed using the standard Python package managers `pip` or `conda` and contains a set of command-line utilities for compiling the book from Markdown text (`.md`) and Jupyter notebook (`.ipynb`) files. Building off of Markdown files is advantageous due to the familiarity of the Markedly Structured Text (MyST) syntax[36] to researchers and the simplicity of the syntax, which is still powerful enough to create rich content pages with text, figures, citations, and embedded files and videos. Content that cannot be easily created in Markdown (e.g., presentation slides) can be created externally and then embedded into Markdown files using standard HTML tags. All of the content (presentations, lecture notes, Python notebooks, etc.) for each day was then grouped to form each "chapter" of the book. In this way, the digital textbook served as a centralized location where students can go to access all of the resources throughout the research internship. Once the pages are complete, the book can be compiled into a static PDF file for offline reading or a collection of HTML web pages for dynamic viewing, all without the need for instructors to learn additional frameworks like JavaScript or CSS. The online book can be accessed using any modern web browser, which frees instructors and students from compatibility concerns across different operating systems and Python environments. Further information on getting started with Jupyter Book can be found in the detailed tutorials on their website,[27] which requires familiarity with using the command line and text editors. In addition to detailed instructions on setup and content creation, the documentation describes several options for sharing the book, which are also discussed below.

Table 1 summarizes the contents of the book, which was used by six undergraduate students for a research module on

Table 1. Brief Summary of the Contents of the Jupyter Book[a]

Section	Topics Covered
Preamble	Module overview, software setup
Week 1	Intro to Python/Jupyter, intro to MI, data management, intro to ML, dielectric materials, and MI research
Week 2	Self-directed research
Week 3	Self-directed research, GitHub pull requests, project presentations, reflections on the module

[a]The first week is a series of tutorials to introduce students to the research topic and software tools, and the remaining time is for their self-directed research projects (MI = materials informatics, ML = machine learning).

the data-driven design of dielectric materials for microelectronics applications. The module was part of a summer internship program supported through a partnership between the College of Engineering at the University of California, Berkeley, and the Materials Sciences Division at Lawrence Berkeley National Laboratory. Briefly, students learned data management best practices and applied them to mine data for dielectric materials from the Materials Project,[37,38] an online database for materials properties computed using high-throughput density functional theory (DFT). Students used this DFT-generated data to train ML models to predict a material's dielectric constant (κ) and then used the ML model, along with other criteria of their choosing, to screen for the "best" high-κ dielectric material for nanoscale transistors. This research question is an active area of research[39,40] with documented impact for the microelectronics industry.[41,42] The content also aligned well with the two other modules in the summer research internship, which shows how a modular design can be flexibly sequenced with other curricula, consistent with the recommendations of other educators.[8,9,43] All files are freely available on GitHub under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license, and the textbook is hosted for free on GitHub Pages.[28] GitHub Pages was chosen for its simplicity and flexibility, and step-by-step instructions for hosting Jupyter Books on a variety of platforms can be found in the documentation.[27]

The first iteration of the internship was structured to feature a full week of tutorials about MI followed by 2 weeks of self-directed research in teams, all performed remotely in Summer 2021. The text begins with an overview of the research module, including the learning outcomes, schedule of events, and account setup instructions (e.g., Materials Project, GitHub). Most of the Python notebooks are introduced in the first week, starting with a complete introduction to the Python language and followed by tutorials on data management, data visualization, ML, dielectrics, and MI research. Weeks 2 and 3 have fewer prescribed notebooks and more supporting content such as additional reading, check-in discussions, and presentation tips for students. Interspersed throughout the weeks are mini-

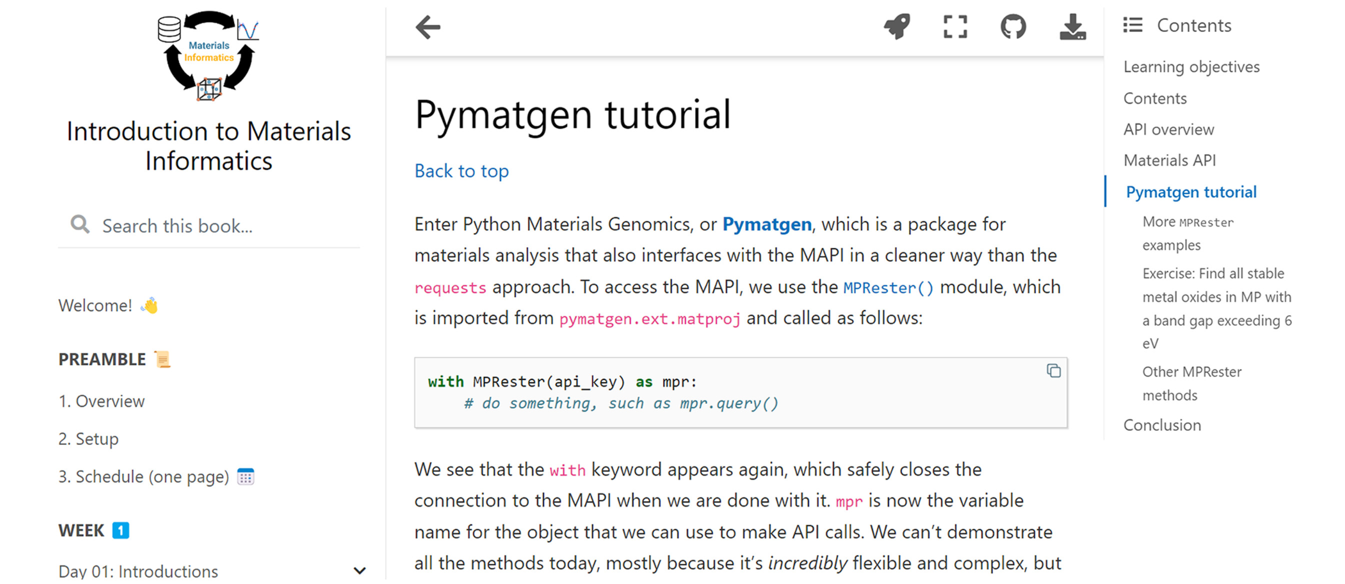


Figure 1. A screenshot of the web interface for the Jupyter Book.[28] The left margin contains a table of contents and a search bar for ease of navigation through the entire book, while the right margin contains section headers for the specific page. The content in the page, here rendered from a Jupyter notebook,[15] contains a mix of text, Python code, and images. Hovering over the rocket icon in the top-right lists options for launching the same page in an interactive environment such as JupyterHub[35] or Google Colaboratory.[24]

lessons on how to read papers and documentation, graduate school applications, and other topics to support their educational development. There are no graded assessments or exams, and the final summative assessment was a presentation of their research and experience. While live mentoring took place over instant messaging and video conferencing platforms, students used the Jupyter Book interface to access all lecture content, assigned readings, and Python exercises (Figure 1). The minimalist style of the web interface features several navigation shortcuts and a familiar rendering of Markdown and notebook content that facilitated students' daily use. One can use arrow keys at the bottom of each page to proceed sequentially through the book or use the navigation headings to jump to specific pages. The MyST syntax makes it simple for instructors to incorporate content from external sources by embedding them in the page or linking them, as shown in Figure 1.

Several interactive options exist for readers to write and execute Python code within the notebooks. UC Berkeley affiliates can use the campus JupyterHub[44] to open the same web page in a Python kernel (by hovering over the rocket icon in the top of Figure 1). The JupyterHub instance syncs with the GitHub repository for the Jupyter Book so that each student has an updated copy of the files that they can work on at their own pace. There are many advantages to using the JupyterHub platform, including a uniform computing environment (required packages can be preinstalled), more powerful resources (extra compute and/or memory can be allocated upon request), and preservation of their work (the modified files are saved to their account). These are significant benefits of JupyterHub that can only be partially satisfied with other cloud services, although a large-scale deployment of JupyterHub may require institutional support and funding (see refs 35 and 44 for deployment details). For non-UC Berkeley affiliates, there is also a Colab option that allows any user to complete the exercises with minimal changes (e.g., a few package installations and file paths), and this may be preferred if teaching at scale without JupyterHub. Nevertheless, the seamless integration of Jupyter Book with cloud-based Python kernels is one of the principal advantages of using the Jupyter Book software to design MI curricula as it enhances the scalability and accessibility of these interactive learning experiences.

RESULTS AND DISCUSSION

This curriculum was taught to students from a variety of backgrounds: Some of them were new to Python (but had other programming experience), many were new to research, only one had a materials science background, and all were new to data science. These individual differences further validated the use of the Jupyter ecosystem to design tutorials during the first week to establish the requisite knowledge to carry out a successful MI research project. To maximize effective usage of the software, a walk-through of the Jupyter Book and JupyterHub interfaces was given on the first day, which is recommended when teaching with any new technology. Subsequent usage included a mix of group instruction (e.g., presentations, live coding) and individual work using the digital textbook. In addition to the final presentation, feedback was collected continuously throughout the internship in the form of daily check-ins and three anonymous surveys (made using Google Forms) distributed at the beginning, middle, and end (see Supporting Information for example questions).

Consistent with our expectations, the students found the Jupyter Book straightforward to navigate and helpful as a centralized resource for the entire module. They no longer had to juggle multiple files and links (only the single URL to the online text) and instead could focus on learning the material. They enjoyed having the interactive exercises alongside the reading to strengthen comprehension and welcomed the opportunity to carefully work through the exercises at their own pace, demonstrating a greater sense of agency. By the end of the second week, most of the students were not only comfortable editing the existing notebooks, but also capable of creating new Jupyter notebooks on the JupyterHub to carry out their self-directed research. Their feedback on the merits of

https://doi.org/10.1021/acs.jchemed.2c00640
J. Chem. Educ. 2022, 99, 3601−3606

the Jupyter Book-powered curriculum aligns with the results reported in previous research on the benefits of interactive textbooks.[31,32]

Figure 2 shows the results from the final student survey on how well the curriculum achieved the learning outcomes of the

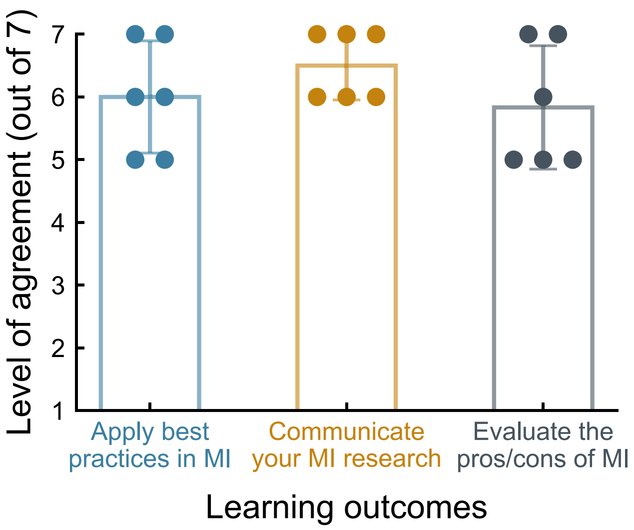


Figure 2. Student feedback on how well the module's learning outcomes were achieved. Questions are scored on a Likert-type scale where 7 indicates strong agreement and 1 indicates strong disagreement. The bars and lines indicate the mean and one standard deviation ($n = 6$), respectively, and the individual scores are separated for clarity.

module. There was a strong level of agreement, consistent with our observations of their final presentations (e.g., hearing comments about the importance of cleaning data, or how ML can be a helpful guide for experiments—not a replacement). While these learning outcomes are not directly tied to the software, the success of this module would not have been possible without the structure and interactivity enabled by the Jupyter Book. Giving students hands-on practice with informatics workflows and open-ended, group research projects are authentic assessments that cultivate problem solving skills in realistic contexts,[10] which will better prepare them to meet the demands of the next-generation workforce. Although the impacts of this internship model will be evaluated in a separate communication, there were a few pieces of feedback directly regarding the software. As the students worked in groups on their research projects, a few of them wished that JupyterHub had the ability to collaborate in real time on the same notebook. Such a possibility was not explored for this iteration, although the suggestion of pair programming[45] through video conferencing software was found to be helpful for sharing programmatic solutions and strengthening group dynamics. Another collaboration tool available in Jupyter Book (that was not used) is Hypothesis,[46] which allows users to directly annotate the HTML for asynchronous group discussions, particularly if Jupyter Book is used for a course that involves more reading. Many students also desired the capability to automatically check the correctness of their code, which might be possible with autograding software.[47,48] Such tools would definitely be valuable if Jupyter Book is used to design a large-scale, semester-long course where graded assessments carry greater importance. These suggestions, in addition to others like better scaffolding of the tutorial exercises, will help focus curricular improvements for future iterations of this module.

From the perspective of instructors, we are particularly excited to see computational software and infrastructure mature to the point where an entire MI research curriculum can be implemented online and taught remotely. These open-source, online platforms markedly increase the accessibility of this knowledge and continue to lower the barrier for new learners, whether that involves better content, more effective pedagogy, or fewer headaches with software installation. The fact that participants were gathered across three time zones shows how cloud-based solutions can scale far more effectively than traditional materials science education can, reaching larger and more diverse populations. The synergy between the Jupyter environment and robust GitHub workflows also lowers the barrier to creating educational content for scientific domain experts, as they can continue to use familiar tools from their research background. Moreover, creating open-source curricula invites participation from the broader community to collaborate on their development,[6] as instructors can adapt these materials for their own needs or add chapters based on their areas of expertise. The use of GitHub presents an opportunity to teach students about team-based open science[49] and each student was assigned to make a small contribution to the Jupyter Book to showcase their research accomplishments. Although none of the students had prior experience with `git`, they were able to successfully clone the repository, update their presentation abstracts, and submit the changes through pull requests, demonstrating the facility of collaborative development with Jupyter Book. We anticipate the increasing prevalence of Jupyter Books (for other examples, see refs 29 and 30 and the public gallery[27]) and other open-source computational tools in instructional design will continue to bolster teaching and learning of informatics in the materials and chemical sciences.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available at https://pubs.acs.org/doi/10.1021/acs.jchemed.2c00640.

Survey questions (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Enze Chen – *Department of Materials Science and Engineering, University of California, Berkeley, California 94720, United States; Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States;* orcid.org/0000-0002-7621-115X; Email: chenze@berkeley.edu

Author

Mark Asta – *Department of Materials Science and Engineering, University of California, Berkeley, California 94720, United States; Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States;* orcid.org/0000-0002-8968-321X

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jchemed.2c00640

https://doi.org/10.1021/acs.jchemed.2c00640
J. Chem. Educ. 2022, 99, 3601−3606

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Ryan Miyakawa, Sinéad Griffin, Teresa Calarco, and Tiffany Reardon for their assistance in the design and support of the internship program. We also thank the DataHub developers at UC Berkeley for computing resources and technical support. E.C. thanks Michelle Wilkerson for helpful discussions and acknowledges support through the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1752814. M.A. acknowledges support for his contributions by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division, under Contract No. DE-AC02-05-CH11231 within the Materials Project program (KC23MP).

■ REFERENCES

(1) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559*, 547−555.

(2) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545−2559.

(3) Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehiers, P. P.; Byington, J.; Piotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, *365*, No. eaax1566.

(4) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360−365.

(5) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discovery Today* **2018**, *23*, 1241−1250.

(6) Kim, S.; Bucholtz, E. C.; Briney, K.; Cornell, A. P.; Cuadros, J.; Fulfer, K. D.; Gupta, T.; Hepler-Smith, E.; Johnston, D. H.; Lang, A. S.; Larsen, D.; Li, Y.; McEwen, L. R.; Morsch, L. A.; Muzyka, J. L.; Belford, R. E. Teaching Cheminformatics through a Collaborative Intercollegiate Online Chemistry Course (OLCC). *J. Chem. Educ.* **2021**, *98*, 416−425.

(7) Pernaa, J. Possibilities and Challenges of Using Educational Cheminformatics for STEM Education: A SWOT Analysis of a Molecular Visualization Engineering Project. *J. Chem. Educ.* **2022**, *99*, 1190−1200.

(8) McDowell, D. L. Gaps and Barriers to Successful Integration and Adoption of Practical Materials Informatics Tools and Workflows. *JOM* **2021**, *73*, 138−148.

(9) The Minerals Metals & Materials Society (TMS). *Creating the Next-Generation Materials Genome Initiative Workforce*; 2019.

(10) Wieman, C.; Price, A.; Kim, C. Instructional Design Based on the Problem-Solving Decisions of Scientists and Engineers. *ETH Learning and Teaching Journal* **2020**, *2*, 10−17.

(11) Perkel, J. M. Programming: Pick up Python. *Nature* **2015**, *518*, 125−126.

(12) van Staveren, M. Integrating Python into a Physical Chemistry Lab. *J. Chem. Educ.* **2022**, *99*, 2604.

(13) Weiss, C. J. Scientific Computing for Chemists: An Undergraduate Course in Simulations, Data Processing, and Visualization. *J. Chem. Educ.* **2017**, *94*, 592−597.

(14) Weiss, C. J. A. Creative Commons Textbook for Teaching Scientific Computing to Chemistry Students with Python and Jupyter Notebooks. *J. Chem. Educ.* **2021**, *98*, 489−494.

(15) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C. *Jupyter Development Team, Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas*; IOS Press: Netherlands, 2016; pp 87−90.

(16) Menke, E. J. Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course. *J. Chem. Educ.* **2020**, *97*, 3899−3903.

(17) De Haan, D. O.; Schafer, J. A.; Gillette, E. I. Using a Modular Approach to Introduce Python Coding to Support Existing Course Learning Outcomes in a Lower Division Analytical Chemistry Course. *J. Chem. Educ.* **2021**, *98*, 3245−3250.

(18) Lafuente, D.; Cohen, B.; Fiorini, G.; García, A. A.; Bringas, M.; Morzan, E.; Onna, D. A Gentle Introduction to Machine Learning for Chemists: An Undergraduate Workshop Using Python Notebooks for Visualization, Data Processing, Analysis, and Modeling. *J. Chem. Educ.* **2021**, *98*, 2892−2898.

(19) Strachan, A.; Desai, S. *Hands-On Data Science and Machine Learning Training Series*. nanoHUB, 2020; https://nanohub.org/groups/ml/handsontraining, Accessed May 2022.

(20) National Institute of Standards and Technology. *Machine Learning for Materials Research Bootcamp & Workshop on Machine Learning Microscopy Data*. 2021; https://www.nanocenter.umd.edu/events/mlmr-2021/, Accessed May 2022.

(21) Munroe, R. *Python Environment*. https://xkcd.com/1987/, 2018; Accessed May 2022.

(22) Engelberger, F.; Galaz-Davison, P.; Bravo, G.; Rivera, M.; Ramírez-Sarmiento, C. A. Developing and Implementing Cloud-Based Tutorials That Combine Bioinformatics Software, Interactive Coding, and Visualization Exercises for Distance Learning on Structural Bioinformatics. *J. Chem. Educ.* **2021**, *98*, 1801−1807.

(23) Bussonnier, M.; Forde, J.; Freeman, J.; Granger, B.; Head, T.; Holdgraf, C.; Kelley, K.; Nalvarte, G.; Osheroff, A.; Pacer, M.; Panda, Y.; Perez, F.; Ragan-Kelley, B.; Willing, C. Jupyter Project. Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale. *Proceedings of the 17th Python in Science Conference* **2018**, 113−120.

(24) *Google Research, Google Colaboratory*. 2018; https://colab.research.google.com/, Accessed May 2022.

(25) Markham, K. *Six Easy Ways to Run Your Jupyter Notebook in the Cloud*; Data School, 2019; https://www.dataschool.io/cloud-services-for-jupyter-notebook/.

(26) Hutchison, G. R. *Teaching Programming across the Chemistry Curriculum*; ACS Symposium Series; American Chemical Society, 2021; Vol. *1387*; Chapter 9, pp 123−134.

(27) *Executable Books Project*; Jupyter Book. Zenodo, 2020; https://jupyterbook.org/, Accessed Apr 2022.

(28) Chen, E.; Asta, M. *Introduction to Materials Informatics*; GitHub, 2021.

(29) White, A. D. *Deep Learning for Molecules and Materials*; GitHub, 2021.

(30) Adhikari, A.; DeNero, J.; Wagner, D. *Computational and Inferential Thinking: The Foundations of Data Science*, 2nd ed.; GitHub, 2021.

(31) Miller, B. N.; Ranum, D. L. Beyond PDF and ePub: Toward an interactive textbook. *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*; New York City, NY, USA, 2012; pp 150−155.

(32) Smith, D. H.; Hao, Q.; Hundhausen, C. D.; Jagodzinski, F.; Myers-Dean, J.; Jaeger, K. Towards Modeling Student Engagement with Interactive Computing Textbooks: An Empirical Study. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. New York City, NY, USA, 2021; pp 914−920.

(33) Seymour, E.; Hunter, A.-B.; Laursen, S. L.; DeAntoni, T. Establishing the Benefits of Research Experiences for Undergraduates in the Sciences: First Findings from a Three-Year Study. *Science Education* **2004**, *88*, 493−534.

(34) Linn, M. C.; Palmer, E.; Baranger, A.; Gerard, E.; Stone, E. Undergraduate Research Experiences: Impacts and Opportunities. *Science* **2015**, *347*, 1261757.

(35) *Project Jupyter*; JupyterHub, 2021; https://jupyter.org/hub.

(36) *Executable Books Project*; MyST—Markedly Structured Text. 2022; https://myst-parser.readthedocs.io/en/latest/index.html.

(37) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Materials* **2013**, *1*, 011002.

(38) Petousis, I.; Mrdjenovich, D.; Ballouz, E.; Liu, M.; Winston, D.; Chen, W.; Graf, T.; Schladt, T. D.; Persson, K. A.; Prinz, F. B. High-Throughput Screening of Inorganic Compounds for the Discovery of Novel Dielectric and Optical Materials. *Scientific Data* **2017**, *4*, 160134.

(39) Morita, K.; Davies, D. W.; Butler, K. T.; Walsh, A. Modeling the dielectric constants of crystals using machine learning. *J. Chem. Phys.* **2020**, *153*, 024503.

(40) Gopakumar, A.; Pal, K.; Wolverton, C. Identification of High-Dielectric Constant Compounds from Statistical Design. arXiv 2022, arXiv:2206.04750.

(41) Kingon, A. I.; Maria, J.-P.; Streiffer, S. K. Alternative dielectrics to silicon dioxide for memory and logic devices. *Nature* **2000**, *406*, 1032−1038.

(42) Robertson, J. High dielectric constant oxides. *European Physical Journal Applied Physics* **2004**, *28*, 265−291.

(43) Enrique, R. A.; Asta, M.; Thornton, K. Computational Materials Science and Engineering Education: An Updated Survey of Trends and Needs. *JOM* **2018**, *70*, 1644−1651.

(44) Division of Computing, Data Science, and Society. *University of California, Berkeley DataHub*. 2020; https://datahub.berkeley.edu/.

(45) Williams, L.; Kessler, R.; Cunningham, W.; Jeffries, R. Strengthening the case for pair programming. *IEEE Software* **2000**, *17*, 19−25.

(46) Hypothesis team. *hypothes.is*; 2022; https://web.hypothes.is/.

(47) Jupyter, Project; Blank, D.; Bourgin, D.; Brown, A.; Bussonnier, M.; Frederic, J.; Granger, B.; Griffiths, T. L.; Hamrick, J.; Kelley, K.; Pacer, M.; Page, L.; Pérez, F.; Ragan-Kelley, B.; Suchow, J. W.; Willing, C. nbgrader: A Tool for Creating and Grading Assignments in the Jupyter Notebook. *Journal of Open Source Education* **2019**, *2*, 32.

(48) Pyles, C. *UC Berkeley Data Science Education Program, Otter-Grader: A Python and R Autograding Solution*; Zenodo, 2022.

(49) Vargas, S.; Zamirpour, S.; Menon, S.; Rothman, A.; Häse, F.; Tamayo-Mendoza, T.; Romero, J.; Sim, S.; Menke, T.; Aspuru-Guzik, A. Team-Based Learning for Scientific Computing and Automated Experimentation: Visualization of Colored Reactions. *J. Chem. Educ.* **2020**, *97*, 689−694.

https://doi.org/10.1021/acs.jchemed.2c00640
J. Chem. Educ. 2022, 99, 3601−3606