Interfacce utente (GUI) moderne e accattivanti senza sforzo in Python con “Delphi VCL” e “Delphi FMX” 🐍 🏛 …

Marco Breveglieri

Software Developer | Trainer and Consultant | Tech Content Creator | Community lover

Marco Breveglieri

Software Developer | Trainer and Consultant | Tech Content Creator | Community lover

Developer, Trainer, Consultant

@ABLS Team snc - Reggio Emilia

- Homepage 👉 **https://www.breveglieri.it**
- Blog 👉 **https://www.compilaquindiva.com**
- Twitch 👉 **https://www.twitch.tv/compilaquindiva**
- YouTube 👉 **https://www.youtube.com/@compilaquindiva**
- LinkedIn 👉 **https://www.linkedin.com/in/marcobreveglieri**
- GitHub 👉 **https://github.com/marcobreveglieri**
- Delphi Podcast 👉 **https://www.delphipodcast.com**
- Canale Telegram 👉 **https://t.me/compilaquindiva**

Delphi Podcast

twitch

Prometheus Client for Delphi

MURPHY FOR DELPHI

EMBRACE THE CHAOS

Agenda

- Python ♥ Delphi
- Sviluppo GUI con Python
- Delphi VCL e Delphi FMX
- Demo, demo, altri demo!
- Conclusioni
- Q & A

Una premessa...

Il mio incontro con Python

- Sintassi immediata, chiara e leggibile
- Adatto a una vasta gamma di applicazioni
- Librerie e framework per tutti gli usi
- Lo “standard de facto” per il data science
- Ampiamente utilizzato nell’ambito AI & ML
- Utile per task automatizzati e scripting
- Cross-platform e cross-device

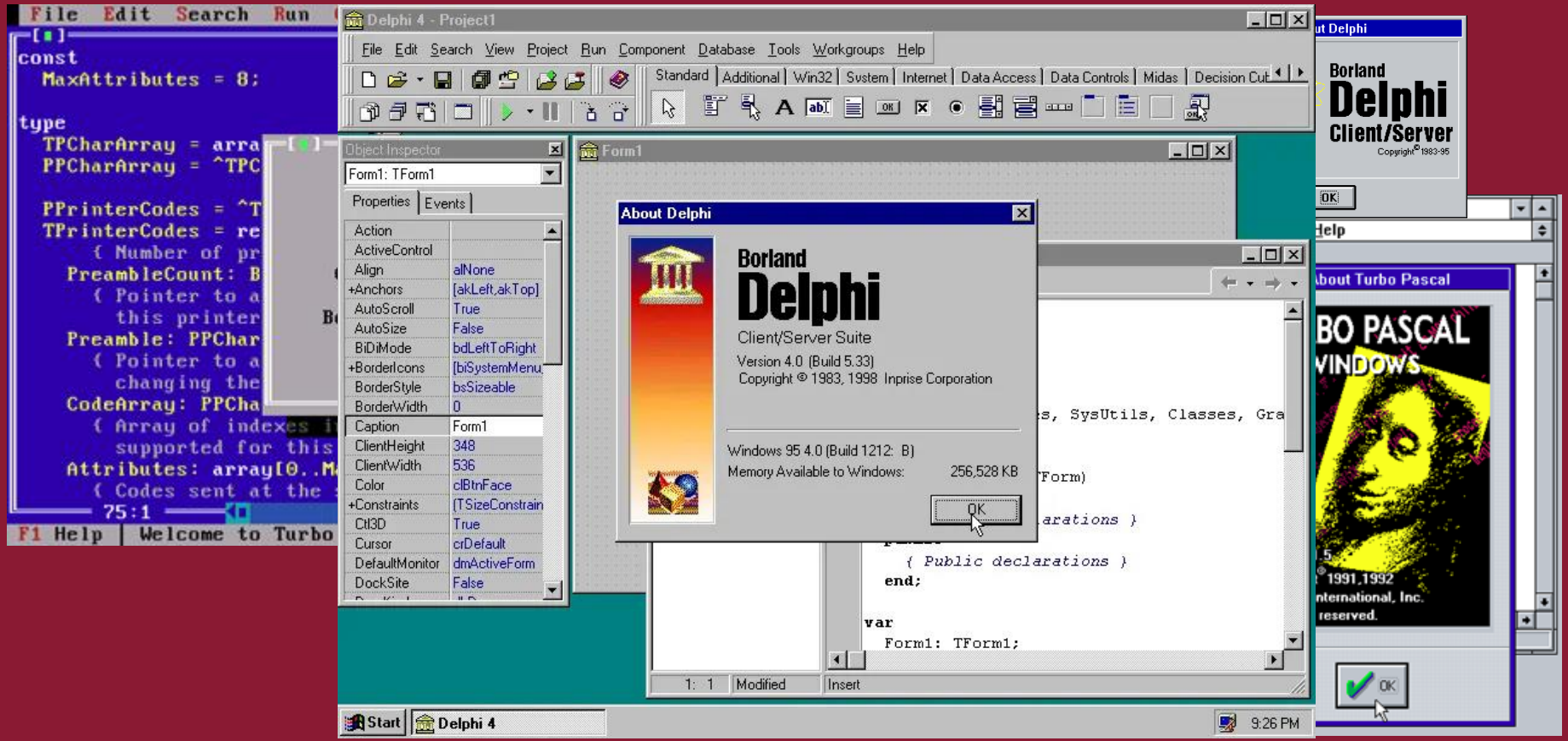
...e me lo chiedevano insistentemente per corsi di formazione.

Sono anche (*) un “Delphi Developer”! 🏛

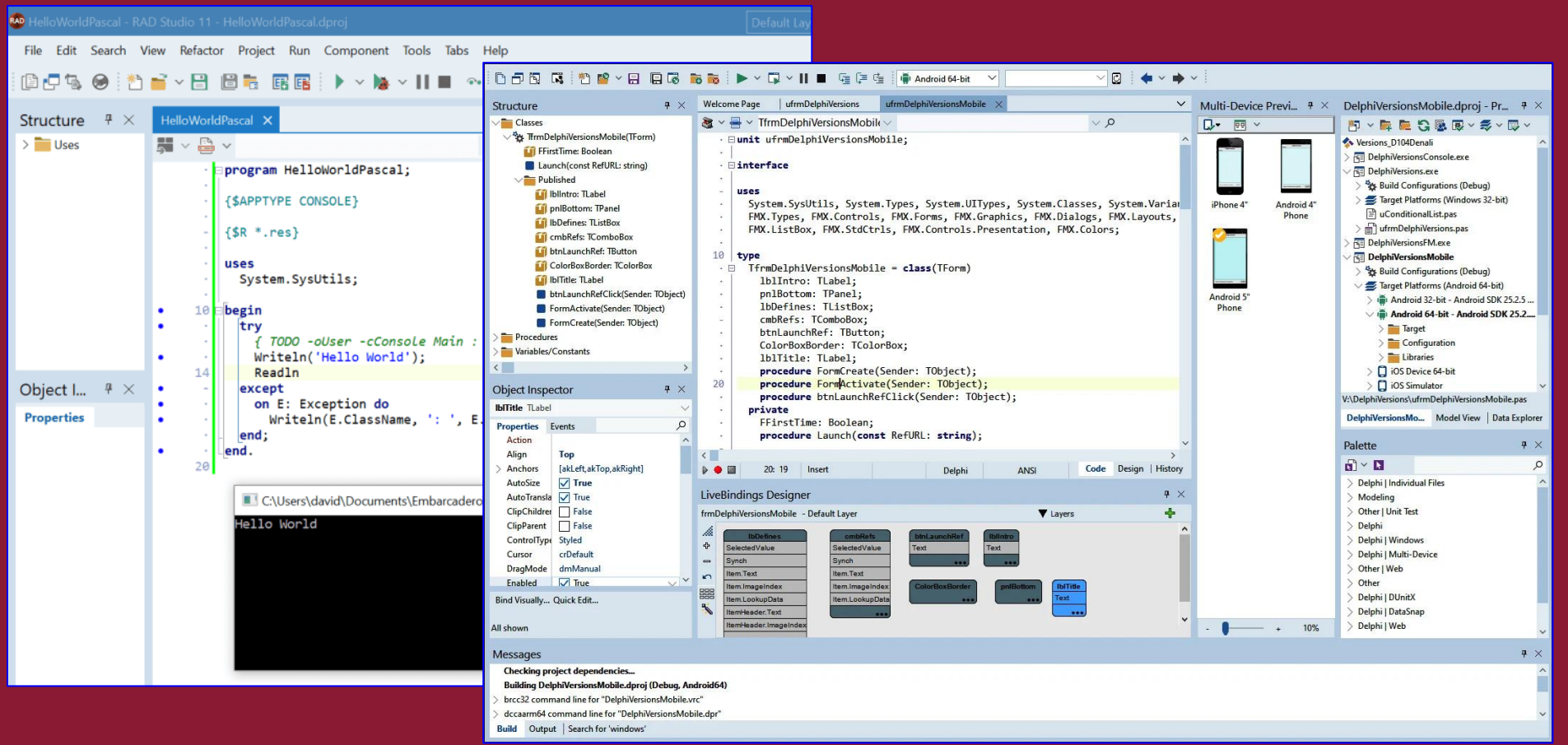
Delphi® è uno dei tool di sviluppo più avanzati al mondo per la realizzazione di applicazioni native multipiattaforma e multidevice ad alte prestazioni, utilizzando strumenti per la progettazione visuale dell’interfaccia utente e funzionalità RAD, unitamente alla scrittura accelerata del codice sorgente usando Pascal, uno dei linguaggi più eleganti e leggibili.

** sviluppo anche in C#, JavaScript e altri linguaggi, ma a Delphi sono affezionato.* 🐶

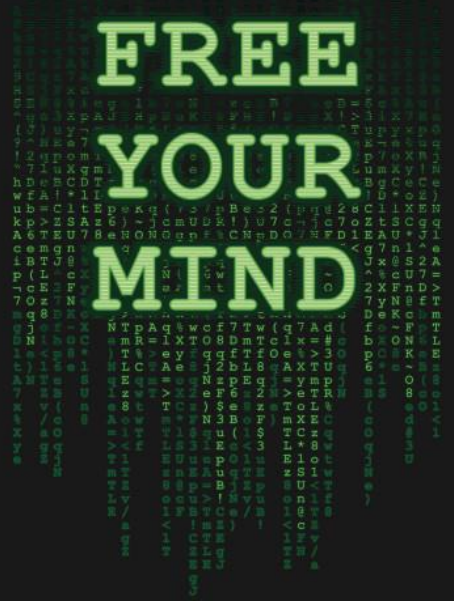
Hai detto Delphi?



Delphi NON è questo...



Questo è 🏛 Delphi!


FREE
YOUR
MIND

Delphi ~~vs~~  Python

Non è una competizione

- Non esiste un tool adatto a tutti gli scopi
- Gli sviluppatori devono avere più tool nella propria cassetta degli attrezzi
- Il gioco è trovare il tool giusto per risolvere uno specifico compito
- Avere tool specifici per ciascuna esigenza è del tutto normale (e pure consigliato)

Delphi 4 Python

- **Moduli free** che rendono disponibili le librerie GUI di Delphi agli sviluppatori Python
  - Mature, ricche di feature, native e cross-platform
  - Non richiedono Delphi per funzionare
- E' basato sul framework open source *Python4Delphi* (lo stesso sfruttato da *PyScripter*)
- Disponibile su GitHub e PyPI
- Due librerie differenti tra cui scegliere
  - **DelphiVCL for Python** (supporta Windows 32/64 bit)
  - **DelphiFMX for Python** (supporta Windows 32 e 64 bit, Mac OS, Android, Linux 64 bit)
- Parte integrante di un **"bridge" bidirezionale** tra Delphi e Python

Timeline for Delphi & Python

Niklaus Wirth

Anders Hejlsberg

Guido van Rossum

1970 — Pascal

Created by Niklaus Wirth and Influenced by ALGOL 68.

1983 — Turbo Pascal

Developed by Anders Hejlsberg and promoted by Borland. The definitive Pascal.

1987 — ABC Programming Language

An imperative general-purpose programming language influenced by ALGOL 68. Guido van Rossum worked on ABC system for several years.

1991 — Python Introduced

Developed by Guido van Rossum as the successor to ABC.

1995 — Delphi Introduced

The successor to Turbo Pascal, includes the VCL. Using an evolved version of Object Pascal.

2000 — Python 2.0

Adding list comprehensions, reference counting, & cycle-detecting garbage collection

2008 — Python 3.0

A Major revision with many improvements.

2011 — Delphi XE adds Mac OS

Introducing FireMonkey, adding Windows 64-bit and Mac OS as natively supported platforms.

2013 — Delphi XE4 & XE5

Compiles to native ARM supporting, iPhone, iPad, and Android.

2017 — Delphi 10.2 *Tokyo*

Support for natively compiled x86 64-bit Linux.

2018 — Guido van Rossum's Permanent Vacation

Stepping down as Python's benevolent dictator for life (BDFL).

Today — Delphi for Python

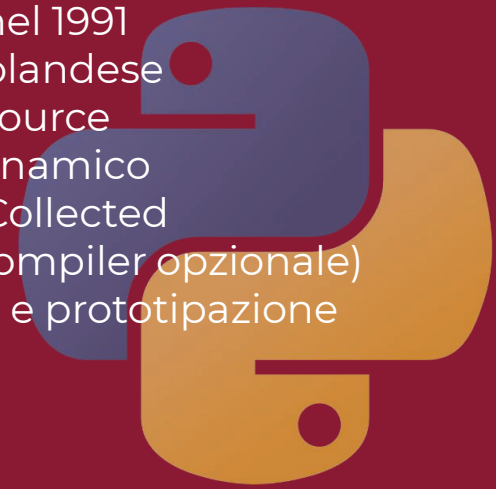
Brings Delphi's VCL & FMX frameworks to Python

Photo of Guido by D. Stroud,et al https://w.wiki/4fSN
Photo for Wirth by Tyomitch https://w.wiki/4fSS
Photo of Hejlsberg by DBegley https://w.wiki/4fST

Comparazione

Presentato nel 1995
Architetto danese
Licenza commerciale (CE free)
Static typing
Memory management manuale
Compilato (interprete opzionale)
Utilizzato in applicazioni business

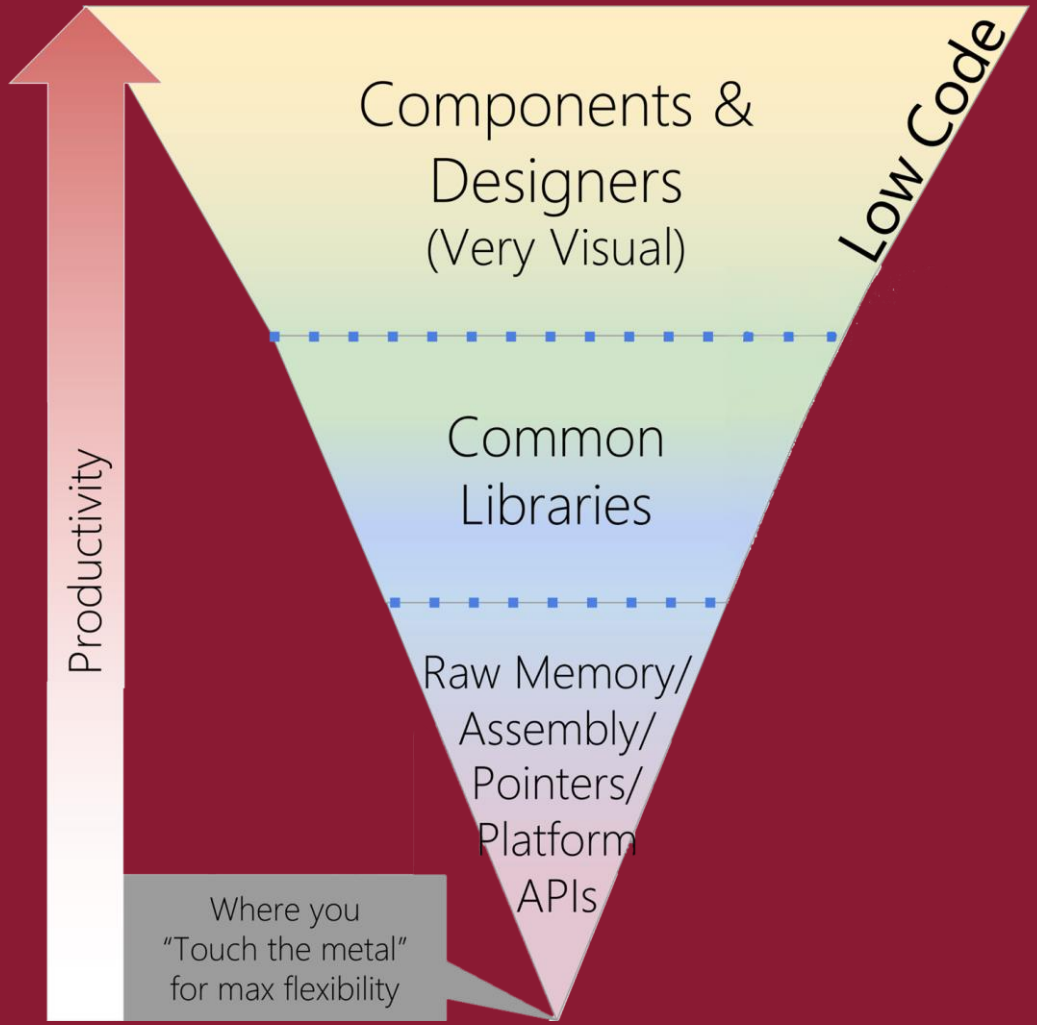
Creato nel 1991
Autore è olandese
Open source
Typing dinamico
Garbage Collected
Interpretato (JIT o compiler opzionale)
Popolare in ricerca e prototipazione



Incoraggiano best practice e leggibilità
Includono librerie e componenti riutilizzabili
Ricco ecosistema di sviluppatori e librerie
Supportano la programmazione strutturata, procedurale, funzionale e OOP
Community di sviluppatori appassionata

Delphi DNA

- **Produttività!** - Persegue l'obiettivo "getting things done quickly"
- **Manutenibilità** - Il codice è facile da leggere e capire, ma ben strutturato
- **App native e compilate** - Le applicazioni native e compilate sono veloci!
- **Accesso ai dati** - Un set completo di componenti per l'accesso ai dati
- **Platform API** - Non occorre chiamare le API di sistema, ma puoi farlo se vuoi
- **Proprietà, metodi ed eventi** - Sviluppo rapido basato su modello a componenti
- **Designer visuali** - Interfaccia drag & drop per la prototipazione e progettazione
- **Affidabilità** - Gestione eccezioni, controllo della memoria, deterministico
- **Compatibilità all'indietro** - Elevata portabilità del codice in versioni aggiornate
- **Ricco ecosistema di componenti** - C'è letteralmente un componente per tutto!


Components &
Designers
(Very Visual)
Low Code
Common
Libraries
Raw Memory/
Assembly/
Pointers/
Platform
APIs
Productivity
Where you
"Touch the metal"
for max flexibility

Zen of Python

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

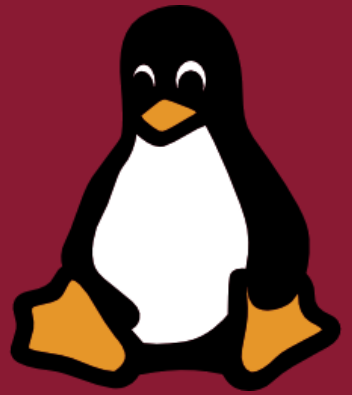
Parliamo di GUI...

Sviluppo app desktop native per Windows

- **Performance**: Windows fornisce accelerazione hardware per controlli nativi
- **Windows Handle**: i controlli nativi per Windows hanno degli handle che forniscono una migliore integrazione con l'OS e con altre applicazioni
- **Consistenza**: l'uso di controlli nativi fornisce aspetto e comportamenti uniformi alle altre applicazioni utilizzate dall'utente
- **Accessibilità**: framework e tool (es. MSAA) consentono l'interfacciamento di screen reader e altri supporti



...e le altre piattaforme?

Python GUI Frameworks

Framework più diffusi

- **Tkinter**: binding per i widget del toolkit Tk
- **PyQt**: binding per la libreria cross-platform Qt

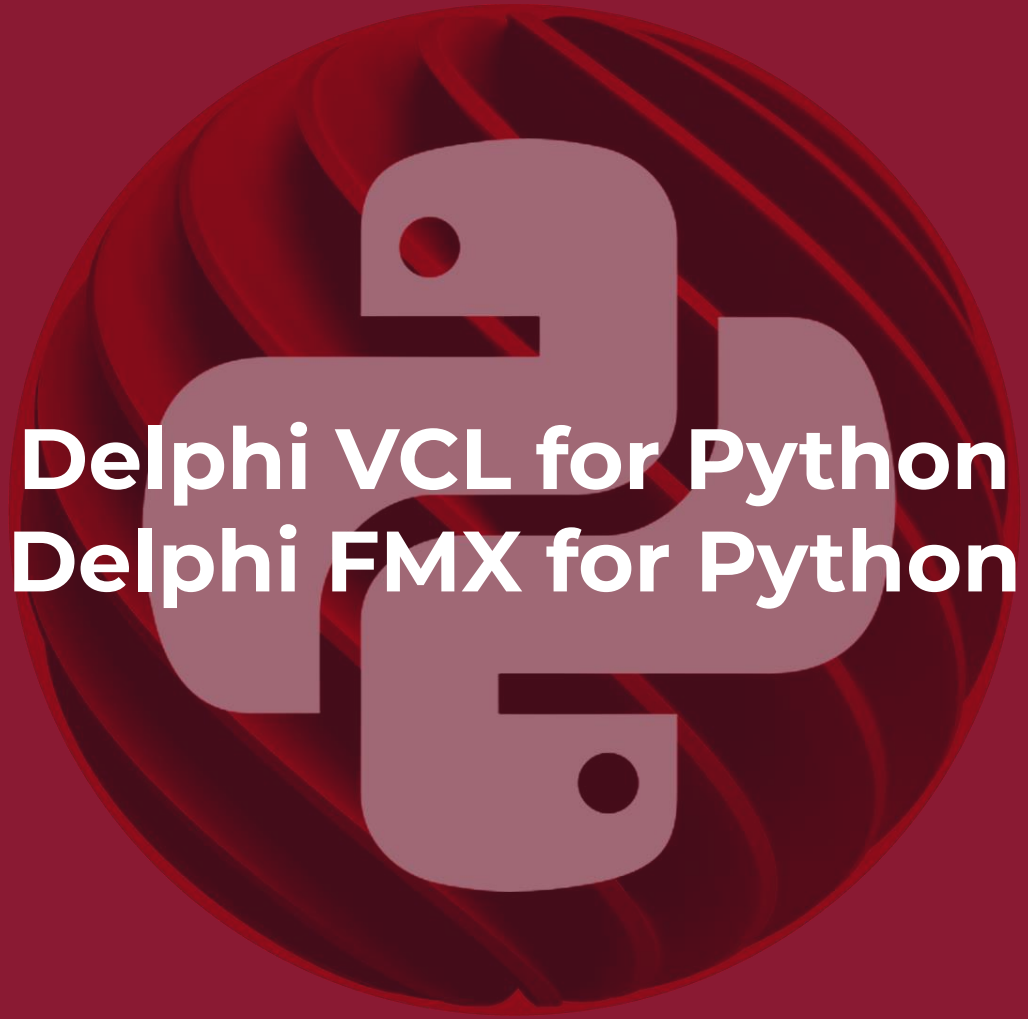
Altri framework

- **PySide**: utilizza anch'esso Qt
- **wxPython**: binding per la libreria wxWidgets
- **PyGTK**: binding per GTK (Gnome)
- **Kivy**: focalizzata su cross-platform e mobile

Nessuna di queste si basa su controlli nativi Windows o su integrazioni specifiche con l'OS.

Alcuni sono troppo "old style", altri poco integrati, altri complessi, altri verbosi, altri incompleti...

...vediamo un'alternativa.



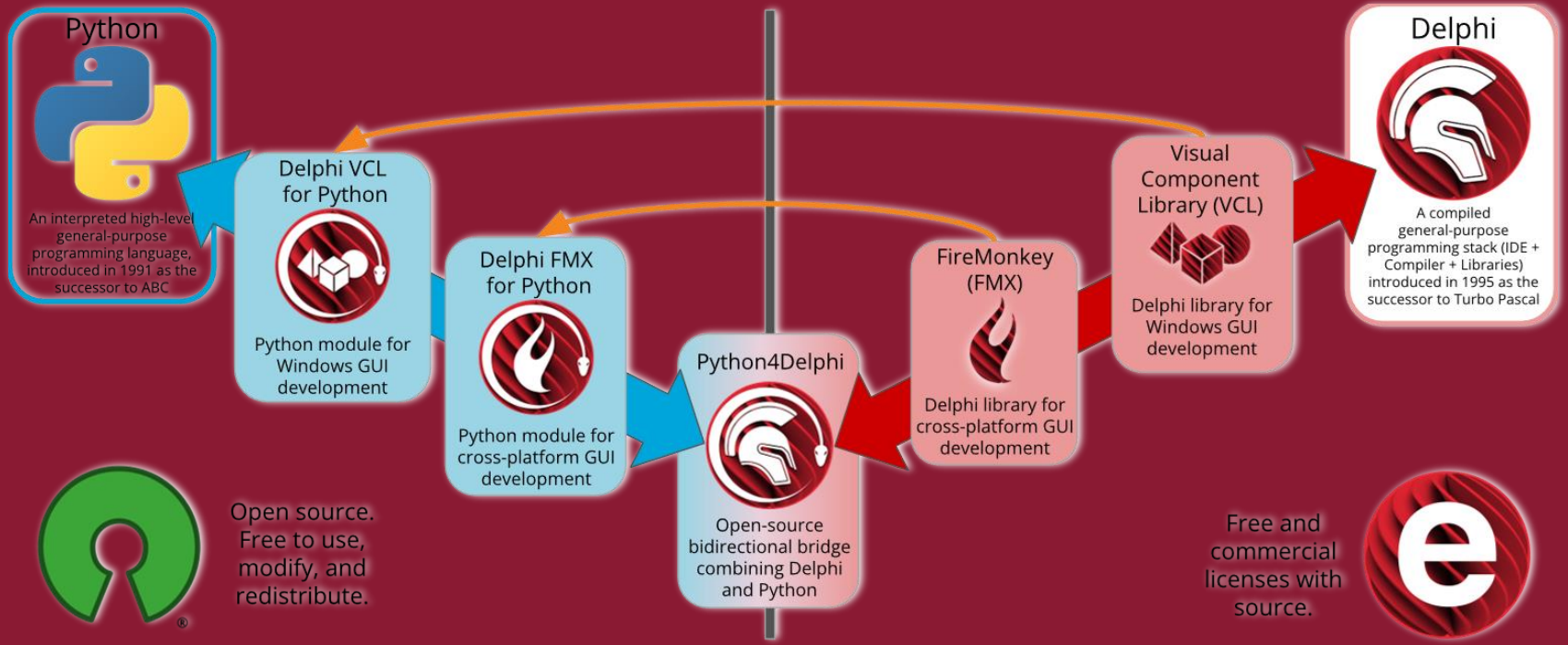
Delphi VCL for Python
Delphi FMX for Python

VCL - Visual Component Library for Windows

- Progettata per Windows e in evoluzione con ogni versione del SO, supportando le funzionalità, i controlli, i tempi e i design più recenti
- Il codice sorgente viene fornito con ogni versione di Delphi (a partire dalla prima del ‘95)
- Principalmente si tratta di un wrapper attorno a widget e controlli della piattaforma nativa
- Include anche molti controlli “disegnati a mano”
- Fornisce un accesso facile agli handle di Windows, ai messaggi del SO, ecc.
- Semplifica notevolmente lo sviluppo su Windows fornendo l’accesso a tutte le funzionalità e API della piattaforma
- Aspetto nativo di Windows con “look & feel” integrato e sistema di temi
- Piattaforma matura con un ricco ecosistema di componenti di terze parti

FMX - Crossplatform FireMonkey Framework

- Sfrutta le librerie GPU per fornire una interfaccia utente ricca con accelerazione hardware che si presenta veloce e dall'aspetto ottimale su tutte le piattaforme
  - Supporta i sistemi operativi Windows, Mac OS, iOS, Android, Linux
  - Utilizza DirectX su Windows, OpenGL su Linux, OpenGL-ES su Android e Metal su iOS e Mac OS
  - L'ultima versione di Delphi si integra con Skia (by Google) e supporta Vulkan
- E' simile alla VCL (con alcuni componenti in comune con essa) ma non è studiata per essere compatibile
  - E' stata progettata da zero come piattaforma universale
- Supporta effetti e animazioni e si appoggia a un sistema di stili gestiti tramite GPU
- Grazie ai "Platform Services", astrae l'accesso all'hardware e alle funzionalità della piattaforma adattando UI e UX alle specificità del device e del sistema operativo
- Anch'essa ricca di componenti (in parte condivisi con la VCL) e controlli visuali molto flessibili (e soprattutto nidificabili: fai di più con meno componenti)



Technology Stack

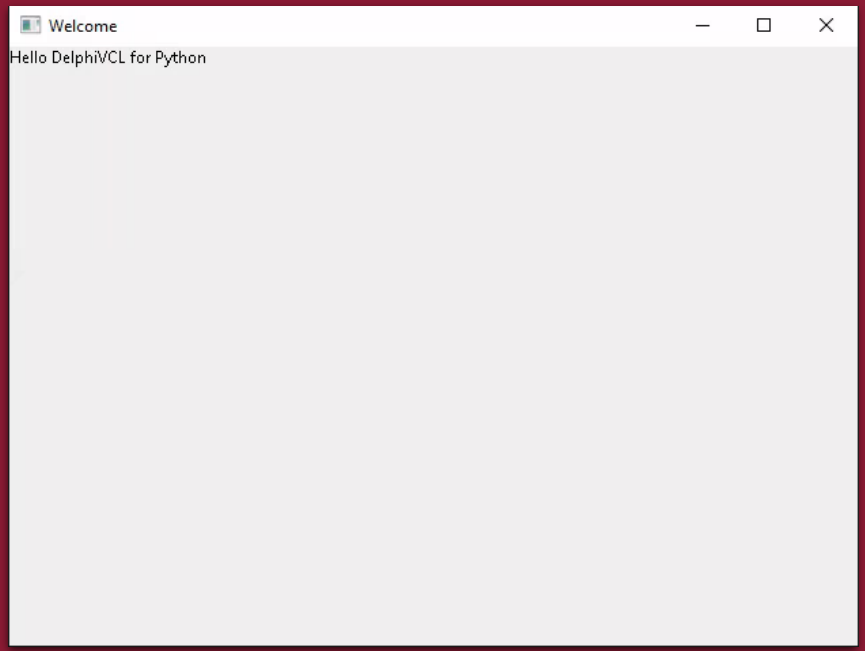
Usiamo PyScripter

- IDE per Python gratuito e open source, sponsorizzato da Embarcadero
- Tutte le funzionalità previste in un ambiente Python, pur rimanendo leggero e molto veloce
- Compilato nativamente per Windows, con uso minimo della memoria e prestazioni massime
- Debug di Python locale e remoto
- Integrazione con strumenti di terze parti come *PyLint*, *TabNanny*, *Profile* e tanti altri...

👉 **https://embarcadero.com/free-tools/pyscripter/free-download**

Demo Time! ⏰

Hello World!




```python
from delphivcl import *


class MainForm(Form):

    def __init__(self, owner):
        self.Caption = "A VCL Form..."
        self.SetBounds(10, 10, 500, 400)
        self.Position = "poScreenCenter"
        self.OnClose = self.__on_form_close

        self.lblHello = Label(self)
        self.lblHello.SetProps(Parent=self, \
            Caption="Hello DelphiVCL for Python")
        self.lblHello.SetBounds(10, 10, 300, 24)


    def __on_form_close(self, sender, action):
        action.Value = caFree


def main():
    Application.Initialize()
    Application.Title = "Hello Python"
    Main = MainForm(Application)
    Main.Show()
    FreeConsole()
    Application.Run()
    Main.Destroy()


main()
```

Class is a VCL Form

Owner

Configure the form

Create and configure the label

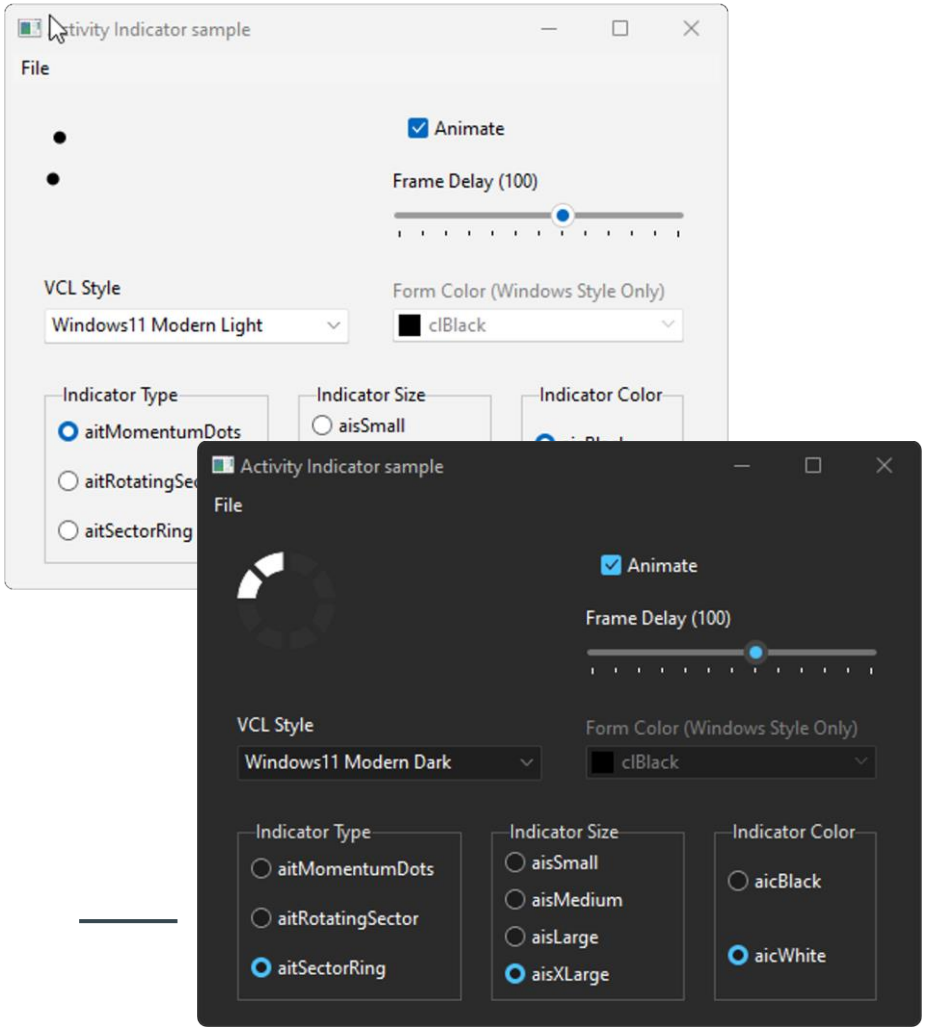
Event handler for the form's close event

Initialization the application and creating the form.

The program loop

Più demo!

Temi visuali
Controlli grafici
Funzionalità



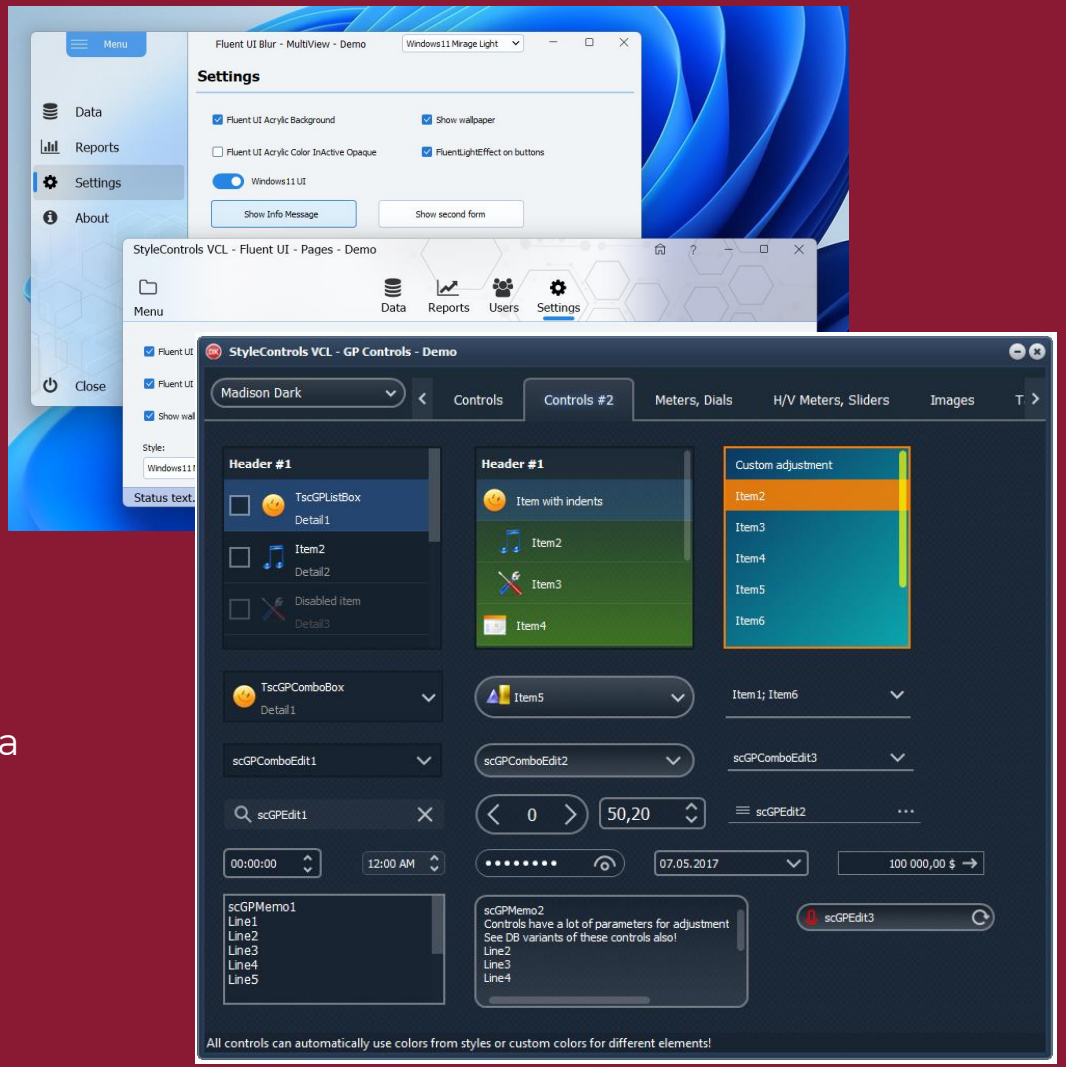
Python on Android!

- Delphi compila nativamente per la piattaforma Android
- Il runtime di Python è incorporato (embedded) nell'app
- Il codice Python è interpretato ed eseguito a runtime
- Tutta la libreria Delphi FMX è disponibile per l'uso
- Completamente in locale! (accesso a rete non richiesto)

Python *powered by* 🏛 Delphi !

Disegna la tua UI

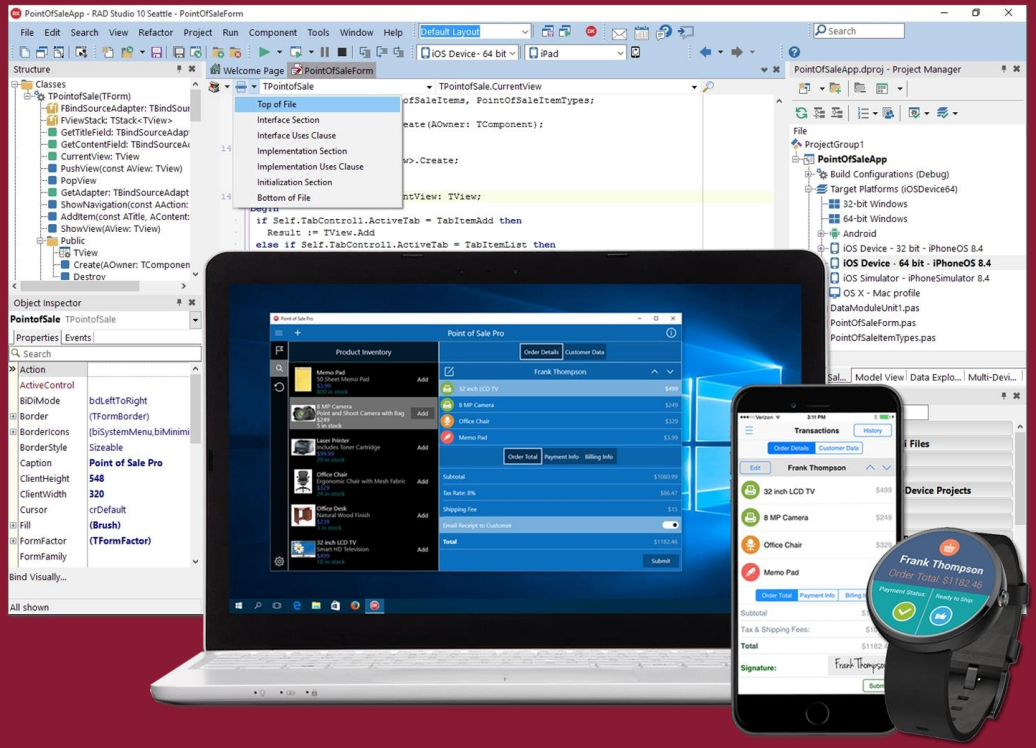
- Usa i designer e gli editor visuali di Delphi per progettare la tua UI
- Crea la tua interfaccia usando un approccio WYSIWYG
- Esporta le finestre (Form) per utilizzarle direttamente in Python senza scrivere (altro) codice
- Non richiede conoscenza del linguaggio Object Pascal
- Continua a scrivere il codice della tua applicazione in Python
- L'esportazione è possibile con un semplice expert e un solo clic



Combina Delphi e Python! 🤝

- Python 4 Delphi è una tecnologia “bidirezionale”
- Sviluppa parte del tuo progetto con Delphi e parte in Python
  - Sfrutta ciascun linguaggio al meglio del suo potenziale
- Fondi le due parti in una singola soluzione coesa e robusta
- Crea prototipi rapidamente in Python e ottimizza moduli con Delphi eliminando i colli di bottiglia

- Sfrutta Delphi per creare moduli nativi in Python
  - Molti moduli Python sono scritti in C/C++ e compilati nativamente
  - Il codice Delphi - come quello Python - è votato a leggibilità e chiarezza, e può risultare più chiaro di C/C++ (ma con gli stessi benefit!)
- Affianca l’utilizzo di altri tool (come PyPy e Cython)



Esploriamo Delphi 🏛 in 5 minuti!

Q & A

Libri e guide

Delphi Succinctly (e-book free, lingua inglese, 100 pagine)
https://www.syncfusion.com/succinctly-free-ebooks/delphi

Risorse e approfondimenti

- **Python GUI** - https://pythongui.org
- **Python 4 Delphi** - https://github.com/embarcadero/python4delphi
- **Delphi VCL 4 Python** - https://github.com/Embarcadero/DelphiVCL4Python
- **Delphi FMX 4 Python** - https://github.com/Embarcadero/DelphiFMX4Python
- **PyScripter IDE** - https://github.com/pyscripter
- **Delphi Homepage** - https://delphi.embarcadero.com
- **Learn Delphi** - https://learndelphi.org
- **Cool Apps Delphi showcase** - https://blogs.embarcadero.com/?s=cool+app
- **Delphi Club Italia** - https://www.facebook.com/groups/delphiclubitalia

Grazie!