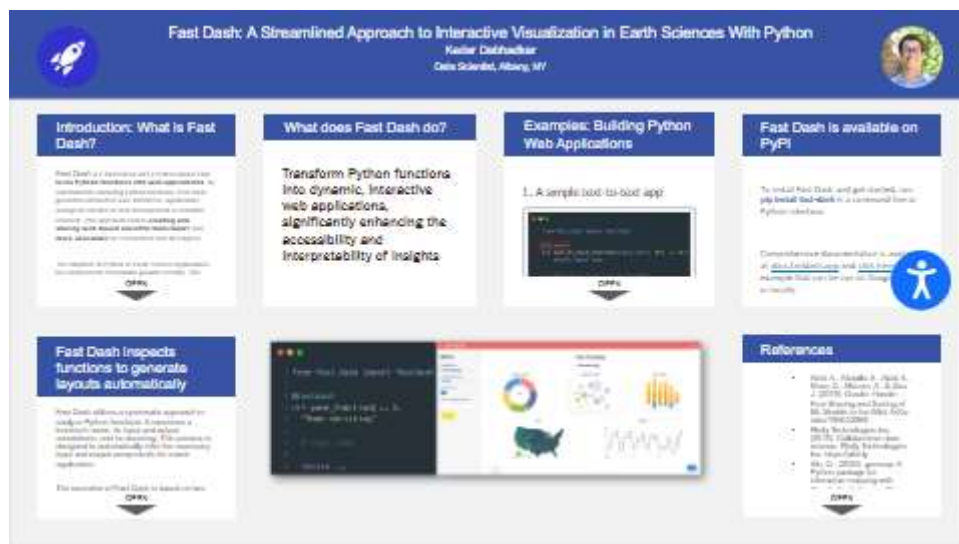


Fast Dash: A Streamlined Approach to Interactive Visualization in Earth Sciences With Python



Kedar Dabhadkar

Data Scientist, Albany, NY



PRESENTED AT:



INTRODUCTION: WHAT IS FAST DASH?

Fast Dash is a framework and a Python library that **turns Python functions into web applications**. By automatically analyzing Python functions, Fast Dash generates interactive user interfaces, significantly easing the burden of web development in scientific research. This approach makes **creating and sharing web-based scientific tools faster** and **more accessible** for researchers and developers.

The adoption of Python in Earth Science applications has experienced remarkable growth recently. This surge is largely due to the integration of Python into various scientific, machine learning, and data visualization tools and libraries. This integration has made Python a preferred choice for Earth Scientists who wish to conduct complex experiments without the need to learn multiple programming languages.

However, Python's application in web development has not kept pace with its growth in other areas. This is primarily because Python was not originally designed with web development as a core focus. Popular frameworks like Django and Flask, while powerful, often require additional languages like HTML and JavaScript for front-end development, complicating the process for those focused on Python.

To address this challenge, frameworks such as Dash, Streamlit, and Gradio have been developed. These tools offer more Python-centric approaches to building web applications but often have steep learning curves. As a result, scientists and developers need to learn additional tools outside their primary scientific programming expertise to effectively share their work with a wider audience.

Fast Dash is introduced as a solution to this problem. It is currently based on Plotly's Dash but can be easily extended to a different web development framework.

WHAT DOES FAST DASH DO?

Transform Python functions into dynamic, interactive web applications, significantly enhancing the accessibility and interpretability of insights

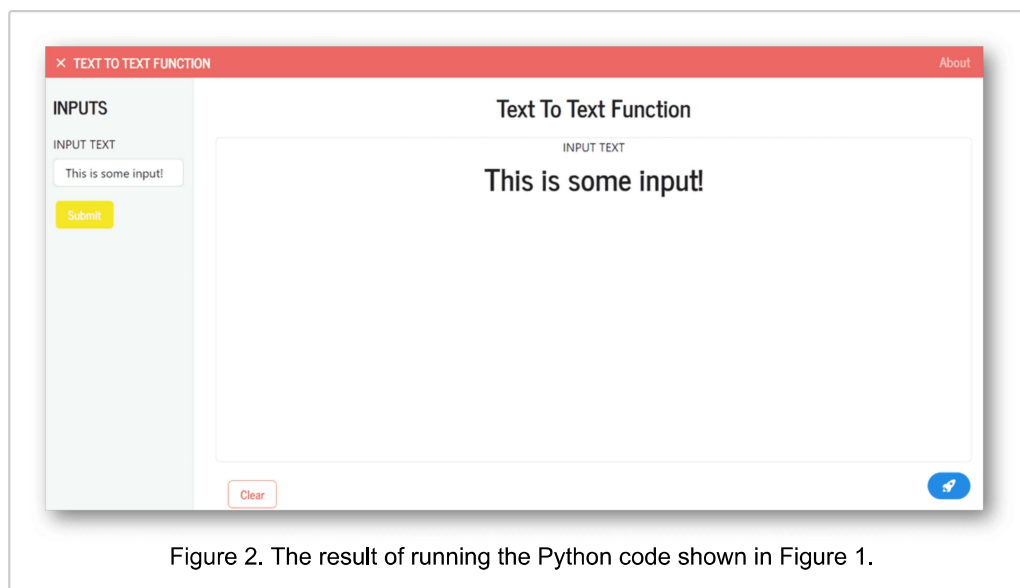
EXAMPLES: BUILDING PYTHON WEB APPLICATIONS

1. A simple text-to-text app



In this simple example, adding a "@fastdash" to your regular Python function and running the script deploys a simple Fast Dash app!

The result of running this function is:



2. Deploying Geemap

Similarly, any map based on Geemap, which extends Google Earth Engine's functionality to a Python interface, can be deployed using the code shown in Figure 3.

```
1 import ee
2 import geemap.foliumap as geemap
3
4 from fast_dash import fastdash, html
5
6 # Authenticate Google Earth Engine
7 ee.Authenticate()
8
9 @fastdash
10 def geemap_map() → html.Iframe(height="100%"):
11
12     # Define your Geemap code here
13     Map = geemap.Map(center=(40, -100), zoom=4, height=600)
14
15     # Additional logic as required
16
17     # Convert map to HTML to render in the app
18     map = Map.to_html()
19     return map
```

Figure 3. Python code to build a simple Geemap-based mapping application

The result of running code shown in Figure 3 can be seen in Figure 4.

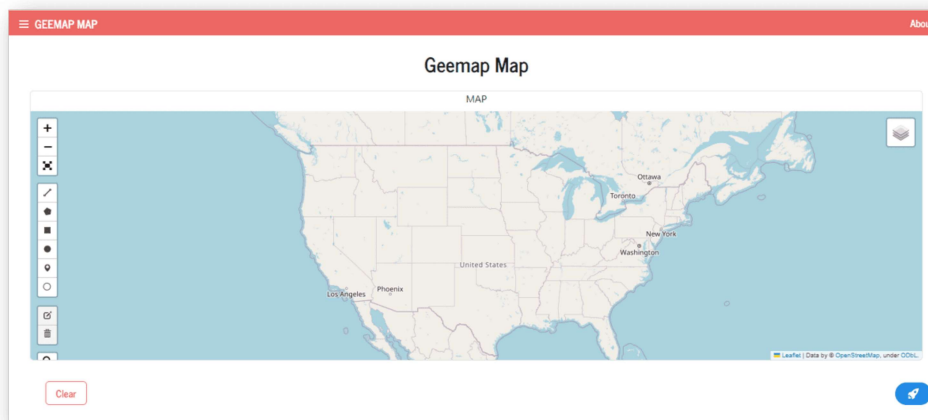


Figure 4. The result of running the code shown in Figure 3

FAST DASH IS AVAILABLE ON PYPI

To install Fast Dash and get started, run **pip install fast-dash** in a command line or Python interface.

Comprehensive documentation is available at docs.fastdash.app (<https://docs.fastdash.app>) and click here (https://docs.fastdash.app/Examples/01_simple_text_to_text/) for example that can be run on Google Colab or locally.

FAST DASH INSPECTS FUNCTIONS TO GENERATE LAYOUTS AUTOMATICALLY

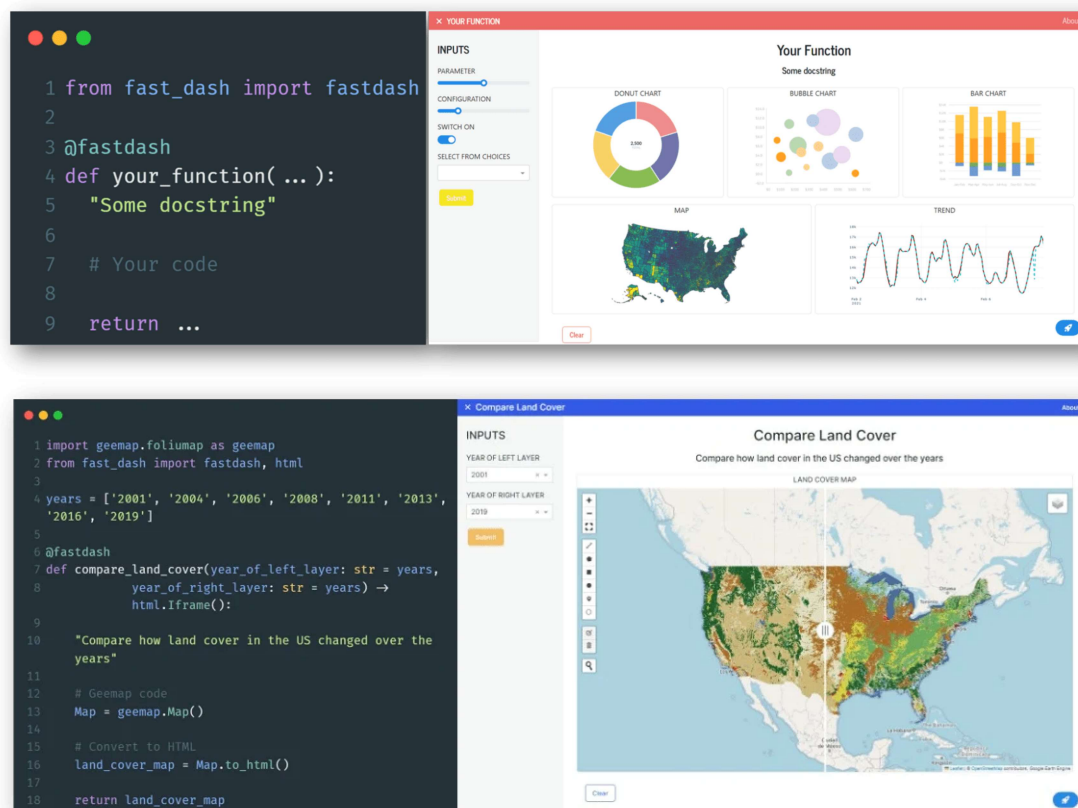
Fast Dash utilizes a systematic approach to analyze Python functions. It examines a function's name, its input and output annotations, and its docstring. This process is designed to automatically infer the necessary input and output components for a web application.

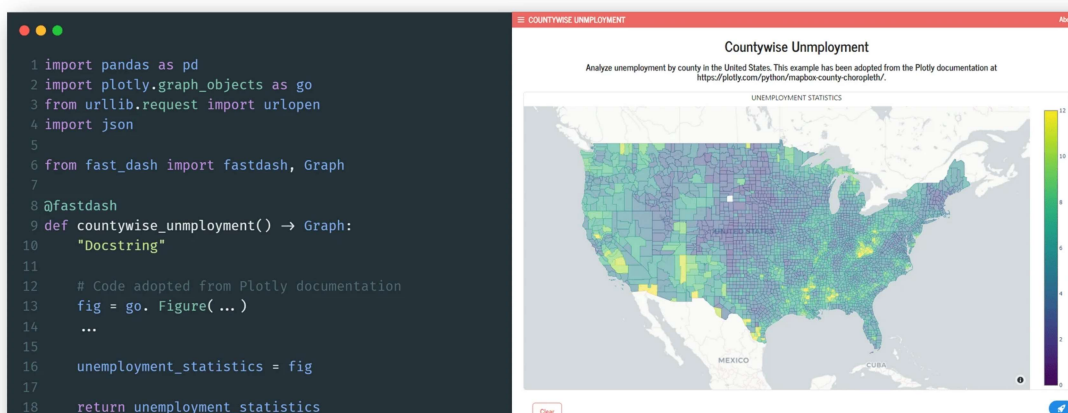
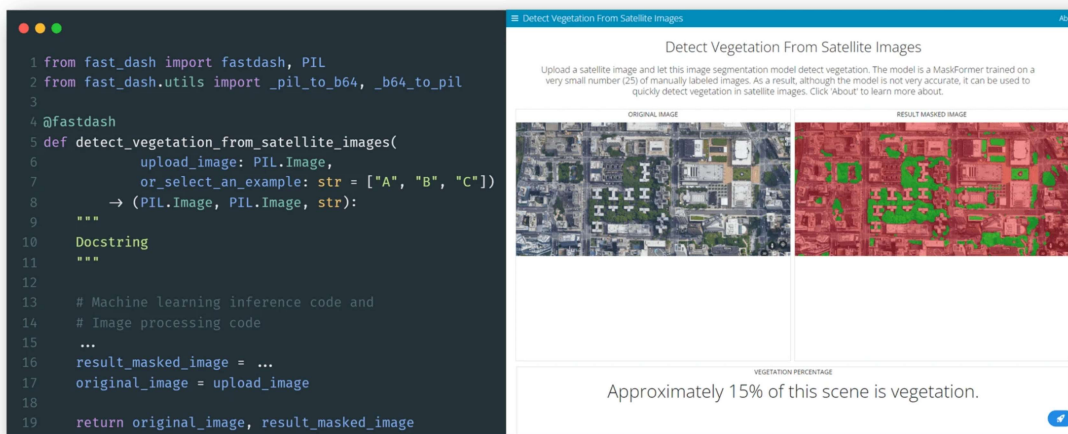
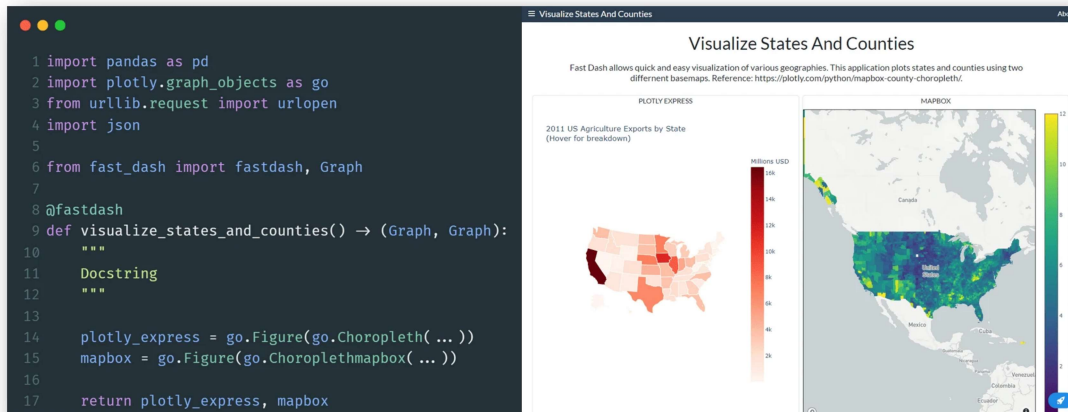
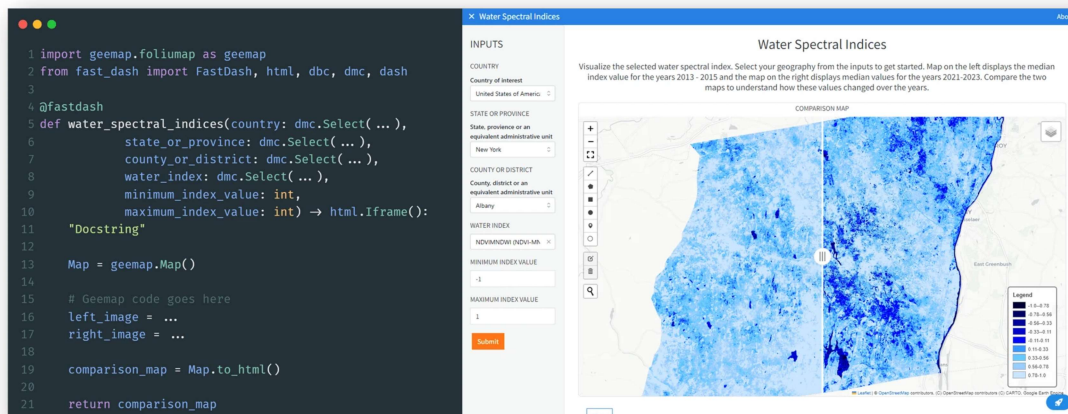
The operation of Fast Dash is based on two core principles:

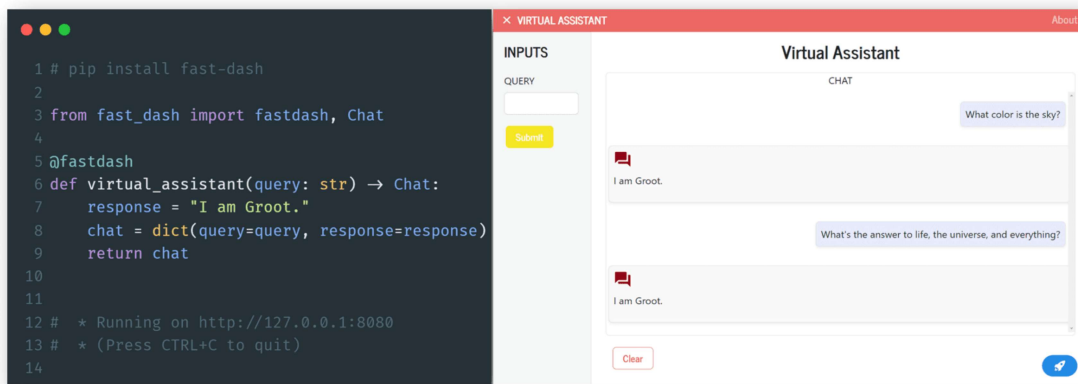
1. **Origin of Web Applications:** It posits that every Python-based web application is derived from a single Python function. This principle emphasizes the potential of a Python function as the foundational element of web application development.
2. **Function Annotations as Information Source:** The framework recognizes that a well-annotated Python function contains all the essential information needed to create an interactive web application. Annotations provide insights into the function's purpose, inputs, outputs, and operational specifics.

In adherence to these principles, Fast Dash meticulously extracts details from a Python function. It analyzes aspects such as the function's name, its docstring, and the type hints associated with inputs and outputs. This comprehensive examination enables Fast Dash to automatically generate a user-friendly layout and create an interactive web application.

Moreover, the synergy between the callback function and the configuration settings of the Fast Dash app plays a pivotal role in determining the application's deployment and interactive capabilities. A function with detailed annotations often minimizes the need for extensive customization of app configurations, simplifying the development process and enhancing user interaction.







REFERENCES

- Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. ArXiv. /abs/1906.02569
 - Plotly Technologies Inc. (2015). Collaborative data science. Plotly Technologies Inc. <https://plot.ly>
 - Wu, Q., (2020). geemap: A Python package for interactive mapping with Google Earth Engine. The Journal of Open Source Software, 5(51), 2305. <https://doi.org/10.21105/joss.02305>
 - Wu, Q., Lane, C. R., Li, X., Zhao, K., Zhou, Y., Clinton, N., DeVries, B., Golden, H. E., & Lang, M. W. (2019). Integrating LiDAR data and multi-temporal aerial imagery to map wetland inundation dynamics using Google Earth Engine. Remote Sensing of Environment, 228, 1-13. <https://doi.org/10.1016/j.rse.2019.04.015> (pdf | source code)
-

AUTHOR INFO

Reach out on LinkedIn: <https://linkedin.com/in/dkedar7>

TRANSCRIPT

ABSTRACT

The exponential growth and complexity of Earth Science data call for innovative, adaptable, and user-friendly visualization tools. This study introduces 'Fast Dash', an open-source Python package that efficiently converts Python functions into interactive web applications, emphasizing geospatial mapping through integrating libraries such as Matplotlib, Geoplot, Plotly, and Leaflet.

The novelty of Fast Dash lies in its capacity to intuitively infer input and output layout components from Python function type hints. For instance, string type hints are transformed into text inputs, float hints manifest as sliders, and list hints become dropdowns. In a similar vein, the output layout is guided by the function outputs' type hints, with an option for visually appealing arrangements via mosaic ASCII art representations.

Grounded in the robust architecture of Plotly Dash and Flask, Fast Dash offers scalability, supports the visualization of large datasets via Datashader, and provides a streamlined process for deployment and sharing.

Fast Dash is underpinned by the understanding that well-annotated Python functions encapsulate all necessary information for designing interactive visualizations. In the domain of Earth observation, marked by a multitude of tools and a lack of standardized data visualization practices, Fast Dash presents a compelling solution. By significantly reducing the time between development and deployment, Fast Dash fosters a climate of accelerated innovation in Earth Science data visualization, further enhancing the interpretation and accessibility of complex data sets.

