



## **LabVIEW and Python**

BudLUG Santa - 2023, László Balogh

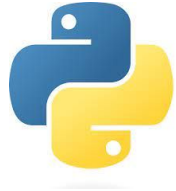
[laszlo.balogh@prodsp.hu](mailto:laszlo.balogh@prodsp.hu)

# Content

- Why?
- How?
  - Embedding Labview
  - Processing TDMS File
  - Labview Node
  - Python via REST API
  - Scripting

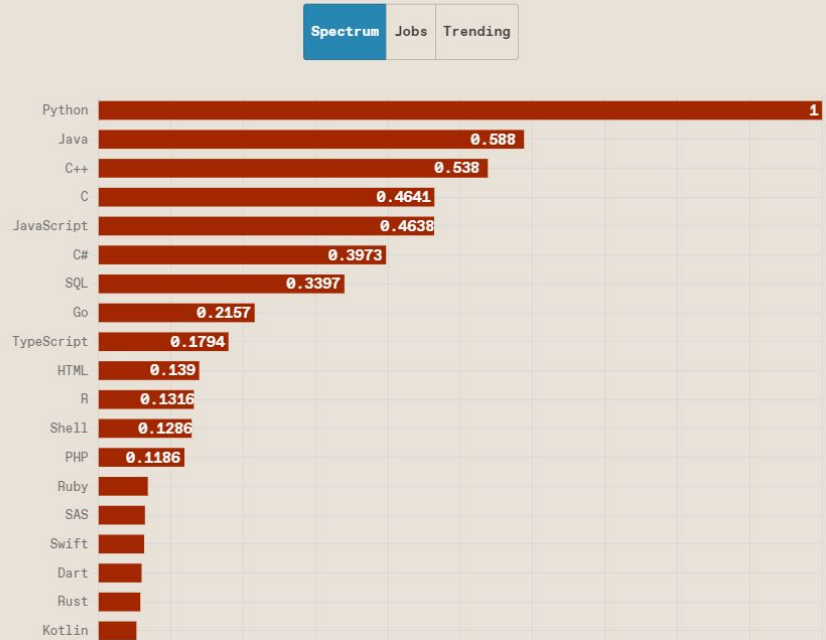
# Why Python?

## The Top Programming Languages 2023 > Python and SQL are on top, but old languages shouldn't be forgotten



### Top Programming Languages 2023

Click a button to see a differently weighted ranking



Source:

[IEEE Spectrum](#)

## Pros

- Easy to learn
- Easy to read
- Text based
- All the platforms are supported (Win, Linux, Mac, etc.)
- Rich set of functionality

## Cons

- Indentation
- Not type safe
- Different style of packages
- Deployment

```
1 def FuncHelloWorld():
2     print("Hello world!")
3
4
5 if __name__ == "__main__":
6     FuncHelloWorld()
7
```

# Python and AI



Keras



Scikit-learn



TensorFlow



Pandas



PyTorch



NumPy



Theano



SciPy



Matplotlib



## AI is the new electricity



- 100 years ago, electricity transformed every industry.
- AI is bringing an equally big transformation.

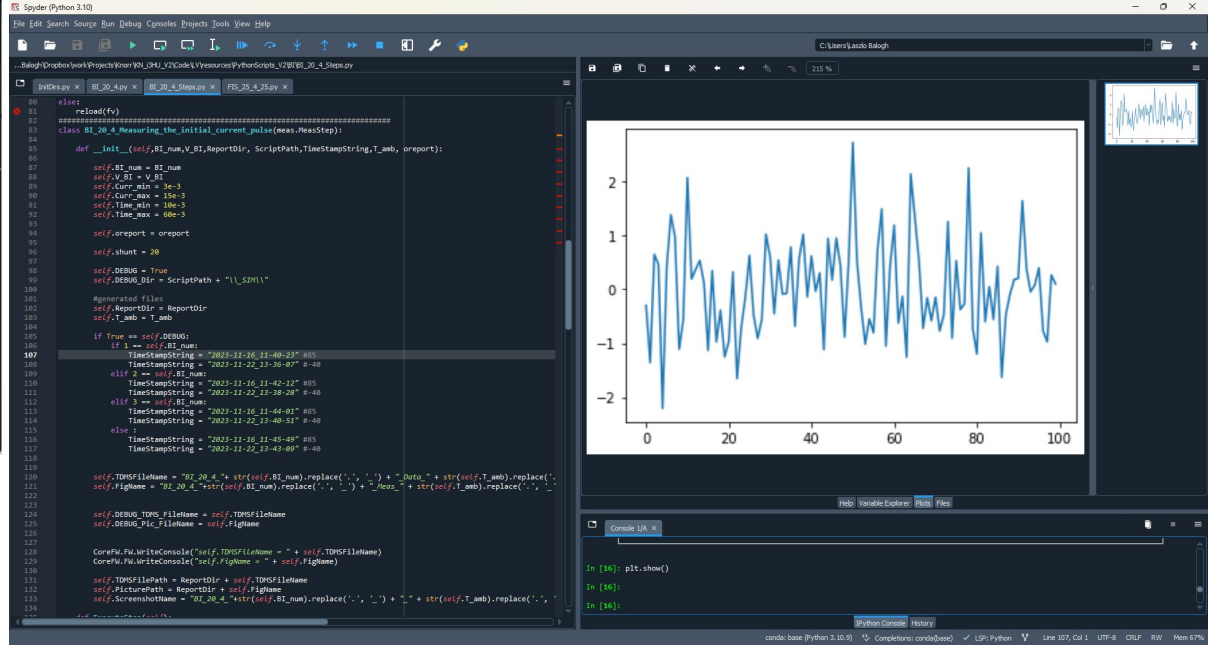
Andrew Ng

Stanford

# Editors

```
def FuncHelloWorld():
    print("Hello world!")

if __name__ == "__main__":
    FuncHelloWorld()
```



The screenshot shows the Spyder Python IDE interface. The left pane contains Python code for a class named `BI_20_4_Measuring_The_Initial_Current_pulse(MeasMeasStep)`. The code includes initialization parameters like `BI_num`, `BI_min`, `BI_max`, `Time_min`, and `Time_max`, along with a `__init__` method and a `load(fv)` method. The right pane displays a line plot of a signal over time, with the x-axis ranging from 0 to 100 and the y-axis from -2 to 2. The plot shows a noisy signal with several peaks. Below the plot is a console window showing the execution of `plt.show()` and the plot's display.

# Python Notebooks

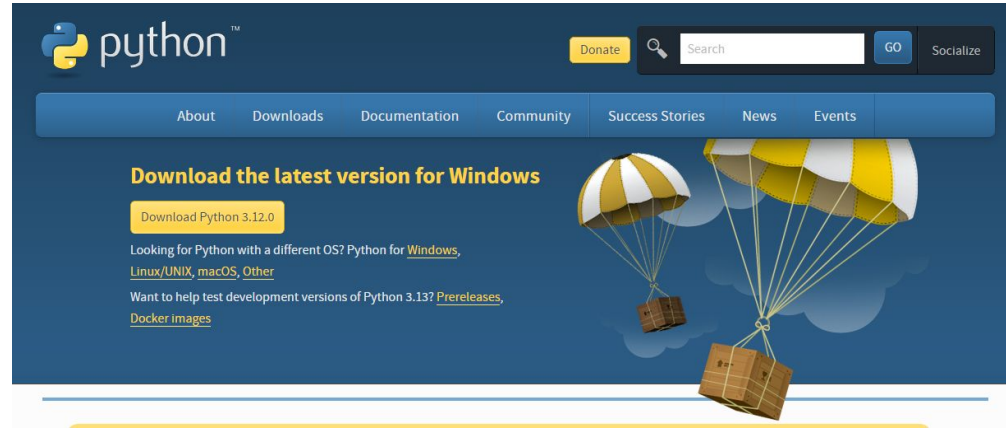


- Jupyter
- JupyterLab

A screenshot of the JupyterLab web interface. The interface is divided into several panes. On the left is a file browser showing a directory structure with folders like 'audio', 'images', 'data', and 'notebooks'. The main area contains several open notebooks. One notebook titled 'In Depth: Linear Regression' is open, showing text and code. Another notebook titled 'Simple' is also open, showing code. A third notebook titled 'Julia' is open, showing a scatter plot. A fourth notebook titled 'python notebook' is open, showing code and a plot. A fifth notebook titled 'R' is open, showing a scatter plot. In the background, a notebook titled 'Seattle Weather: 2012-2015' is open, showing a scatter plot of 'Maximum Daily Temperature (F)' over time, with a bar chart below it showing the number of records for each month. The interface includes a top menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tools', 'Settings', and 'Help'. The bottom status bar shows 'Mode: Command' and 'Ln 7, Col 1 - Lorenz.jupyter'.

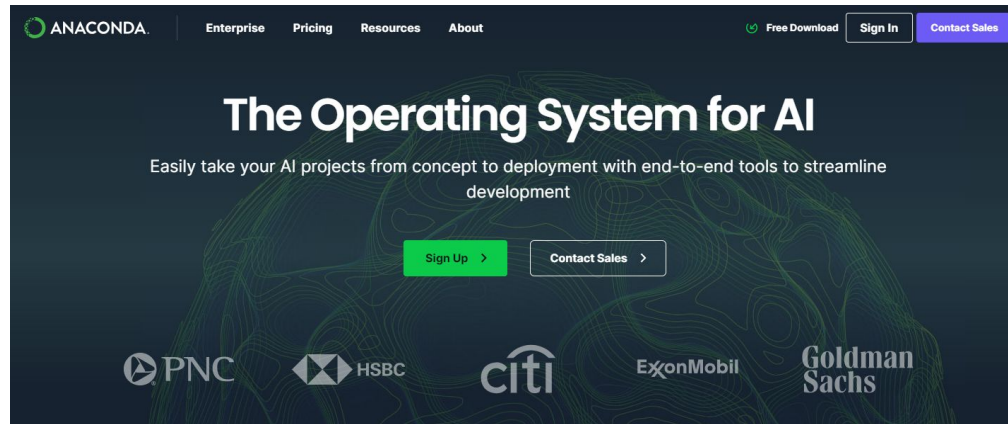
# Python Distributions

www.python.org



The screenshot shows the Python.org homepage. At the top left is the Python logo and the word "python" with a trademark symbol. To the right is a "Donate" button, a search bar with a "GO" button, and a "Socialize" button. Below this is a navigation menu with links for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The main content area features a large heading "Download the latest version for Windows" and a prominent yellow button labeled "Download Python 3.12.0". Below the button, there are links for "Linux/UNIX, macOS, Other" and "Prereleases", along with a link for "Docker images". The background of the main content area is dark blue with an illustration of two parachutes and a cardboard box falling from the sky.

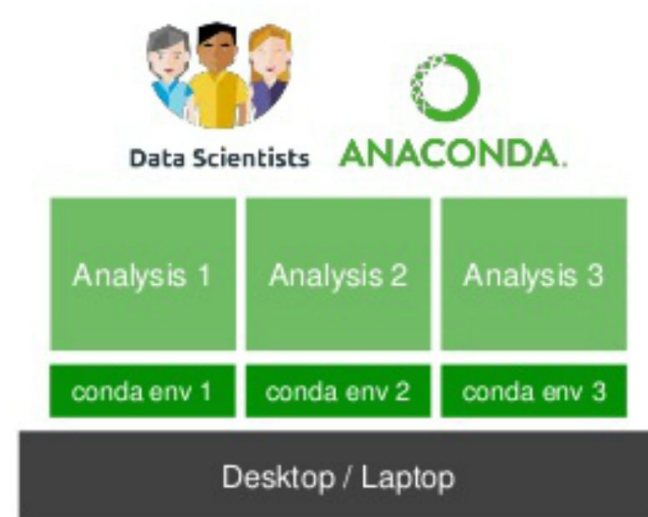
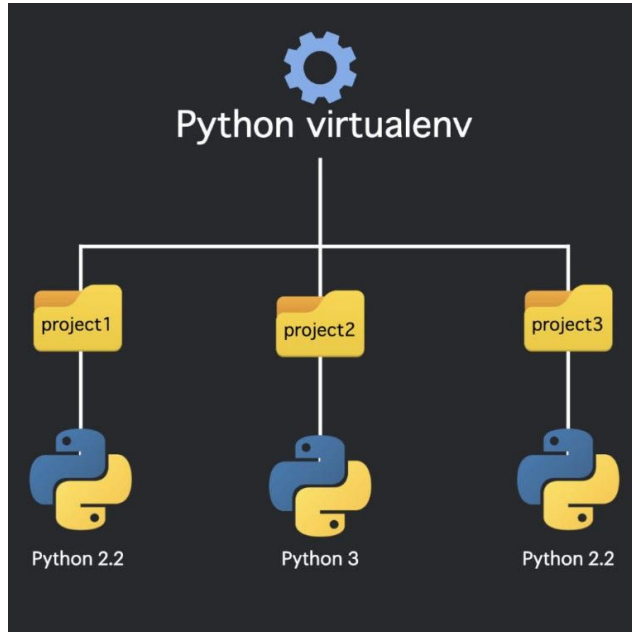
www.anaconda.com



The screenshot shows the Anaconda website homepage. At the top left is the Anaconda logo and the word "ANACONDA". To the right are navigation links for "Enterprise", "Pricing", "Resources", and "About". Further right are links for "Free Download", "Sign In", and "Contact Sales". The main content area features a large heading "The Operating System for AI" and a subheading "Easily take your AI projects from concept to deployment with end-to-end tools to streamline development". Below the subheading are two buttons: a green "Sign Up >" button and a white "Contact Sales >" button. At the bottom of the page are logos for partner companies: PNC, HSBC, citi, ExxonMobil, and Goldman Sachs.



# venv and conda



## Using a NI DAQ Device with Python and NI DAQmx in Windows

Updated Aug 29, 2023



### Environment +

#### Driver

- NI-DAQmx

#### Programming Language

- Python

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> import nidaqmx
>>> with nidaqmx.Task() as task:
        task.ai_channels.add_ai_voltage_chan("Dev1/ai0")
        task.read()

AIChannel(name=Dev1/ai0)
0.0005428027459117857
>>>
```

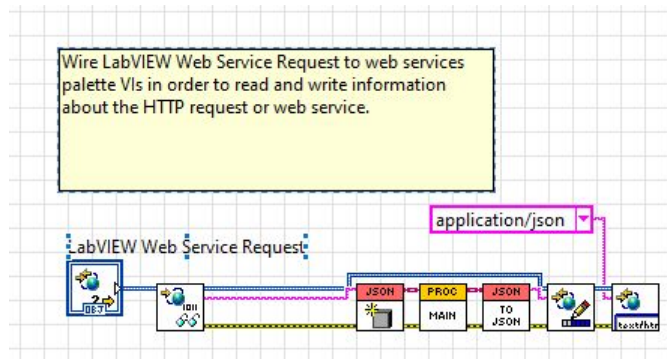
# Embedding Labview

- Calling Labview generated dll:

```
1 from ctypes import *
2 from sys import exit
3
4 lvdll = WinDLL("labview-testdll.dll")
5 add = lvdll.Addition
6 add.argtypes = (c_int, c_int)
7 add.restype = c_int
8 print("Addition of 3 and 3:", add(3, 3))
9 exit()
```



- Web service in Labview



# Processing TDMS File in Python



```
from nptdms import TdmsFile
import matplotlib.pyplot as plt
import pandas as pd

fn = "TEST LOW SPEED-20200410-114451-Part 0.tdms"

tdms_file = TdmsFile.read(fn)
group_names = []
for group in tdms_file.groups():
    group_name = group.name
    group_names.append(group_name)

#-----
# LMA v Time plot
group_LMA = tdms_file.groups()[group_names.index("LMA")]

LMA_channels = []
for ch in group_LMA.channels():
    channel_name = ch.name
    LMA_channels.append(channel_name)

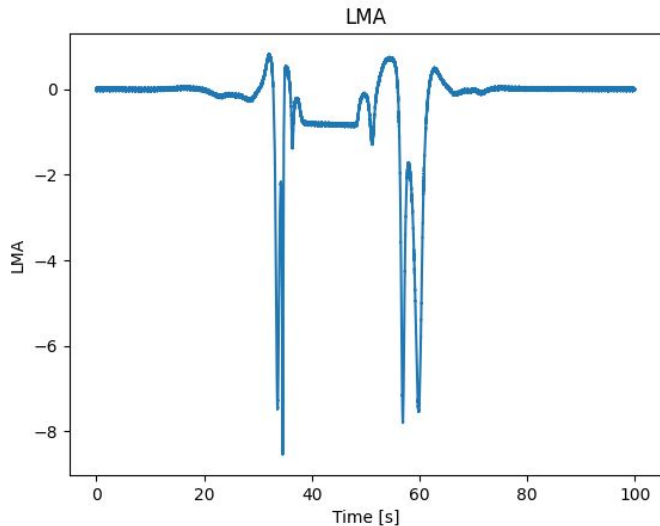
LMA_ch = group_LMA.channels()[LMA_channels.index("LMA filtered")]

Data = LMA_ch[:]
Time = LMA_ch.time_track()

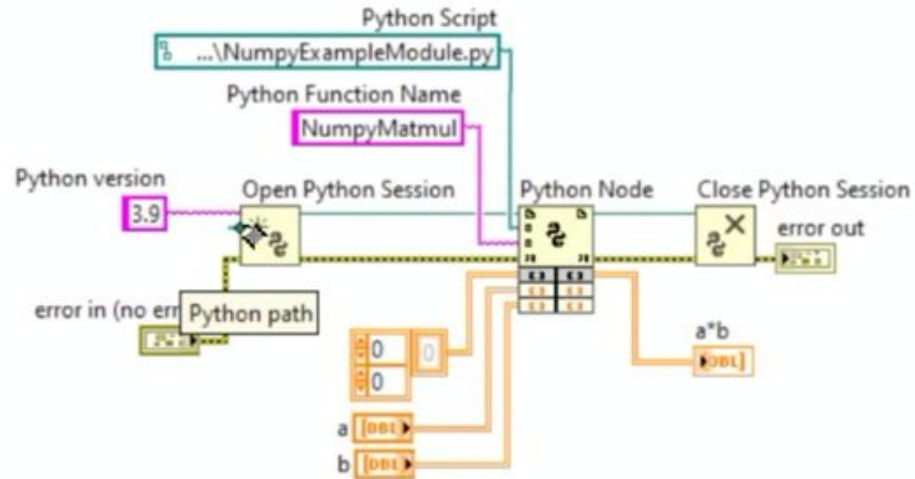
plt.plot(Time,Data)
plt.title("LMA")
plt.xlabel("Time [s]")
plt.ylabel("LMA")
plt.show()
```



```
!pip install npTDMS
```



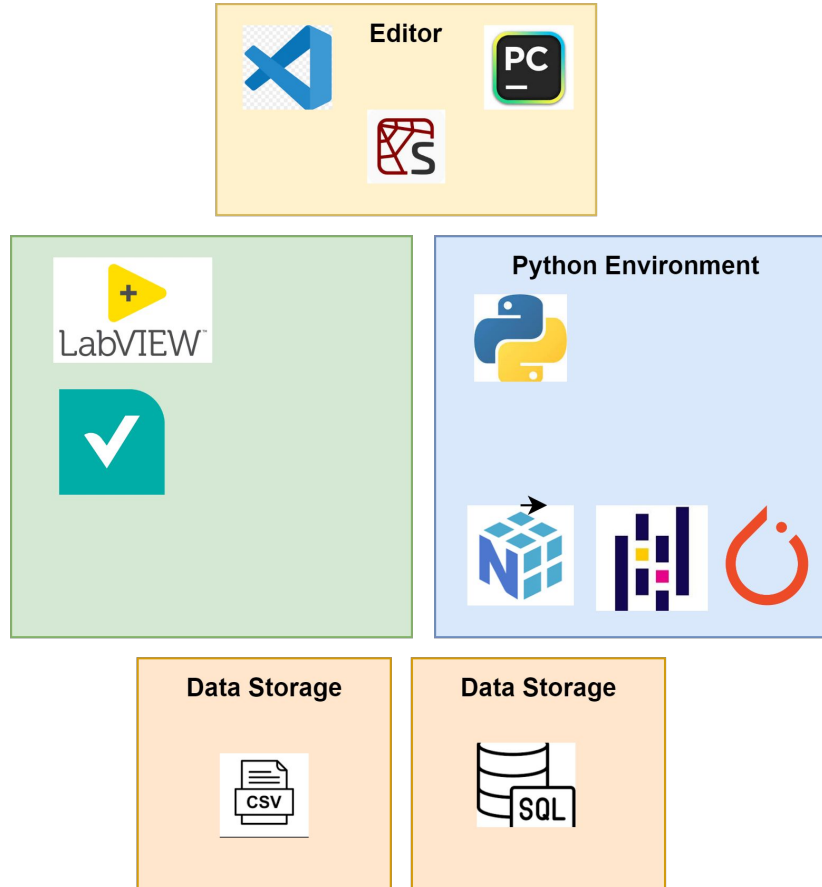
# First Step: Python Node



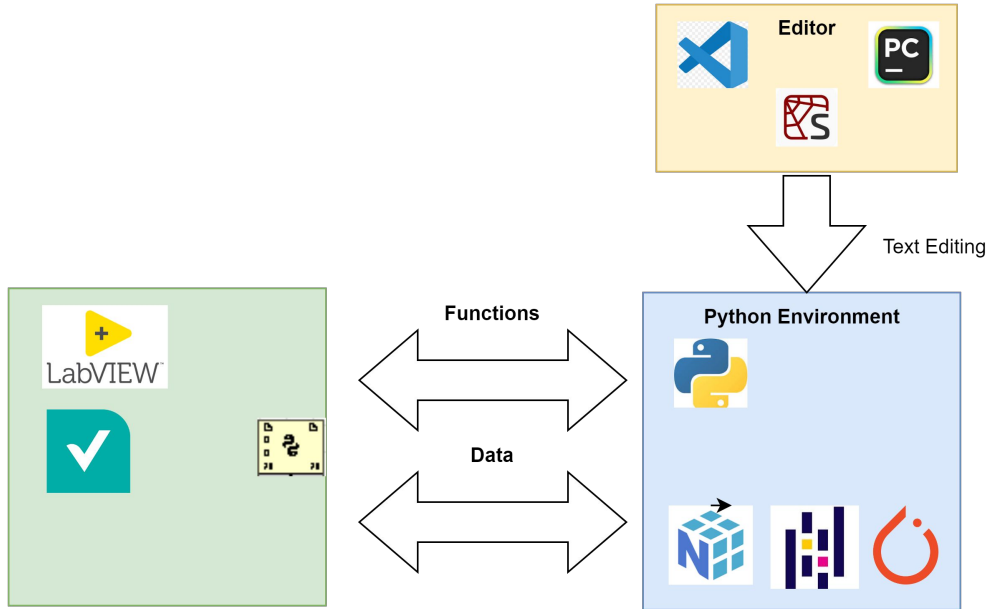
**New Features**

- Python versions 3.6-3.9 now supported
- New input "Python Path" for custom Python directory

# Labview and Python - Components



# Using Labview Python Node



## Pros:

- Simple and quick

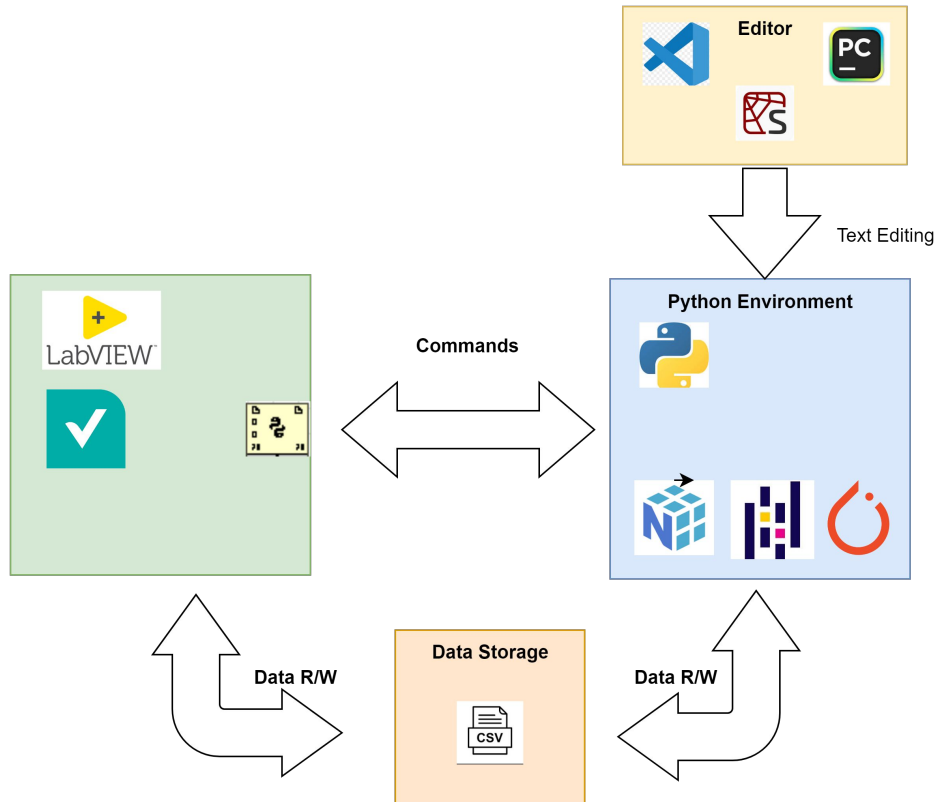
## Cons:

- Limited type in data exchange
- No control in Python execution
- Big amount of data can be problematic
- Python environment is independent
- Deployment

## Note:

- TestStand support Python (connector)

# Using Python Node and File



## Pros:

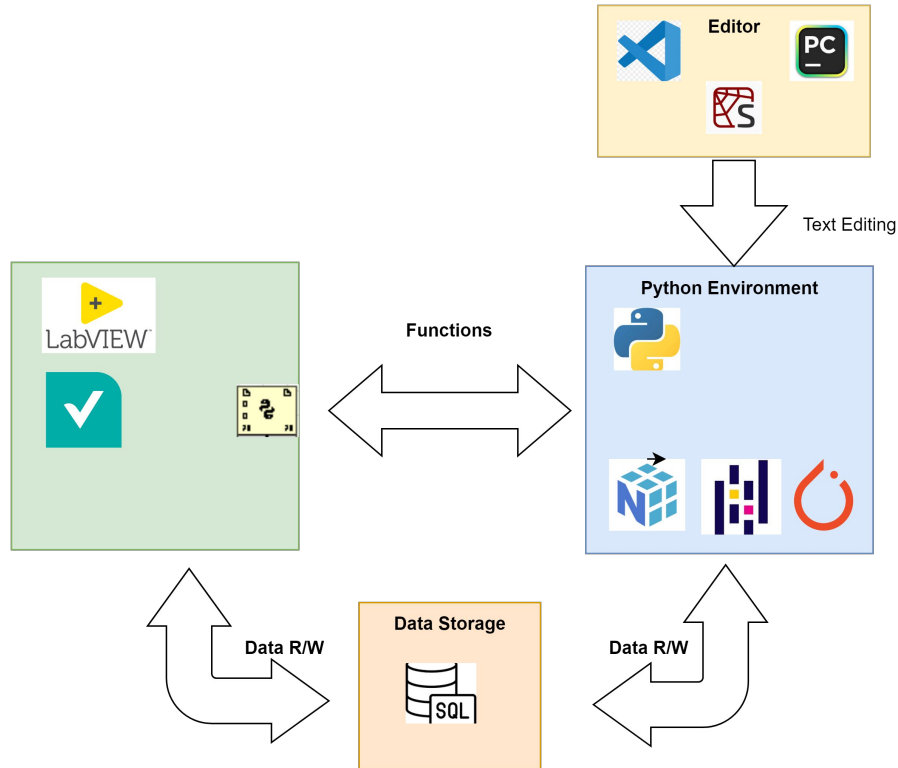
- Simple
- Big data in files

## Cons:

- No control in Python execution
- Maintenance: weak



# Using Python Node and Database

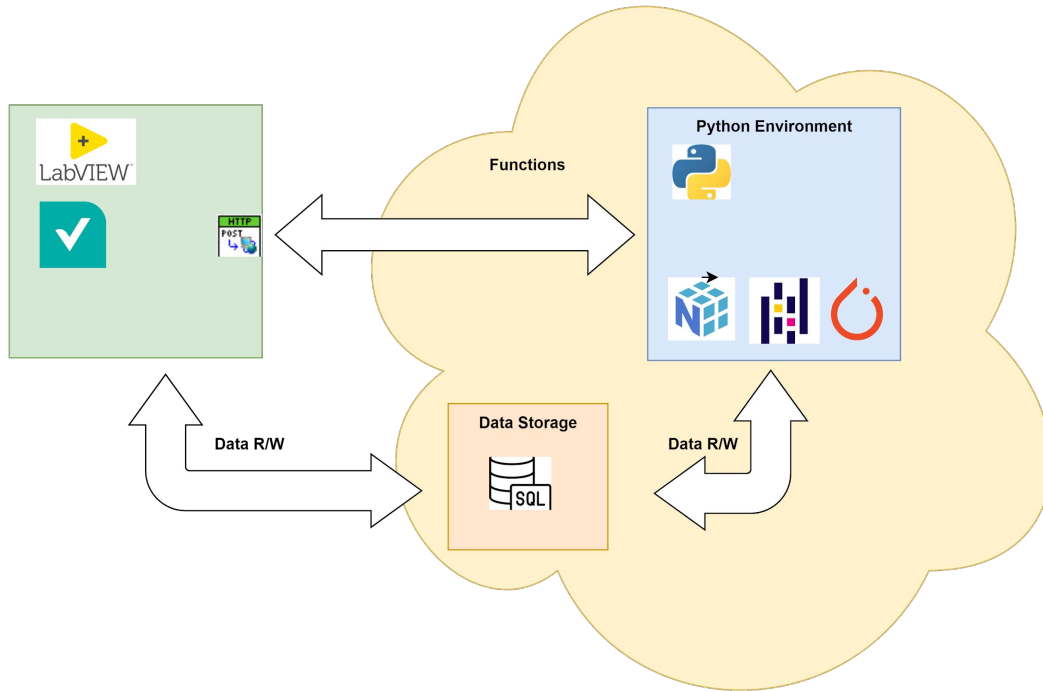


## Pros:

- Big data in files
- Proper data management

## Cons:

- No control in Python execution



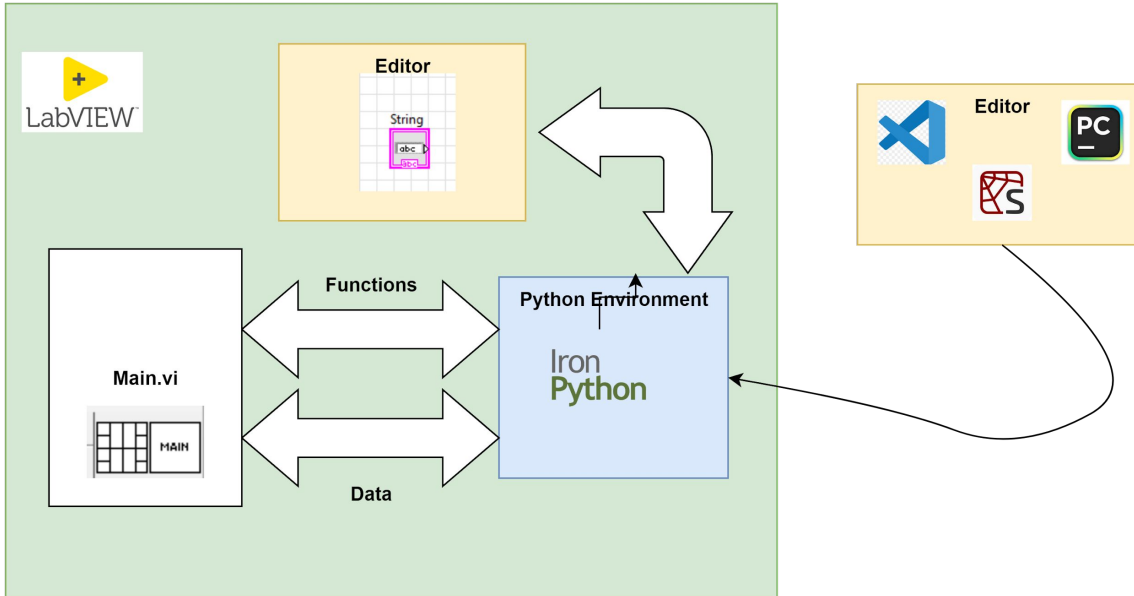
## Pros:

- Big data in files
- Proper data management
- Independent data handling
- REST API

## Cons:

- Deeper knowledge in cloud architecture

# Embedding Python in Labview



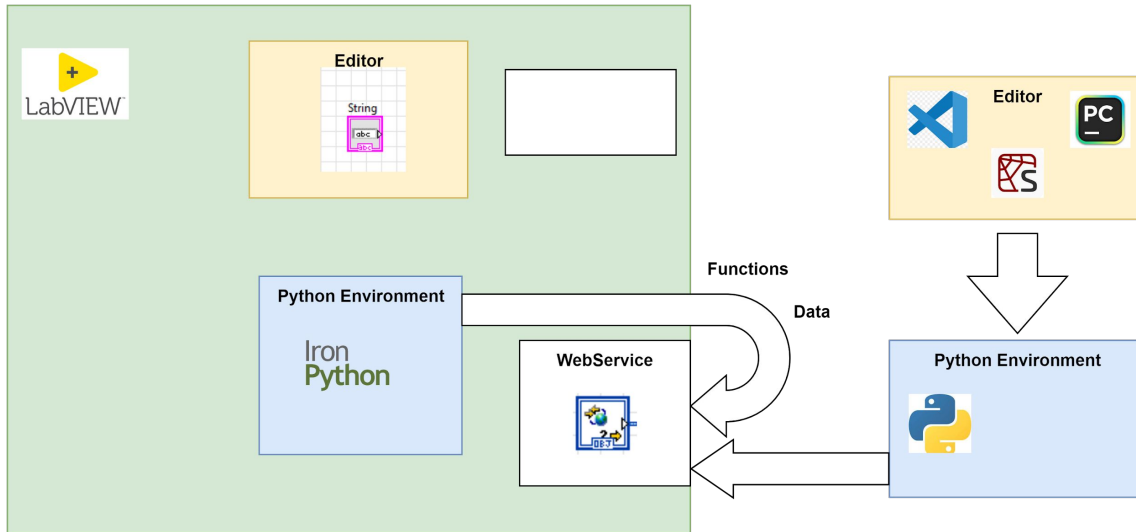
## Pros:

- Full control
- Custom function
- Easy deployment

## Cons:

- Limited library (just for scripting)
- Limited debug (yet)

# Embedding Python in Labview



## Pros:

- Full control
- Full debug
- Full featured environment

## Cons:

- Web service is required

## Scripting, TestStand vs Embedded Python

	Extensible	Modules	Speed	Format	Price
TestStand	Yes	Yes	OK	Binary	High
Python Scripting	Yes	No	OK	Text	Low

- Lightweight solution for scripting
- Competitive
- Language: Python

## Summary



**NetworkChuck** ✓

3.54M subscribers

One of the best ROI