PythonConnected GrasshopperTemplate

Documentation

Developer: Jonas FERON (UCLouvain, Belgium)

Opensource: https://github.com/JonasFeron/PythonConnectedGrasshopperTemplate

PythonConnected GrasshopperTemplate

Overview

Introduction

Why using PythonConnectedGrasshopperTemplate ?

•**Seamless Grasshopper Plugin Development** – Combine C# and Python 3 (latest version) to create multiple custom Grasshopper components that leverage the strengths of both languages.

•**Optimized Execution and Data Exchange** – Run Python scripts within Grasshopper components, efficiently transfer data, and use Python libraries like NumPy and Pandas without delays.

•**A Practical Alternative to Traditional Grasshopper Scripting** – Provides a structured and scalable approach to integrating Python into Grasshopper beyond basic scripting.

•**Ideal for Complex and Scalable Grasshopper Plug-ins** – Easily manage and develop multiple interconnected components within Visual Studio, improving organization and maintainability.

•**Conclusion**: PythonConnectedGrasshopperTemplate simplifies Grasshopper plugin development, making it easier to integrate and manage Python scripts within C# workflows.

C# (in Visual Studio) for custom Grasshopper Plug-In

But what about Python ?

Manage multiple grasshopper components written in C# in Visual Studio:

- Follow the official tutorial : Grasshopper - Your First Component
- Use the official Visual Studio Grasshopper Template
- Tip: In Grasshopper Developer Settings: do not forget to add path to your plug in

→ What if you need specific Python Librairies (like NumPy for scientific computing) ?

→ What if you already have your custom python scripts,
and want to use Grasshopper as a user interface ?

Python for custom Grasshopper component(s)

But not in Visual Studio and hence not for complex plug-in

- Follow the official tutorial Grasshopper Scripting: Python to develop your custom Grasshopper components in Python
- Python scripts are written directly within Grasshopper, not Visual Studio

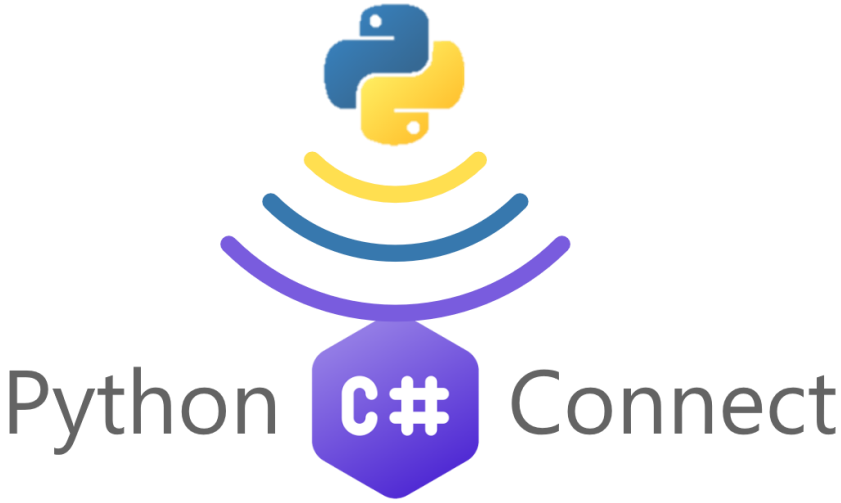
→ Difficulties to manage a full plug-in made of multiple components

→ How to develop Grasshopper plug-in in Visual Studio for Python ? (no solution)

→ PythonConnectedGrasshopperTemplate

PythonConnectedGrasshopperTemplate

is based on



+

GrasshopperTemplate

opensource:
https://github.com/JonasFeron/PythonConnect

GrasshopperTemplate for Visual Studio

Visual Studio

Custom Plug-in

custom components

C# Code in Visual Studio for this custom component

call python

```
using System;
using System.Collections.Generic;
using System.IO;
using Grasshopper.Kernel;
using Grasshopper.Kernel.Data;
using Grasshopper.Kernel.Types;
using MyGrasshopperPlugIn.PythonConnectedComponents.TwinObjec
using Newtonsoft.Json;
using PythonConnect;

namespace MyGrasshopperPlugIn.PythonConnectedComponents
{
    1 référence
    public class aPythonConnectedGHComponent : GH_Component
    {
        private static readonly log4net.ILog log = LogHelper.

        0 références
        public aPythonConnectedGHComponent()
          : base...
        {
        }

        0 références
        protected override void RegisterInputParams(GH_InputP
        {
            pManager.AddNumberParameter("A List of Numbers",
            pManager.AddIntegerParameter("Row Number", "rowNu
            pManager.AddIntegerParameter("Column Number", "co
        }

        0 références
        protected override void RegisterOutputParams(GH_Outpu
        {
            pManager.AddNumberParameter("Grasshopper tree", "
        }


        0 références
        protected override void SolveInstance(IGH_DataAccess
        /// <summary> Provides an Icon for the component.
        0 références
        protected override System.Drawing.Bitmap Icon...

        /// <summary> Gets the unique ID for this component.
        0 références
        public override Guid ComponentGuid...
```

```
# complex_script.py
import numpy as np


def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

PythonConnected GrasshopperTemplate

Explained

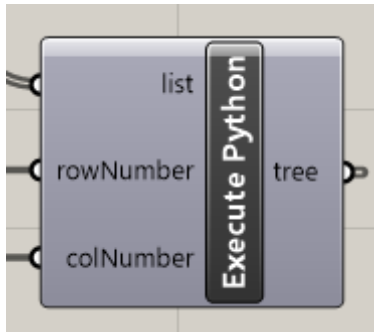
PythonConnectedGrasshopperTemplate

Manages multiple C# and Python Components in Visual Studio

Visual Studio

a Grasshopper component written in C#

is connected to Python and execute « complex_script.py »



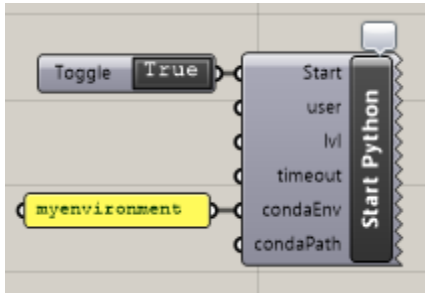

```python
# complex_script.py
import numpy as np


def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

Step 1) Start Python (from Grasshopper)

Prerequisite: Download and Install Anaconda

This Grasshopper component is written in C# and launches Python

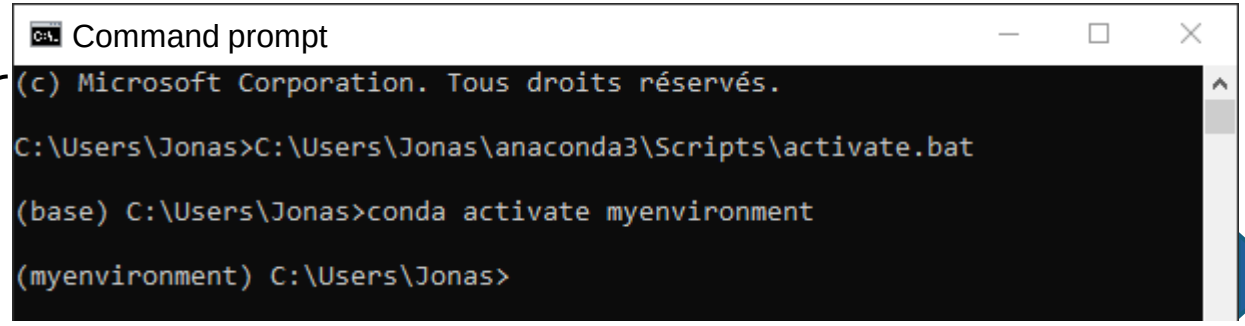


transforms into

Python thread

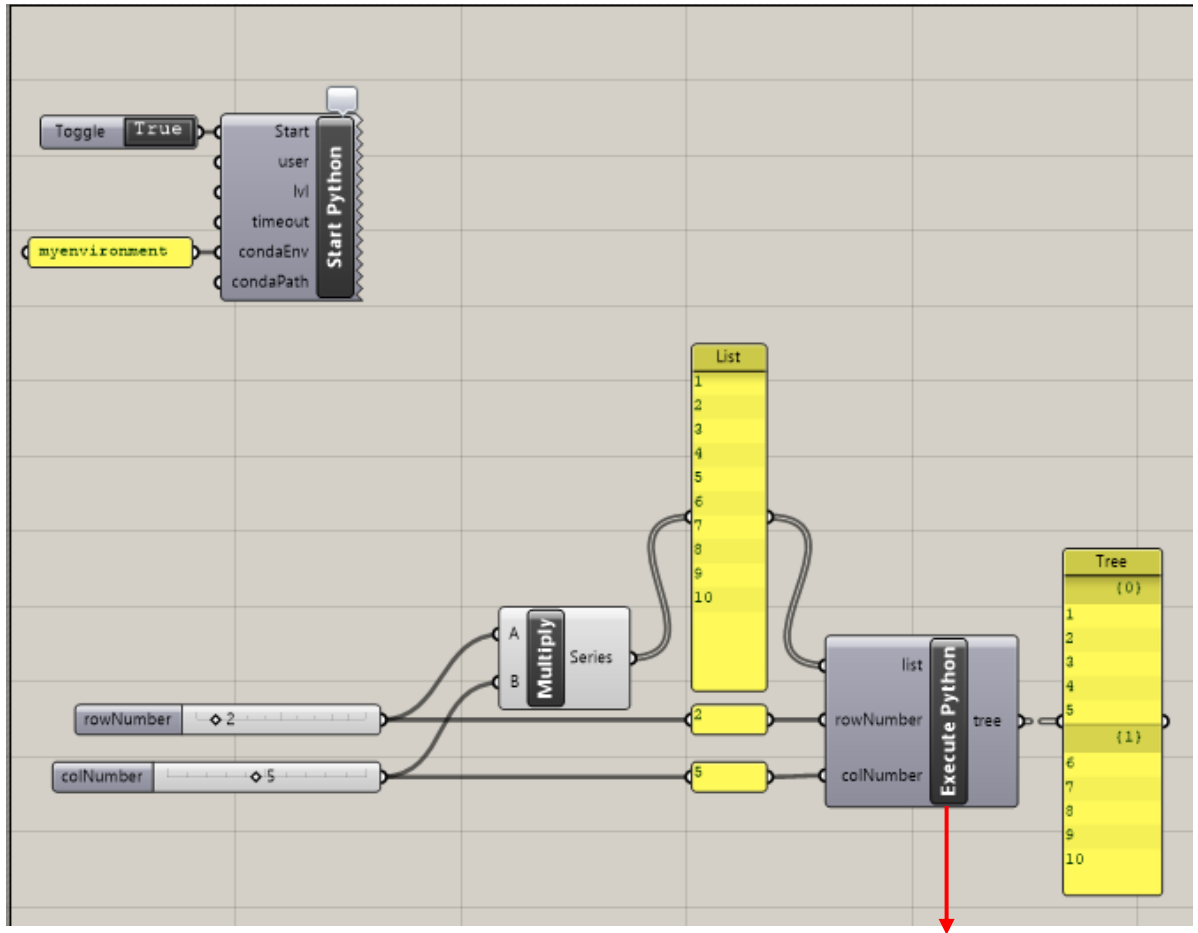
1. C# launches in parallel a new subprocess « Command Prompt »

2. Command Prompt is turned into a Python interpreter by activating Anaconda environment



Cmd is running behind the scene, waiting for new python commands to execute.....

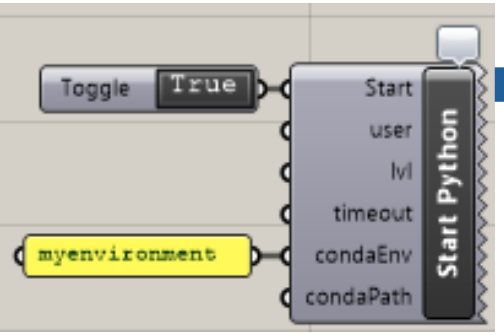
Step 2) Execute Python Scripts (from Grasshopper)




```python
# complex_script.py
import numpy as np


def main(List, rowNumber, colNumber):
    return np.array(List).reshape(rowNumber, colNumber)
```

Step 2) Execute Python Scripts (from Grasshopper)



Cmd is running behind the scene, waiting for new python commands to execute.....

Activate Anaconda
(Slow but only once)

Command prompt

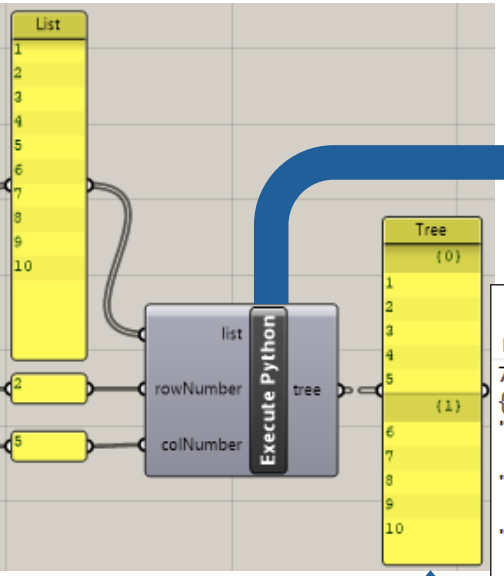
```
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Jonas>C:\Users\Jonas\anaconda3\Scripts\activate.bat

(base) C:\Users\Jonas>conda activate myenvironment

(myenvironment) C:\Users\Jonas> python  complex_script.py  path\to\Data.txt
```

Python thread



In C#

- Write Data.txt

New signal:
ExecuteCommand

Wait signal..... :
Ended execution

In C#

- Read Result.txt

Data.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
73f16727-afbb-4e12-87d7-80f39ab02b21
{
"TypeName":"TwinData",

"List":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0],

"rowNumber":2,

"colNumber":5
}
```

```
# complex_script.py

import numpy as np


def main(List, rowNumber, colNumber):

    return np.array(List).reshape(rowNumber, colNumber)
```

Result.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
73f16727-afbb-4e12-87d7-80f39ab02b21
{
"TypeName":"TwinResult",

"Array":[[1.0, 2.0, 3.0, 4.0, 5.0],
         [6.0, 7.0, 8.0, 9.0, 10.0]]

}
```

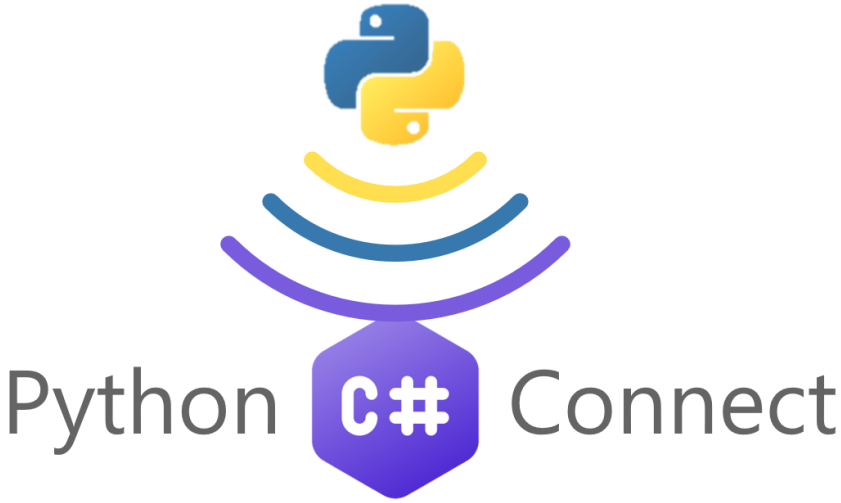
PythonConnected GrasshopperTemplate

Conclusion

PythonConnectedGrasshopperTemplate

simplifies Grasshopper plugin development,

making it easier to integrate and manage Python scripts within C# workflows.



+

GrasshopperTemplate

opensource:
https://github.com/JonasFeron/PythonConnect

GrasshopperTemplate for Visual Studio

PythonConnected GrasshopperTemplate

Appendices

Getting started with PythonConnect

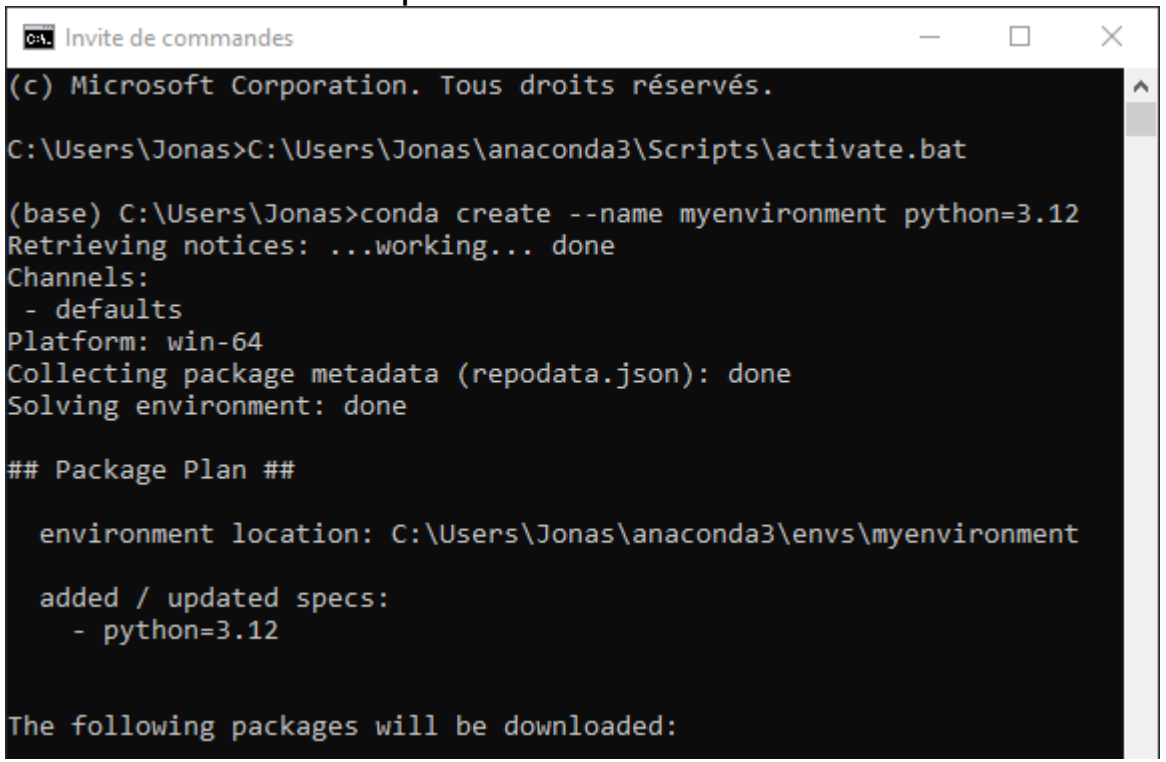
1) Download and Install Anaconda

Getting started with PythonConnect

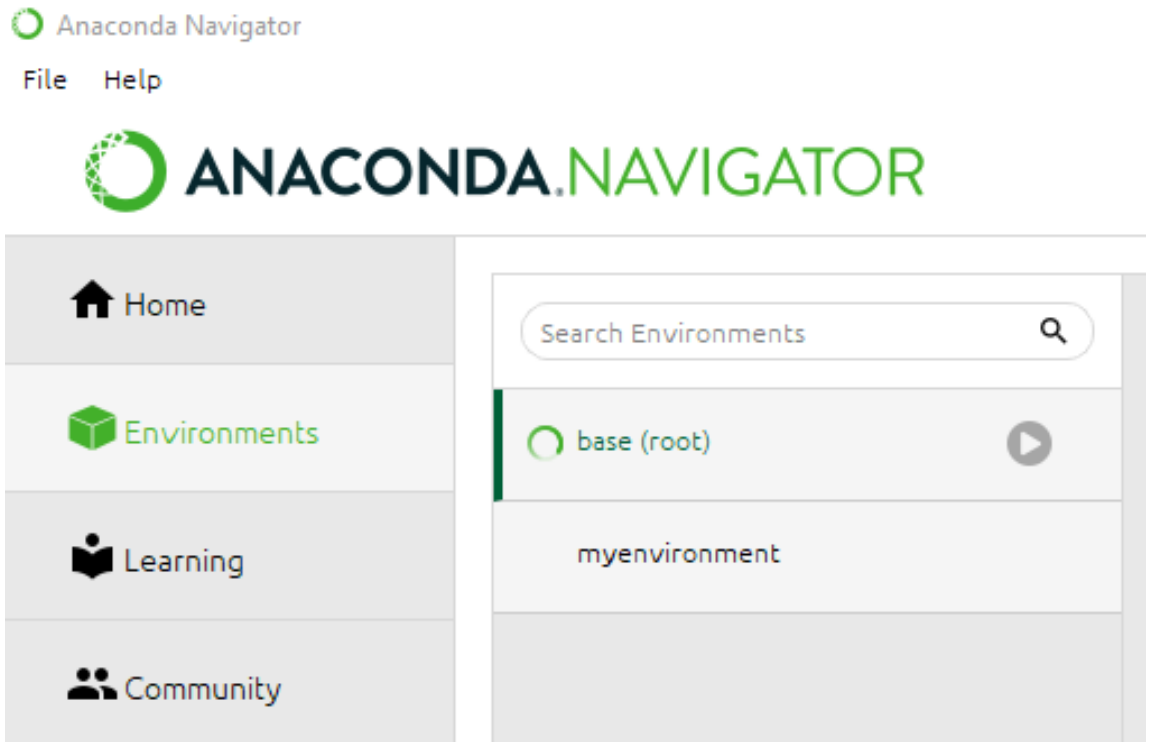
2) Manage python virtual environment

1. Use (base) conda environment
2. Or create a new environment for specific python version

via Command Prompt



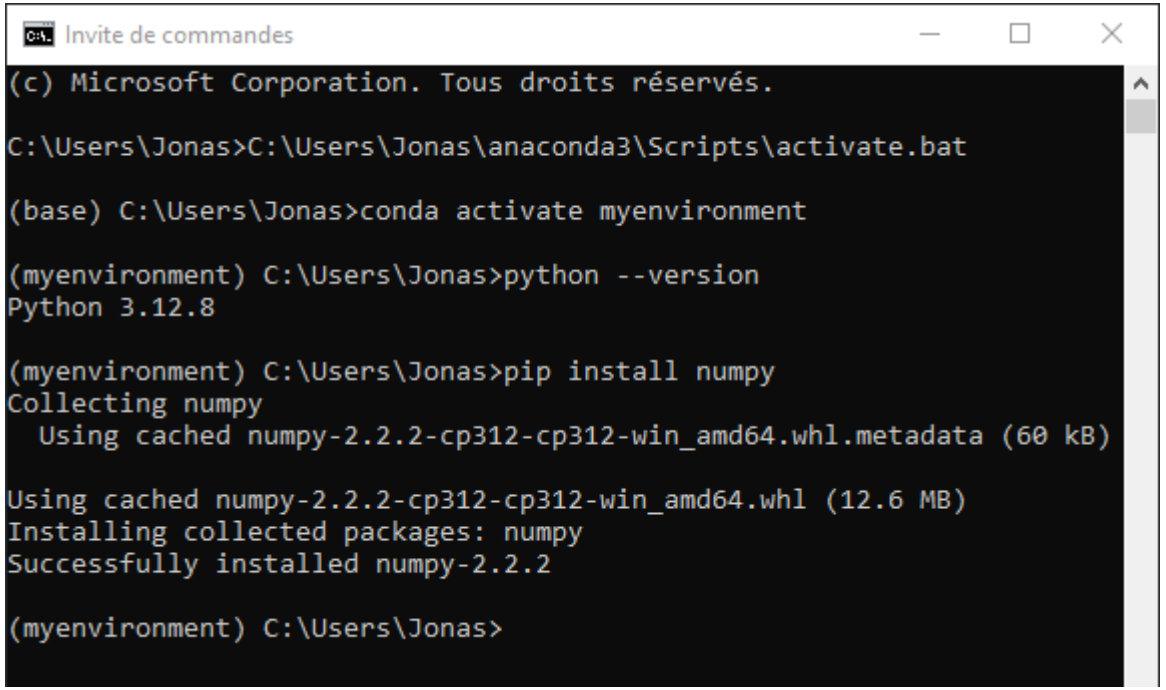
Or via Anaconda Navigator App



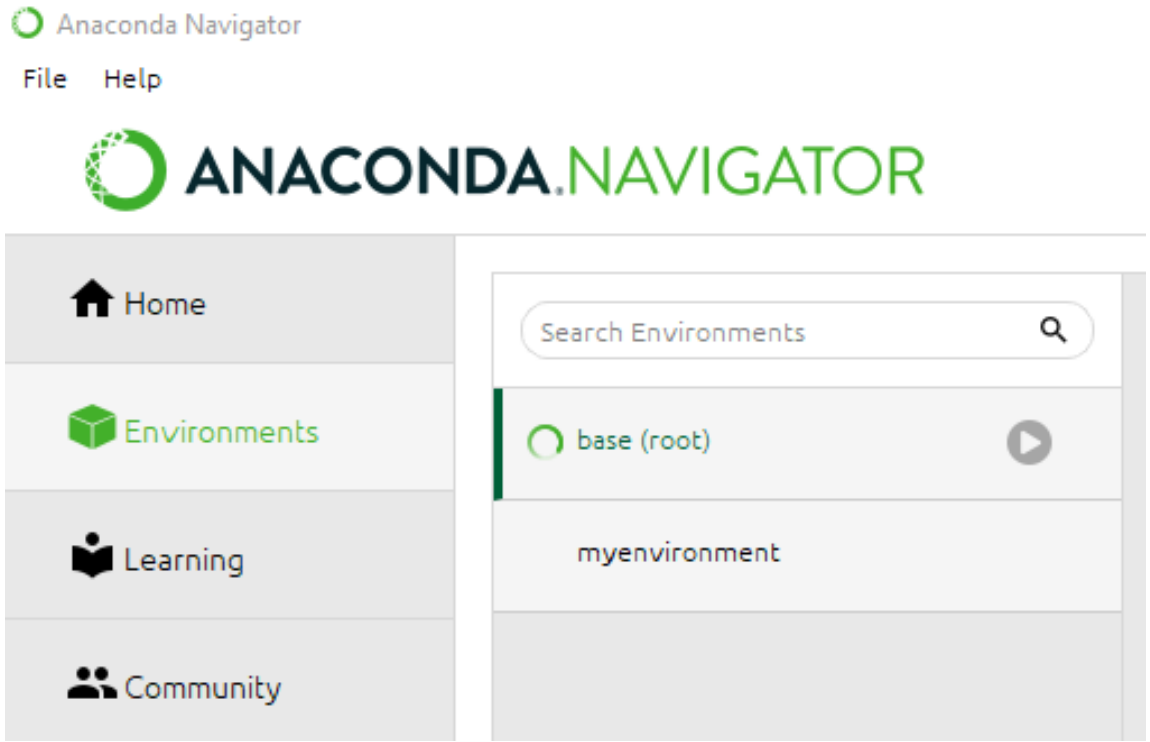
Getting started with PythonConnect

3) Install required python librairies in the environment

via Command Prompt



Or via Anaconda Navigator App



Getting started with PythonConnectedGrasshopperT.

4) From JonasFeron/PythonConnectedGrasshopperTemplate

- Fork the main branch of the Github repository, locally on your computer (GitHub Desktop helps)
- Open file src/PythonConnectedGrasshopperTemplate.sln using Visual Studio
- Run the project
- Follow the official tutorial : Grasshopper - Your First Component