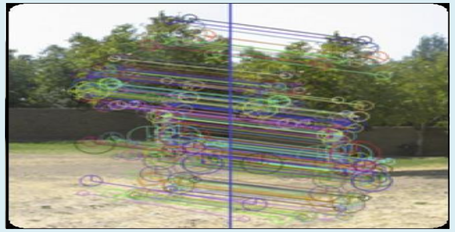
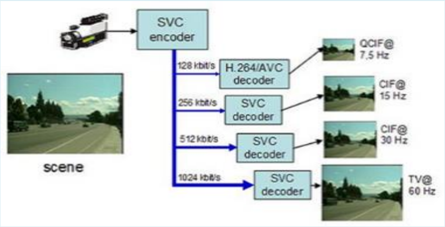
Open-Source Programming

#2: Introduction of Anaconda Platform & Programming (1)

2025 Spring

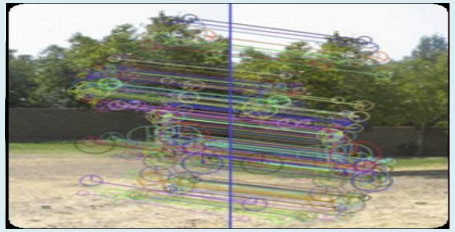
Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
http://ivpl.sookmyung.ac.kr
Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@sookmyung.ac.kr

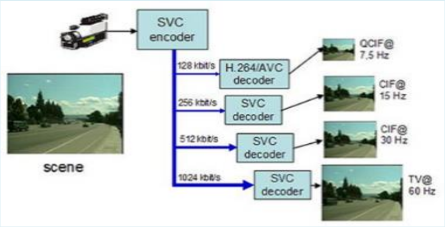




Contents

- Examples in Open-source SW (OSS)
- Intro. Of Anaconda Platform
- Exercise of Anaconda Environment





Contents

- **Examples in Open-source SW (OSS)**
- **Intro. Of Anaconda Platform**
- **Exercise of Anaconda Environment**

Examples in Open-source SW (OSS) (1)

❖ Linux

- Linux is **the best-known and most-used** open source **operating system**.
- As an operating system, **Linux is software that sits underneath all of the other software on a computer**, receiving requests from those programs and relaying these requests to the computer's hardware.

Examples in Open-source SW (OSS) (2)

❖ Linux

- **What is the difference between Unix and Linux?**

Unix vs Linux

- Unix is an operating system developed in the 1970s at Bell Labs by Ken Thompson, Dennis Ritchie, and others.
- Unix and Linux are similar in many ways, and
- In fact, Linux was originally created to be indistinguishable from Unix.
- **Linux was created in 1991 by Linus Torvalds**, a then-student at the University of Helsinki. Torvalds built Linux as a free and open source alternative to Minix, another Unix clone that was predominantly used in academic settings.

Examples in Open-source SW (OSS) (3)

❖ Python

- Python is an interpreted, interactive, object-oriented programming language.
- Python has advanced as **an open source programming language** by managing public discussion through Python Enhancement Proposals (PEPs).

- **Is Python open source?**
  - all modern versions of Python are copyrighted under a GPL-compatible license certified by the Open Source Initiative (SOI).

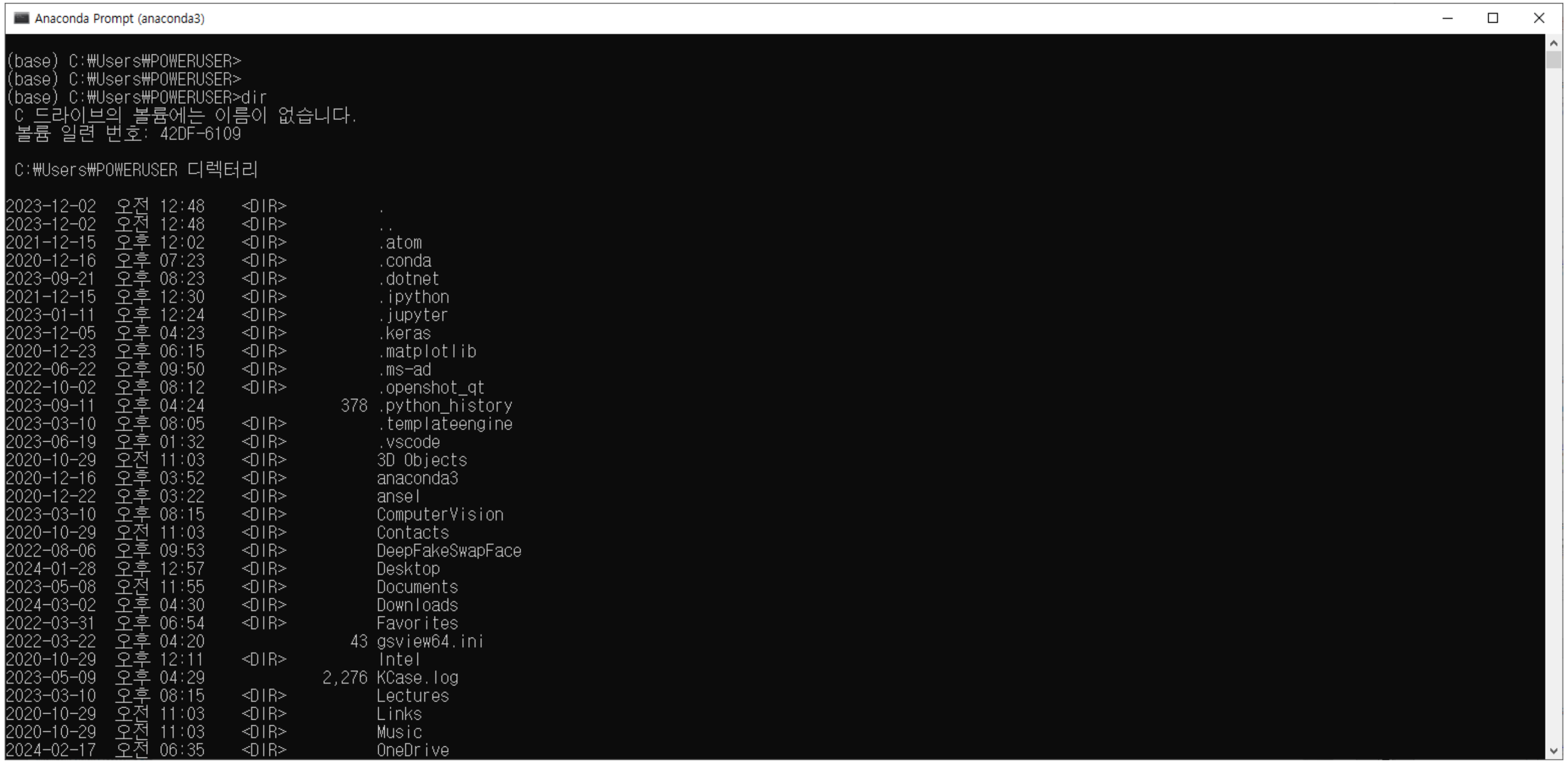
Examples in Open-source SW (OSS) (4)

❖ Bash Shell

- Shell is for **Unix computers** that operates outside of the kernel (or around the kernel, like a shell in nature) and allows **humans** to interact with the computer whenever they want to.
- Bash is one of the most popular, the most powerful, and the most friendly.
- **Bash is an application**
  - Its primary job is to **run *other* applications** (in the form of commands) that are installed on the same system.
  - When you start a terminal (such as the GNOME Terminal or Konsole on Linux or iTerm2 on macOS) running the Bash shell, you're greeted with a ***prompt***. **A prompt** is a symbol, usually a dollar sign (**$**), indicating that the shell is waiting for your input.

Examples in Open-source SW (OSS) (5)

- **Bash is an application**
  - Its primary job is to **run *other* applications** (in the form of commands) that are installed on the same system.
  - When you start a terminal (such as the GNOME Terminal or Konsole on Linux or iTerm2 on macOS) running the Bash shell, you're greeted with a ***prompt***. **A prompt** is a symbol, usually a dollar sign (**$**), indicating that the shell is waiting for your input.



```
(base) C:₩Users₩POWERUSER>
(base) C:₩Users₩POWERUSER>
(base) C:₩Users₩POWERUSER>dir
 C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 42DF-6109

 C:₩Users₩POWERUSER 디렉터리

2023-12-02  오전 12:48    <DIR>          .
2023-12-02  오전 12:48    <DIR>          ..
2021-12-15  오후 12:02    <DIR>          .atom
2020-12-16  오후 07:23    <DIR>          .conda
2023-09-21  오후 08:23    <DIR>          .dotnet
2021-12-15  오후 12:30    <DIR>          .ipython
2023-01-11  오후 12:24    <DIR>          .jupyter
2023-12-05  오후 04:23    <DIR>          .keras
2020-12-23  오후 06:15    <DIR>          .matplotlib
2022-06-22  오후 09:50    <DIR>          .ms-ad
2022-10-02  오후 08:12    <DIR>          .openshot_qt
2023-09-11  오후 04:24               378 .python_history
2023-03-10  오후 08:05    <DIR>          .templateengine
2023-06-19  오후 01:32    <DIR>          .vscode
2020-10-29  오전 11:03    <DIR>          3D Objects
2020-12-16  오후 03:52    <DIR>          anaconda3
2020-12-22  오후 03:22    <DIR>          ansel
2023-03-10  오후 08:15    <DIR>          ComputerVision
2020-10-29  오전 11:03    <DIR>          Contacts
2022-08-06  오후 09:53    <DIR>          DeepFakeSwapFace
2024-01-28  오후 12:57    <DIR>          Desktop
2023-05-08  오전 11:55    <DIR>          Documents
2024-03-02  오후 04:30    <DIR>          Downloads
2022-03-31  오후 06:54    <DIR>          Favorites
2022-03-22  오후 04:20                43 gsview64.ini
2020-10-29  오후 12:11    <DIR>          Intel
2023-05-09  오후 04:29             2,276 KCase.log
2023-03-10  오후 08:15    <DIR>          Lectures
2020-10-29  오전 11:03    <DIR>          Links
2020-10-29  오전 11:03    <DIR>          Music
2024-02-17  오전 06:35    <DIR>          OneDrive
```

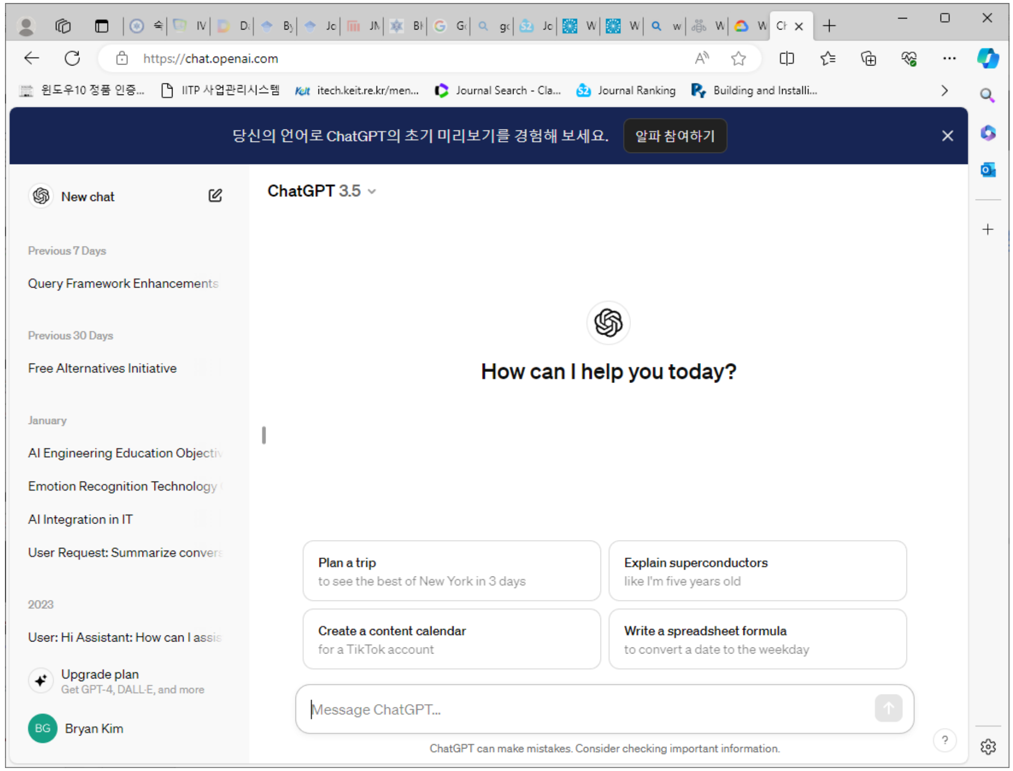
Examples in Open-source SW (OSS) (6)

Prompt Engineering

Generative AI models operate based on natural language processing (NLP) and use natural language inputs to produce complex results.

- What is prompt engineering?
  - a discipline focused on designing and optimizing prompts to enhance the performance and reliability of AI models, particularly LLMs.

- What is a prompt for AI?
  - A prompt for artificial intelligence refers to a statement or instruction provided to an AI model to guide its response or action. Prompts serve as a means of communication between humans and AI systems, enabling users to convey their requests or queries effectively.



당신의 언어로 ChatGPT의 초기 미리보기를 경험해 보세요.

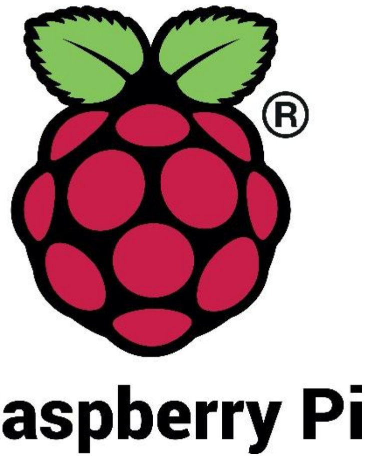
How can I help you today?

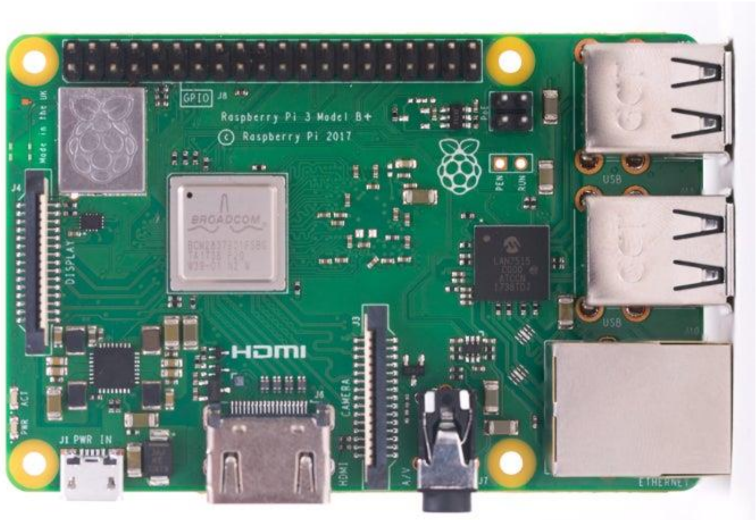
ChatGPT can make mistakes. Consider checking important information.

Examples in Open-source SW (OSS) (7)

❖ Raspberry Pi

- Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.
- **Is the Raspberry Pi open source?**
  - The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Pi OS, is open source and runs a suite of open source software.
  - The Raspberry Pi Foundation contributes to the Linux kernel and various other open source projects as well as releasing much of its own software as open source.
- Code Club and CoderDojo





Examples in Open-source SW (OSS) (8)

- **Want to read more Raspberry Pi articles?**
  - How piwheels will save Raspberry Pi users time in 2020
  - 10 Hello World programs for your Raspberry Pi
  - Play retro video games on the Pi
  - The physical computing capabilities of the Raspberry Pi
  - Top 10 Raspberry Pi add-on boards
  - GPIO Zero and Raspberry Pi programming starter projects
  - Exploring the Raspberry Pi Sense HAT
  - Pi in the sky: High altitude ballooning with Raspberry Pi

Examples in Open-source SW (OSS) (9)

❖ Java

- commonly used to refer to the **Java platform**, **a set of tools allowing for easy cross-platform application development, as well as the Java programming language**, which is a general-purpose programming language often used to develop programs for this platform.
- It is designed such that code written in Java can be run on any system that a **Java virtual machine (JVM)** can run on.

- **Is Java open source?**
  - today, **most major components of Java are available under open source licenses**, and those which are not available under open licenses typically have drop-in replacements which are open.

Examples in Open-source SW (OSS) (10)

- **Where can I learn more?**

In addition to following the Java tag here on Opensource.com, here are some resources you might want to check out.

- AdoptOpenJDK (Java User Group)
- Java programming language (Wikipedia)
- Java software platform (Wikipedia)
- Java homepage (Java.com)
- Java questions and discussion (Stack Overflow)

Examples in Open-source SW (OSS) (11)

❖ Anaconda Platform

- A distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
- The distribution includes data-science packages suitable for Windows, Linux, and macOS.
- It is developed and maintained by **Anaconda, Inc.**, which was founded by Peter Wang and Travis Oliphant in 2012.
- Open-source?
  - **Anaconda Distribution** or **Anaconda Individual Edition** as open-source
  - **Anaconda Team Edition** and **Anaconda Enterprise Edition** as non-free (not opne-source)

Examples in Open-source SW (OSS) (12)

❖ Python

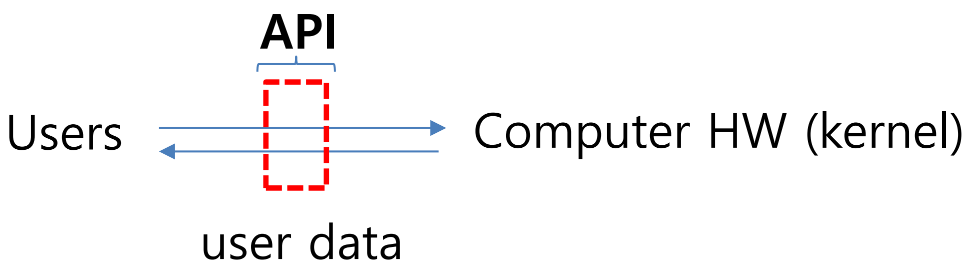
- an **interpreted, object-oriented, high-level programming language with dynamic semantics**. It was created by **Guido van Rossum** and released in 1991.
- Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it attractive for Rapid Application Development.

- Python is an open-source software?
  - Python is indeed open-source (https://www.python.org/about/).
  - Python is developed under an **OSI-approved open source license**, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

Checking point (1)

❖ What is API?

- application programming interface (API)



API

Users

Computer HW (kernel)

user data

```
#include <stdio.h>
int main(int argc, char *argv[])
{
            int i; for(i=1;i<argc;i++) {
                        printf("%s\n",argv[i]);
            }
            return 0;
}
```

Checking point (2)

❖ What is API?

- Examples of Application Programming Interface (API)
  - There are many standard input & output functions in C language.
  - These **standard input & output functions** can be considered as kinds of **APIs responsible for the interaction between a C program and the operating system**.

1.Input/Output Functions:

- **printf**: Outputs formatted text to the standard output stream.
- **scanf**: Reads formatted input from the standard input stream.
- **getchar**, **putchar**: Reads and writes a single character to the standard input/output stream.

2.String Manipulation Functions:

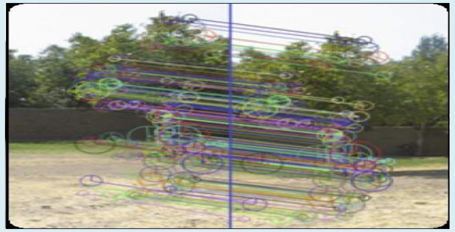
- **strcpy**, **strncpy**: Copies strings.
- **strlen**: Returns the length of a string.
- **strcmp**, **strncmp**: Compares strings.
- **strcat**, **strncat**: Concatenates strings.
- **sprintf**, **snprintf**: Writes formatted data to strings.

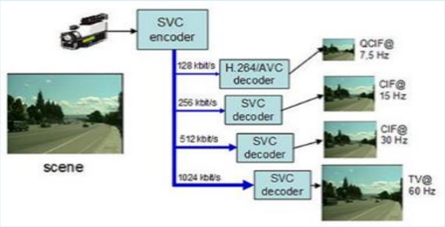
3.Memory Allocation Functions:

- **malloc**, **calloc**: Allocates memory on the heap.
- **free**: Deallocates memory previously allocated by **malloc** or **calloc**.
- **realloc**: Changes the size of the memory block.

4.Mathematical Functions:

- **sqrt**: Computes the square root.
- **sin**, **cos**, **tan**: Trigonometric functions.
- **pow**: Computes the power of a number.



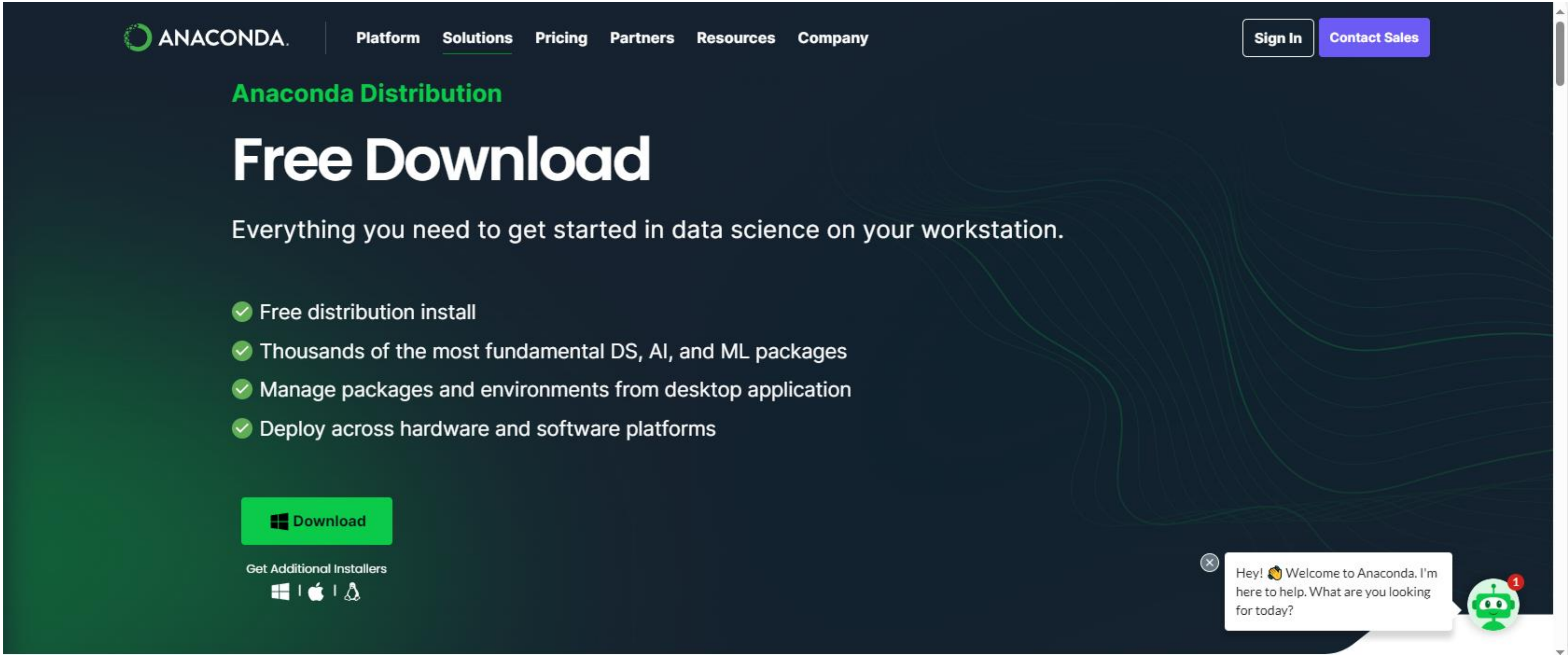


Contents

- Examples in Open-source SW (OSS)
- **Intro. Of Anaconda Platform**
- Exercise of Anaconda Environment

Introduction to Anaconda Platform (1)

- Anaconda (https://www.anaconda.com/)



Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

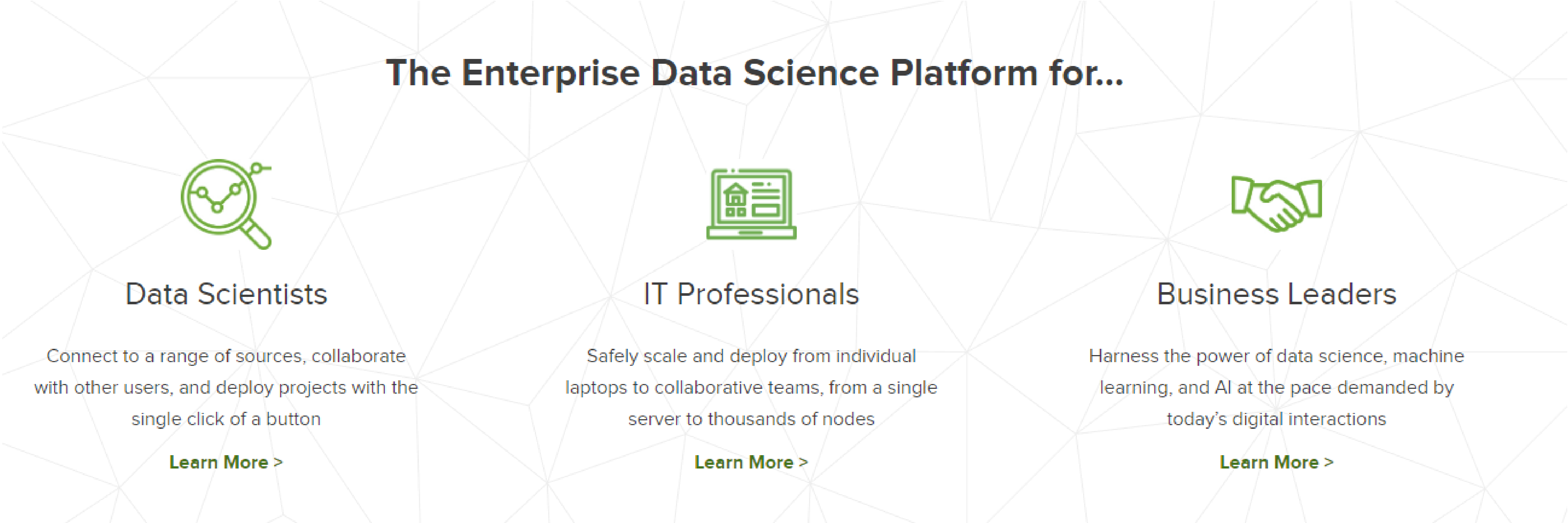
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Hey! 👋 Welcome to Anaconda. I'm here to help. What are you looking for today?

Introduction to Anaconda Platform (2)

❖ Definition

- **Data Science and Analysis Platform** (by Python)
- The World's Most Popular Python/R Data Science Platform



The Enterprise Data Science Platform for...

Data Scientists

Connect to a range of sources, collaborate with other users, and deploy projects with the single click of a button

Learn More >

IT Professionals

Safely scale and deploy from individual laptops to collaborative teams, from a single server to thousands of nodes

Learn More >

Business Leaders

Harness the power of data science, machine learning, and AI at the pace demanded by today's digital interactions

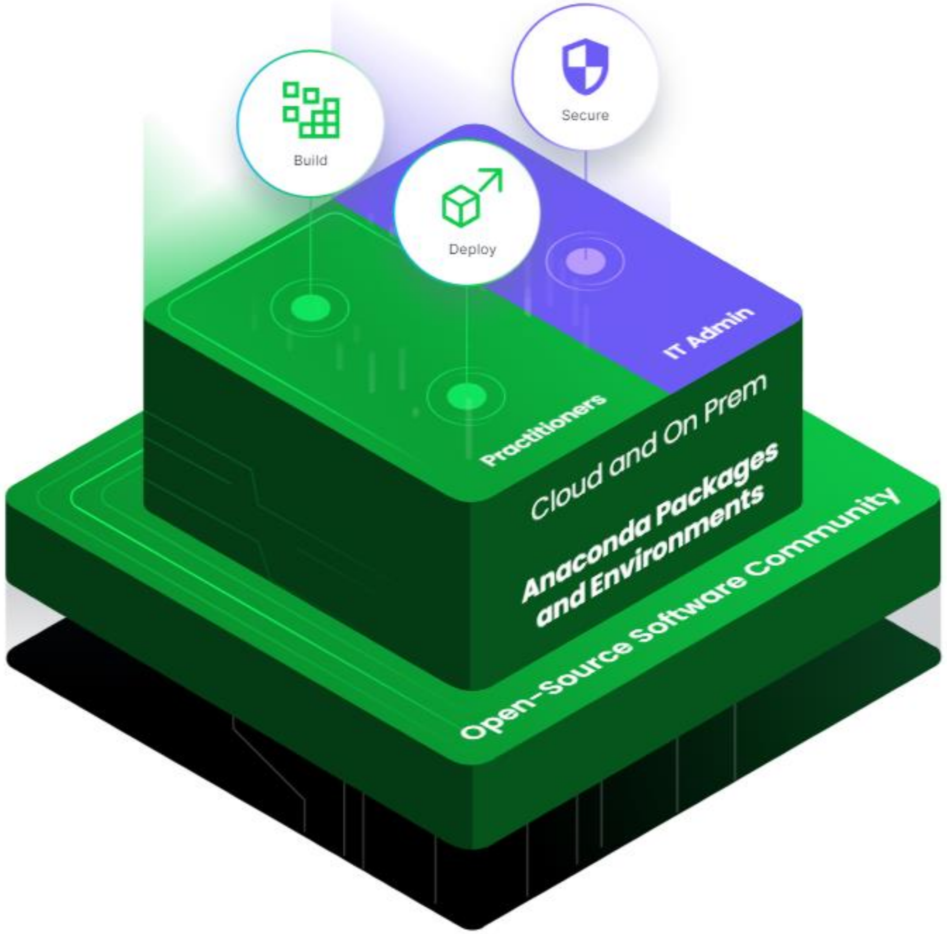
Learn More >

Introduction to Anaconda Platform (3)

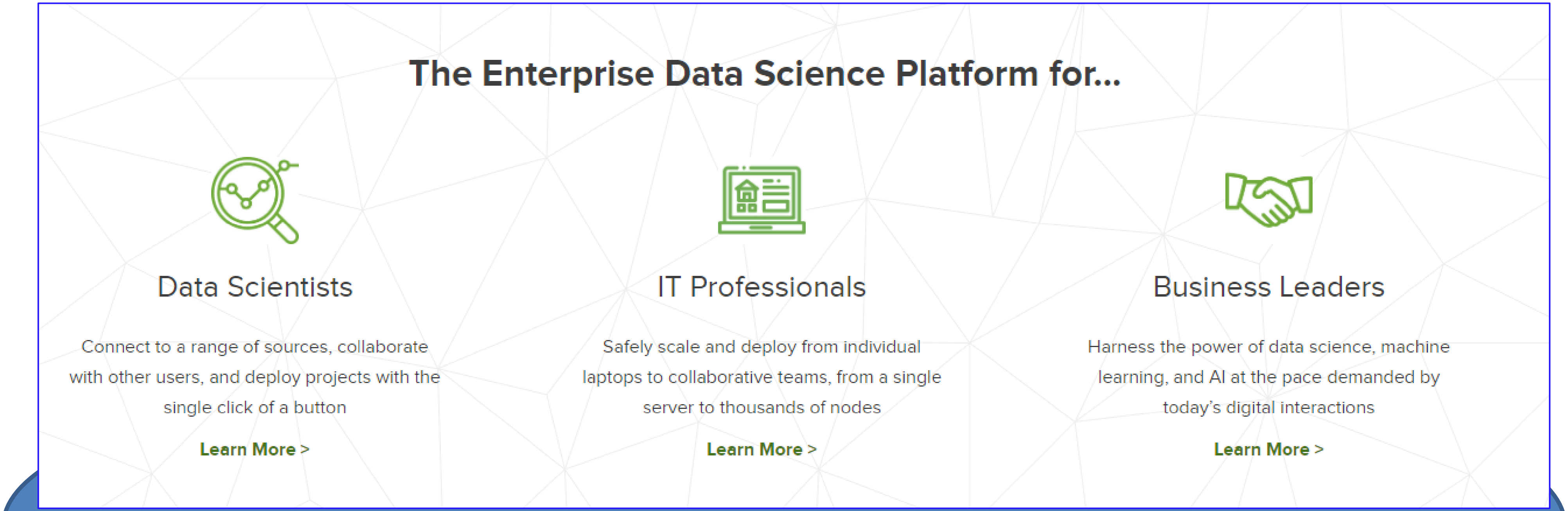
Build, Iterate, and Deploy at Scale

Python Users
Move from ideation to production faster.

Teams Using Python
Achieve scale, collaboration, and rapid deployment.


Build
Secure
Deploy
IT Admin
Practitioners
Cloud and On Prem
Anaconda Packages and Environments
Open-Source Software Community

Introduction to Anaconda Platform (4)



The Enterprise Data Science Platform for...

Data Scientists

Connect to a range of sources, collaborate with other users, and deploy projects with the single click of a button

Learn More >

IT Professionals

Safely scale and deploy from individual laptops to collaborative teams, from a single server to thousands of nodes

Learn More >

Business Leaders

Harness the power of data science, machine learning, and AI at the pace demanded by today's digital interactions

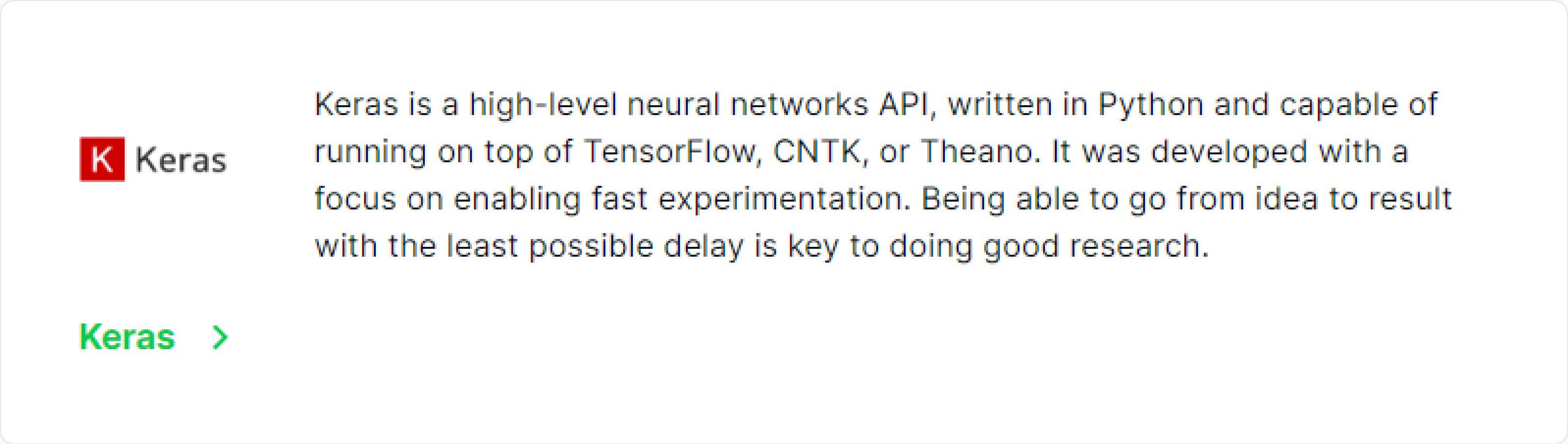
Learn More >

Anaconda platform

Introduction to Anaconda Platform (5)

❖ Some Resources

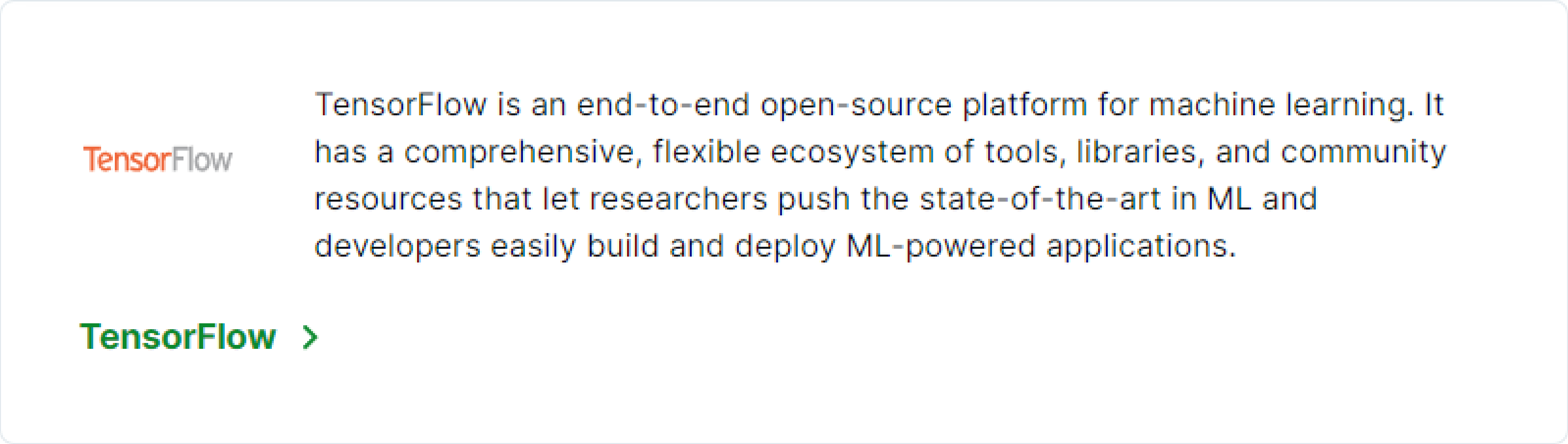
- Machine Learning



Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Keras ›

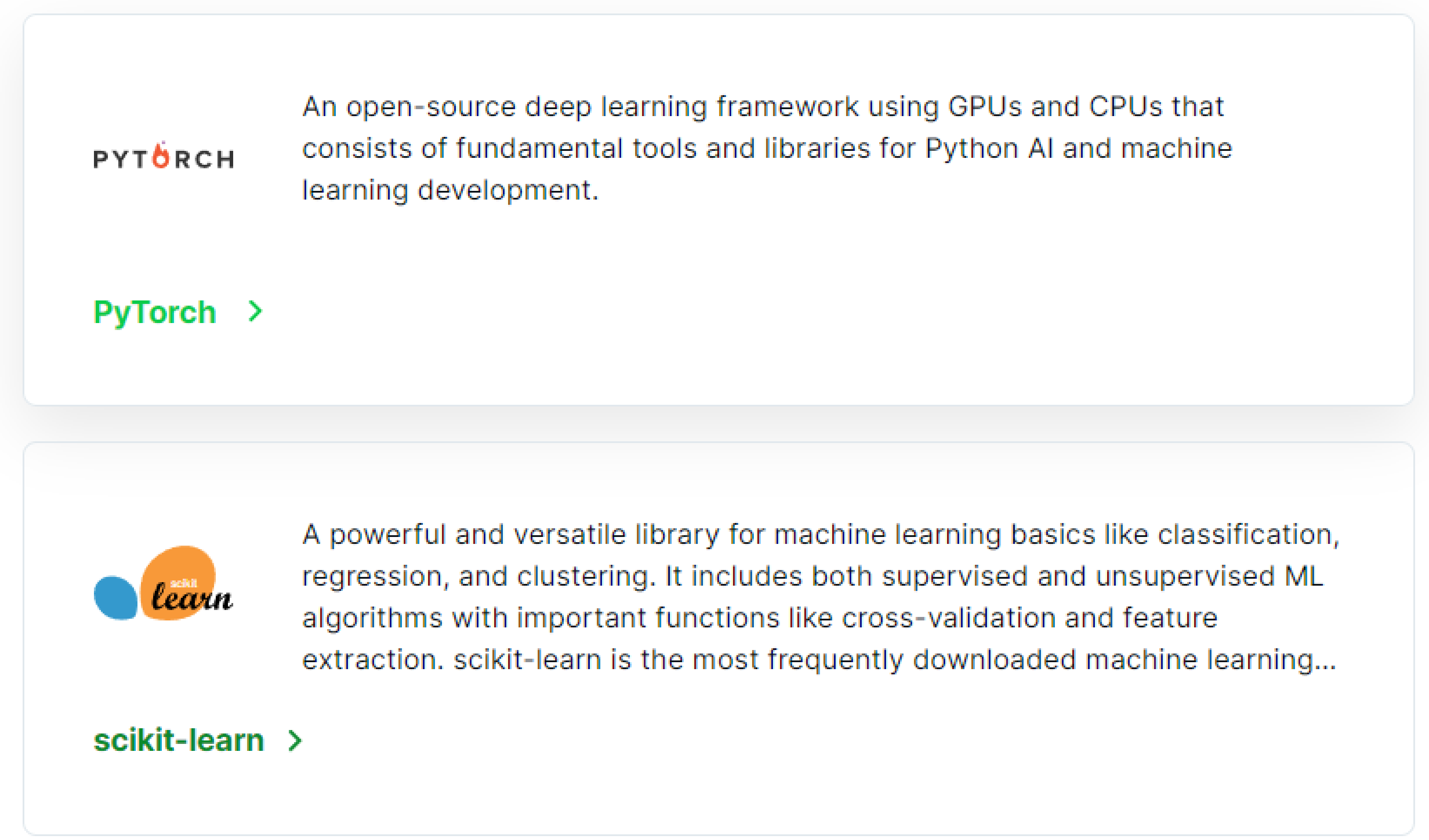


TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that let researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

TensorFlow ›

Introduction to Anaconda Platform (6)



An open-source deep learning framework using GPUs and CPUs that consists of fundamental tools and libraries for Python AI and machine learning development.

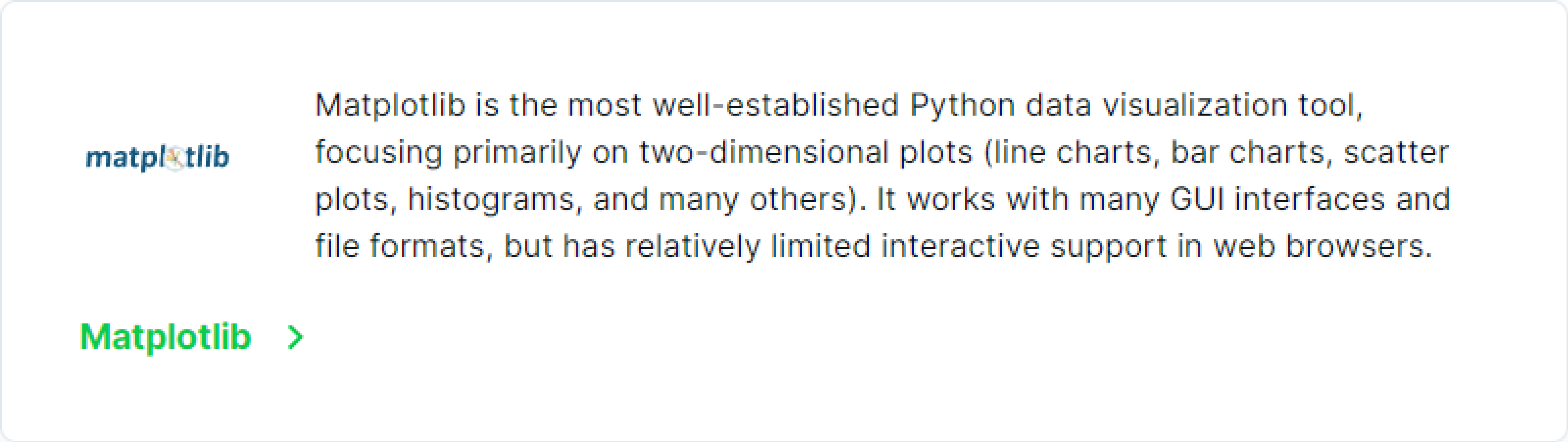
PyTorch >

A powerful and versatile library for machine learning basics like classification, regression, and clustering. It includes both supervised and unsupervised ML algorithms with important functions like cross-validation and feature extraction. scikit-learn is the most frequently downloaded machine learning...

scikit-learn >

Introduction to Anaconda Platform (7)

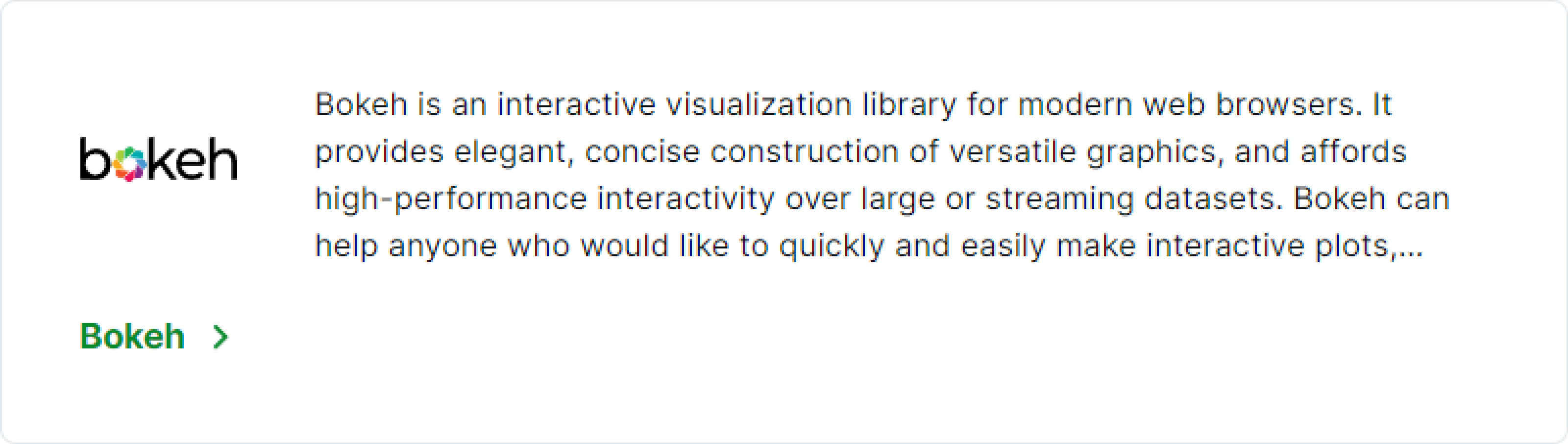
- **Data Visualization**



matplotlib

Matplotlib is the most well-established Python data visualization tool, focusing primarily on two-dimensional plots (line charts, bar charts, scatter plots, histograms, and many others). It works with many GUI interfaces and file formats, but has relatively limited interactive support in web browsers.

Matplotlib >

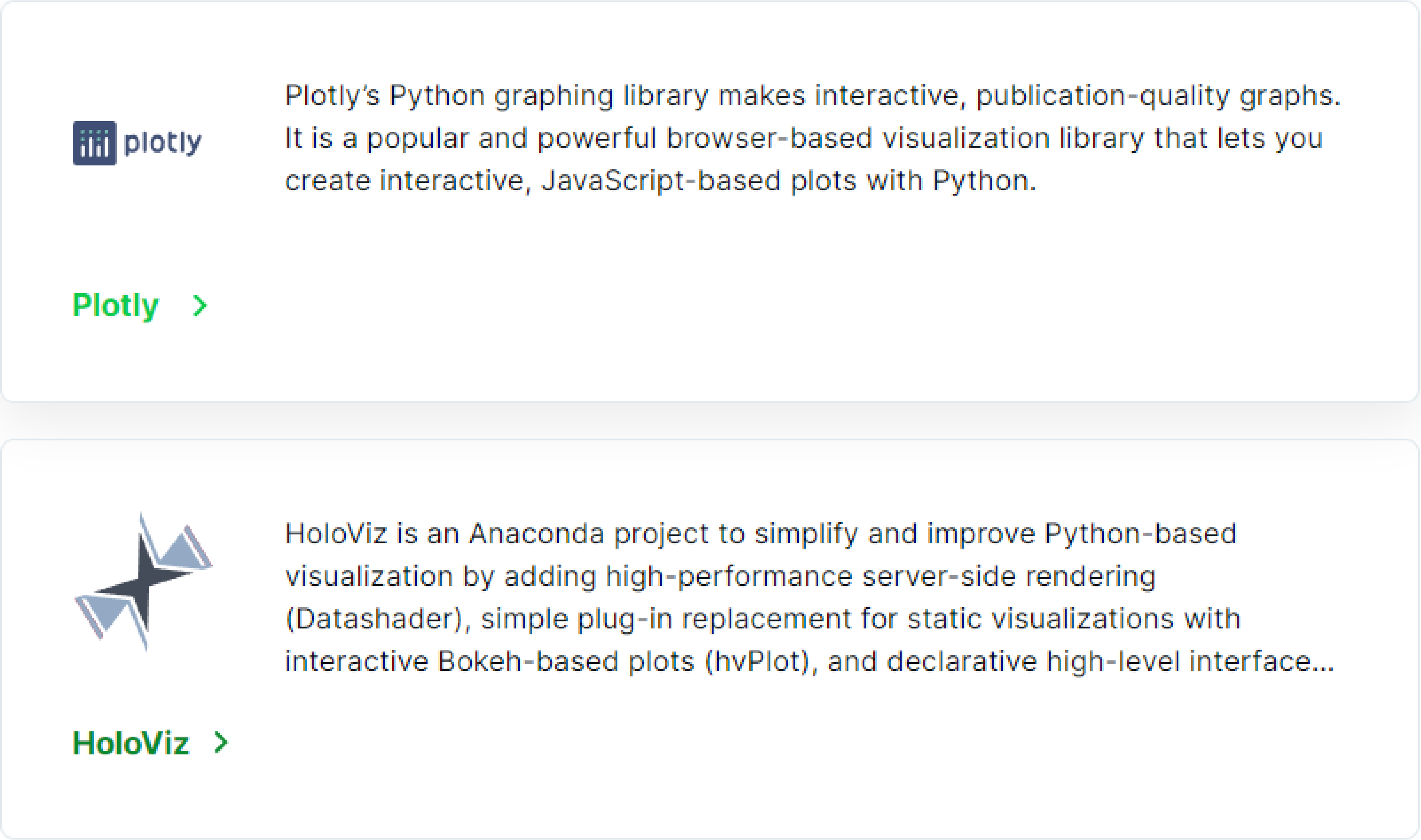


bokeh

Bokeh is an interactive visualization library for modern web browsers. It provides elegant, concise construction of versatile graphics, and affords high-performance interactivity over large or streaming datasets. Bokeh can help anyone who would like to quickly and easily make interactive plots,...

Bokeh >

Introduction to Anaconda Platform (8)

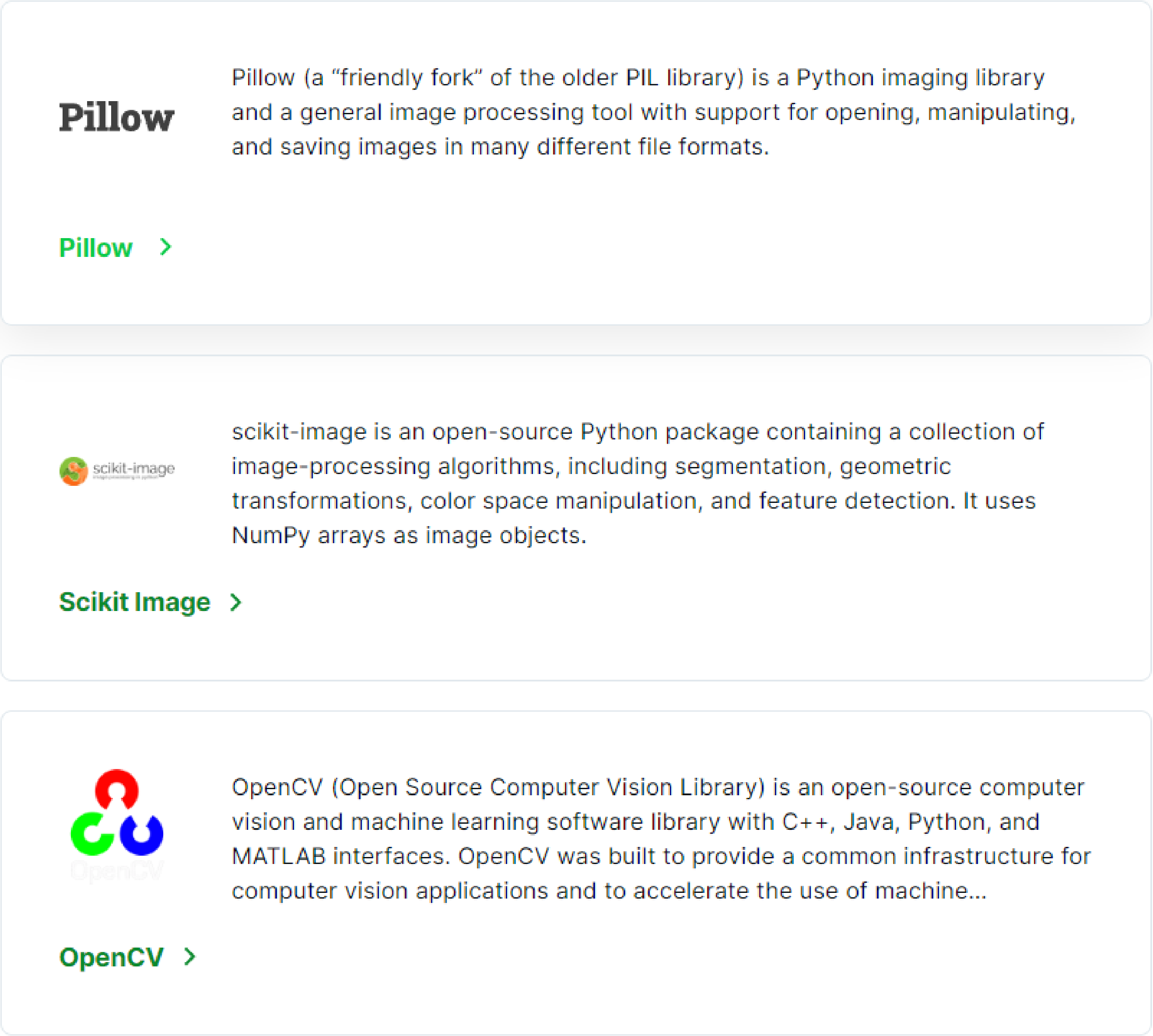


Plotly's Python graphing library makes interactive, publication-quality graphs. It is a popular and powerful browser-based visualization library that lets you create interactive, JavaScript-based plots with Python.

HoloViz is an Anaconda project to simplify and improve Python-based visualization by adding high-performance server-side rendering (Datashader), simple plug-in replacement for static visualizations with interactive Bokeh-based plots (hvPlot), and declarative high-level interface...

Introduction to Anaconda Platform (9)

- **Image Processing**



Pillow (a “friendly fork” of the older PIL library) is a Python imaging library and a general image processing tool with support for opening, manipulating, and saving images in many different file formats.

Pillow >

scikit-image is an open-source Python package containing a collection of image-processing algorithms, including segmentation, geometric transformations, color space manipulation, and feature detection. It uses NumPy arrays as image objects.

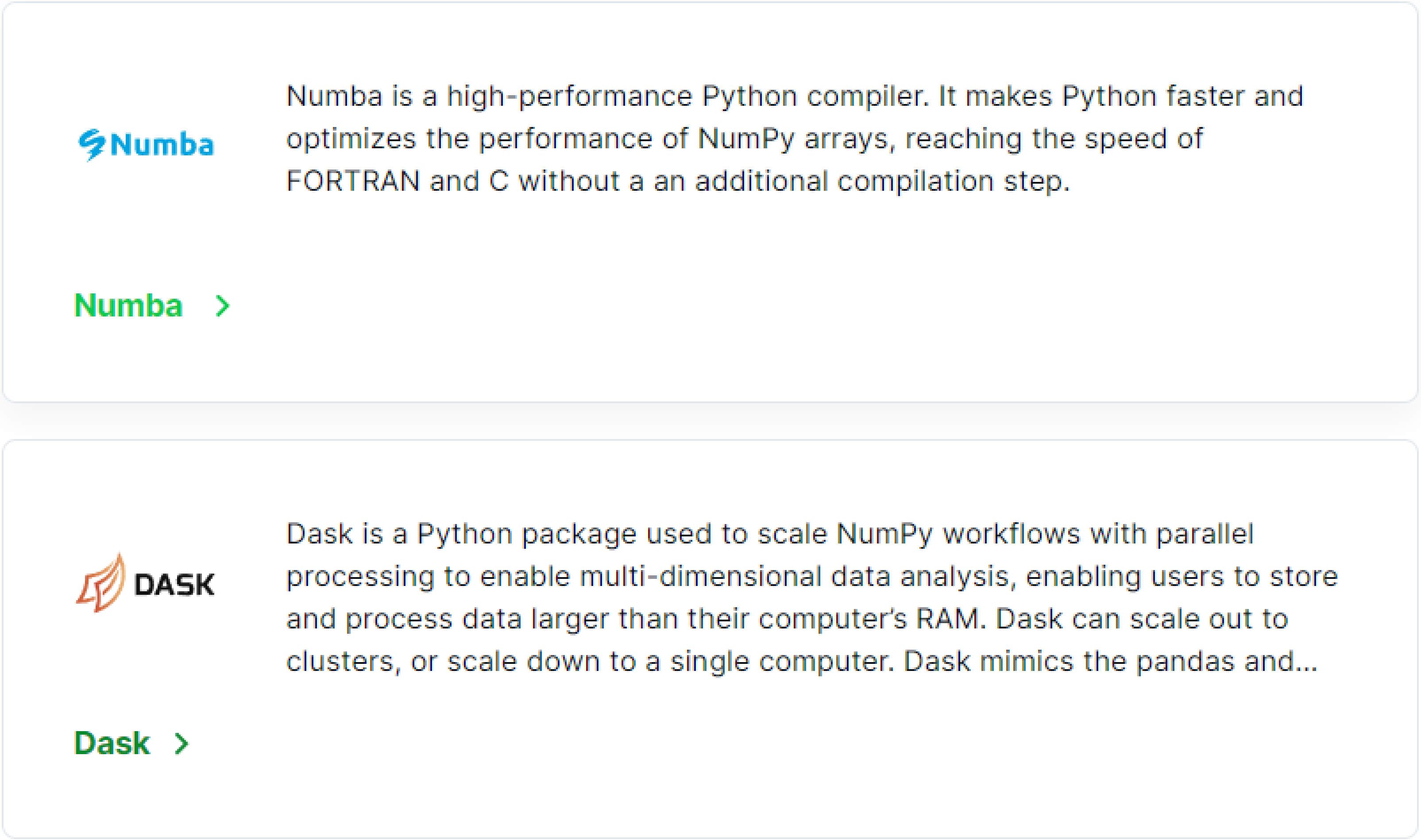
Scikit Image >

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library with C++, Java, Python, and MATLAB interfaces. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine...

OpenCV >

Introduction to Anaconda Platform (10)

- **Scalable Computing**



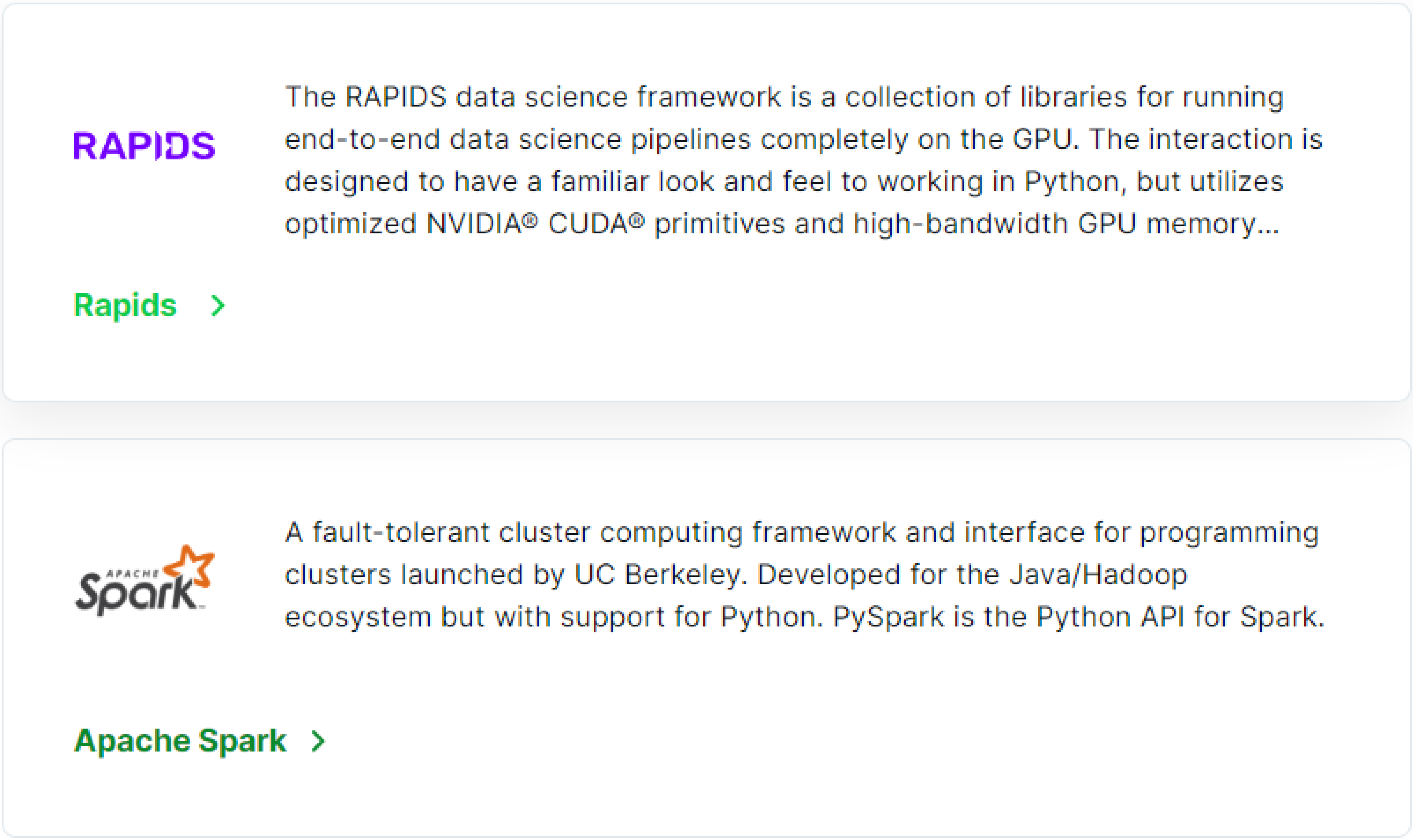
Numba is a high-performance Python compiler. It makes Python faster and optimizes the performance of NumPy arrays, reaching the speed of FORTRAN and C without a an additional compilation step.

Numba >

Dask is a Python package used to scale NumPy workflows with parallel processing to enable multi-dimensional data analysis, enabling users to store and process data larger than their computer's RAM. Dask can scale out to clusters, or scale down to a single computer. Dask mimics the pandas and...

Dask >

Introduction to Anaconda Platform (11)

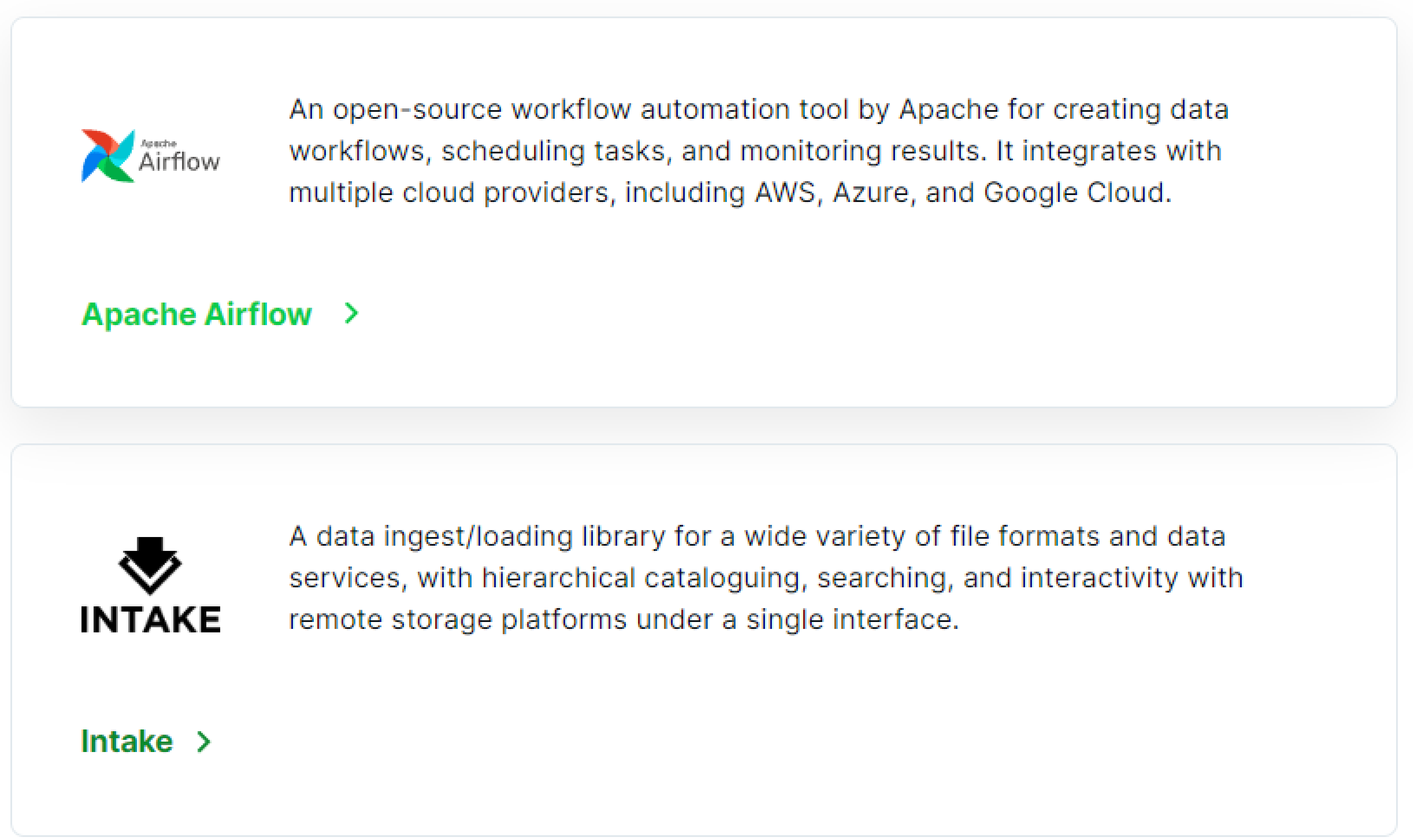


The RAPIDS data science framework is a collection of libraries for running end-to-end data science pipelines completely on the GPU. The interaction is designed to have a familiar look and feel to working in Python, but utilizes optimized NVIDIA® CUDA® primitives and high-bandwidth GPU memory...

A fault-tolerant cluster computing framework and interface for programming clusters launched by UC Berkeley. Developed for the Java/Hadoop ecosystem but with support for Python. PySpark is the Python API for Spark.

Introduction to Anaconda Platform (12)

- **Data Pipelines / ETL**



An open-source workflow automation tool by Apache for creating data workflows, scheduling tasks, and monitoring results. It integrates with multiple cloud providers, including AWS, Azure, and Google Cloud.

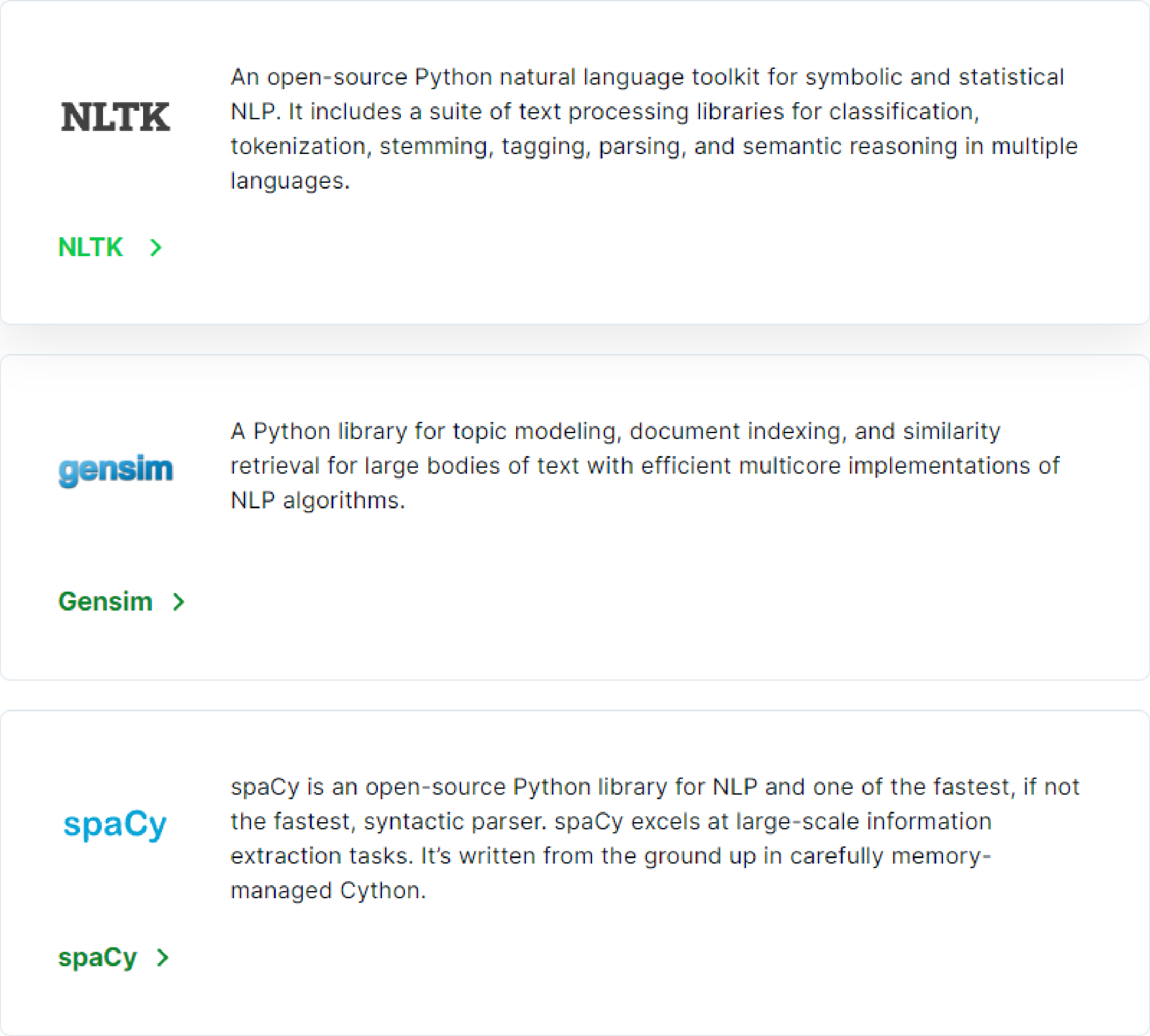
Apache Airflow >

A data ingest/loading library for a wide variety of file formats and data services, with hierarchical cataloguing, searching, and interactivity with remote storage platforms under a single interface.

Intake >

Introduction to Anaconda Platform (13)

- **Natural Language Processing (NLP)**



NLTK

An open-source Python natural language toolkit for symbolic and statistical NLP. It includes a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning in multiple languages.

NLTK ›

gensim

A Python library for topic modeling, document indexing, and similarity retrieval for large bodies of text with efficient multicore implementations of NLP algorithms.

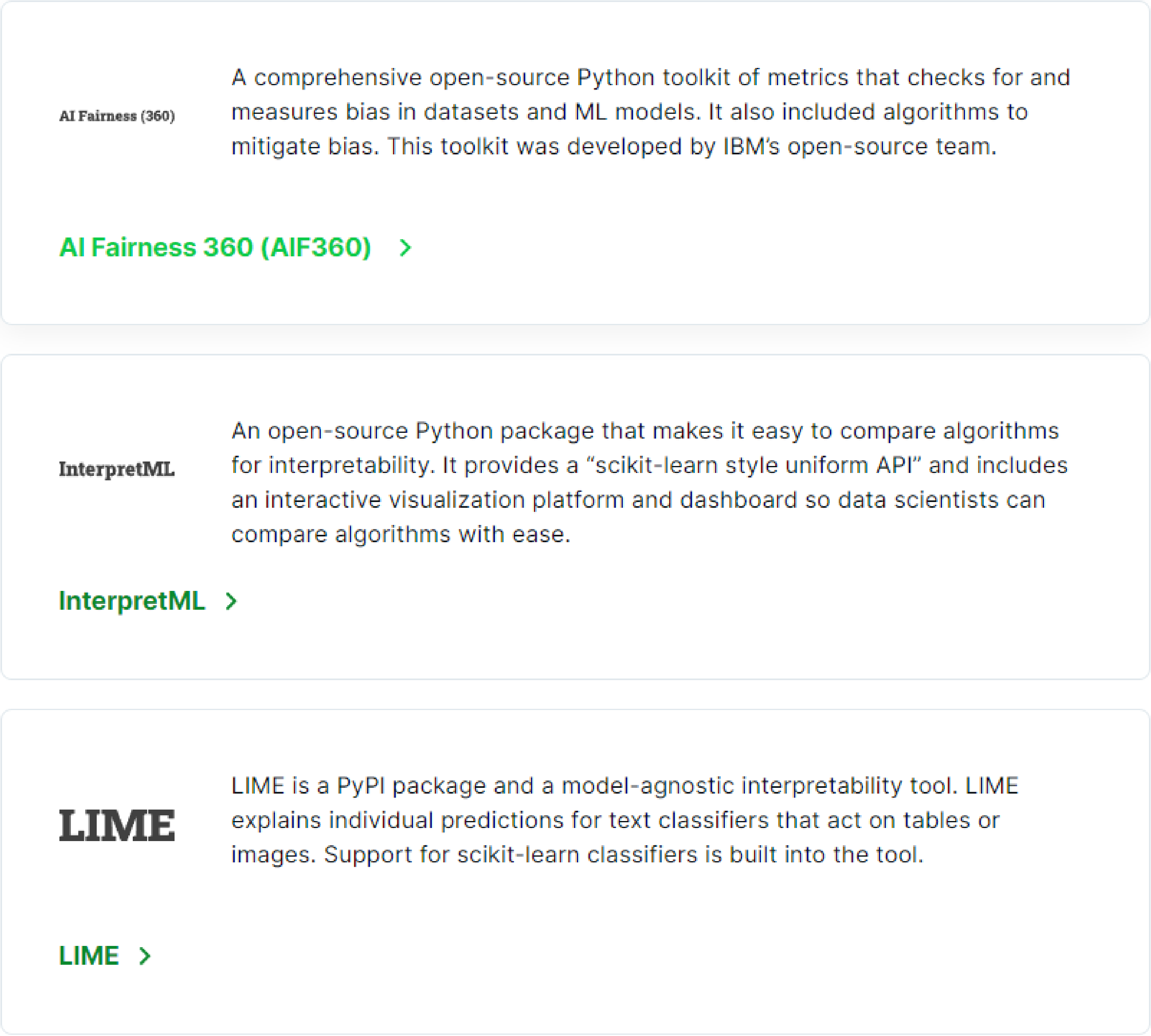
Gensim ›

spaCy

spaCy is an open-source Python library for NLP and one of the fastest, if not the fastest, syntactic parser. spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython.

spaCy ›

Introduction to Anaconda Platform (14)



AI Fairness (360)

A comprehensive open-source Python toolkit of metrics that checks for and measures bias in datasets and ML models. It also included algorithms to mitigate bias. This toolkit was developed by IBM's open-source team.

AI Fairness 360 (AIF360) >

InterpretML

An open-source Python package that makes it easy to compare algorithms for interpretability. It provides a “scikit-learn style uniform API” and includes an interactive visualization platform and dashboard so data scientists can compare algorithms with ease.

InterpretML >

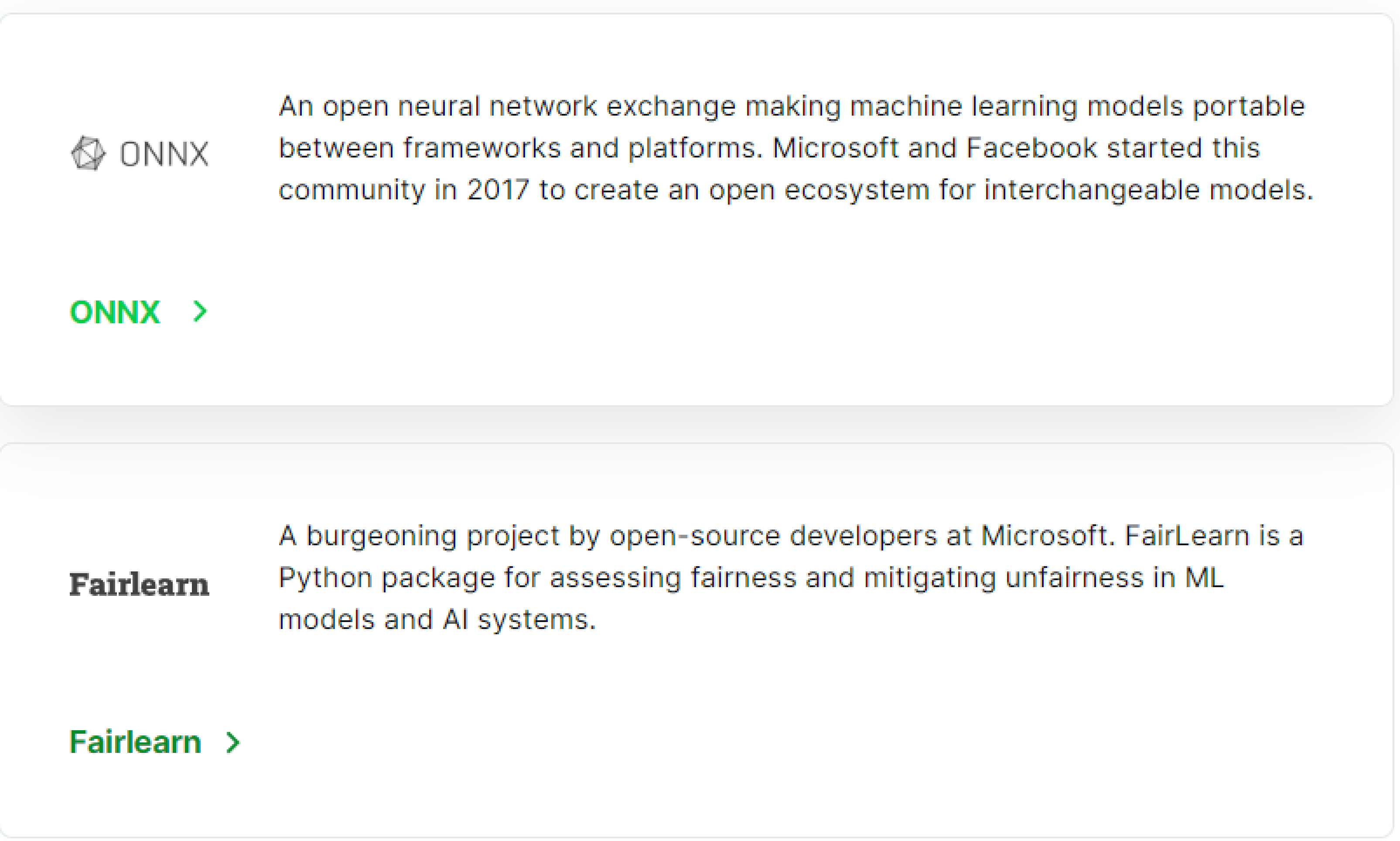
LIME

LIME is a PyPI package and a model-agnostic interpretability tool. LIME explains individual predictions for text classifiers that act on tables or images. Support for scikit-learn classifiers is built into the tool.

LIME >

Introduction to Anaconda Platform (15)

- **Some others**



ONNX

An open neural network exchange making machine learning models portable between frameworks and platforms. Microsoft and Facebook started this community in 2017 to create an open ecosystem for interchangeable models.

ONNX >

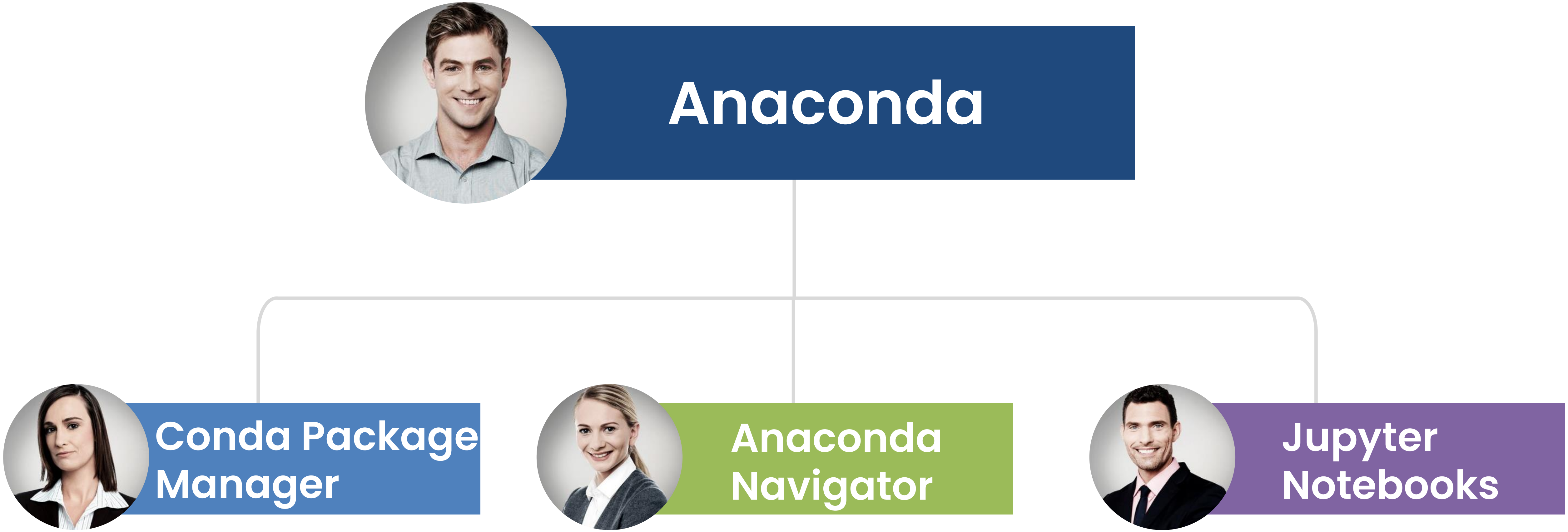
Fairlearn

A burgeoning project by open-source developers at Microsoft. FairLearn is a Python package for assessing fairness and mitigating unfairness in ML models and AI systems.

Fairlearn >

Anaconda Components (1)

❖ Components in Anaconda Platform



Anaconda

Conda Package Manager

Anaconda Navigator

Jupyter Notebooks

Anaconda Components (1)

❖ Conda Package Manager

- **Conda** is an open-source, cross-platform, language-agnostic **package manager** and environment management system.

  가상환경을 만들고 관리함

- Description: Package manager for managing software environments and dependencies.
- Command : “conda create”, “conda install”, “conda update” and etc.

  가상환경 만든 이후
  메모리에 올려 활성화
  (activation)

  모든 가상환경에
  적용하고 싶을 때

<Updating Conda>

```
$ conda update -n base conda
```

```
$ conda install -n base conda=22.11.1
$ conda update conda
```

Anaconda Components (2)

$텍스트 기반으로 명령어 수행

what packages are installed by running:

```
$ conda list
```

설치한 것 보여줌

to see all the packages that are available, use:

```
$ conda search
```

to install a package

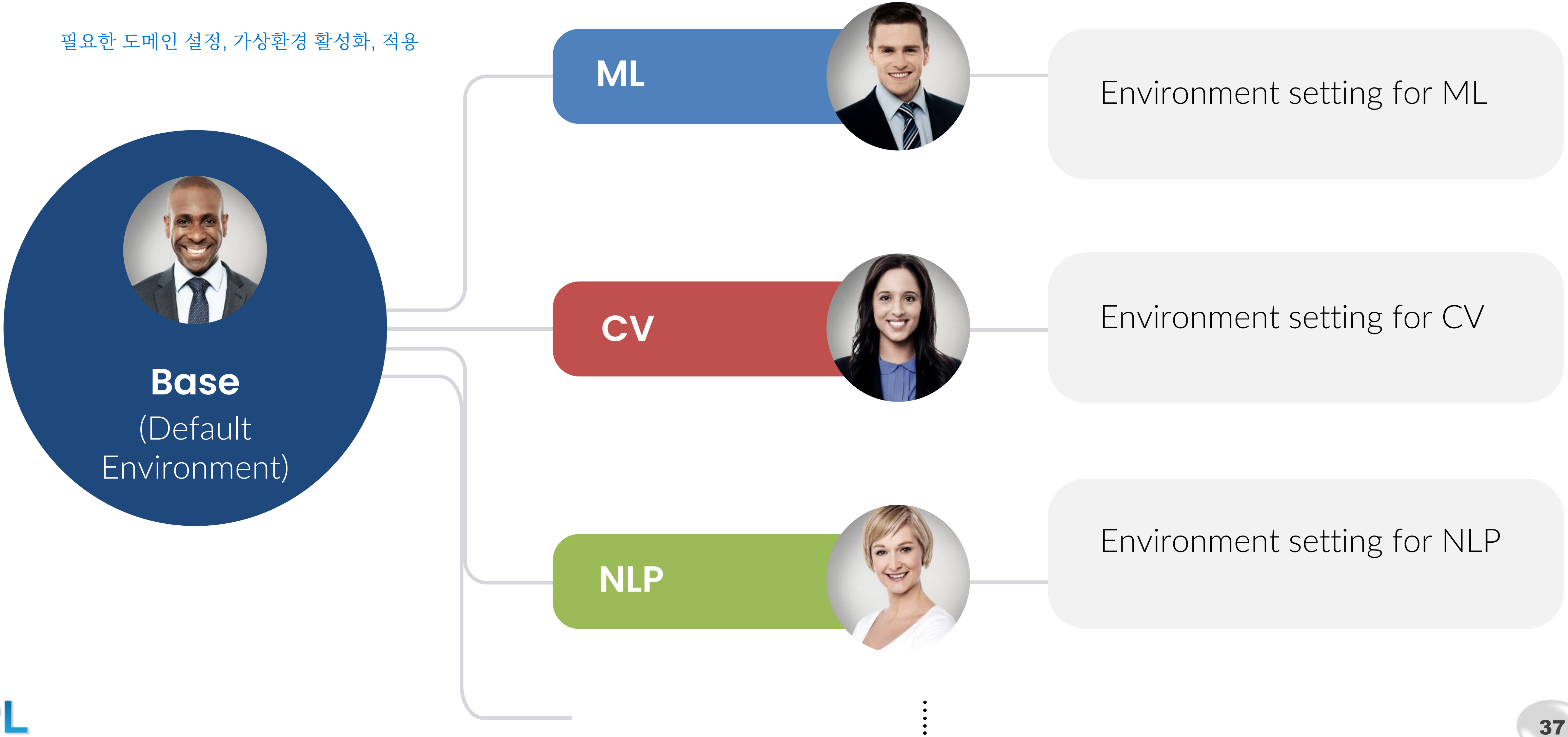
```
$ conda install <package-name>
```

필요한 모듈, 라이브러리 install

Anaconda Components (3)

- The real power of conda comes from its **ability to manage environments**.
  - In **conda**, an environment can be thought of as **a completely separate installation**.

필요한 도메인 설정, 가상환경 활성화, 적용



Base
(Default Environment)
ML
CV
NLP
Environment setting for ML
Environment setting for CV
Environment setting for NLP
⋮

Anaconda Components (4)

Create new environment:

```
$ conda create --name ml-project pytorch    python == 3.8
$ conda create –name <name> package name or version
```

지정 안하면 최신버전이 default

activate this environment:

```
$ conda activate ml-project
```

To go back to the **base environment:**

```
$ conda deactivate
```

To remove the existing environment:

```
$ conda remove --name <name> --all    가상환경 삭제
```

To see the existing environments:

```
$ conda env list    모든 가상환경 보여줌
```

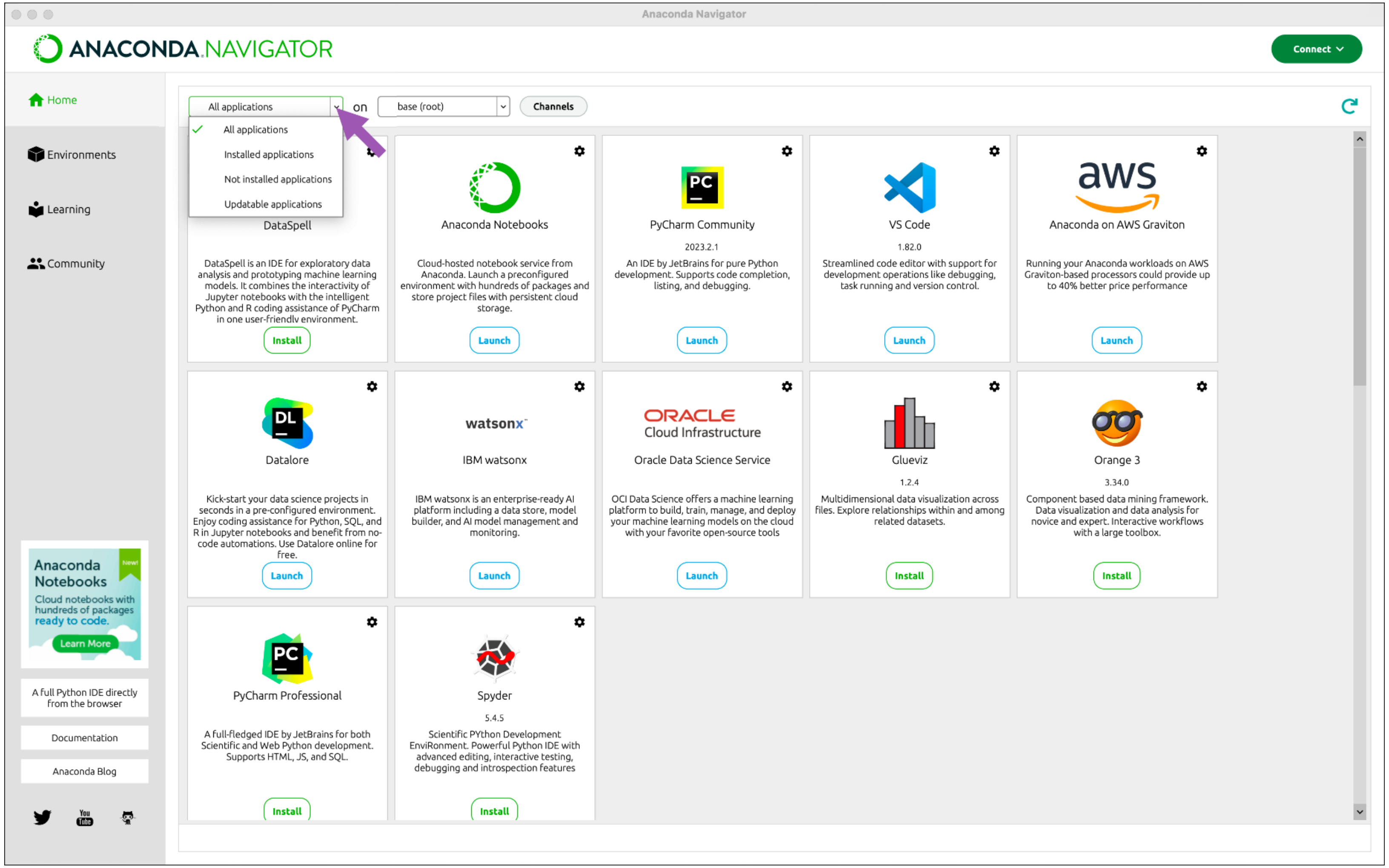
To see the conda information:

```
$ conda info
```

Anaconda Components (5)

❖ Anaconda Navigator

- Overview: a graphical desktop application (GUI) that enables you to use conda without having to run commands at the command line.



Anaconda Notebooks: Cloud notebooks with hundreds of packages ready to code.

A full Python IDE directly from the browser

DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install

Anaconda Notebooks
Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
Launch

PyCharm Community
2023.2.1
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.
Launch

VS Code
1.82.0
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch

Anaconda on AWS Graviton
Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance
Launch

Datalore
Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
Launch

IBM watsonx
IBM watsonx is an enterprise-ready AI platform including a data store, model builder, and AI model management and monitoring.
Launch

Oracle Data Science Service
OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools
Launch

Glueviz
1.2.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.34.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

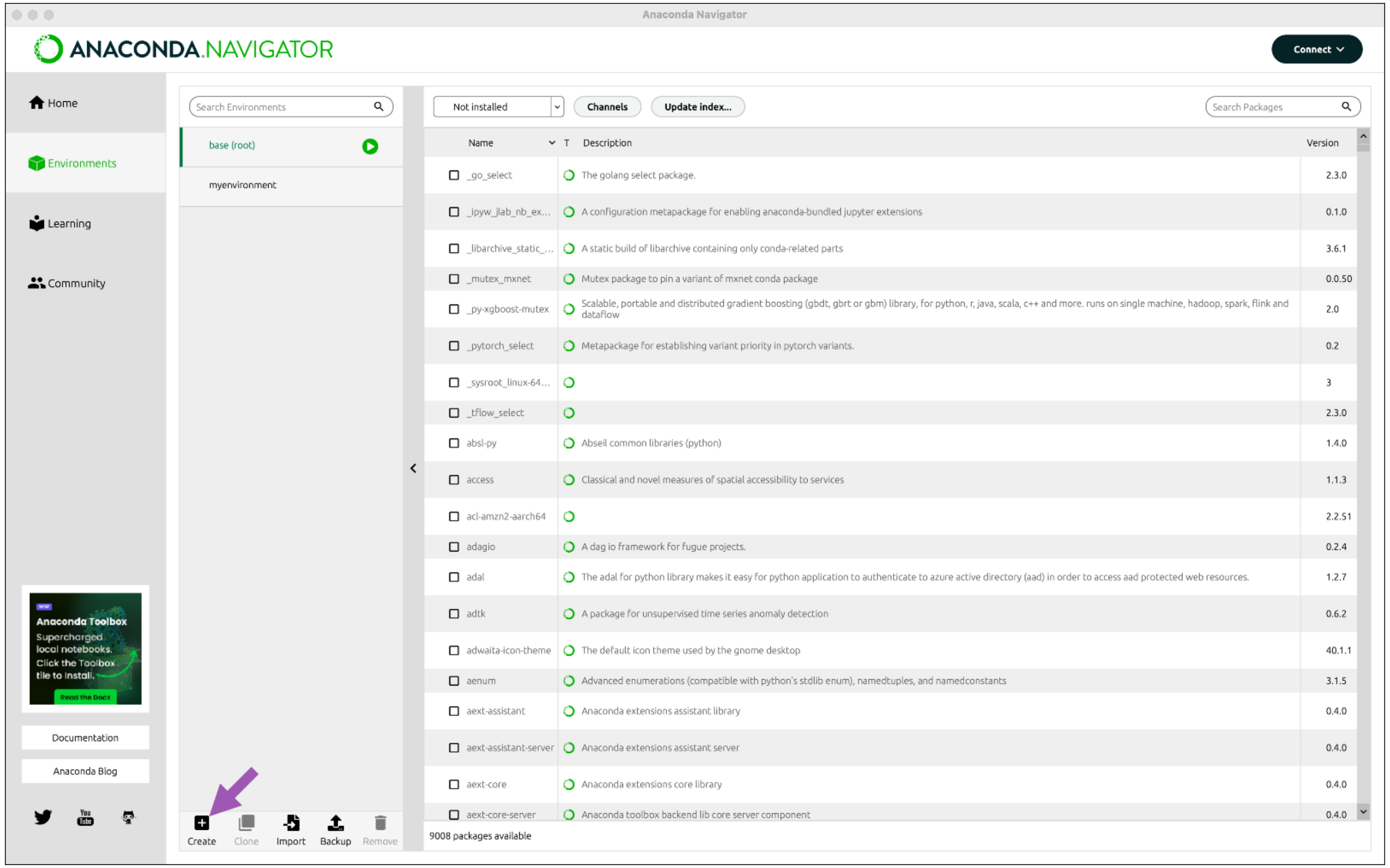
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install

Spyder
5.4.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Install

Anaconda Components (6)

- Managing Environments

1. On the **Environments** page, click **Create**.
2. In the **Environment** name field, type a descriptive name for your environment.
3. Click **Create**. Navigator creates the new environment and activates it.
4. Now you have two environments: the default environment base (root) and myenvironment.

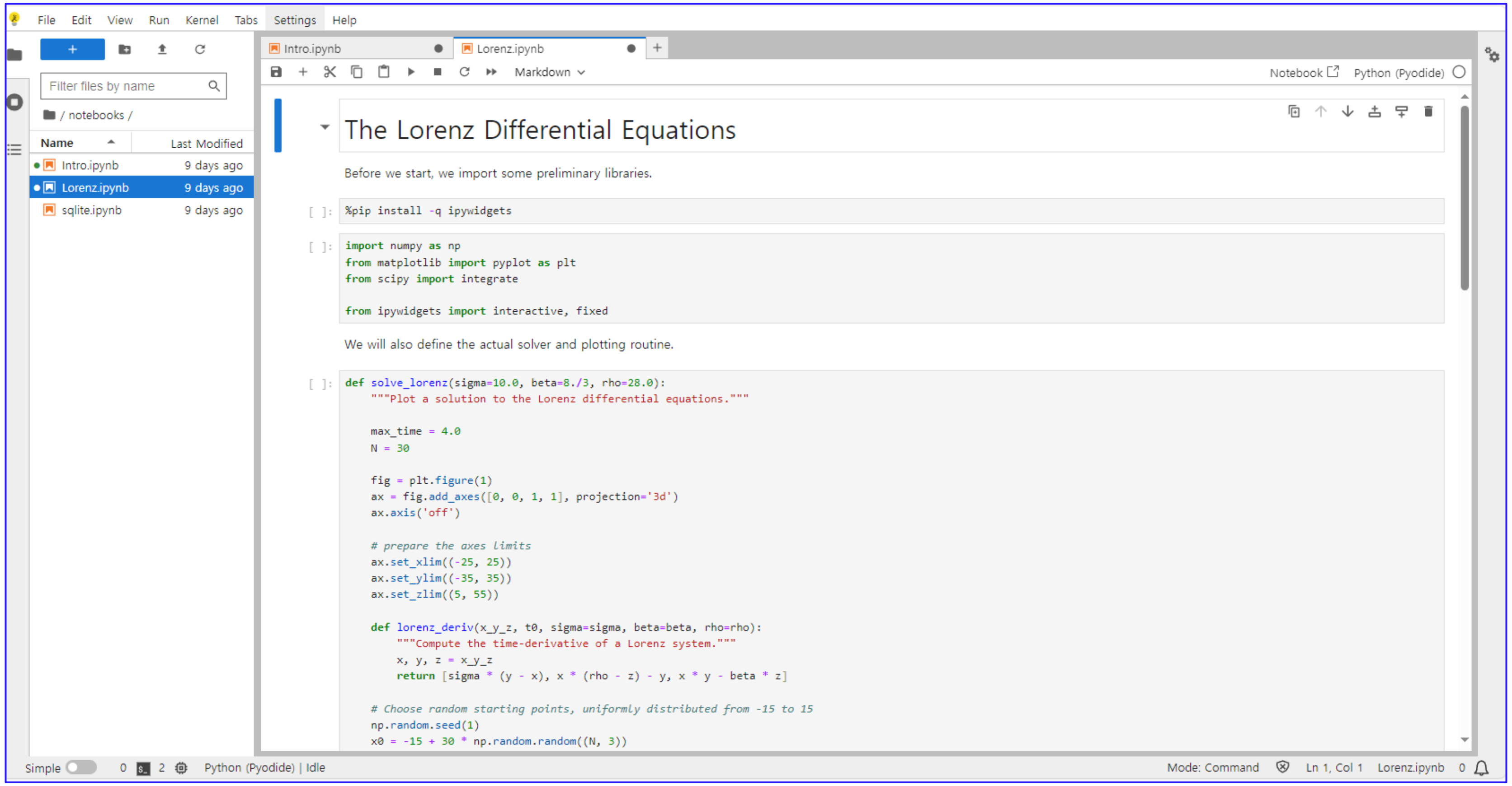


Name	T	Description	Version
□ _go_select	○	The golang select package.	2.3.0
□ _ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
□ _libarchive_static_...	○	A static build of libarchive containing only conda-related parts	3.6.1
□ _mutex_mxnet	○	Mutex package to pin a variant of mxnet conda package	0.0.50
□ _py-xgboost-mutex	○	Scalable, portable and distributed gradient boosting (gbdt, gbrt or gbm) library, for python, r, java, scala, c++ and more. runs on single machine, hadoop, spark, flink and dataflow	2.0
□ _pytorch_select	○	Metapackage for establishing variant priority in pytorch variants.	0.2
□ _sysroot_linux-64...	○		3
□ _tflow_select	○		2.3.0
□ absl-py	○	Abseil common libraries (python)	1.4.0
□ access	○	Classical and novel measures of spatial accessibility to services	1.1.3
□ acl-amzn2-aarch64	○		2.2.51
□ adagio	○	A dag io framework for fugue projects.	0.2.4
□ adal	○	The adal for python library makes it easy for python application to authenticate to azure active directory (aad) in order to access aad protected web resources.	1.2.7
□ adtk	○	A package for unsupervised time series anomaly detection	0.6.2
□ adwaita-icon-theme	○	The default icon theme used by the gnome desktop	40.1.1
□ aenum	○	Advanced enumerations (compatible with python's stdlib enum), namedtuples, and namedconstants	3.1.5
□ aext-assistant	○	Anaconda extensions assistant library	0.4.0
□ aext-assistant-server	○	Anaconda extensions assistant server	0.4.0
□ aext-core	○	Anaconda extensions core library	0.4.0
□ aext-core-server	○	Anaconda toolbox backend lib core server component	0.4.0

9008 packages available

Anaconda Components (7)

- **Jupyter Notebooks** (JupyterLab) (https://jupyter.org/)
  - Open Standards for Interactive Computing
  - web-based interactive development environment for notebooks, code, and data.
    - UI: web-based
    - Workflow: server-based



The Lorenz Differential Equations

Before we start, we import some preliminary libraries.

```
[ ]: %pip install -q ipywidgets
```

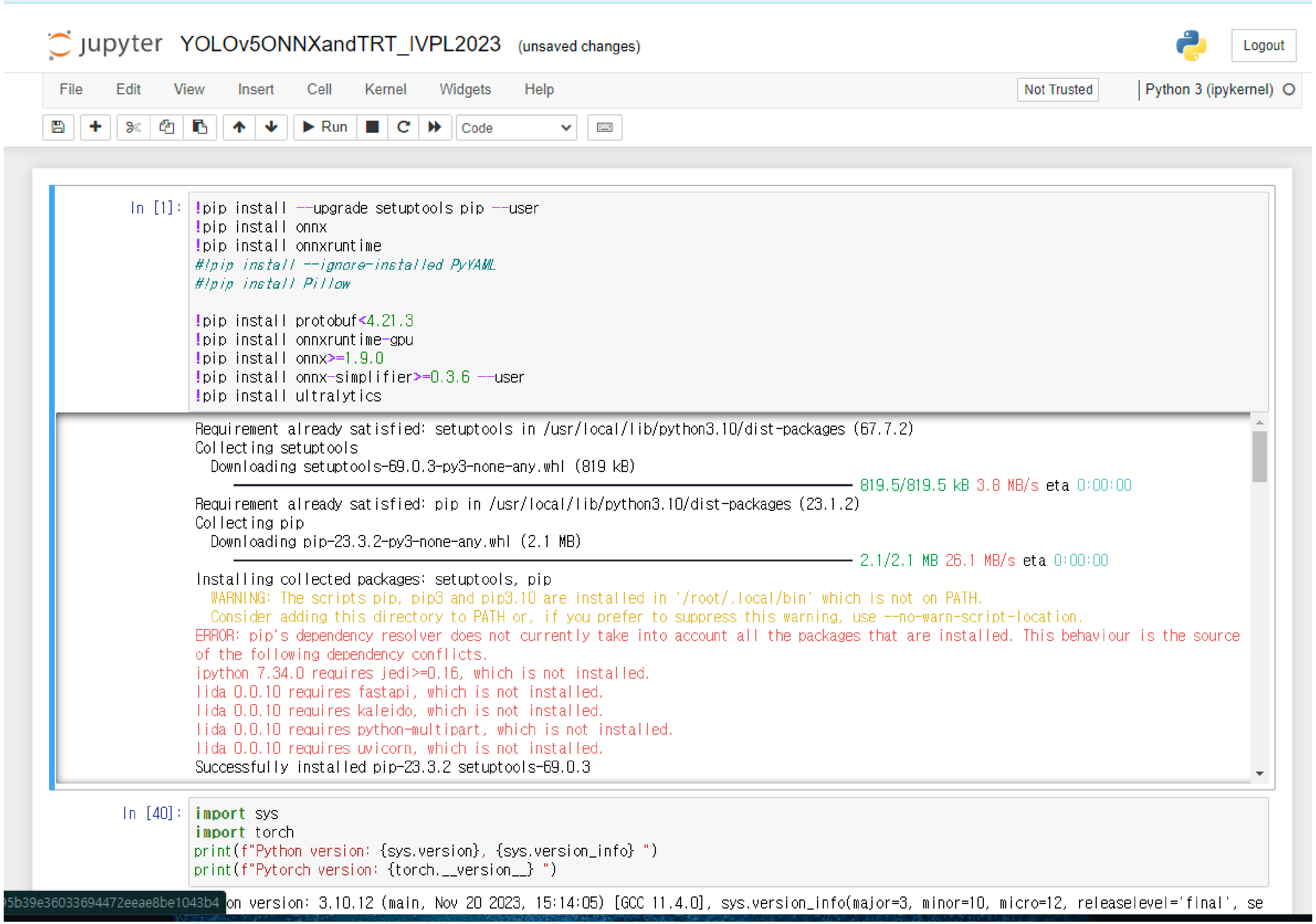
```
[ ]: import numpy as np
     from matplotlib import pyplot as plt
     from scipy import integrate

     from ipywidgets import interactive, fixed
```

We will also define the actual solver and plotting routine.

```
[ ]: def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
         """Plot a solution to the Lorenz differential equations."""

         max_time = 4.0
         N = 30

         fig = plt.figure(1)
         ax = fig.add_axes([0, 0, 1, 1], projection='3d')
         ax.axis('off')

         # prepare the axes limits
         ax.set_xlim((-25, 25))
         ax.set_ylim((-35, 35))
         ax.set_zlim((5, 55))

         def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
             """Compute the time-derivative of a Lorenz system."""
             x, y, z = x_y_z
             return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

         # Choose random starting points, uniformly distributed from -15 to 15
         np.random.seed(1)
         x0 = -15 + 30 * np.random.random((N, 3))
```

IVPL
Intelligent Vision Processing Lab

Anaconda Components (8)



jupyter YOLOv5ONNXandTRT_IVPL2023 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel) ○
Run
Code

In [1]:

```
!pip install --upgrade setuptools pip --user
!pip install onnx
!pip install onnxruntime
#!pip install --ignore-installed PyYAML
#!pip install Pillow

!pip install protobuf<4.21.3
!pip install onnxruntime-gpu
!pip install onnx>=1.9.0
!pip install onnx-simplifier>=0.3.6 --user
!pip install ultralytics
```

```
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (67.7.2)
Collecting setuptools
  Downloading setuptools-69.0.3-py3-none-any.whl (819 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 819.5/819.5 kB 3.8 MB/s eta 0:00:00
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (23.1.2)
Collecting pip
  Downloading pip-23.3.2-py3-none-any.whl (2.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 26.1 MB/s eta 0:00:00
Installing collected packages: setuptools, pip
  WARNING: The scripts pip, pip3 and pip3.10 are installed in '/root/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
of the following dependency conflicts.
ipython 7.34.0 requires jedi>=0.16, which is not installed.
lida 0.0.10 requires fastapi, which is not installed.
lida 0.0.10 requires kaleido, which is not installed.
lida 0.0.10 requires python-multipart, which is not installed.
lida 0.0.10 requires uvicorn, which is not installed.
Successfully installed pip-23.3.2 setuptools-69.0.3
```

In [40]:

```
import sys
import torch
print(f"Python version: {sys.version}, {sys.version_info} ")
print(f"Pytorch version: {torch.__version__} ")
```

```
5b39e36033694472eeae8be1043b4 on version: 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0], sys.version_info(major=3, minor=10, micro=12, releaselevel='final', se
```

IVPL Intelligent Vision Processing Lab

Working with Environments (1)

❖ Workingflow

- **1) Creating Environments:**
  - **Command: conda create --name myenv python=3.8**
  - **Purpose: Isolating project dependencies and versions.**

가상환경 생성
이름 만들기, 파이썬 버전 주기

- **2) Activating and Deactivating Environments:**
  - Commands: conda activate **myenv**, conda deactivate
  - Explanation: Switching between different environment contexts.

대ㅂ부분 pip 사용ㅂ부분 pip 사용

Package Management (1)

❖ Installing Packages:

- Command: “conda install pandas”
- Example: Installing the Pandas library for data manipulation.

```
$ conda install <package name>
```

❖ Updating and Removing Packages:

- Commands: “conda update numpy”, “conda remove matplotlib”
- Best Practices: Keeping packages up-to-date and managing dependencies.

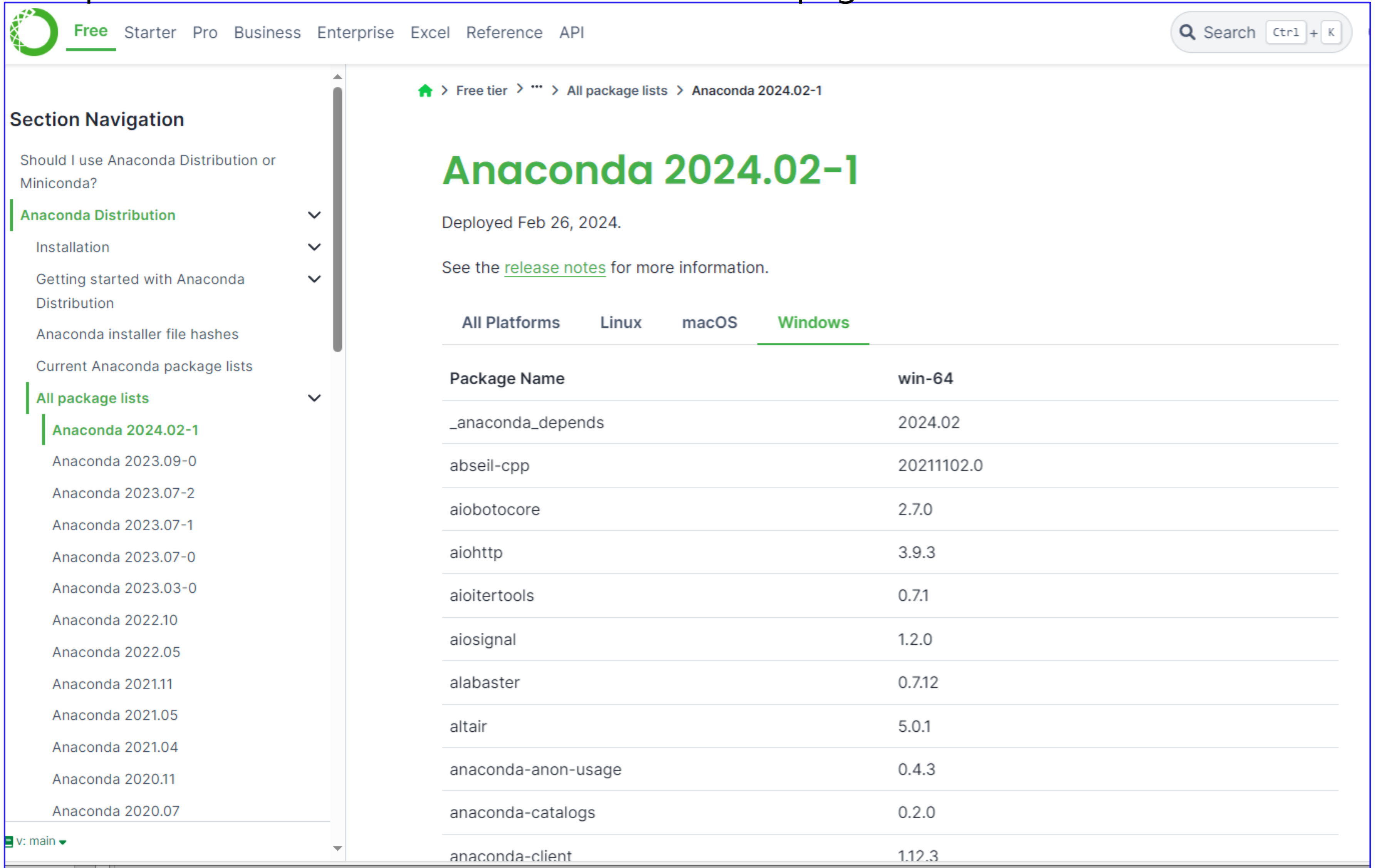
```
$ conda update <package name>
```

```
$ conda remove <package name>
```

Package Management (2)

❖ All lists of Conda packages

* https://docs.anaconda.com/free/anaconda/allpkglists/2024.02-1/



Anaconda 2024.02-1

Deployed Feb 26, 2024.

See the release notes for more information.

Package Name	win-64
_anaconda_depends	2024.02
abseil-cpp	20211102.0
aiobotocore	2.7.0
aiohttp	3.9.3
aioitertools	0.7.1
aiosignal	1.2.0
alabaster	0.7.12
altair	5.0.1
anaconda-anon-usage	0.4.3
anaconda-catalogs	0.2.0
anaconda-client	1.12.3

Anaconda and Data Science Libraries (1)

- Popular Libraries (Packages) in Anaconda:
  - **NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, Pillow.**
  - Description: Core libraries for data manipulation, analysis, and machine learning.

Package	Description
NumPy	Python library that provides a **multidimensional array object**, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
Pandas	It has functions for **analyzing, cleaning, exploring, and manipulating data**.
Matplotlib	a comprehensive library for **creating static, animated, and interactive visualizations in Python.**
Scikit-learn	**a free machine learning library for Python.** It supports both supervised and unsupervised **machine learning,** providing diverse algorithms for classification, regression, clustering, and dimensionality reduction.

Anaconda and Data Science Libraries (2)

❖ Additional Libraries:

- **TensorFlow, PyTorch, NLTK, OpenCV**. 🡐 Specialized tools
- Integration: Installing and managing additional libraries for specialized tasks.

Package	Description
TensorFlow	an open-source library that provides software engineers and data scientists with a tool to build, train, and work with deep learning models in order to draw predictions from data (by Google).
PyTorch	an open-source machine learning library developed using Torch library for python programs. It was developed by Facebook's AI Research lab and released in January 2016 as a free and open-source library mainly used in computer vision, deep learning, and natural language processing applications.
NLTK	**Natural Language Toolkit** (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP)
OpenCV	**OpenCV** (Open Source Computer Vision) is a **programming library** designed for **real-time computer vision** tasks. Originally developed by Intel, it provides a comprehensive set of tools and algorithms for image processing, video manipulation, object detection, and more. OpenCV is **open source**, meaning it is freely available for use and modification by developers and researchers.

Integration with IDEs

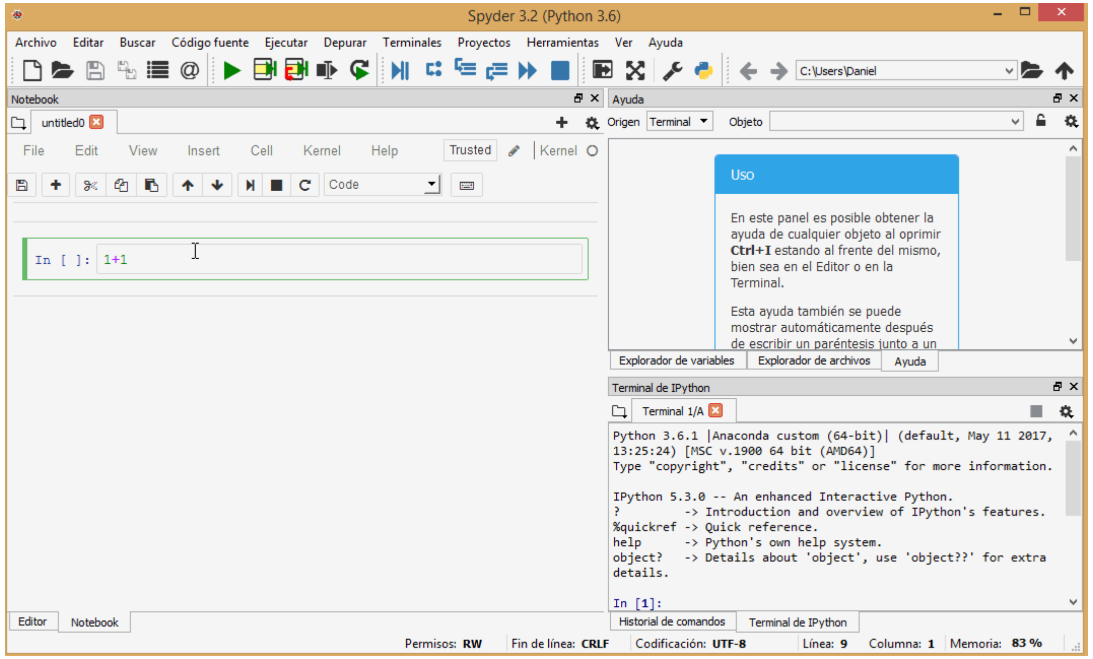
❖ **Jupyter** Integration:

- Opening Jupyter Notebooks in Anaconda Navigator.
- Running Jupyter from the command line.

❖ **IDEs** for Data Science:

free and open-source scientific environment written in **Python**, designed by and for **scientists, engineers, and data analysts**

- Anaconda supports IDEs like Jupyter, Spyder, and Visual Studio Code.
- Configuring and using these IDEs for efficient development.



```
In [ ]: 1+1
```

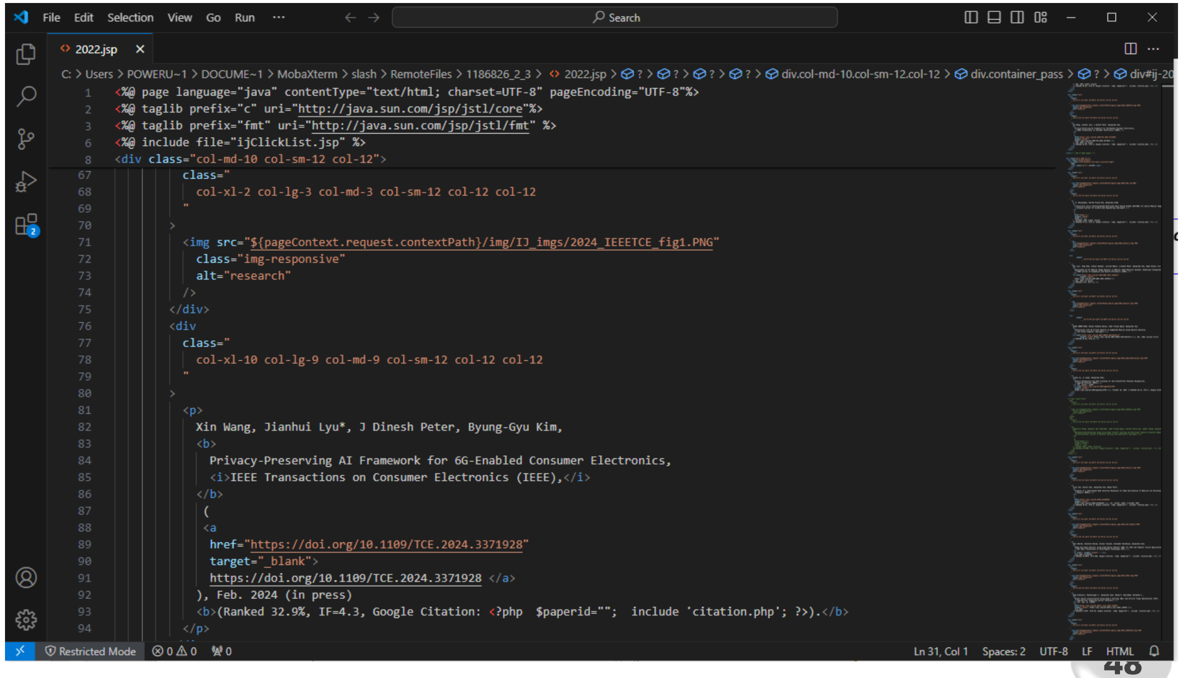
Uso

En este panel es posible obtener la ayuda de cualquier objeto al oprimir **Ctrl+I** estando al frente del mismo, bien sea en el Editor o en la Terminal.

Esta ayuda también se puede mostrar automáticamente después de escribir un paréntesis junto a un

```
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017,
13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra
details.

In [1]:
```

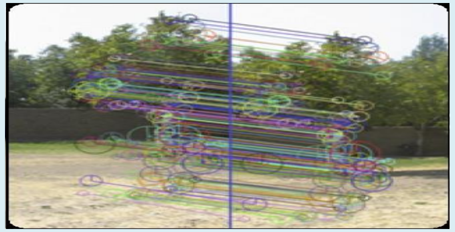
Permisos: RW Fin de línea: CRLF Codificación: UTF-8 Línea: 9 Columna: 1 Memoria: 83 %

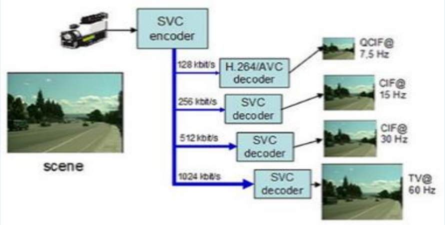

File Edit Selection View Go Run ···
Search
2022.jsp
C: > Users > POWERU~1 > DOCUME~1 > MobaXterm > slash > RemoteFiles > 1186826_2_3 > 2022.jsp > ? > ? > ? > ? > div.col-md-10.col-sm-12.col-12 > div.container_pass > ? > div#ij-20
Restricted Mode
0 0 0

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ include file="ijClickList.jsp" %>
<div class="col-md-10 col-sm-12 col-12">
          class="
            col-xl-2 col-lg-3 col-md-3 col-sm-12 col-12 col-12
          "
        >
          <img src="${pageContext.request.contextPath}/img/IJ_imgs/2024_IEEETCE_fig1.PNG"
            class="img-responsive"
            alt="research"
          />
        </div>
        <div
          class="
            col-xl-10 col-lg-9 col-md-9 col-sm-12 col-12 col-12
          "
        >
          <p>
            Xin Wang, Jianhui Lyu*, J Dinesh Peter, Byung-Gyu Kim,
            <b>
              Privacy-Preserving AI Framework for 6G-Enabled Consumer Electronics,
              <i>IEEE Transactions on Consumer Electronics (IEEE),</i>
            </b>
             (
             <a
              href="https://doi.org/10.1109/TCE.2024.3371928"
              target="_blank">
              https://doi.org/10.1109/TCE.2024.3371928 </a>
            ), Feb. 2024 (in press)
            <b>(Ranked 32.9%, IF=4.3, Google Citation: <?php  $paperid="";  include 'citation.php'; ?>).</b>
          </p>
```

Ln 31, Col 1 Spaces: 2 UTF-8 LF HTML

IVPL
Intelligent Vision Processing Lab





Contents

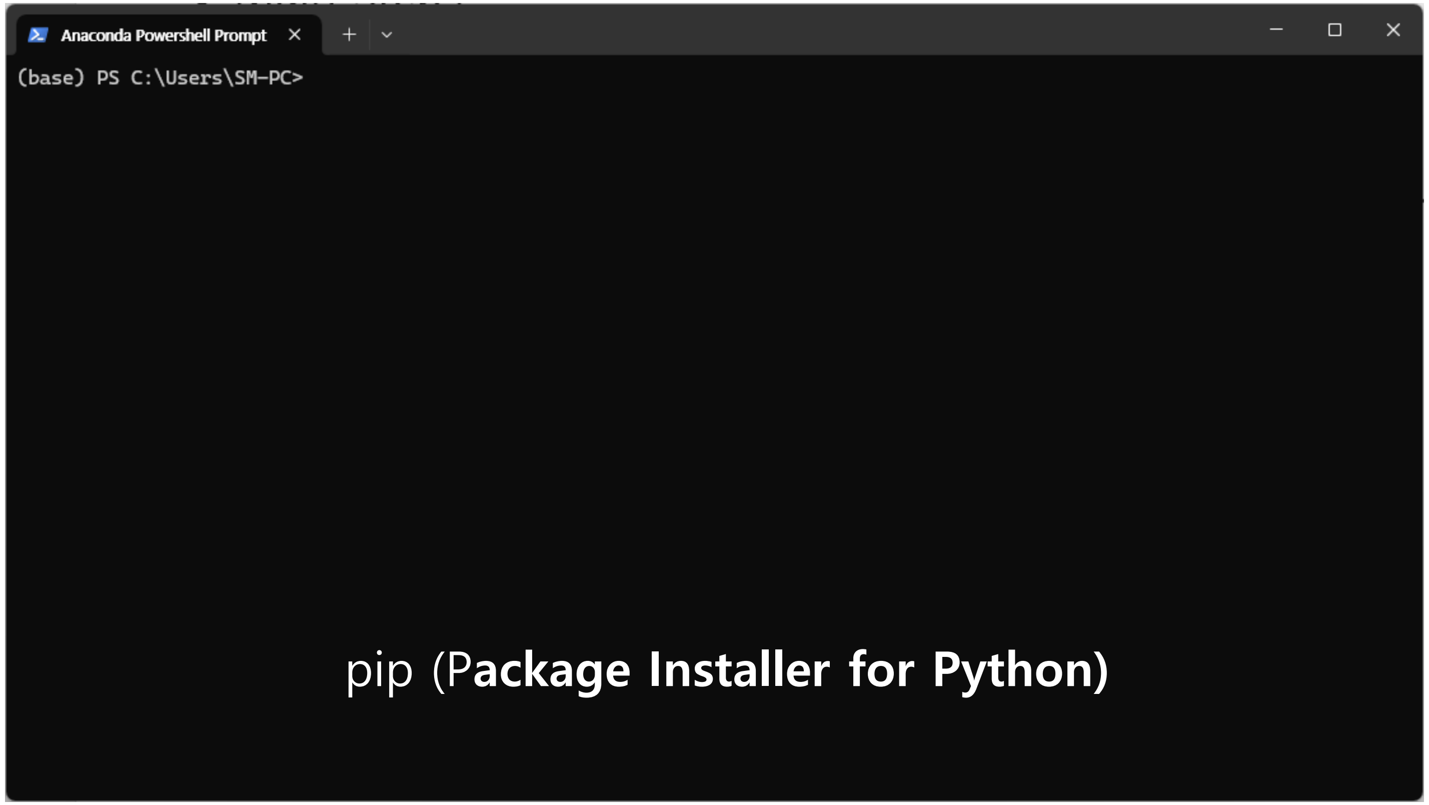
- Examples in Open-source SW (OSS)
- Intro. Of Anaconda Platform
- **Exercise of Anaconda Environment**

Exercise of Anaconda Environment

❖ Main Steps

- 1] Install Anaconda from https://www.anaconda.com.
  - Checking your OS type!

- 2] After finishing the installation, run “Anaconda Powershell Prompt” to take exercise.



```
(base) PS C:\Users\SM-PC>
```

pip (**Package Installer for Python)**

HW#1 – Creation of Your Account in GitHub (가입하여 실습 준비하기)

❖ Contents:

- Make your Github account.

❖ Due:

- ~ before our class on March 25 (Mon.)

Thank you for your attention.!!!
QnA

http://ivpl.sookmyung.ac.kr