

Streamlit

Purpose

Streamlit is an open-source Python framework for rapidly turning data scripts into interactive web apps with only a few lines of code and zero front-end knowledge³⁴. It is purpose-built for data scientists and ML engineers to share results visually and interactively—no HTML, CSS, or JS required.

Core Functionalities

- **Interactive Widgets & Components:** Sliders, inputs, data tables, and more.
- **Live Re-running:** Instant UI updates as the Python script changes.
- **Auto Integration with pandas, matplotlib, Plotly:** Seamless for data vis.
- **Simple API:** Minimal learning curve for anyone familiar with Python.

Project Orientation

Streamlit is ideal for **data exploration tools, ML model prototyping and demos, quick interactive dashboards, and classroom data science demonstrations.**

Model Example

python

```
import streamlit as st
import pandas as pd

st.write("## Hello, Streamlit!")
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
st.line_chart(df)
```

Launch via `streamlit run app.py` and view live updates in your browser.

Advantages

Supremely simple and fast for ML/data prototyping. Learners can go from dataset to web app in minutes. Automatic, responsive UIs are generated for Python functions, aiding collaboration and reproducibility in data science workflows.

Limitations

Not suitable for general-purpose web development; limited customization of UI and navigation. For production-grade or highly trafficked apps, performance can suffer. Streamlit is not meant for apps needing complex authentication or back-end integrations.

Guidance for Learners

For **data science students, ML enthusiasts, researchers, and educators**, Streamlit is a must-learn. Use Streamlit to:

- Share Jupyter-like data apps with stakeholders
- Build real-time dashboards for local businesses or research teams
- Prototype machine learning models in hackathons or university competitions

For full-stack web apps or standardized UIs, switch to Dash, Flask, or Django.

Dash

Purpose

Dash by Plotly is an open-source Python framework specifically tailored for building data-centric, analytical web applications with interactive visualizations. It is highly favored in business intelligence, scientific computation, and enterprise reporting.

Core Functionalities

- **Component-based UI:** Reusable components rendered as React.js behind the scenes.
- **Plotly Graph Integration:** Advanced charts, maps, 3D plots.
- **Real-time UI Updates:** Callback-driven, interactive user inputs.
- **Pure Python Development:** No JS or CSS knowledge required.
- **Production Deployment:** Dash Enterprise for enterprise features.

Project Orientation

Dash is built for **interactive dashboards, reporting apps, advanced analytics, data science portals, and internal business tools**.

Model Example

python

```
import dash
from dash import html

app = dash.Dash(__name__)
app.layout = html.Div(children=[
    html.H1('Hello Dash!'),
    html.Div('Dash: A web application framework for Python.')
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

Advantages

Dash comes equipped with easy linking of interactive charts and UI widgets, facilitating rapid analytical app development. No need for bespoke web skills—ideal for data-driven organizations. Dash’s Plotly underpinnings give it unmatched data vis and real-time updates³⁸.

Limitations

Limited UI customization compared to general web frameworks. Advanced layouts or navigation can be complex, and learning Dash’s callback architecture takes effort. For projects with no analytical component, it’s overkill.

Guidance for Learners

Data analysts, scientists, and business-savvy developers should **use Dash for client dashboards, academic presentations, NGO data portals, and any scenario where visual analytics are paramount.** For mobile-first or generic web products, use another framework.