

IDL-Python Bridge

ENVI and IDL are trademarks of L3HARRIS, Inc.
http://www.harrisgeospatial.com



Intro

본 문서에서는 IDL-Python Bridge 실습을 진행하고자 합니다.

IDL-Python Bridge

IDL-Python Bridge 기능은 IDL 8.5 버전부터 제공되고 있습니다. 이는 IDL에서 Python의 기능을 불러오거나 (IDL to Python to IDL), Python에서 IDL의 기능을 불러오도록 (Python to IDL) 합니다. IDL-Python Bridge에 대한 자세한 설명은 IDL 도움말 또는 아래의 링크에서 확인할 수 있습니다.

⇒ <http://blog.daum.net/swrush/560>

Tutorial

본 실습에서는 IDL-Python Bridge 를 위한 환경 구축 후, IDL to Python 및 Python to IDL을 모두 실행해볼 예정입니다.

Setting Setup

본 실습에서는 Python 사용을 위해 아나콘다 (Anaconda) 패키지를 설치한 후, 가상환경을 생성하여 실습을 진행하고자 합니다. 아래는 IDL-Python Bridge의 기본 설정 과정에 대한 설명입니다.

1) 아나콘다 패키지 설치

Anaconda 홈페이지에서 운영체제에 맞는 Python Anaconda 배포판 다운로드 및 설치를 진행합니다. 이때, Advanced Option에서 "Add Anaconda3 to my PATH environment variable"는 선택하지 않았습니다.

2) 가상환경 생성

다음으로, Anaconda Prompt 상에서 실습용 가상환경을 생성하였습니다. 이때, Python 3.6 버전을 설치하였습니다. 현재, IDL-Python Bridge은 IDL8.7의 경우, Python 3.6 그리고 IDL8.8의 경우, Python 3.6, 3.7, 3.8 버전에서 호환됩니다.

```
(base) C:\Users\USER>conda create -n bridge_test python=3.6
```

3) 환경변수 설정

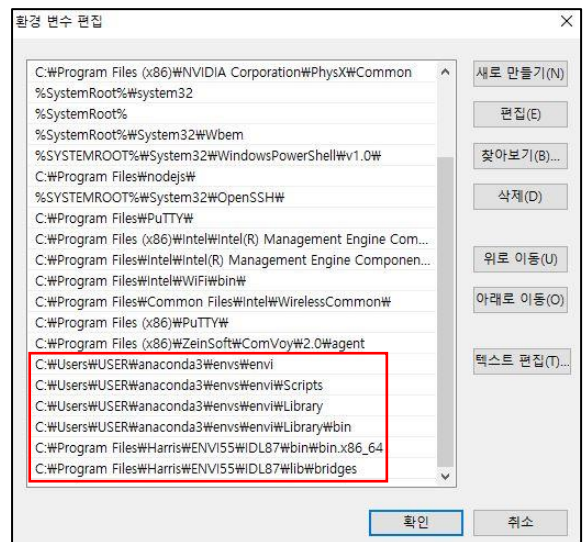
IDL-Python Bridge를 위하여 관련 경로가 환경변수로 설정되어야 합니다.

```
⇒ C:\Anaconda3
⇒ C:\Anaconda3\Scripts
⇒ C:\Anaconda3\Library\bin
⇒ C:\Program
Files\Harris\ENVI55\IDL87\bin\bin.x86_64
⇒ C:\Program
Files\Harris\ENVI55\IDL87\lib\bridges
```

이때, 환경변수 설정은 윈도우의 [제어판]-[시스템 및 보안]-

[시스템]-[고급 시스템 설정]-[고급]-[환경 변수]에서 가능하며, [시스템 변수] 내 [Path]에 위의 경로를 추가하면 됩니다.

아래의 그림은 환경변수 설정으로, 위에서 언급한 Anaconda3 관련 경로와 다소 차이가 있는 것을 확인할 수 있습니다. 현재, Anaconda를 통해 Python 3.8 버전이 설치되었으나, 본 실습에서는 Python 3.6 버전에 대한 실습용 가상환경을 생성하였습니다. 따라서, 아래와 같이 Anaconda3 설치 경로 하부 내 가상환경 경로로 환경변수 설정을 수행해야 합니다. 이에 따라, IDL-Python Bridge를 통해 Python 3.6 버전의 모듈을 사용할 수 있으며, 만약 앞서 언급한대로 환경변수를 설정한다면, Python 3.8 버전과 연동이 되기 때문에 호환성 문제로 IDL-Python Bridge는 실행할 수 없습니다 (IDL8.7의 경우, 3.6 버전과 연동).



3) Setup 파일 설치

생성된 가상환경에서 IDL 설치경로 내 setup 파일을 설치합니다.

```
(base) C:\Users\USER>conda activate bridge_test
(bridge_test) C:\Users\USER>cd C:\Program Files\Harris\ENVI55\IDL87\lib\bridges
(bridge_test) C:\Program Files\Harris\ENVI55\IDL87\lib\bridges>python setup.py install
```

4) numpy 설치

IDL의 array에 해당하는 배열을 선언하기 위하여 Python의 numpy 라이브러리가 필요합니다.

```
(bridge_test) C:\Users\USER>pip install numpy
```

앞선 작업환경 설정을 완료한 후 IDL을 import하면 다음과 같이 메시지가 뜹니다.

```
>>> from idlpy import IDL
IDL 8.7.3 (Win32 x86_64 m64).
(c) 2020, Harris Geospatial Solutions, Inc.

A new version is available: IDL 8.8
https://harrisgeospatial.flexnetoperations.com

Licensed for use by: SELab, Inc.
License: 705826
```

Python Python to IDL Bridge

본 실습에서는 Anaconda Navigator를 통해 Jupyter Notebook 설치 후, Python을 실행하였습니다. 아래는 Python to IDL Bridge의 수행 방법입니다. 우선적으로, 패키지 모듈을 불러옵니다. 이때, 기본적으로 idlpy 및 numpy가 필요합니다.

```
from idlpy import IDL as IDL
import numpy as np
```

앞서 언급하였듯이, Python에서는 numpy 패키지를 통해 IDL의 array에 해당하는 배열을 선언할 수 있습니다.

```
print, range(10)
<function print>, range(0, 10)

print, np.array(range(10), dtype = np.uint16)
<function print>, array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=uint16)

print, IDL.INDGEN(10)
<function print>, array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int16)
```

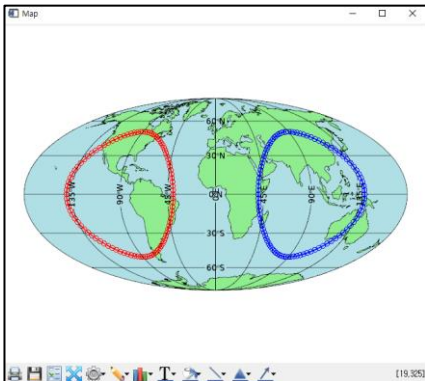
또한, MAP 함수를 사용하여 지도 표출 또한 가능합니다.

```
from idlpy import *
arr = IDL.findgen(100)/50*3.14159
x = 50*IDL.sin(arr)
y = 50*IDL.cos(arr)
m = IDL.map(test=1)
p = IDL.plot(x-90, y, 'r-o2', overplot=1)
p = IDL.plot(x+90, y, 'b-o2', overplot=1)
```

이를 아래와 같은 형태로 변형하여 사용할 수도 있습니다.

```
from idlpy import *
IDL.run("arr = findgen(100)/50*3.14159")
IDL.run("x = 50*sin(arr)")
IDL.run("y = 50*cos(arr)")
IDL.run("m=MAP(test=1)")
IDL.run("p = plot(x-90, y, 'r-o2', overplot=1)")
IDL.run("p = plot(x+90, y, 'b-o2', overplot=1)")
```

처리 결과는 IDL에서와 동일한 GUI를 통해 표출됩니다. 아래의 그림 확인 부탁드립니다.



IDL IDL to Python Bridge

다음으로는, IDL to Python Bridge의 수행 방법으로, IDL에서 Python의 패키지를 불러오려고 합니다. 본 실습에서는 numpy 패키지를 호출하여, 간단한 배열 생성 및 함수 설정을 수행하도록 하겠습니다. 우선, 패키지 호출을 위한 첫 번째 방법은 Python.Import()로, 괄호 안에 패키지 입력을 수행합니다. 본 실습에서는 numpy 패키지를 불러와 np로 선언하였습니다.

```
IDL> np = Python.Import('numpy')
% Loaded DLM: PYTHON36.
IDL> coords = np.random.random([10,2])
IDL> x = coords[0,*]
IDL> y = coords[1,*]
IDL> r = np.sqrt(x^2+y^2)
IDL> t = np.arctan2(y,x)
```

위의 코드는 ① np.random.random을 통해 [10,2]의 랜덤 배열 선언 ② 각 열을 x, y로 선언 ③ numpy 내 sqrt, arctan2 기능 선언 순으로 이루어집니다. 한 가지 주의할 점은 행렬 선언시, IDL과 Python의 배열이 서로 반대이며, 각각 [Column, Row], [Row, Column]입니다. 또 다른 방법으로는, >>>을 통해 IDL Command Line 환경에서 실행하는 방법으로, Python에서와 동일하게 스크립트를 작성하면 됩니다. 이때, 배열의 순서는 Python과 동일하게 진행하면 됩니다.

```
IDL> >>>
>>> import numpy as np
>>> coords = np.random.random((10,2))
>>> x,y = coords[:,0], coords[:,1]
>>> r = np.sqrt(x**2+y**2)
>>> t = np.arctan2(y,x)
```

추가적으로, 아래의 그림은 matplotlib 패키지를 호출하여 sin, cosine 함수를 plot한 결과입니다.

```
IDL> >>>
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x = np.arange(0,10,0.01)
>>> y1 = np.sin(x)
>>> y2 = np.cos(x)
>>> plt.plot(x,y1,c='r')
[<matplotlib.lines.Line2D object at 0x0000000050976C88>]
>>> plt.plot(x,y2,c='b')
[<matplotlib.lines.Line2D object at 0x0000000050976D68>]
>>> plt.xlabel('x')
Text(0.5, 0, 'x')
>>> plt.ylabel('y')
Text(0, 0.5, 'y')
>>> plt.legend(['y=sin(x)', 'y=cos(x)'])
<matplotlib.legend.Legend object at 0x000000005098E208>
>>> plt.grid(True)
>>> plt.show()
IDL>
```

