




PlixLab: A Python package for browser-based interactive presentations

Giuseppe Romano ¹

¹ Massachusetts Institute of Technology, Cambridge MA, United States 

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Disseminating scientific results increasingly requires interactivity with figures and data. For example, enabling the rotation of a molecule within a presentation slide allows biologists to illustrate molecular docking mechanisms critical to drug discovery. In computational fluid dynamics, key insights are often encoded in spatial maps; allowing presenters to interactively explore these maps can lead to more engaging and effective communication. Despite these needs, mainstream presentation tools primarily support static content. PlixLab addresses this gap by combining the power of Python with modern JavaScript visualization libraries. PlixLab is a Python-based framework that generates JSON-encoded data, which is rendered interactively in the browser.

Statement of need

PlixLab is a Python library for creating browser-based interactive presentations. It supports a range of plugins, including interfaces to 3jmol ([Rego & Koes, 2015](#)) for protein visualization, plotlyjs ([Plotly, Inc, 2023](#)) and Bokeh ([Bokeh Development Team, 2018](#)) for dynamic plots, and web embedding for integrating full scientific web applications directly into slides. Notably, when embedding a JupyterLite application ([JupyterLite Contributors, 2021](#)), PlixLab enables the use of an in-slide REPL or even a full Jupyter Notebook ([Kluyver et al., 2016](#)), with Python running locally in the browser via Pyodide ([Pyodide development team, 2021](#)). These capabilities make PlixLab a unified platform for building data-centric presentations, eliminating the need to juggle multiple tools. Standard presentation features, such as animations, can also be implemented programmatically.

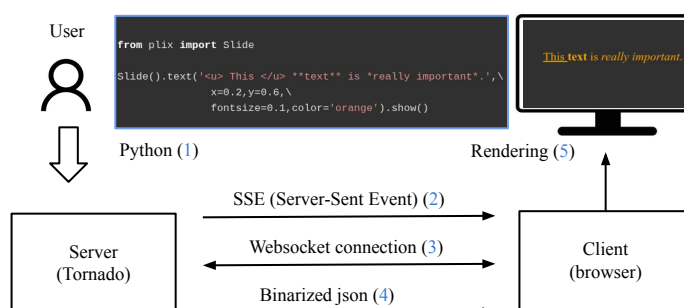


Figure 1: Real-time update flow in PlixLab. When a Python file is modified, the server sends an SSE (1–2) to the browser, which opens a WebSocket (3) to receive binarized JSON data (4) and re-render the presentation (5), enabling seamless hot-reloading.

The graphical user interface (GUI), written in JavaScript, offers three navigation modes:

single-slide view, grid view, and fullscreen presentation mode. To streamline slide development, PlixLab includes hot-reload functionality (see Figure 1). When a user modifies the Python source file, a local server is triggered, which uses Server-Sent Events (SSE) to notify the browser client. The client then opens a WebSocket connection to receive the updated binary JSON data and renders the new content. This design avoids inefficient client-side polling, resulting in a responsive and seamless development experience.

References

- Bokeh Development Team. (2018). *Bokeh: Python library for interactive visualization*. <https://github.com/bokeh/bokeh>.
- JupyterLite Contributors. (2021). *JupyterLite: JupyterLab distribution that runs in the browser*. <https://github.com/jupyterlite/jupyterlite>.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., & others. (2016). Jupyter notebooks—a publishing format for reproducible computational workflows. In *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS press. <https://ebooks.iospress.nl/doi/10.3233/978-1-61499-649-1-87>
- Plotly, Inc. (2023). *Plotly.js: Open-source JavaScript charting library behind Plotly and Dash*. <https://github.com/plotly/plotly.js>
- Pyodide development team. (2021). *Pyodide/pyodide* (Version 0.27.6). Zenodo. <https://doi.org/10.5281/zenodo.5156931>
- Rego, N., & Koes, D. (2015). 3Dmol. Js: Molecular visualization with WebGL. *Bioinformatics*, 31(8), 1322–1324.