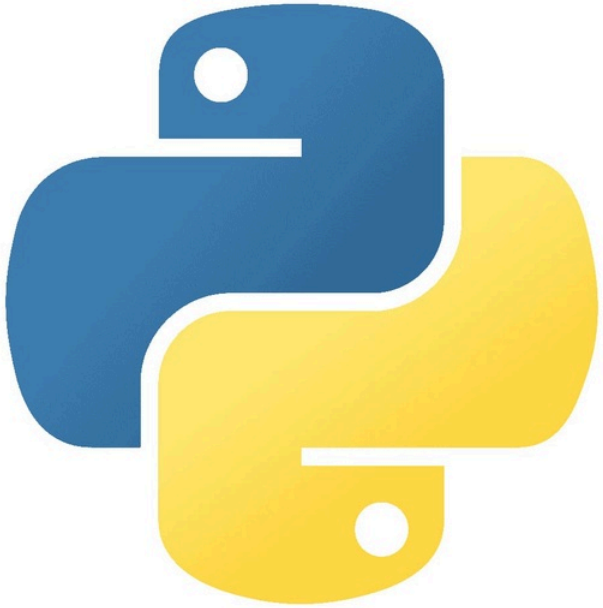


python™

DEEP DIVE ON PANEL LIBRARY

Andres Gomez

AGENDA

1. Introduction to Panel
2. Why Use Panel?
3. Installation and Setup
4. Core Concepts and Structure
5. Types of Widgets
6. Visualization Capabilities
7. Tutorial Walkthrough
8. Code Examples
9. Advanced Functionalities
10. Comparison
11. Best Practices

INTRODUCTION TO PANEL

Open-source Python library

Developed by HoloViz team

Build interactive web apps and dashboards

Works seamlessly with Jupyter, VS Code, or deployed on servers

Supports static and live apps

Allows for quick prototyping and full deployment

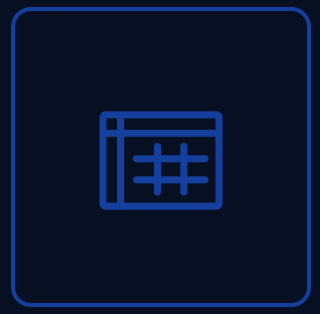
Works with tools you already use

Integrates with pandas, matplotlib, Bokeh, Plotly, and more

No need for JS/HTML/CSS knowledge

Focus on your Python skills to build interactive applications

WHY USE PANEL?

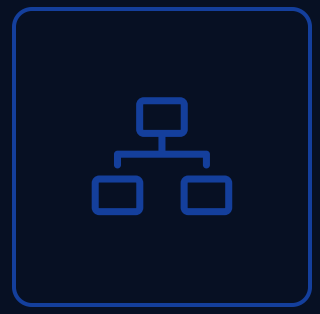


Rapidly Build Interactive Dashboards

Panel allows data scientists to quickly build interactive web applications and dashboards without having to learn frontend frameworks like JavaScript, HTML, or CSS.

Leverage Existing Data Science Tools

Panel seamlessly integrates with popular data science libraries like pandas, matplotlib, Bokeh, and Plotly, allowing data scientists to leverage their existing skills and workflows.



Streamline Prototyping to Production

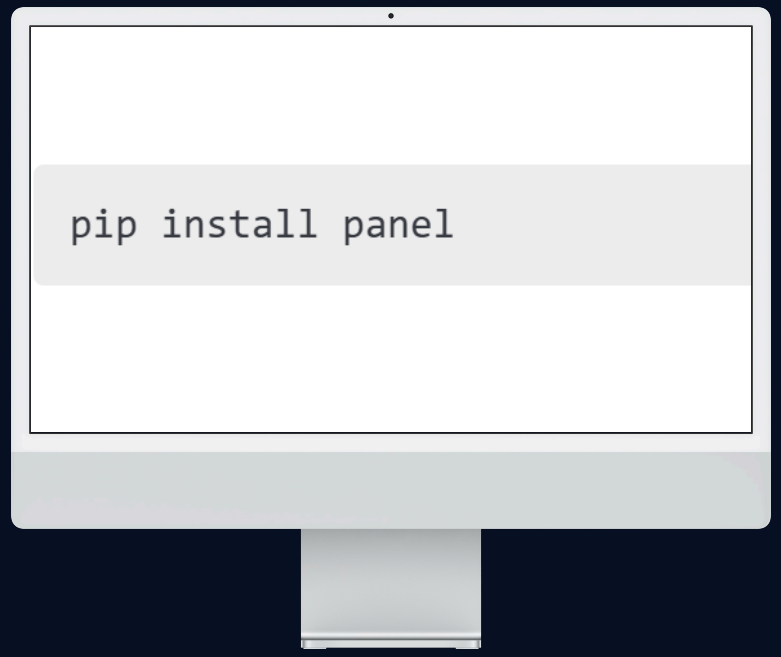
Panel makes it easy to start with simple prototypes and then scale up to full-fledged applications that can be deployed on servers, shared with stakeholders, or embedded in Jupyter Notebooks.

Reactive Programming Capabilities

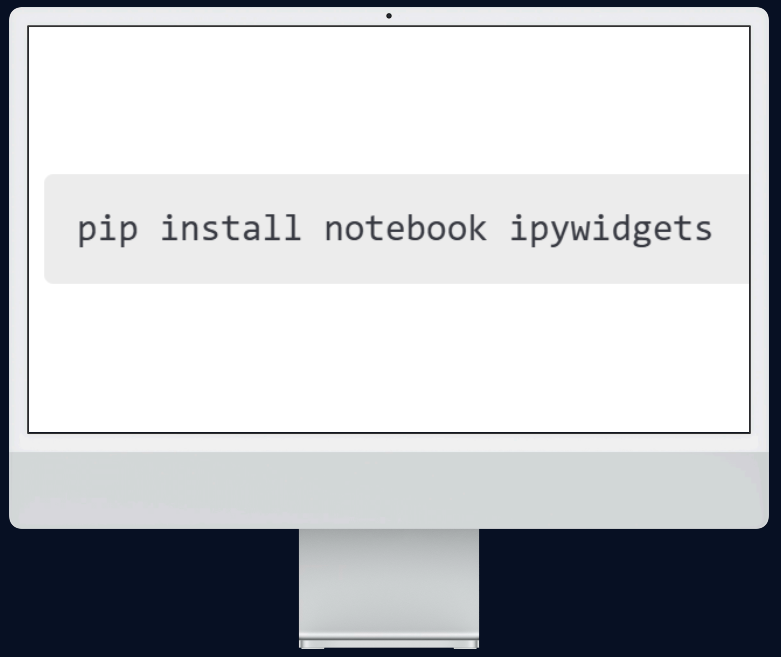
Panel's reactive programming model allows you to easily bind widgets to functions, creating interactive visualizations and dashboards that update in real-time.

Panel is a powerful and Pythonic library that empowers data scientists to rapidly build, deploy, and share interactive applications without having to learn complex frontend technologies.

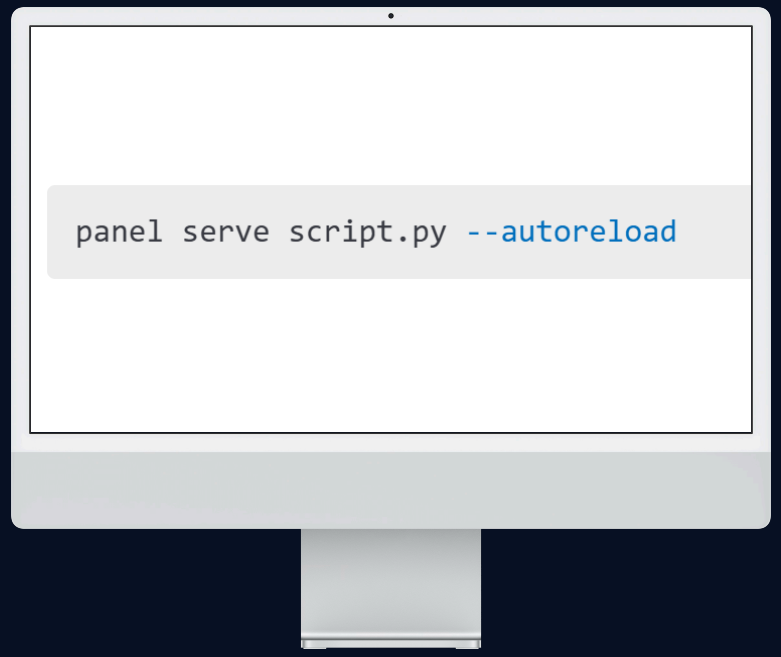
INSTALLATION AND SETUP



Installs the Panel library from PyPI using pip



Required if using Panel inside Jupyter Notebooks to ensure widgets render properly



Starts a local web server to run your Panel app in script.py, and --autoreload watches your file and auto-refreshes your app whenever changes are saved

CORE CONCEPTS AND STRUCTURE

Pane: Display components

Panes are used to display different types of content in your Panel app, such as Markdown, Plotly, Matplotlib, etc.

Widgets: User inputs

Widgets are interactive UI elements that allow users to provide input, such as sliders, dropdowns, buttons, etc.

Layout: Organize UI

Layouts are used to arrange and organize the different components of your Panel app, such as rows, columns, tabs, etc.

Reactive Programming

Panel uses reactive programming principles to bind widgets to functions, allowing for dynamic updates and interactivity.

These core concepts and structures are the building blocks of creating interactive data science applications with Panel.

TYPES OF WIDGETS IN PANEL

- **Basic Inputs**: TextInput, Checkbox, Select, RadioButtonGroup
- **Sliders**: IntSlider, FloatSlider, RangeSlider
- **Pickers**: DatePicker, ColorPicker, FileInput
- **Buttons & Actions**: Button, Toggle, LiteralInput
- **Complex**: DataFrame, AutocompleteInput, MultiSelect

VISUALIZATION CAPABILITIES

Matplotlib
Static plots with full control

Plotly
Interactive visualizations

Bokeh
Linked brushing, streaming

Altair
Declarative statistical graphics

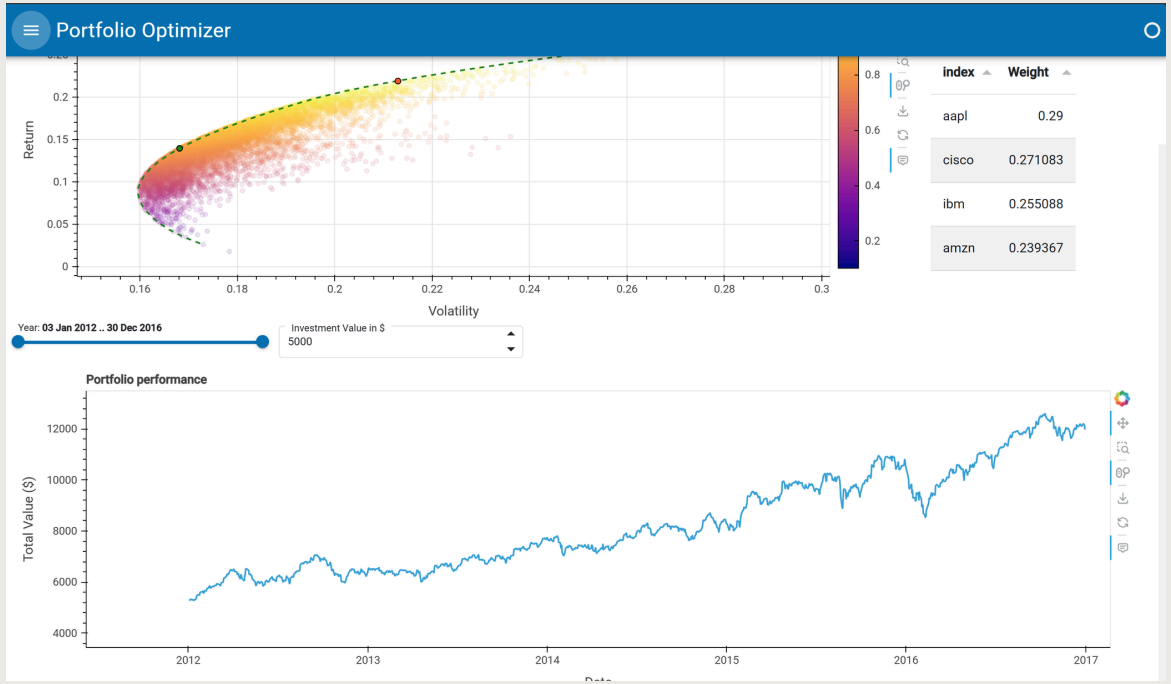
HoloViews
High-level compositional plots

Vega/vega-lite
JSON-based visualization

Other media
HTML, Markdown, PNG, GIF, PDF

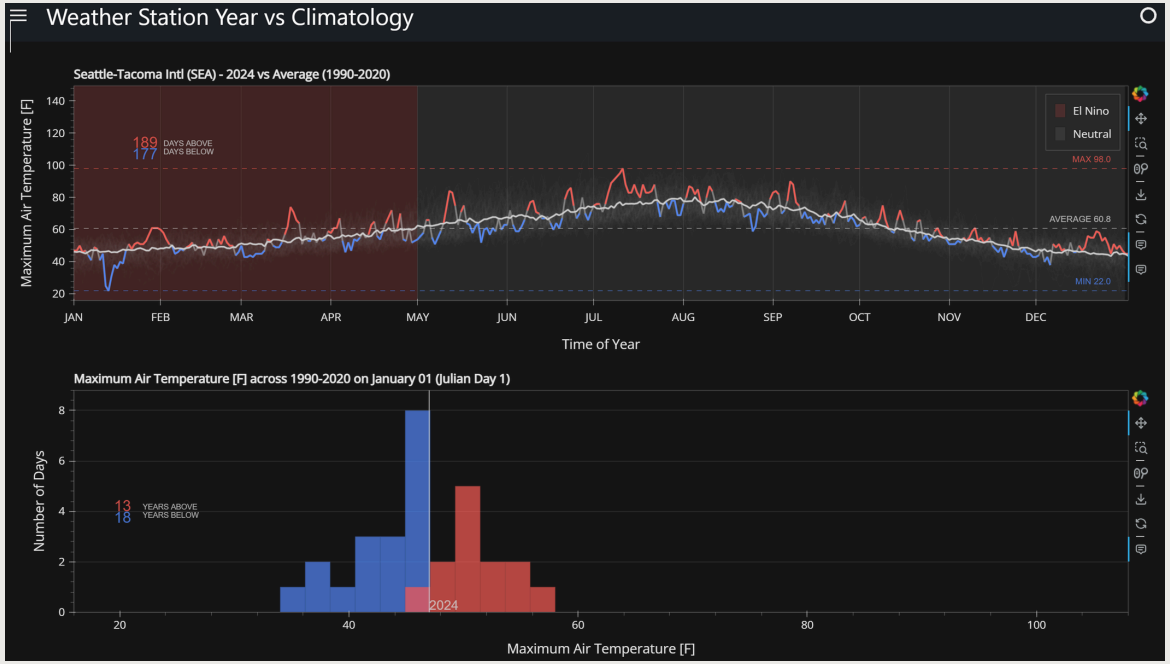
CODE EXAMPLES

ADVANCED EXAMPLES



Portfolio Optimizer

Optimize asset allocations interactively using financial data



Year vs Climatology Dashboard

Explore climate pattern comparisons

ADVANCED FUNCTIONALITIES

Theming (Dark/Light)

Ability to switch between dark and light themes for your application's user interface

Embedding apps in Jupyter or web

Easily embed your Panel applications within Jupyter Notebooks or deploy them as standalone web applications

Deploy with Binder, Heroku, or static export

Multiple options to deploy your Panel apps, including Binder for interactive demos, Heroku for web hosting, or static HTML export

Integrate with HoloViews, Bokeh, Plotly, Vega

Seamless integration with other popular data visualization libraries, allowing you to leverage their features within your Panel apps

Panel's advanced functionalities provide a versatile and extensible platform for building interactive data applications, empowering data scientists to create sophisticated and visually compelling dashboards and reports.

VS OTHER LIBRARIES

	JS knowledge	Custom Layouts	Jupyter Native	Widgets
Panel				Extensive
Streamlit		Limited		Good
Dash	Some			Good

BEST PRACTICES

Follow PEP-8 guidelines

Adhere to the official Python style guide for consistent and readable code

Use docstrings for documentation

Provide clear and comprehensive documentation for functions and modules

Modularize your code

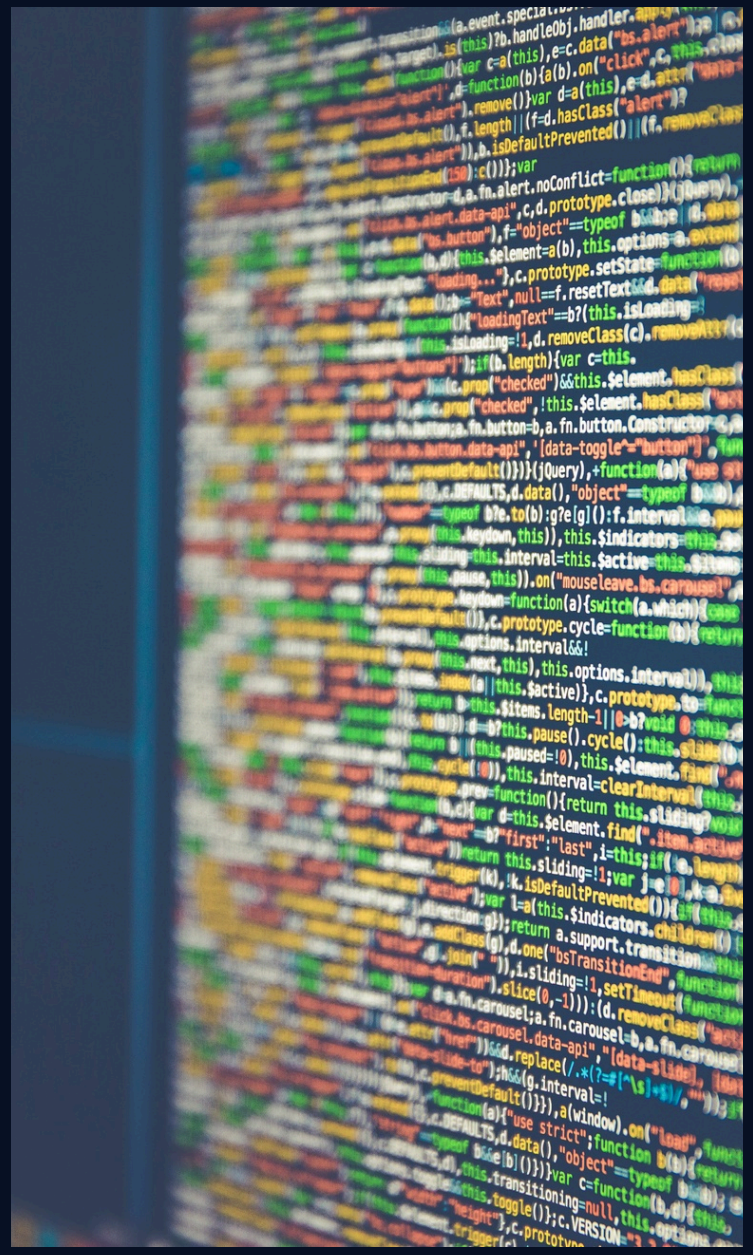
Separate widgets, logic, and UI components for better organization and maintainability

Use meaningful variable names

Choose descriptive names to improve code readability and understanding

Limit inline comments

Use inline comments only when necessary, prefer self-explanatory code



CONCLUSION

Panel is a powerful open-source Python library developed by HoloViz that enables data scientists to build interactive web apps and dashboards. It works seamlessly with existing data science tools and libraries, allowing you to create sophisticated visualizations and GUIs without the need for frontend web development expertise.

QUESTIONS

RESOURCES & DOCUMENTATION

- Panel Docs
  https://panel.holoviz.org/
- GitHub
  https://github.com/holoviz/panel
- Examples Gallery
  https://awesome-panel.org/
- PEP-8 Guide
  https://peps.python.org/pep-0008/