

# Python on Android with Delphi FMX

The Cross Platform GUI Framework  
Part 2



**Jim McKeeth**  
Chief Developer Advocate  
**Embarcadero Technologies**  
jim.mckeeth@embarcadero.com  
@JimMcKeeth



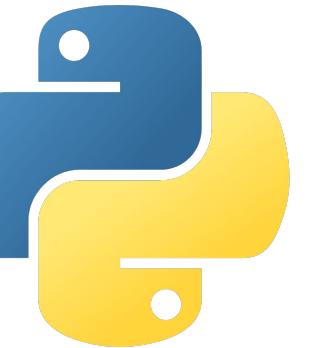
# Agenda

- Introduction to Delphi FMX
- Architecture and platforms
- Installing and using Delphi FMX for Python
- Demonstrations & Code
- Python on Android
- Going Beyond - Mixing Delphi & Python
- More information, Next steps, Q&A



# About Jim McKeeth

- Chief Developer Advocate & Engineer for Embarcadero
- Long time software developer
  - Delphi, C/C++, Python, Java, JavaScript, Ruby, etc.
- Invented and patented pattern and swipe to unlock
  - e.g. US Patents # 8352745 & 6766456
- Built thought controlled drone with Google Glass and wireless EEG headset
- Contributor to *Internet of Things and Data Analytics Handbook*
- Blogger, podcaster, conference speaker, webinar host, etc.
- Twitter, TikTok, YouTube, etc. @JimMcKeeth



# Who are you?

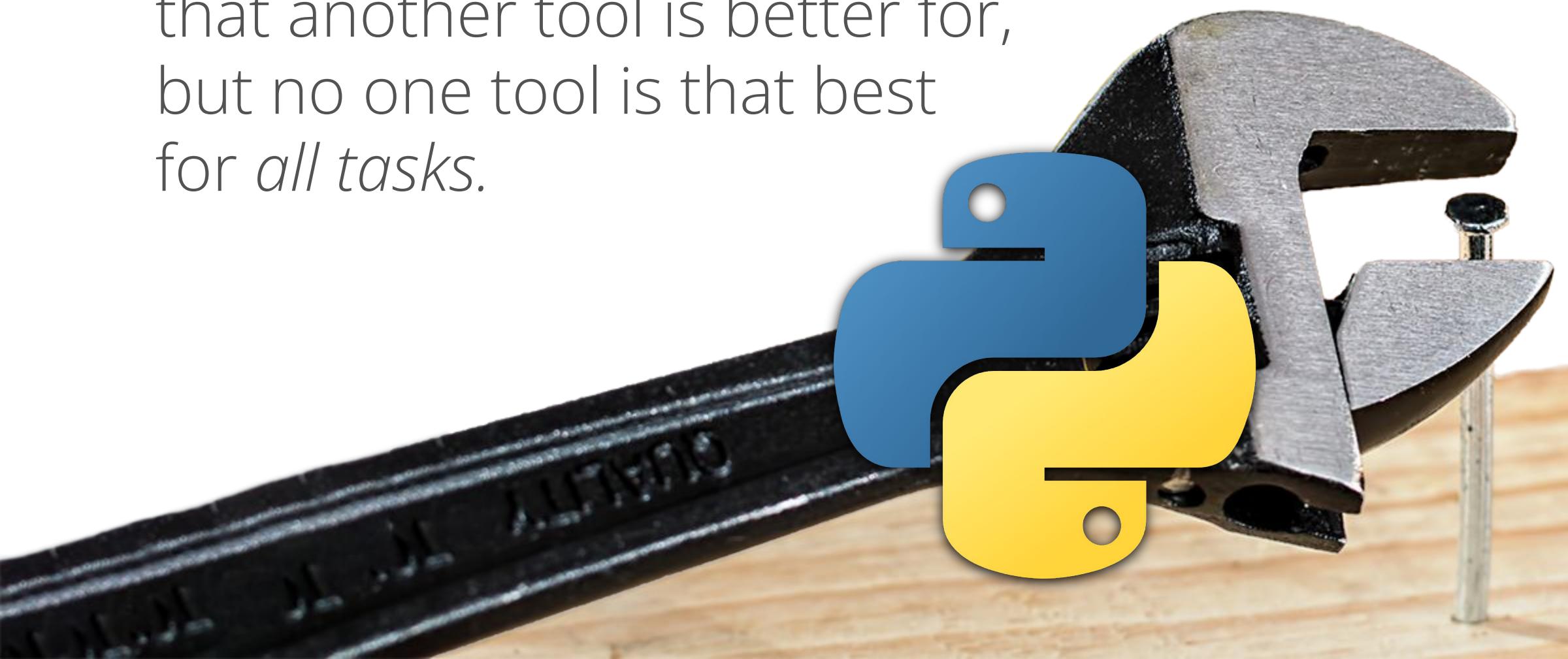
- Python devs who want a nice GUI
- Python devs curious about Delphi
- Delphi devs who want to use Python
- Delphi devs curious about what's new
- Other devs curious about Delphi & Python

*I'll do my best not to assume too much familiarity with either Delphi or Python, but also include code and technical details.*



# It's not a Competition

- Developers have multiple tools on their workbench
- It is about finding the right tool for each task
- Having specialized tools for different tasks doesn't detract from *favorite* tools
- You can always find a specific task that another tool is better for, but no one tool is that best for *all* tasks.

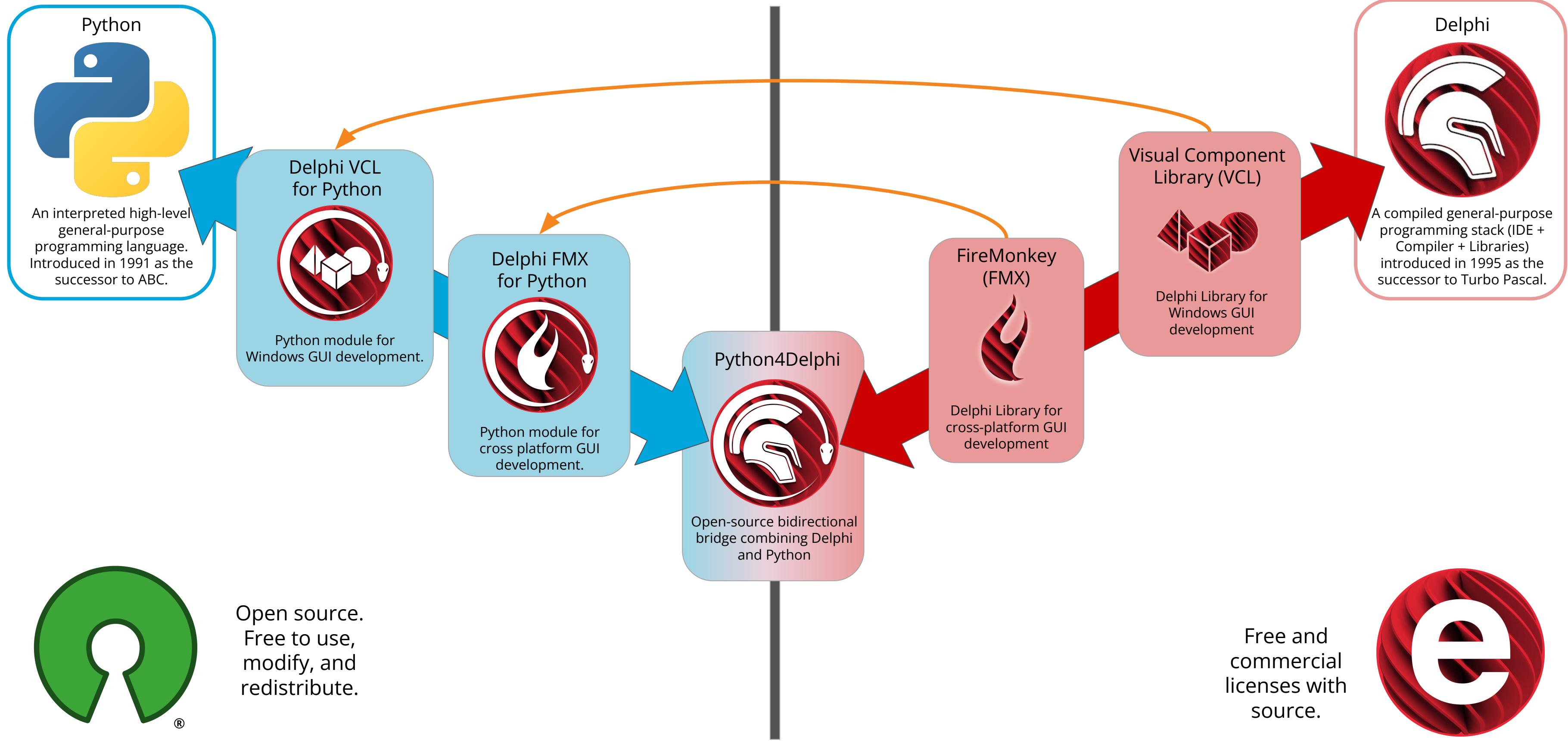


# What is Delphi for Python?

- Set of free Python modules bringing Delphi's GUI libraries to Python developers
  - Mature, feature rich, native & cross-platform
  - *Does not require Delphi to use*
- Based on the open source **Python4Delphi** (same technology that powers the *PyScripter IDE*)
- Available today on GitHub & PyPi
  - Currently in *beta*, but ready for use
- **DelphiVCL for Python** supports Windows (32-bit & 64-bit)
- **DelphiFMX for Python** adds Linux, Android, & Mac OS
- Part of a bidirectional bridge between Delphi and Python



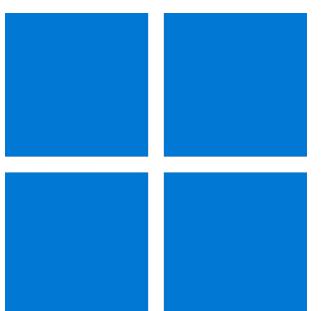
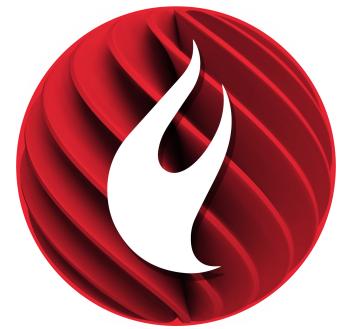
# Dissecting the Technology Stacks



# FMX

# The Cross-Platform FireMonkey Framework

- Takes advantage of GPU libraries to provide a hardware accelerated, rich user interface that is fast and looks great across multiple platforms:
  - Windows, macOS, iOS, Android, and Linux
  - Uses DirectX on Windows, OpenGL on Linux, OpenGL-ES on Android, and Metal on iOS and macOS
- Similar to VCL, but not designed to be compatible
  - Designed as cross platform from the ground up
- Integrated GPU effects, animations, and robust styling system
- Platform services abstract the access to platform hardware and functionality to intelligently adapt the UI & UX to platform specifics
- Very flexible component system - do more with fewer components (nestable & styleable)





# Getting Started with Delphi FMX for Python

Rich GUI Framework for Python  
on Windows, Mac, Linux, & Android



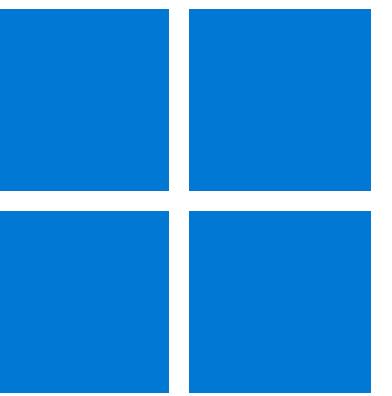
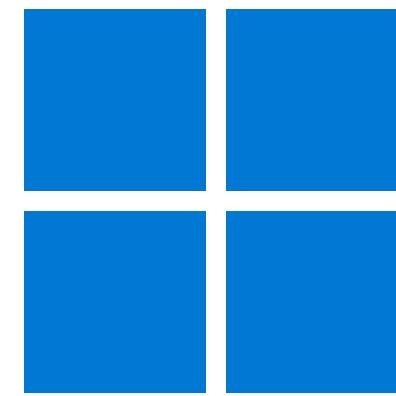
Covered in  
Part 1

# Delphi for Python



## Delphi VCL for Python

- Windows 32-bit and 64-bit only
- Windows 8.1 through Windows 11
  - (Earlier versions may work, but not supported.)
- Based on native Windows components
- Includes Windows Handles, Messages, Accessibility, etc.
- Styling system



# Delphi FMX for Python Installation

- Install via pip
  - `pip install delphifmx`
- Supports:
  - Win32 x86, Win64 x86, Linux64 x86, Android64, Mac OS x64 (Intel) and Mac M1 (Arm) architectures
  - Python cp3.6, cp3.7, cp3.8, cp3.9 and cp3.10 (excluding cp3.6 on Linux and macOS)
- Conda support:
  - Win x86 and x64 from Python cp3.6 to cp3.10
  - Linux x86\_64 from Python cp3.7 to cp3.9 (*Recommended!*)
  - macOS not supported yet
- Details and downloads
  - [github.com/Embarcadero/DelphiFMX4Python](https://github.com/Embarcadero/DelphiFMX4Python)
  - [pypi.org/project/delphifmx/](https://pypi.org/project/delphifmx/)



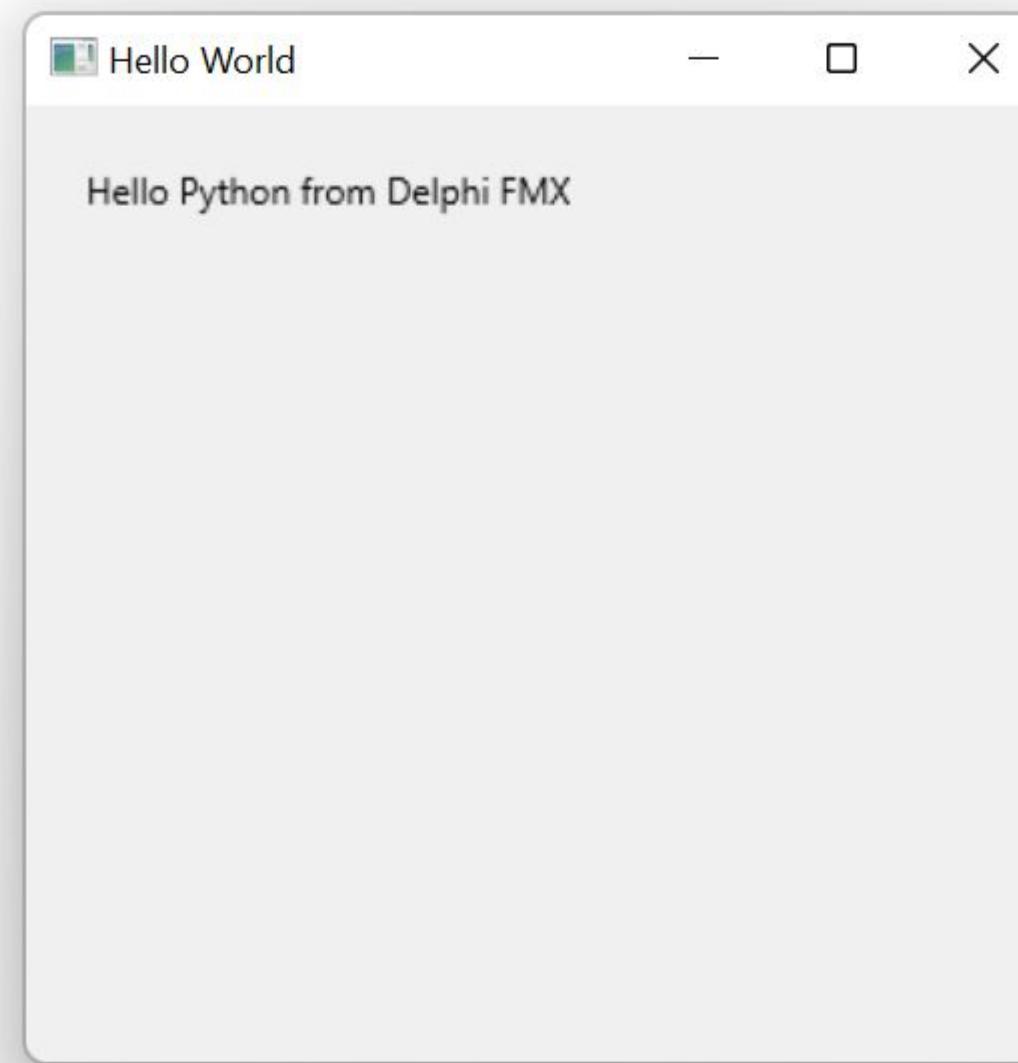


# Hello Delphi FMX for Python



# Hello World

The simplest example



```
from delphifmx import *
```

```
Application.Initialize()  
Application.Title = "Hello Delphi FMX"
```

Configure the Application

```
Application.MainForm = Form(Application)  
Application.MainForm.SetProps(Caption = "Hello World")
```

Configure the form

```
msg = Label(Application.MainForm)  
msg.SetProps(Parent = Application.MainForm,  
             Text = "Hello Python from Delphi FMX",  
             Position = Position(PointF(20, 20)),  
             Width = 200)
```

Create and configure the label

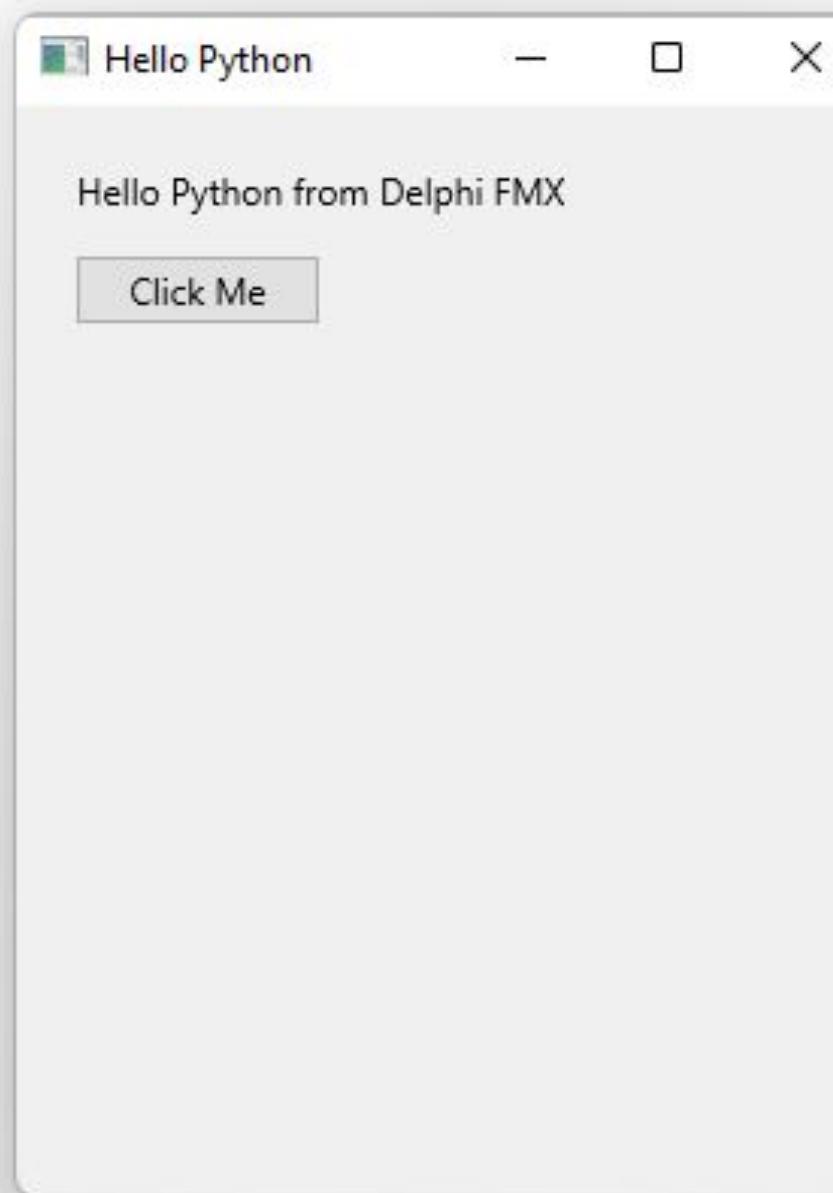
```
Application.MainForm.Show()  
Application.Run() # This is the main loop  
Application.MainForm.Destroy()
```

Show the form and start the main loop

<https://github.com/Embarcadero/DelphiFMX4Python/blob/main/samples/Simplest.py>

# Hello World

An Object Oriented example



```
from delphifmx import *
class HelloForm(Form):
    def __init__(self, owner):
        self.SetProps(Caption = "Hello Python",
                      Position = "poScreenCenter", OnShow = self.__form_show)

        self.hello = Label(self)
        self.hello.SetProps(Parent = self,
                           Text = "Hello Python from Delphi FMX",
                           Position = Position(PointF(20, 20)), Width = 200)

        self.clickme = Button(self)
        self.clickme.SetProps(Parent = self, Text = "Click Me",
                             Position = Position(PointF(20, 50)), OnClick = self.__button_click)

    def __form_show(self, sender):
        self.SetProps(Width = 300, Height = 400)

    def __button_click(self, sender):
        self.hello.Text = "Thanks!"
        self.Width = 300

def main():
    Application.Initialize()
    Application.Title = "Hello Delphi FMX"
    Application.MainForm = HelloForm(Application)
    Application.MainForm.Show()
    Application.Run()
    Application.MainForm.Destroy()

if __name__ == '__main__':
    main()
```

HelloForm is an FMX Form object

Button has an event handler assigned

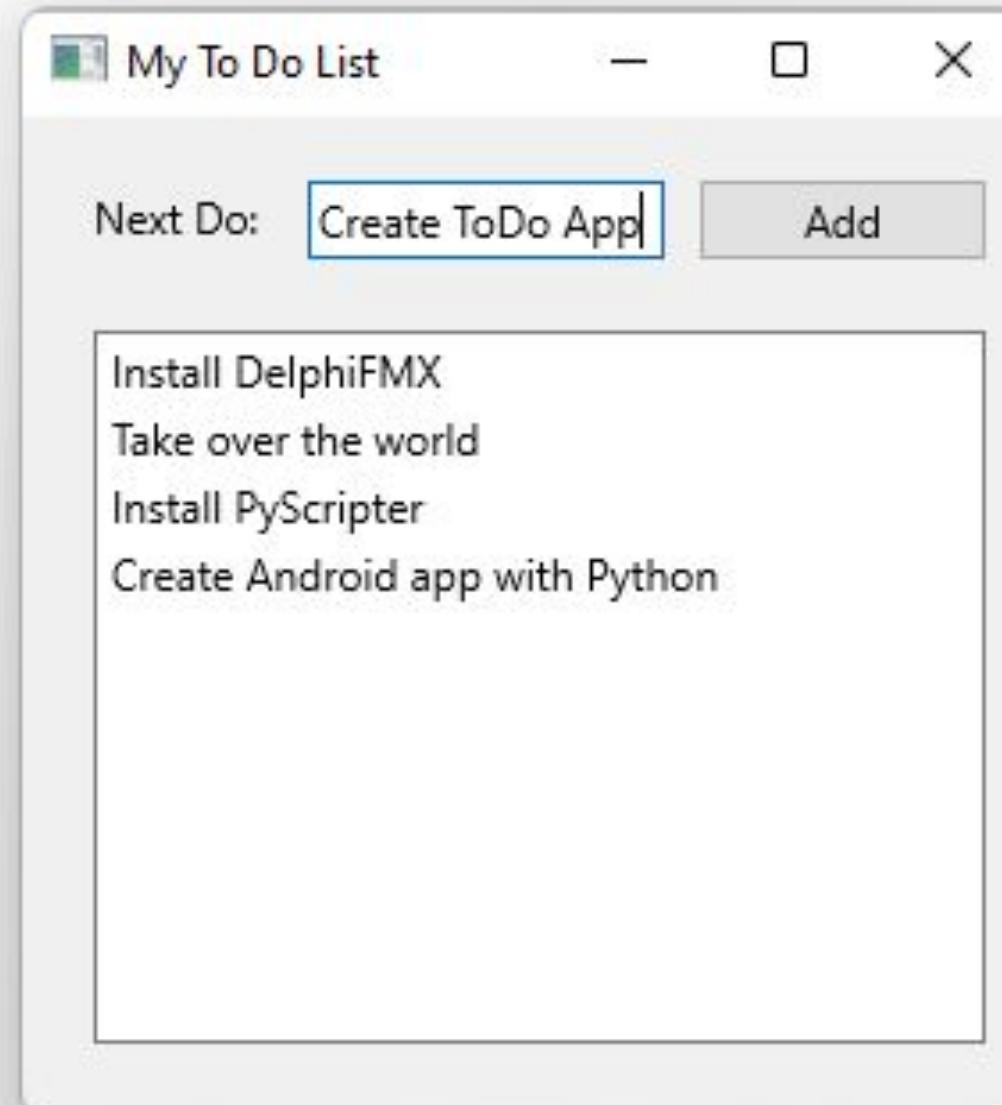
Event handler for button click

Show the form and start the main loop

<https://github.com/Embarcadero/DelphiFMX4Python/blob/main/samples/HelloDelphiFMX.py>

# ToDo App

Starting to get useful....



```
from delphifmx import *
from os.path import exists

class HelloForm(Form):

    def __init__(self, owner):
        self.SetProps(Caption = "My To Do List", OnShow = self.__form_show, OnClose = self.__form_close)

        self.hello = Label(self)
        self.hello.SetProps(Parent = self, Text = "Next Do: ", Position = Position(PointF(20, 20)))

        self.edit = Edit(self)
        self.edit.SetProps(Parent = self, Position = Position(PointF(80,18)))

        self.clickme = Button(self)
        self.clickme.SetProps(Parent = self, Text = "Add",
                             Position = Position(PointF(190, 18)), Width = 80, OnClick = self.__button_click)

        self.list = ListBox(self)
        self.list.SetProps(Parent = self, Position = Position(PointF(20, 60)), Width = 250, OnClick = self.__list_item_click)

    def __list_item_click(self, sender):
        if (self.list.itemindex > -1):
            self.list.items.delete(self.list.itemindex)

    def __form_show(self, sender):
        self.SetProps(Width = 300, Height = 320)
        if exists("todo.txt"):
            self.list.items.loadfromfile("todo.txt")

    def __form_close(self, sender, action):
        self.list.items.savetofile("todo.txt")
        action = "caFree"

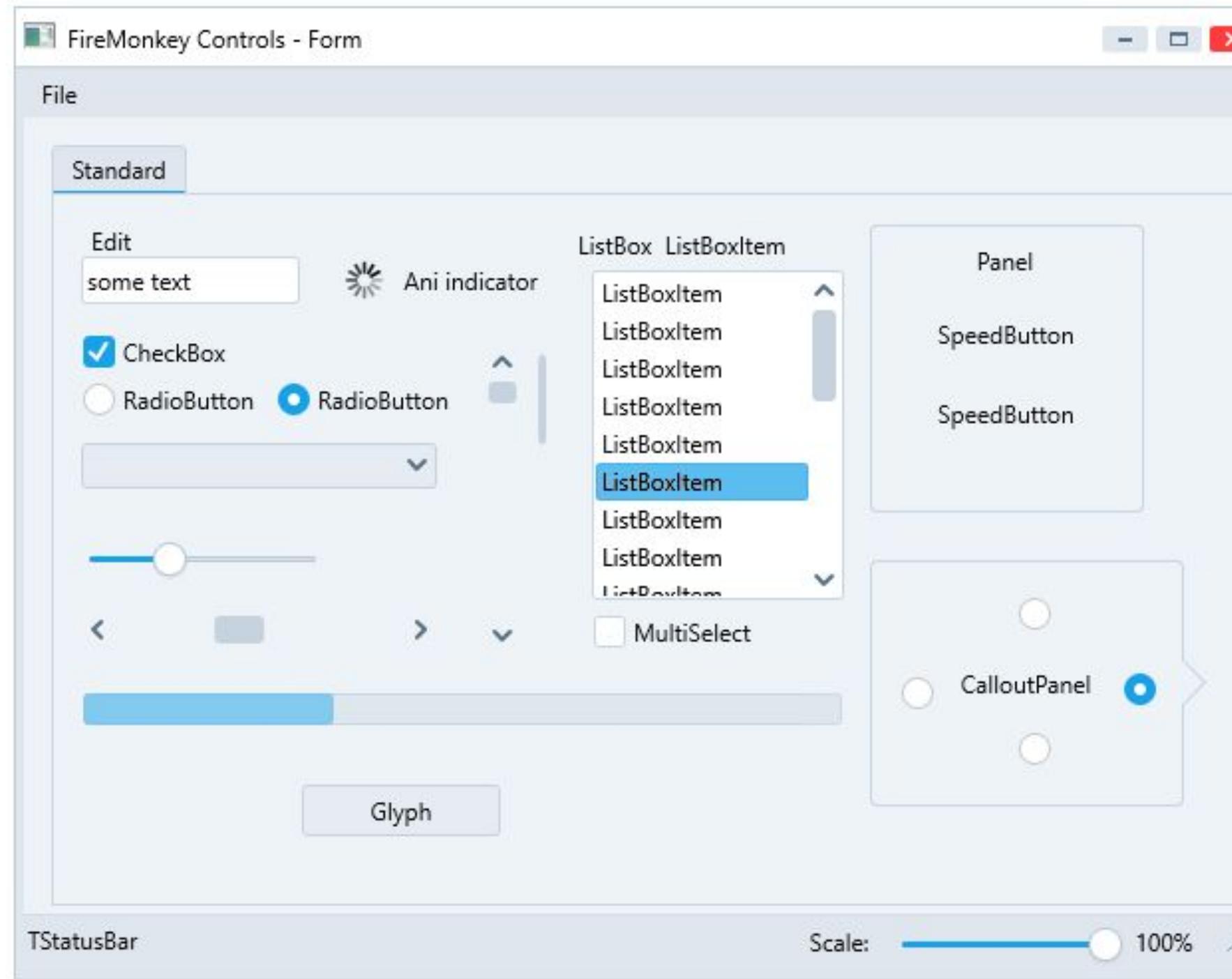
    def __button_click(self, sender):
        self.list.items.add(self.edit.text)
        self.edit.text = ""
```

<https://github.com/Embarcadero/DelphiFMX4Python/blob/main/samples/ToDoList.py>

Left main() off for space.

# With Style

And more controls



Load and apply style

```
StyleManager().SetStyle(  
    StyleStreaming().LoadFromFile("StyleFile.style"))
```

*Embarcadero is working on a  
free style bundle for Python*

<https://github.com/Embarcadero/DelphiFMX4Python/tree/main/samples/ControlsDesktop>

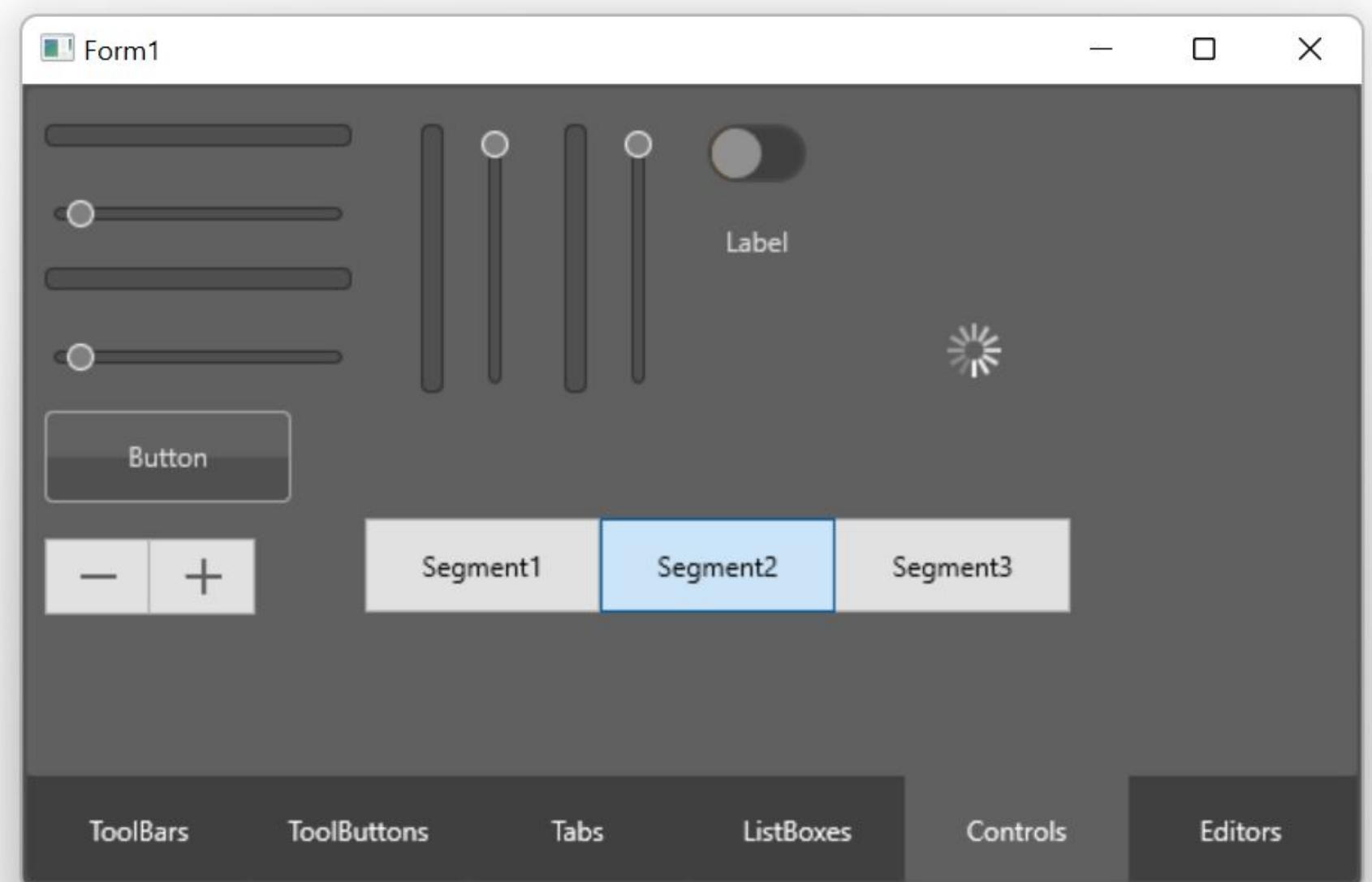
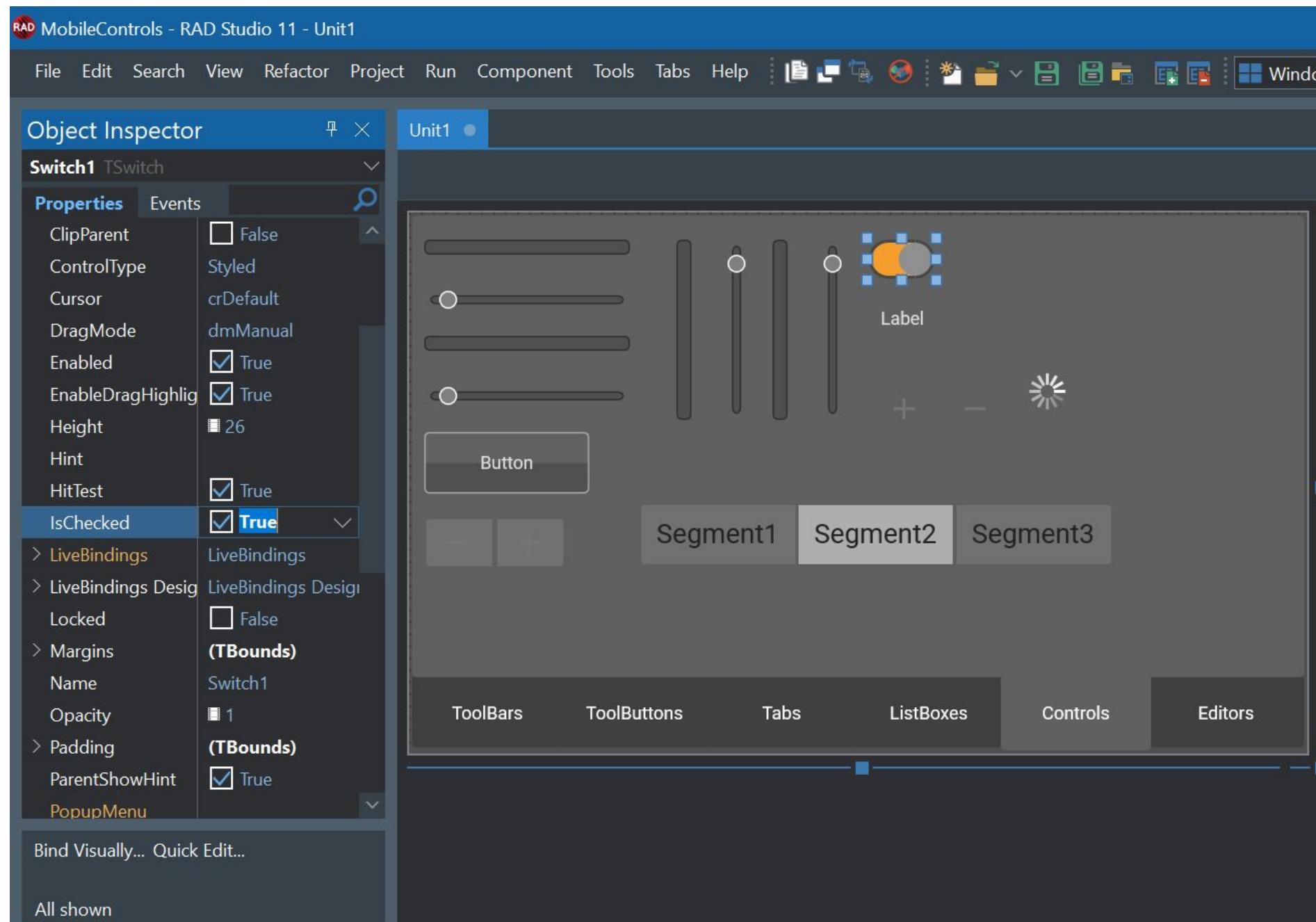


# *Design Your UI* in the Delphi IDE

- Take full advantage of the Delphi IDE designers and property editors
- WYSIWYG preview with *styles*
- Export the form for use in Python then write Python code
- Just right-click and export with the IDE add-in
- Works with any Delphi edition requires no Object Pascal
- [github.com/Embarcadero/DelphiFMX4Python/tree/main/experts](https://github.com/Embarcadero/DelphiFMX4Python/tree/main/experts)
- *Currently uses a binary .pydfm file, but text version coming*

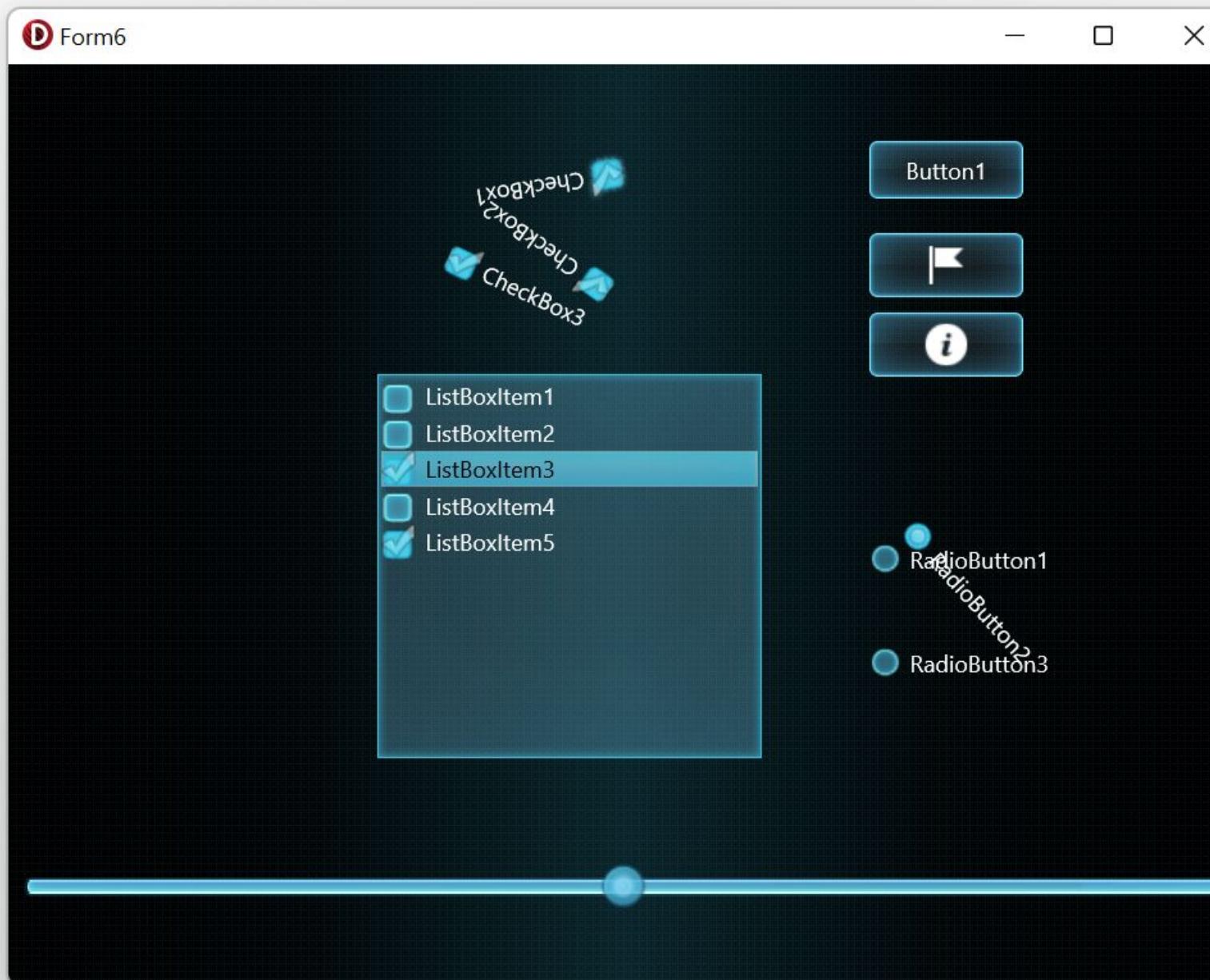


# Exported from Delphi's FMX Designers to Python *with Styles*

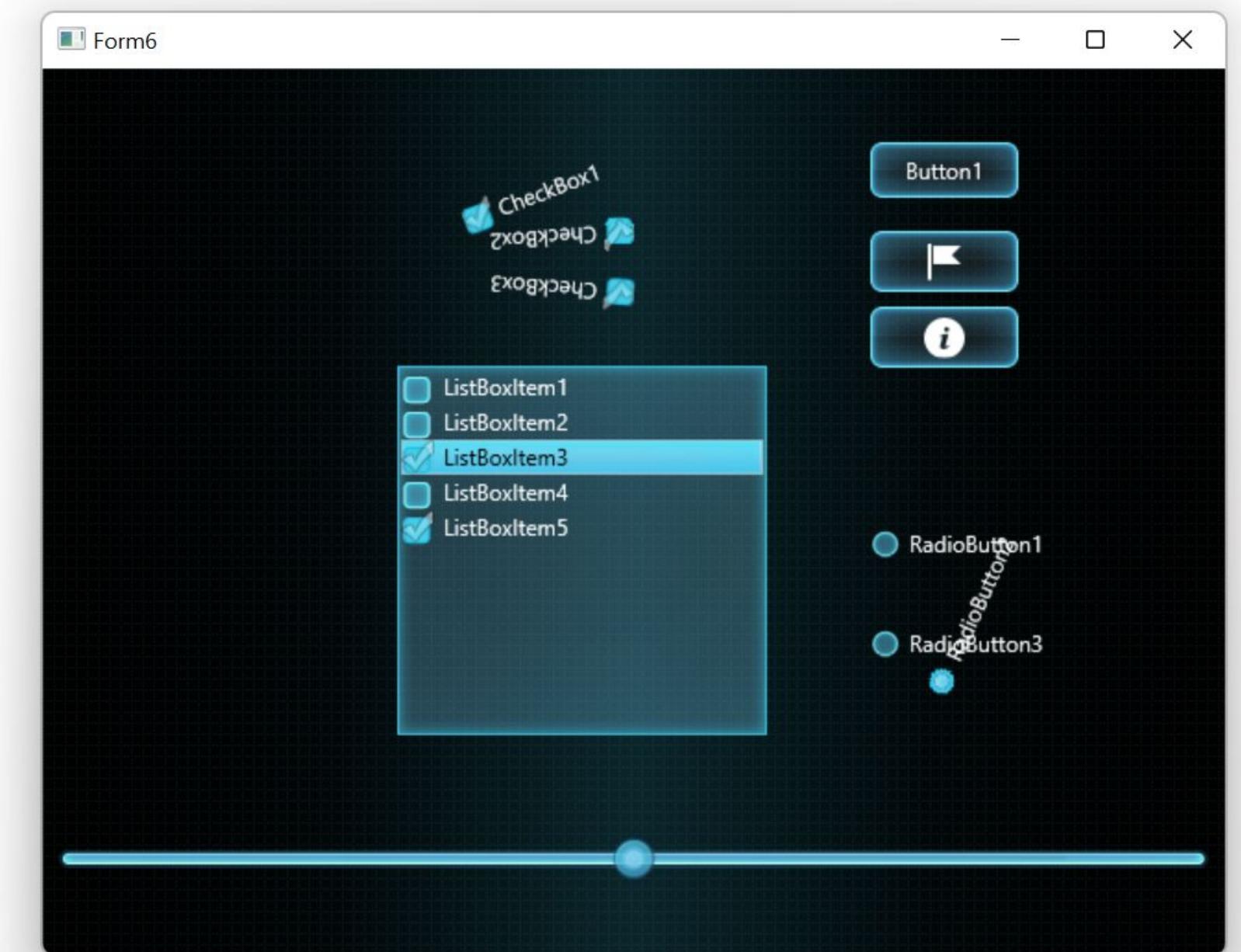


# Exported from Delphi's FMX Designers to Python *with Styles and Animations*

Native Delphi on Windows



Pure Cross-Platform Python App



*The spinning animations are just because I could...*

# Deploying Python to Android

*Yes, Python  
on Android!*



# The Architecture

## Fundamentals

- Delphi FMX supports Android ARM64 as a target
- The Python4Delphi library supports Android
- Allows building Delphi app to run Python on Android

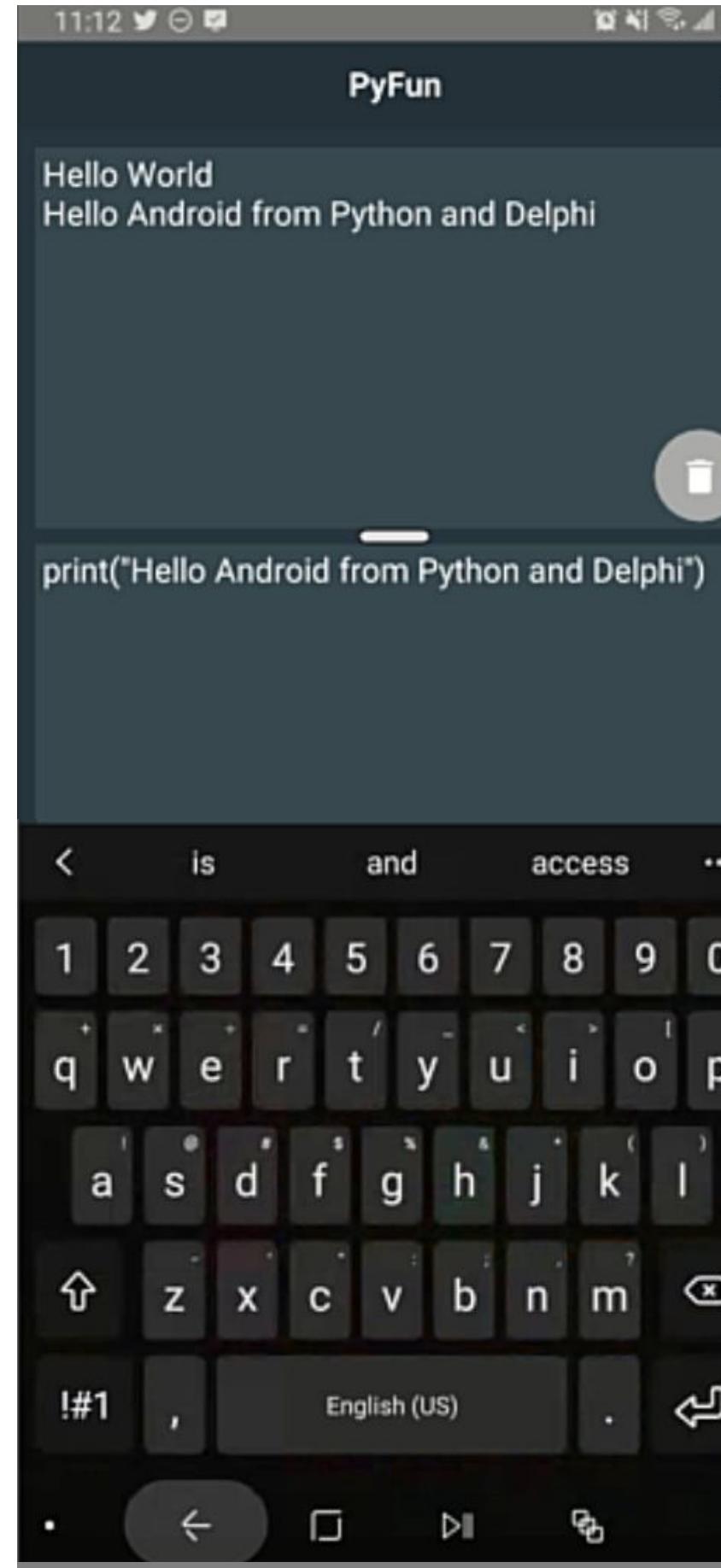
## Solution

- A pre-built Delphi application with Python enabled
- User modifiable Python script payload that runs automatically
- Automated tool to modify the Android Manifest and build final APK
  - On GitHub [github.com/Embarcadero/PythonFMXBuilder](https://github.com/Embarcadero/PythonFMXBuilder) (alpha?)
  - Or resign modified APK manually

## To Do

- Live preview before deploying to Android
- Bundling multiple Python scripts & modules
- Customize icons, splash screens, loading, etc.
- More testing, debugging, and your suggestions....





# Running Python on Android

- Delphi compiles natively for Android
- The Python runtime is embedded into the app
- Python is dynamically interpreted and executed at runtime
- All of Delphi FMX for Python is available
- Supports additional Python modules
- Completely local - no network access necessary



The screenshot shows the PyAppBuilder IDE interface. On the left is a preview window displaying a mobile screen with the title "Hello Android with Python and Delphi FMX". The main area contains the Python code for the application:

```
#-----
# Name: Simplest.py
# Purpose: The simplest demo of Delphi FMX
#
# Author: Jim McKeith
#
# Created: 21/01/2022
# Copyright: (c) Embarcadero Technologies 2022
#-----
from delphifmx import *

MainForm = Form(Application)
MainForm.SetProps(Caption = "Hello World")
msg = Label(MainForm)
msg.SetProps(Parent = MainForm,
             Text = "Hello Android with Python and Delphi FMX",
             Position = Position(PointF(50, 50)),
             Width = 400)
MainForm.Show()
```

Below the code editor, a terminal window displays the command-line output of the build process:

```
ExecCmd: "C:\Users\Public\Documents\Embarcadero\Studio\22.0\CatalogRepository\AndroidSDK-2525-22.0.42600.6491\build-tools\30.0.3\apksigner.bat" sign --ks-key-alias PyApp --ks cert\A...
ExecCmd: C:\Users\Public\Documents\Embarcadero\Studio\22.0\CatalogRepository\AndroidSDK-2525-22.0.42600.6491\platform-tools\Adb.exe -s 99051FFBA00C04 install -r C:\Users\Jim\Do...
Serving...
Performing Incremental Install
Success
Install command complete in 1469 ms

ExecCmd: C:\Users\Public\Documents\Embarcadero\Studio\22.0\CatalogRepository\AndroidSDK-2525-22.0.42600.6491\platform-tools\Adb.exe -s 99051FFBA00C04 shell am start -n com.embarcadero.PyAppTestLive/com.embarcadero.firemonkey.FMXNativeActivity
Starting: Intent { cmp=com.embarcadero.PyAppTestLive/com.embarcadero.firemonkey.FMXNativeActivity }
```

[github.com/Embarcadero/PythonFMXBuilder](https://github.com/Embarcadero/PythonFMXBuilder)

# Going Further...



Exploring the  
Delphi for Python  
Bridge

# Customizing Python on Android

*Take your  
app further  
with Delphi*



# Combine Delphi and Python

- The Python4Delphi library is a bidirectional bridge
- Develop parts of your solution in Delphi, and part in Python
  - Play to the strengths of each
- Merge them together into a single cohesive solution
- Find samples, tutorials, and videos
  - [github.com/pyscripter/python4delphi](https://github.com/pyscripter/python4delphi)



# Use Delphi to Create Native Python Modules

- Many Python modules are written in C/C++ and natively compiled
- Delphi also creates natively compiled Python modules via Python4Delphi
- Prototype rapidly in Python, and then create optimized modules in Delphi to clear bottlenecks
- Much like Python, Delphi code is focused on readability and clear structure and may be easier for you than using C/C++
- Augment your use of PyPy or Cython



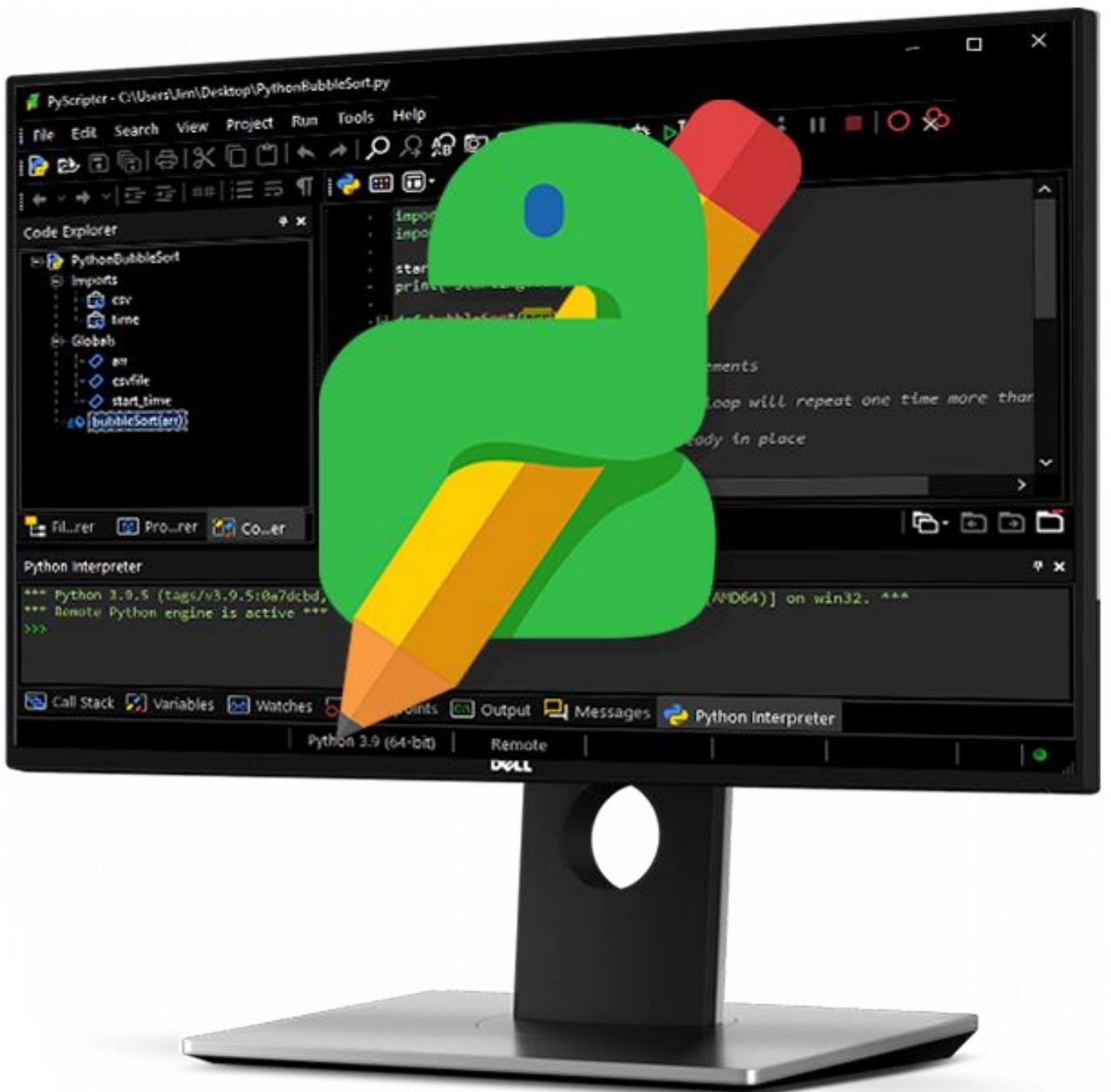
# Documentation and Library References

- Main Delphi documentation
  - Main [docwiki.embarcadero.com/RADStudio/en/](https://docwiki.embarcadero.com/RADStudio/en/)
  - VCL [docwiki.embarcadero.com/RADStudio/en/VCL\\_Overview](https://docwiki.embarcadero.com/RADStudio/en/VCL_Overview)
  - FMX [docwiki.embarcadero.com/RADStudio/en/FireMonkey](https://docwiki.embarcadero.com/RADStudio/en/FireMonkey)
- Library Reference
  - Main [docwiki.embarcadero.com/Libraries/en/](https://docwiki.embarcadero.com/Libraries/en/)
  - VCL [docwiki.embarcadero.com/Libraries/en/Vcl](https://docwiki.embarcadero.com/Libraries/en/Vcl)
  - FMX [docwiki.embarcadero.com/Libraries/en/FMX](https://docwiki.embarcadero.com/Libraries/en/FMX)
- Delphi prefixes type names with a “T”
  - *TEdit* in Delphi is an **Edit** in Python
  - It is just a naming convention



# About PyScripter

- Popular open-source Python IDE sponsored by Embarcadero
- All the features expected in a modern Python IDE while being lightweight and very fast
- Natively compiled for Windows to use minimal memory with maximum performance
- Full Python debugging with remote debugging
- Integration with Python tools like PyLint, TabNanny, Profile, etc.
- Run or debug files from memory.
- [embarcadero.com/free-tools/pyscripter/free-download](https://www.embarcadero.com/free-tools/pyscripter/free-download)



# About UltraEdit

- High performance text editor for programmers.
- Industry's best large file handling: 10+ GB and beyond.
- Syntax highlighting for nearly any language or data format.
- Smart templates.
- Hex editing. Column / block mode editing.
- Part of Idera family of developer tools.
- [ultraedit.com/products/ultraedit/](https://ultraedit.com/products/ultraedit/)

```

File Home Edit Format View Coding Project Layout Window Advanced
index-re.html x info.php x simpleslideview.js x main.css x
0 10 20 30 40 50 60 70
214 *:after,
215 *:first-letter,
216 *:first-line {
217 background: transparent !important;
218 color: #000 !important; /* Black prints faster:
219 http://www.sanbeiji.com/archives/953 */
220 box-shadow: none !important;
221 text-shadow: none !important;
222 }
223
224 a,
225 a:visited {
226 text-decoration: underline;
227 }
228 a[href]:after {
229 content: "(" attr(href) ")";
230 }
231
232

```

**Find and Replace**

Find Replace Find in Files Replace in Files

Find what: <div class="small-12 .? columns">(?=\\s\*<h)

In: Current file Selected text

Next Previous Count all

Match whole word Match case Highlight all items found List lines containing string Bookmark matching lines In column:

Regular expressions: Perl Filter lines Show Hide Close after find

```

361 <li>Customizable, configurable, beautiful themes</li>
362 <li>OS integration (command line, shell extension)</li>
363 <li>Blazing fast find / replace and find in files</li>
364 <li>Fully integrated file compare, instant diff op</li>
365 <li><a href="/products/ultraedit/ultraedit-features"></a></li>
366 </ul>
367 </div>
368 <div class="small-12 medium-6 columns">
369 <h4><i class="fa fa-building-o fa-sm idm-with-text-l</i>
370 <ul class="list-benefits">
371 <li>Secure, trusted, proven, dependable, stable</li>
372 <li>Digitally signed installers and libraries</li>
373 <li>Easy / silent deployment with MSI installers</li>
374 <li>Professional support and in-house call center</li>
375 <li>Multi-platform: Windows, Mac, and Linux</li>
376 <li><a href="/register-buy/corporate-solutions.htm</a></li>
377 </ul>
378 </div>
379 </div>
380
381 <div class="bgpicture"></div>
382
383

```

# Next Steps

- Install Delphi FMX for Python [github.com/Embarcadero/DelphiFMX4Python](https://github.com/Embarcadero/DelphiFMX4Python)
  - See the samples, ***Star the Repository***, file issues, and make feature requests
- Read the blog post (links, replays) [blogs.embarcadero.com/?p=130176](https://blogs.embarcadero.com/?p=130176)
- See part 1 on Delphi VCL [blogs.embarcadero.com/?p=128183](https://blogs.embarcadero.com/?p=128183)
- Start a 30-day Delphi trial [embarcadero.com/products/delphi/start-for-free](https://embarcadero.com/products/delphi/start-for-free)
- Subscribe on YouTube  [youtube.com/c/EmbarcaderoTechnologies](https://youtube.com/c/EmbarcaderoTechnologies)
- Follow us on Twitter  [twitter.com/embarcaderotech](https://twitter.com/embarcaderotech)
- Like us on Facebook  [facebook.com/embarcaderotech](https://facebook.com/embarcaderotech)
- Follow us on LinkedIn  [linkedin.com/company/embarcadero-technologies](https://linkedin.com/company/embarcadero-technologies)
- Read our blog  [pythongui.org](https://pythongui.org)

# Python on Android with Delphi FMX

The Cross Platform GUI Framework  
Part 2

## Q&A



**Jim McKeeth**  
Chief Developer Advocate  
**Embarcadero Technologies**  
jim.mckeeth@embarcadero.com  
@JimMcKeeth

