App Development Tools

Easiest, Quickest Dev Time, Powerful

Streamlit

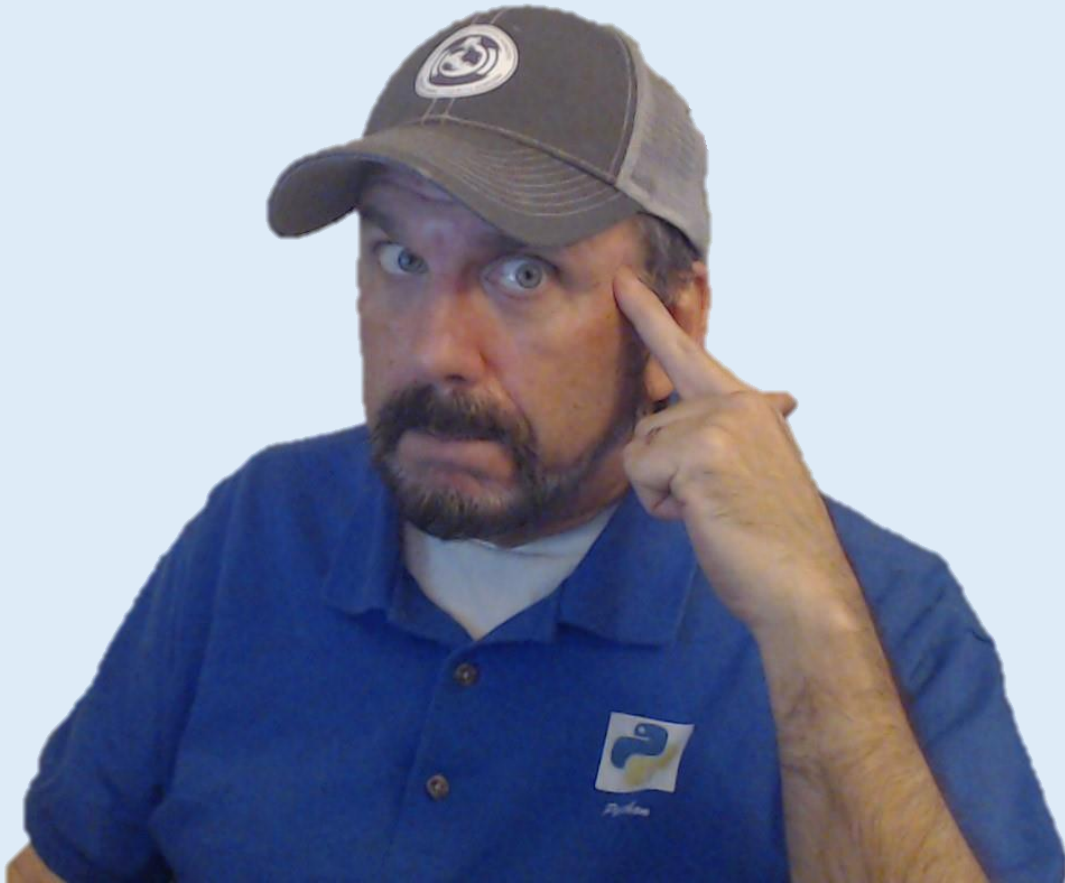
python™ django

Flask
web development,
one drop at a time

Shiny

Flutter

FlutterFlow
flutterflow.io



https://www.youtube.com/c/BryanCafferky/videos

Where We're Going

- **Why Dev Tools?**
- **Two Types**
- **Goals**
- **Quick List**
- **Service By Service Highlights**

Why Dev Tools

- Build/Deploy Your Career Portfolio
- Apps are Incorporating Data & AI
- Build an Application to Support a Need You Have

The Two Types

- **Web App Development Tools**
  - Can Run Anywhere (on Mobile, Desktop, & Laptops)
  - Easy Maintenance & Deployment and No Need to Push Updates
  - Can Integrate Almost Anything
  - No Need to Deal with an App Store (You're In Control)
- **Cross Platform Mobile App Development Tools**
  - May Be Able to Be Deployed as Web App
  - Deploy to App Store (Monetize)
  - Run Super Fast
  - Full Application Interface including hand gestures like swiping

My Goals

- Ease of Development
- Ease of Deployment
- Power & Extensibility of Service
- Scalability
- Maintainability (Configurable & Support High Reusability)
- Preference – No Code or Low Code (or Python based)
- Python

Quick List

Streamlit

R Based

Kivy

Not Being Covered

LiveScript

.Net Cross
Platform Dev

Worldwide, May 2024 :

Rank	Change	Language	Share
1		Python	28.98 %
2		Java	15.97 %
3		JavaScript	8.79 %
4		C#	6.78 %
5		C/C++	6.46 %
6	↑	R	4.76 %
7	↓	PHP	4.55 %
8		TypeScript	3.03 %
9		Swift	2.76 %
10		Rust	2.6 %
11		Objective-C	2.41 %
12		Go	2.25 %
13		Kotlin	1.97 %
14		Matlab	1.52 %
15	↑↑↑↑	Dart	1.0 %
16	↓	Ruby	0.94 %

https://pypl.github.io/PYPL.html

Web App Development

Service By Service

Streamlit

python

Pros

- Open-Source Python Fully Interactive Development Framework (React.js under the covers)
- Super Easy and Intuitive to Develop Apps
- Ideal for Data Science and AI apps
- Built-In Web Server for Testing

Cons

- Not Designed for Large Website Applications
- To scale, probably need to create a backend using something like Flask.
- State Management is a pain.

```python
import streamlit as st
from streamlit.logger import get_logger

LOGGER = get_logger(__name__)


def run():
    st.set_page_config(
        page_title="Hello",
        page_icon="👋",
    )

    st.write("# Welcome to Streamlit! 👋")

    st.sidebar.success("Select a demo above.")
```

python

Pros

- Tagline "A Web Dev Framework for Perfectionists with Deadlines".
- Sophisticated and Logically laid out framework.
- Built-In Services like Database, Security Admin, data table crud, data ORM, local webserver.
- Great for any size websites but especially good for large ones.
- Highly Configurable and Parameterizable (Jinja). Inherit pre-built forms.
- Maximum Reusability Support!

Cons

- Steep Learning Curve.
- No One-Click Deployment.
- Heavy on code.

anvil
WEB APPS MADE SIMPLE

python

https://anvil.works/

Pros

- All Python (yet interactive – translates to React.js)
- GUI Development
- Easy Deployment to the anvil cloud service

Cons

- Commercial and Limited to their Cloud Service
- No Local Development IDE Client Tool
- Unless You Go Open Source Only and Self Host, very limited scalability.
- Costs Rise Quickly When You Use Their Hosting.

Pros

- Python based.
- Very lightweight Web Development Framework.
- Just Code for what you need.
- Integrates with just about anything.

Cons

- Not much included out of the box.
- Not suited for large apps (meaning many web pages)
- You have to code everything you need.

For the R Language

Pros

- Like Streamlit for R. Best for R Developers.
- Intuitive syntax and concepts.
- Easy to deploy to Shiny service. Right from RStudio.
- Ideal for Data Science and AI apps.

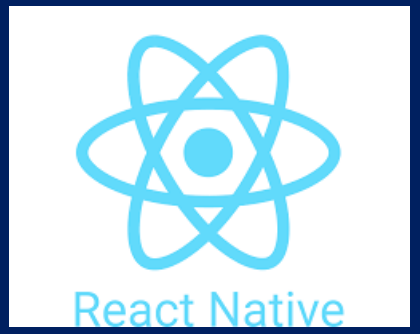
Cons

- Function based means lots of nested calls which is hard to read. (Hint: Use Flexboard)
- Limited to the R language and stack.

Mobile App Development

Service By Service

Skipping For Now



Kivy

Pros

- All Python and Cross Platform. Open Source.
- Flexible Layout Managers.
- Support for Mobile Device actions.

Cons

- Infrequent releases – future support?
- Does not support deployment to App Stores.
- Not ideal for large apps.

Flutter &

FlutterFlow

flutterflow.io

Pros

- FlutterFlow provides full GUI drag and drop development.
- Great Integrated Development Environment.
- Cross Platform and Fast App Execution.

Cons

- Google DART – Yet Another Language.
- If you need custom behavior, you'll need to learn DART.

Flet (Flutter on Python)

Pros

- Python! Wrapper++.
- Leverages Flutter.

Cons

- New Flet 1.0 Planned for 2024.
- Long term support and development?

counter.py

```python
import flet as ft

def main(page: ft.Page):
    page.title = "Flet counter example"
    page.vertical_alignment = ft.MainAxisAlignment.CENTER

    txt_number = ft.TextField(value="0", text_align=ft.TextAlign.RIGHT, width=100)

    def minus_click(e):
        txt_number.value = str(int(txt_number.value) - 1)
        page.update()

    def plus_click(e):
        txt_number.value = str(int(txt_number.value) + 1)
        page.update()

    page.add(
        ft.Row(
            [
                ft.IconButton(ft.icons.REMOVE, on_click=minus_click),
                txt_number,
                ft.IconButton(ft.icons.ADD, on_click=plus_click),
            ],
            alignment=ft.MainAxisAlignment.CENTER,
        )
    )

ft.app(main)
```

https://flet.dev/

In Summary

- **Why Dev Tools?**
- **Two Types**
- **Go**
- **Quick List**
- **Service By Service Highlights**

Thank You!