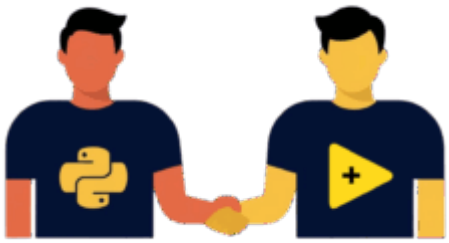
NIDays 2025: IL futuro di LabVIEW e l’amicizia con Python

Category: LabVIEW Talk
scritto da Lorenzo Borghi | Febbraio 19, 2025



Il 18 febbraio 2025, a Milano, presso il Centro Congressi dell’NH Hotel, si sono svolti gli **NI Days**, un evento di rilevanza internazionale organizzato da **National Instruments**.

Questo incontro, che ha visto la partecipazione di professionisti del settore, partner aziendali e appassionati, ha offerto un’importante occasione di aggiornamento sulle ultime tendenze e tecnologie nel campo delle misurazioni, dell’automazione e della progettazione di sistemi.

Un tema centrale, emerso già dalla riunione esclusiva per i partner, ha riguardato la crescente **integrazione tra LabVIEW e Python**.

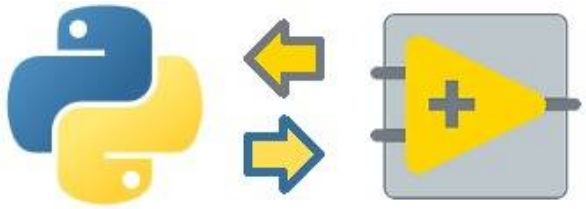
Questo argomento ha suscitato ampie discussioni tra i partecipanti, soprattutto in relazione alla scelta strategica di **National Instruments** di valutare l’integrazione di Python, uno dei linguaggi di programmazione più diffusi e utilizzati negli ultimi anni, al suo software di punta, **LabVIEW**, tradizionalmente utilizzato per la progettazione di sistemi di misurazione e controllo.

Il dibattito ha posto in evidenza le potenzialità e le sfide dell’adottare Python come parte integrante dell’ecosistema **LabVIEW**, esplorando aspetti legati all’efficienza, alla flessibilità e alla compatibilità tra i due strumenti.

L’integrazione di Python in LabVIEW: La Visione di NI dal 2018

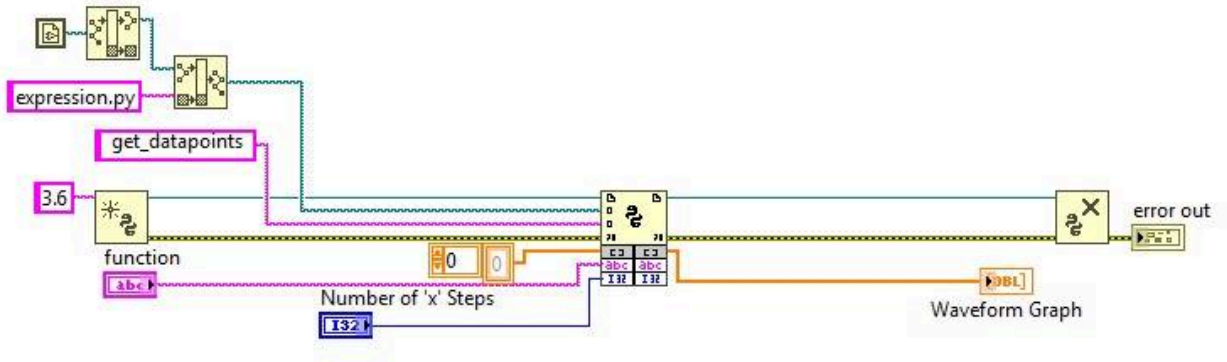
Già nel 2018, **National Instruments** aveva previsto l’espansione del linguaggio di programmazione **Python** nel campo delle applicazioni di misurazione e automazione.

Per rispondere a questa tendenza crescente, NI ha sviluppato una **libreria Python** che consente di integrare Python all’interno di **LabVIEW**.

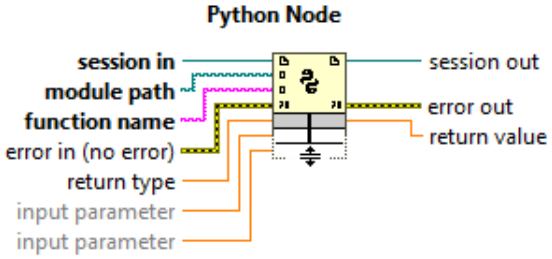


Questa integrazione avviene attraverso una catena di operazioni che include tre componenti principali:

- **Close Python Session**: Termina la sessione Python, liberando le risorse utilizzate.
- **Python Node**: Consente l'esecuzione di script Python direttamente all'interno di un diagramma a blocchi di LabVIEW.
- **Open Python Session**: Inizializza la sessione Python all'interno di LabVIEW.



Questo permette di eseguire script Python presenti nel PC, specificando il loro percorso all'interno del **Python Node**. Inoltre, è possibile ricevere in uscita i parametri passati nel terminale, utilizzando il tipo di ritorno definito.

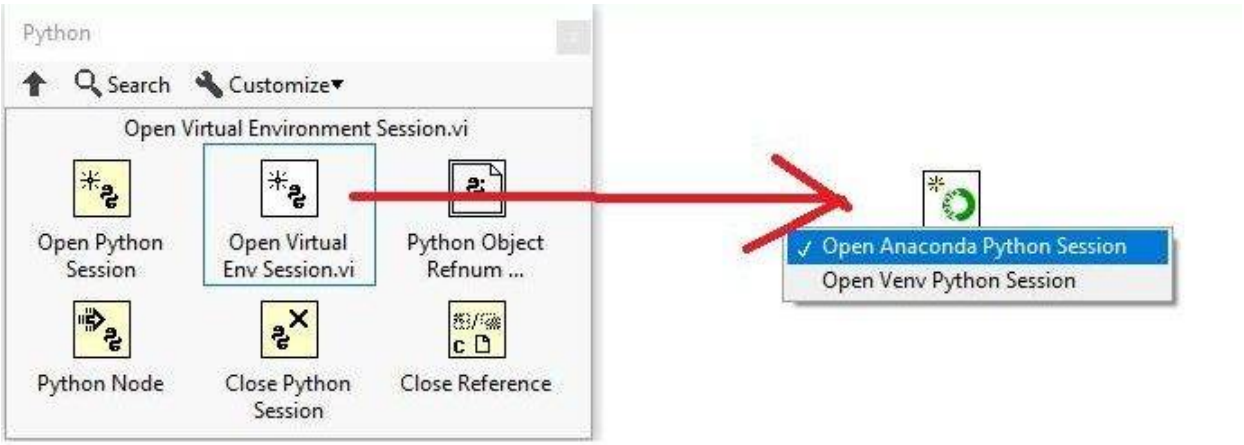


Con il rilascio di LabVIEW 2023 Q1, NI ha esteso il supporto agli ambienti virtuali Python, inclusi **Anaconda** e **venv**.

Questa integrazione permette agli sviluppatori di gestire facilmente le dipendenze e le versioni dei pacchetti Python, garantendo un ambiente di sviluppo più controllato e replicabile.

Per versioni precedenti, potresti trovare interessante il nostro articolo [Python Virtual Environment per LabVIEW](#).

L'uso di ambienti virtuali facilita inoltre la collaborazione tra team e la distribuzione di applicazioni, assicurando che le configurazioni Python siano coerenti su diverse macchine e installazioni.



Nonostante LabVIEW domini il settore dell'acquisizione dati grazie alla sua interfaccia grafica intuitiva e all'integrazione diretta con hardware di misura, negli ultimi anni ha dovuto confrontarsi con la crescente diffusione di **Python**.

Se da un lato LabVIEW eccelle nella gestione del flusso di dati provenienti da sensori e strumenti di misura, dall'altro Python si è imposto come lo standard per l'analisi avanzata, grazie alla sua vasta gamma di librerie per il machine learning, l'elaborazione statistica e la visualizzazione dei dati.

Questo ha portato molti sviluppatori e ricercatori a integrare entrambi gli strumenti, sfruttando la precisione dell'acquisizione di LabVIEW e la potenza analitica di Python.

Un Approccio Ibrido: LabVIEW per l'Acquisizione, Python per l'Elaborazione

Perché Utilizzare LabVIEW per l'Acquisizione Dati?

LabVIEW è lo standard per:

- **Gestione Hardware**: Integra in modo nativo strumenti di misura (oscilloscopi, sensori, attuatori) e sistemi di automazione.
- **Applicazioni in Tempo Reale**: Offre capacità real-time e deterministiche, fondamentali in settori quali controllo qualità, test industriali e ricerca scientifica.
- **Affidabilità**: Riduce il rischio di perdita di dati o errori di sincronizzazione grazie al supporto per moduli FPGA e sistemi embedded.
- Gestione implicita del Multithreading, Sincronizzazione con Code e Notifier, una migliore gestione e performance sulla programmazione asincrona.

- Interfacce **WYSIWYG** (What You See Is What You Get)
- Programmazione Grafica, intuitiva, AutoDocumentata
- Tempi di prototipazione e sviluppo più rapidi
- Curva di apprendimento più veloce

Queste caratteristiche rendono LabVIEW la scelta migliore per acquisire dati con alta precisione e affidabilità.

Python? Complementare al LabVIEW per quello che LabVIEW non fa.

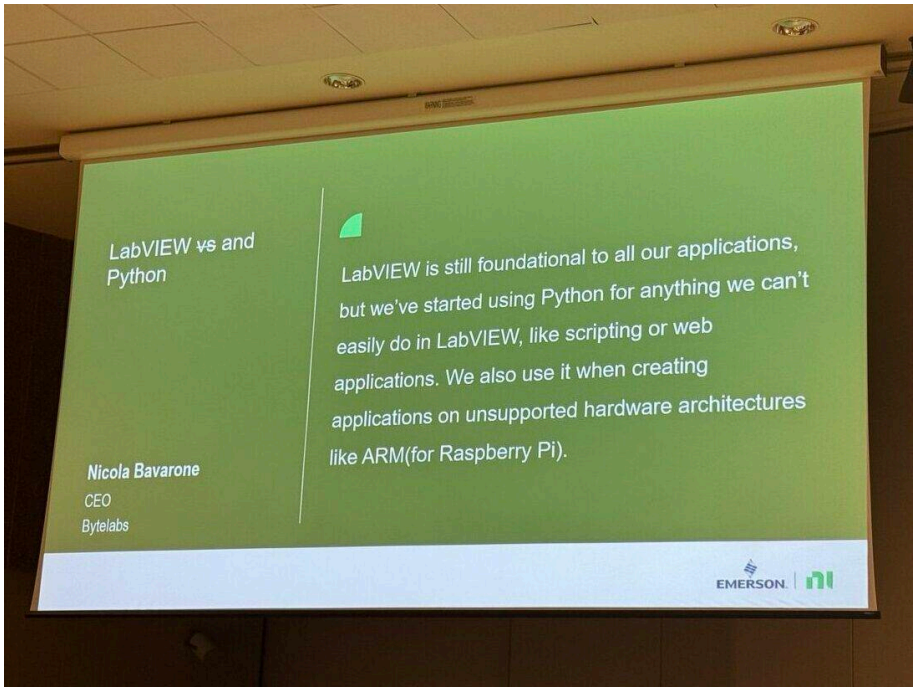
Python può essere considerato come un linguiaggio general purpose, una sua grande applicabilità è l'analisi dati grazie a:

- **Librerie Avanzate**: Strumenti come **NumPy, Pandas, SciPy, Matplotlib, Seaborn, TensorFlow** e **scikit-learn** permettono di effettuare calcoli scientifici, analisi statistiche, machine learning e visualizzazione dei dati.
- **Scalabilità e Automazione**: La capacità di gestire grandi dataset e distribuire il carico computazionale su più core o GPU.
- **Aggiornamenti Costanti e Comunità Attiva**: Python beneficia di una comunità di sviluppatori in continua crescita, che garantisce aggiornamenti e nuove funzionalità per rimanere sempre all'avanguardia.

Grazie a queste potenzialità, Python è perfetto per trasformare i dati acquisiti in informazioni utili e approfondite.

Ancora può essere un valido complemento a LabVIEW per lavorare con database, applicazioni web e altro ancora, dove LabVIEW non è e speriamo non cerchi di andare.

Il nostro punto di vista: LabVIEW con Python (quando serve)



In Bytelabs siamo entusiasti di LabVIEW e rimane per noi lo strumento principale per garantire Software e sistemi di Test e Misura affidabili, con un elevato rapporto Qualità/Costo, riadattabili dai clienti che grazie alla rapidità di apprendimento e all'intuitività del codice, possono sentirsi parte del sistema, lavorando sul sorgente e collaborando a migliorare soluzioni, potendo esprimere il proprio KnowHow.

Ovviamente abbiamo da subito strizzato l'occhio a Python, in primis per realizzare connettività con il cloud, interoperabilità tra i sistemi e in seguito aprendo a tutto quello che è il tema Machine Learning e Intelligenza Artificiale, ma anche più semplicemente alla già citata elaborazione di dati, sfruttandone le librerie e la maggiore quantità di risorse comunitarie.

Noi speriamo quindi che LabVIEW continui ad evolvere a supporto dell' Hardware e della sua enorme capacità, magari portando supporti vettoriali ai controlli per intefacce più performanti, nuovi controlli per sinottici di impianti, sempre maggiore connettività ad Hardware e supporto di protocolli, non vogliamo un labVIEW universale, in un mondo di sistemi sempre più complessi, sappiamo che è necessario implementare tecnologie complementari come in questo caso sta diventando Python.