

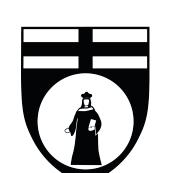
# **Corso di formazione per l'ottimizzazione e la gestione di modelli idrodinamici procedure per l'analisi automatizzata di dati di qualità delle acque marine**

October 2023

Agenzia Regionale per la protezione dell'ambiente ligure

**Andrea Lira Loarca**  
[andrea.lira.loarca@unige.it](mailto:andrea.lira.loarca@unige.it)

[www.meteocean.science](http://www.meteocean.science)  
Dipartimento di Ingegneria Chimica, Civile ed Ambientale  
Università degli studi di Genova



**Università  
di Genova**

**DICCA DIPARTIMENTO  
DI INGEGNERIA CIVILE, CHIMICA  
E AMBIENTALE**



**MeteOcean**  
[meteocean.dicca.unige.it](http://meteocean.dicca.unige.it)  
[meteocean@dicca.unige.it](mailto:meteocean@dicca.unige.it)

# Programma del corso

## Introduzione a Python e Numpy (4 ore)

Venerdì 13 ottobre | 09:00-13:00

- ✓ Introduzione a Python e librerie scientifiche
- ✓ Ambiente di sviluppo (IDE): Spyder, Visual Studio Code, Jupyter notebook
- ✓ Nozioni di base sulla programmazione Python
- ✓ I principali tipi di dato di Python e i loro metodi
- ✓ Introduzione a Numpy
- ✓ Array (1D, 2D, N-D) in Numpy
- ✓ Funzioni matematiche, di ordine e gestione array

## Analisi di serie temporale di dati EMODNET – Pandas e Matplotlib (4 ore)

Lunedì 16 ottobre | 09:00-13:00

- ✓ Lettura files
- ✓ DataFrame e Series e metodi di indicizzazione e modifica per serie temporale
- ✓ Groupby, dati mancanti e operazioni matematiche
- ✓ Gestione delle date e broadcasting
- ✓ Elaborazione dei dati
- ✓ Line plots, scatter plots, bar plots, histograms
- ✓ Scrittura files: csv, json

# Programma del corso

## Analisi dati georeferenziate e temporali Copernicus Marine Service – Xarray, Dask, Cartopy e Pandas (8 ore)

Mercoledì 18 ottobre e Venerdì 20 ottobre 2023 | 09:00-13:00

- ✓ Scaricare dati ed uso Dask
- ✓ Dataarray e Dataset e metodi di indicizzazione e modifica
- ✓ Creazione dataset e dataarray da numpy e pandas
- ✓ Funzioni matematiche, di ordine, gestione e masks
- ✓ NetCDF e GRIB
- ✓ Analisi geospaziale e plots

## Esercitazione su Catena operativa Mar Ligure

Giovedì 26 ottobre | 09:00-13:00

- ✓ Ottimizzazione e gestione della catena operativa del modello idrodinamico Mar Ligure
- ✓ Pianificazione di attività di controllo della catena operativa tramite file batch
- ✓ Gestione dell'archiviazione dei file della catena operativa

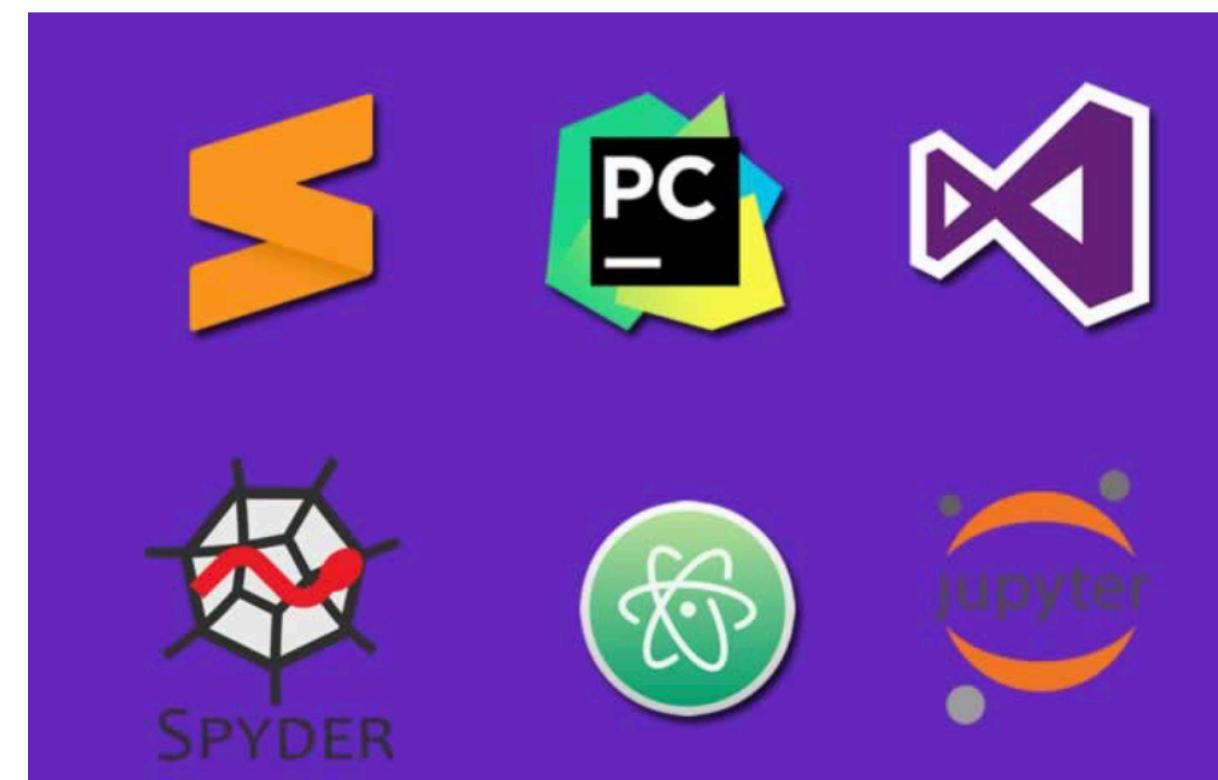
## Esercitazione su file dei monitoraggi marini

Venerdì 27 ottobre | 09:00-13:00

- ✓ Importazione, lettura e modifica di file di testo esistenti
- ✓ Lettura, scrittura ed elaborazioni dati di monitoraggio secondo D.lgs 152/06 e D.lgs n.190/2010 (Decreto Strategia Marina)

# Programmare in Python

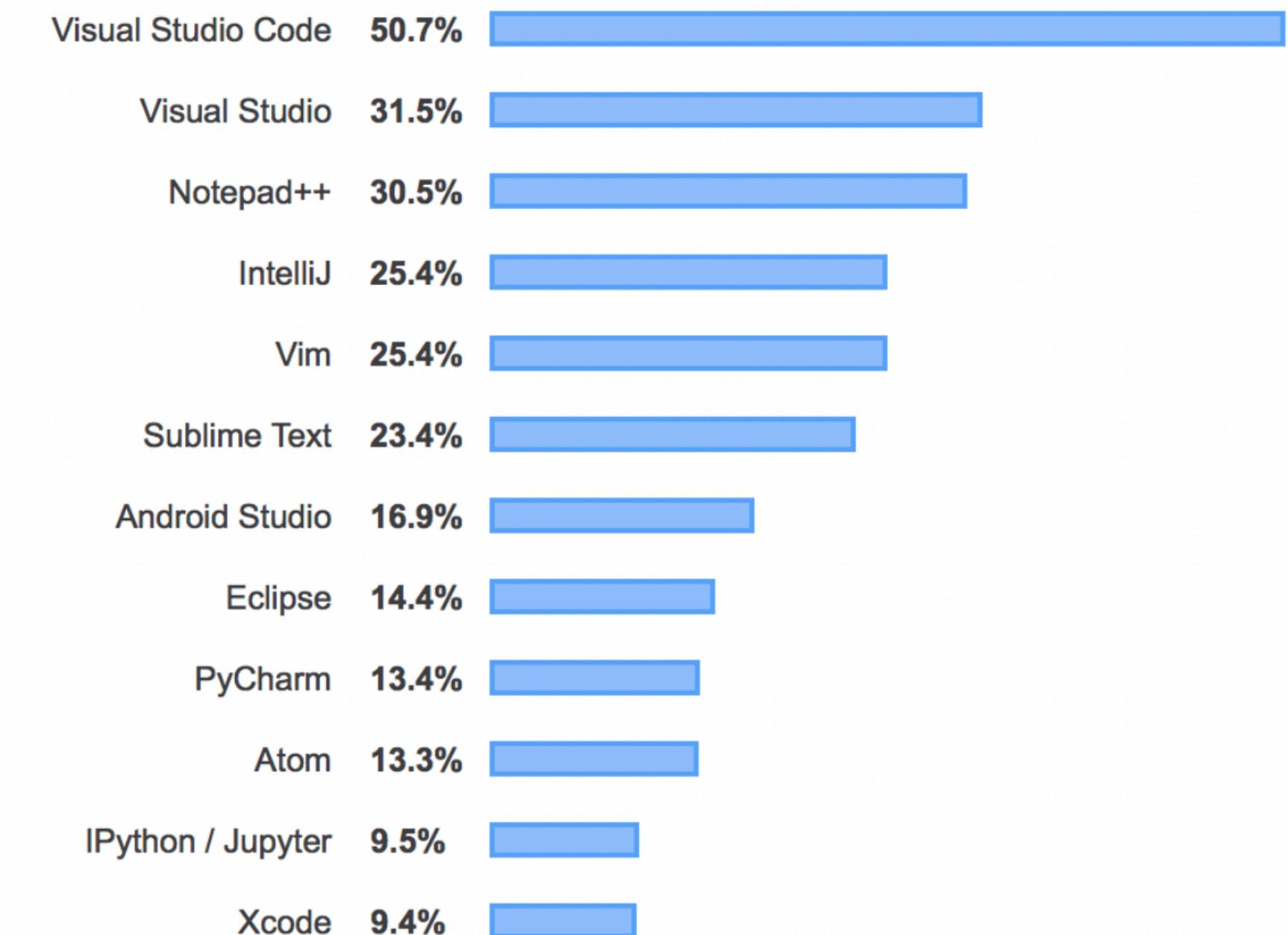
## 1. Distribuzione - librerie, pachetti



## 2. Editore o applicazione

- Notepad++
- Vim
- Sublime Text
- **VS Code**
- **Spyder**
- Atom
- Pycharm
- Jupyter

...



# Distribuzione Python scientifica - Anaconda

[https://www.anaconda.com/  
products/individual](https://www.anaconda.com/products/individual)

The screenshot displays the Anaconda Distribution website's homepage. At the top, the logo "ANACONDA DISTRIBUTION" and the tagline "Most Trusted Distribution for Data Science" are visible. Below this, three main sections are highlighted: "ANACONDA NAVIGATOR" (Desktop Portal to Data Science), "ANACONDA PROJECT" (Portable Data Science Encapsulation), and "DATA SCIENCE LIBRARIES". The "DATA SCIENCE LIBRARIES" section is expanded, showing categories and logos for Data Science IDEs (jupyter, spyder, jupyterlab, R Studio), Analytics & Scientific Computing (NumPy, SciPy, Numba, pandas, DASK), Visualization (Bokeh, HoloViews, DataShader, matplotlib), and Machine Learning (TensorFlow, scikit-learn, H2O.ai, theano). A note "...and many more!" is present at the bottom right of this section. The bottom part of the page features the "CONDA" logo and the text "Data Science Package & Environment Manager".

# Distribuzione Python scientifica - Anaconda

## The Fundamentals



Jupyter is an open-source project created to support interactive data science and scientific computing across programming languages. Jupyter offers a web-based environment for working with notebooks containing code, data, and text. Jupyter notebooks are the standard workspace for most Python data scientists.



The SciPy library consists of a specific set of fundamental scientific and numerical tools for Python that data scientists use to build their own tools and programs. It provides many user-friendly and efficient numerical routines, such as routines for numerical integration, interpolation, optimization, linear algebra, and statistics.



A library for tabular data structures, data analysis, and data modeling tools, including built-in plotting using Matplotlib. pandas aims to be the fundamental high-level building block for doing practical, real world data analysis in Python



A core package for scientific computing with Python. Numpy enables array formation and basic operations with arrays. Numpy is used for indexing and sorting but can also be used for linear algebra and other operations. Many other data-science libraries for Python are built on NumPy internally, including Pandas and SciPy.



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.



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 from Python.

## Data Visualization



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, dashboards, and data applications.



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 interfaces for building large and complex systems (HoloViews and Param).

# Distribuzione Python scientifica - Anaconda

## 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 an additional compilation step.



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 NumPy API, making it more intuitive for Python data scientists.

## RAPIDS

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 under the hood.



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

## Powered by Dask

These software projects are well-integrated with Dask, or use Dask to power components of their infrastructure.



pandas  
Tabular data analysis



NumPy  
Array and numerical computing



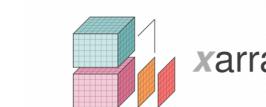
scikit-learn  
Machine learning in Python



scikit-image  
A collection of algorithms for image processing in Python



XGBoost  
Gradient boosted trees for machine learning  
XGBoost can use Dask to bootstrap itself for distributed training



xarray  
Brings the labeled data power of pandas to the physical sciences, by providing N-dimensional variants of the core pandas data structures



RAPIDS  
GPU Accelerated libraries for data science



Iris  
A Python library for analysing and visualising Earth science data



Pangeo  
A community effort for big data geoscience in the cloud



napari  
Multi-dimensional image viewer for Python



Datashader  
Visualization packages for large data



TPOT  
A Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming



Prefect  
A workflow management system, designed for modern infrastructure



Snorkel  
Programmatically build training data for machine learning

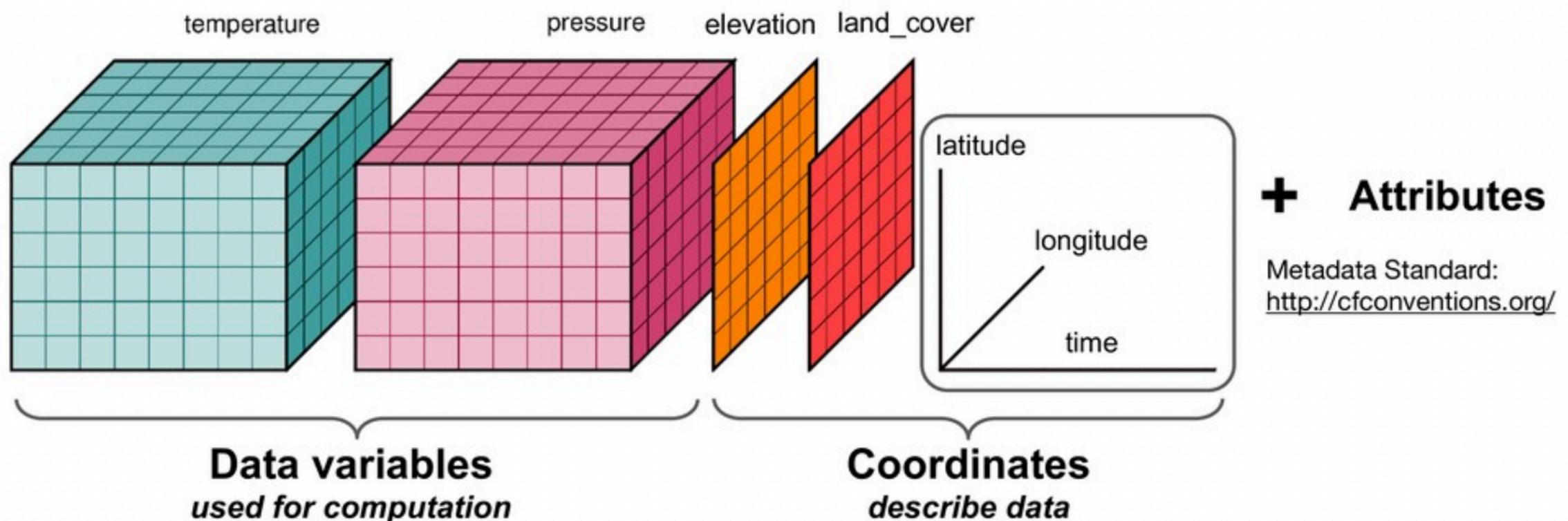
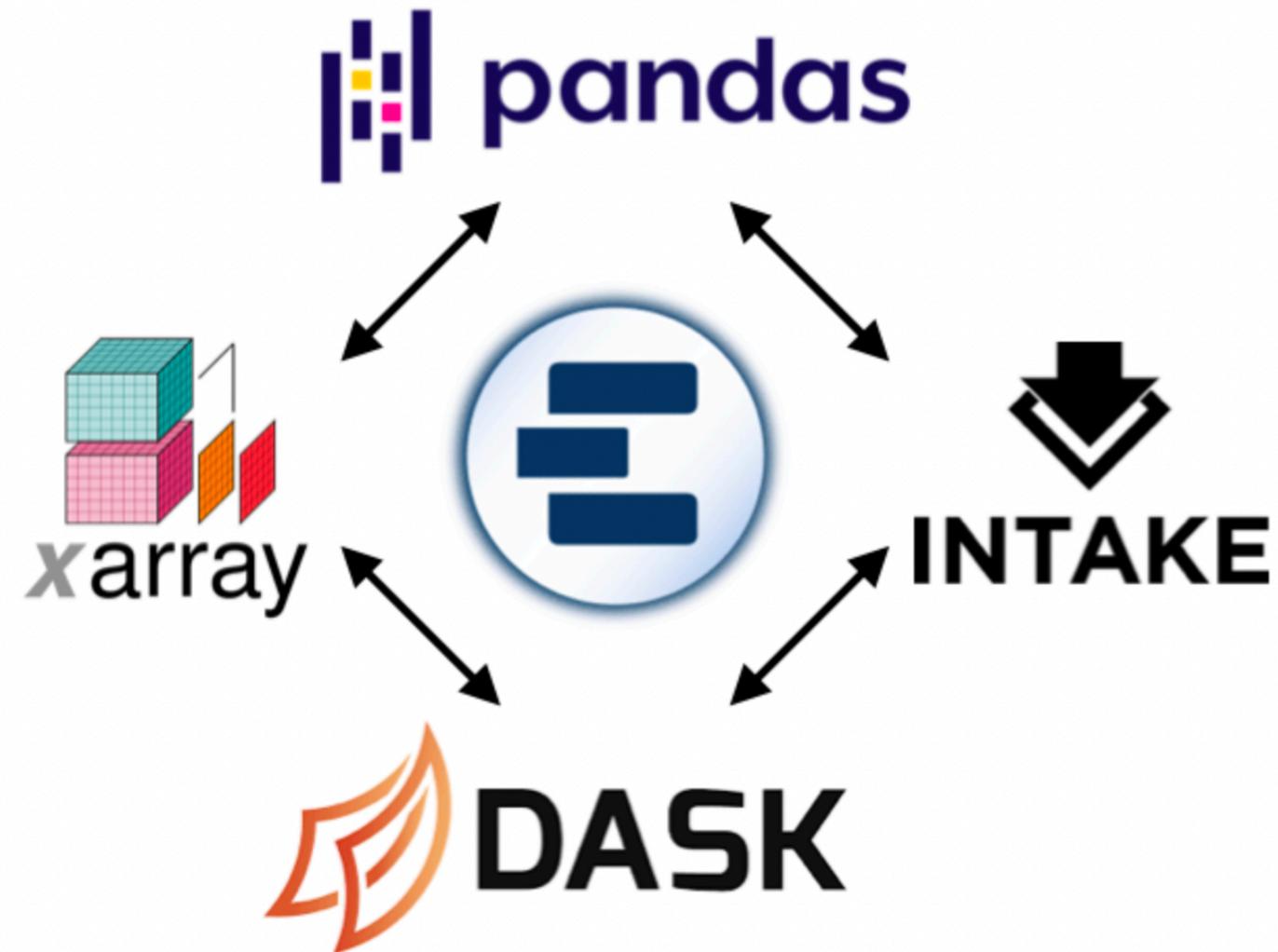


Intake  
A lightweight package for finding, investigating, loading and disseminating data



MDAnalysis  
A Python toolkit to analyze molecular dynamics trajectories generated by a wide range of popular simulation packages

# Distribuzione Python scientifica - Anaconda



## Machine Learning

**K 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.

**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 lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

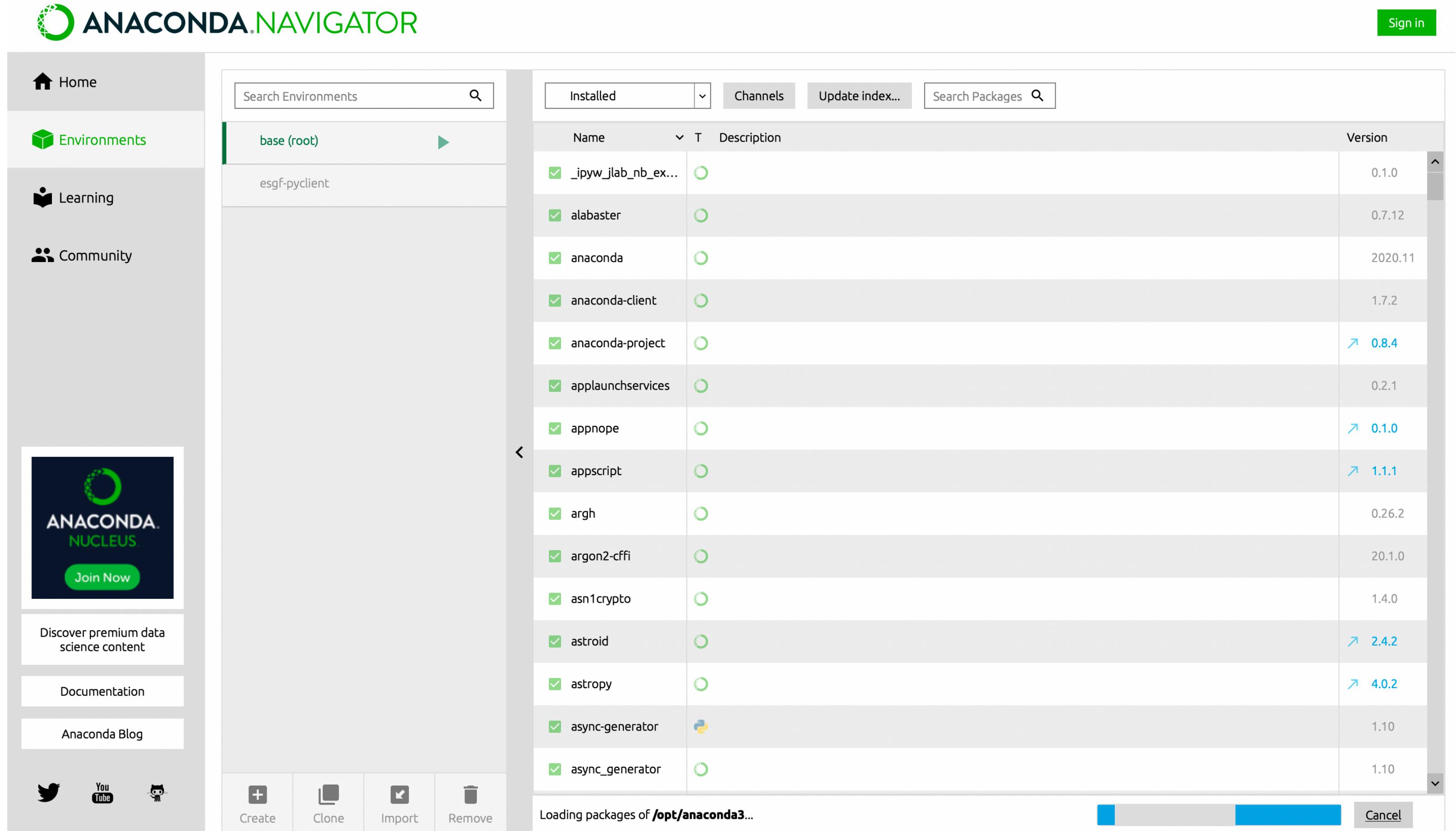
**PYTORCH**

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

**scikit learn**

A powerful and versatile machine learning 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 library.

# Distribuzione Python scientifica - Anaconda



The screenshot shows the Anaconda Navigator application. On the left is a sidebar with icons for Home, Environments, Learning, and Community. A prominent 'ANACONDA NUCLEUS' section with a 'Join Now' button is also present. The main area displays a list of installed packages in the 'base (root)' environment. The packages listed include:

Name	Description	Version
_ipyw_jlab_nb_ex...		0.1.0
alabaster		0.7.12
anaconda		2020.11
anaconda-client		1.7.2
anaconda-project		0.8.4
applaunchservices		0.2.1
appnope		0.1.0
appscript		1.1.1
argh		0.26.2
argon2-cffi		20.1.0
asn1crypto		1.4.0
astroid		2.4.2
astropy		4.0.2
async-generator		1.10
async_generator		1.10

At the bottom, a progress bar indicates 'Loading packages of /opt/anaconda3...'.

OPPURE

Terminale:  
conda install ...  
conda install -c conda-forge ...

# Distribuzione Python scientifica - Anaconda

The screenshot shows the Anaconda Navigator interface. On the left is a sidebar with icons for Home, Environments, Learning, and Community. A large central area displays a grid of application cards. The cards include:

- Datalore: Online Data Analysis Tool with smart coding assistance by JetBrains.
- IBM Watson Studio Cloud: Provides tools for data analysis, machine learning, and model building.
- JupyterLab: An extensible environment for interactive computing.
- Jupyter Notebook: Web-based, interactive computing notebook.
- PyCharm Professional: A full-fledged IDE for both Scientific and Web Python development.
- Qt Console: PyQt GUI for inline figures and multiline editing.
- Spyder: Scientific PYthon Development EnviRonment.
- Glueviz: Multidimensional data visualization.

Each card has a "Launch" button (except for Glueviz which has an "Install" button). The top right corner of the main area has a "Sign in" button and a "Refresh" button.

- Notepad++
- Vim
- Sublime Text
- Spyder
- Atom
- **VS Code**
- Pycharm
- Jupyter
- ...

Editor	Learning curve	Users	Benefits
Spyder	pretty short	Matlab and R background	mature, many features
Jupyter	smooth	teachers	interactive
Visual Studio Code	moderate	scientifics / developers	code quality
PyCharm	steep	developers	professional code

The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a file named `main.py` containing Python code for reading netCDF datasets. The code includes functions for printing variables, extracting station data, and saving matrices. In the center, a help browser for the `DataFrame` class is open, providing its definition, type information, and a detailed description of its two-dimensional tabular nature. Below the help browser is the IPython console, which shows the Python version (2.7.15), the IPython version (5.8.0), and some basic help commands. The bottom status bar indicates permissions (RW), end-of-lines (LF), encoding (UTF-8), and the current line number (28).

```

16 def print_variables(dataset):
17     for d in dataset.variables:
18         desc = dataset.variables[d].name + ' : ' + dataset.variables[d].long_name
19         desc += ' [' + dataset.variables[d].units + ']' if 'units' in dataset.variables[d].ncattrs()
20
21     print desc
22
23
24 def get_value(dataset, variable, cp):
25     # extract index for the station
26     m = cp[0]
27     n = cp[1]
28     stations = pd.DataFrame(dataset.variables['MNSTAT'][0], columns=['M', 'N'])
29     station_index = stations[(stations.M == m) & (stations.N == n)].index[0]
30
31     # extract julian date
32     start_time = datetime.strptime(dataset.variables['time'].units, 'seconds since %Y-%m-%d %H:%M:%S')
33     seconds_offset = np.array(dataset.variables['time'][:, :], dtype='double')
34
35     time_vector = np.array([date2num(start_time + timedelta(seconds=s)) + 366 for s in seconds_offset])
36     time_vector.dtype = 'double'
37
38     # extract values
39     values = np.array([], dtype='double')
40     if variable == 'ZWL':
41         values = np.array(dataset.variables[variable][:, station_index], dtype='double')
42     elif variable == 'ZCURU':
43         values = np.array(dataset.variables[variable][:, 0, station_index], dtype='double')
44
45     data = {'data_nc': OrderedDict([
46         ('X', np.array(dataset.variables['XSTAT'][:, station_index], dtype='double')),
47         ('Y', np.array(dataset.variables['YSTAT'][:, station_index], dtype='double')),
48         ('XUnits', 'm'),
49         ('YUnits', 'm'),
50         ('Val', values),
51         ('Time', time_vector),
52         ('Name', dataset.variables[variable].long_name),
53         ('Units', dataset.variables[variable].units)])
54     }
55
56     return data
57
58
59 def save_mat(data, filename):
60     savemat(filename, data, oned_as='column')
61

```

# Integrated Development Environment - IDEs

Spyder

- Un'interfaccia chiara
- Visualizzazioni di variabili e figure

The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left lists files and folders, including 'test\_summary.py' which is currently selected. The Editor pane in the center displays the code for 'test\_summary.py'. The code imports various modules and defines a try-except block. The Terminal pane at the bottom shows a message in Spanish: 'Hasta el momento, no se encontraron problemas en el área de trabajo.' (Up to now, no problems were found in the workspace area.)

```
#!/usr/bin/env python
# coding: utf-8
from __future__ import (absolute_import, division,
print_function, unicode_literals)

try:
    # noinspection PyUnresolvedReferences, PyCompatibility
    from builtins import * # noqa
except ImportError:
    pass

import os
import matplotlib.pyplot as plt

from climate import summary, tests
from climate.util import plot
from climate.stats import empirical_distributions
from input import saih, tidal_model_driver
```

# Integrated Development Environment - IDEs

VS Code

- Progetti
- Interpreti Python esterni (ssh)
- La possibilità di effettuare il refactoring (rinominare) una variabile in tutto il codice, senza utilizzare una ricerca e sostituzione globale.
- Repository Git completamente integrato per il controllo della versione e la collaborazione del team.

```

1 #!/usr/bin/env python
2 # coding: utf-8
3 from __future__ import (absolute_import, division,
4                         print_function, unicode_literals)
5
6 try:
7     # noinspection PyUnresolvedReferences, PyCompatibility
8     from builtins import * # noqa
9 except ImportError:
10     pass
11
12 import os
13
14 from metoceandataframe.metoceandataframe import MetOceanDF
15 from preprocessing import missing_values
16 from report import latex, tests
17
18 def test_create_latex_document_granada_beach():
19     location = 'granada_beach'
20     drivers = ['wave', 'wind', 'astronomical_tide', 'sea_level_pressure']
21
22     data = []
23     # Data
24     for driver in drivers:
25         modf = os.path.join(tests.current_path, '..', '..', 'inputadapter', 'tests', 'output', 'modf',
26                             '{}_{}.modf'.format(location, driver))
27         data.append(MetOceanDF.read_file(modf))
28
29     # Config report file
30     template = os.path.join(tests.current_path, '..', 'templates', 'latex', '{}.conf'.format(location))
31
32     latex.create_document(data, template, output_title=location)
33
34
35 def test_create_latex_document_cancun():
36     location = 'cancun'
37     drivers = ['wave', 'wind', 'astronomical_tide', 'sea_level_pressure']
38
39     data = []
40     # Data
41     test_create_latex_document_canc...

```

Run console or debugger to view available data

Documentation: protocol x  
No documentation found.

- Progetti
- Interpreti Python esterni (ssh)
- La possibilità di effettuare il refactoring (rinominare) una variabile in tutto il codice, senza utilizzare una ricerca e sostituzione globale.
- Repository Git completamente integrato per il controllo della versione e la collaborazione del team.

# Integrated Development Environment - IDEs

Pycharm

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.

## Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.

# Integrated Development Environment - IDEs

## Jupyter

```
In [57]:  
from sympy import diff, sin, exp  
  
diff(sin(x)*exp(x), x)  
Out[57]:  $e^x \sin(x) + e^x \cos(x)$ 
```

Compute  $\int(e^x \sin(x) + e^x \cos(x)) dx$

```
In [58]:  
from sympy import integrate, cos  
  
integrate(exp(x) * sin(x) + exp(x) * cos(x), x)  
Out[58]:  $e^x \sin(x)$ 
```

Compute  $\int_{-\infty}^{\infty} \sin(x^2) dx$

```
In [59]:  
from sympy import oo  
  
integrate(sin(x**2), (x, -oo, oo))  
Out[59]: 
$$\frac{\sqrt{2}\sqrt{\pi}}{2}$$

```

# Cloud-based software

---

- Kaggle (also GPU)
- Deepnote
- **Google Colab**
- **Github Codespaces**
- Amazon SageMaker Studio Lab

# Cloud-based software | Google Colab

```
# Load the Drive helper and mount
from google.colab import drive
# This will prompt for authorization.
drive.mount('/content/drive')

# After executing the cell above, Drive
# files will be present in "/content/drive/My Drive".
!ls "/content/drive/My Drive"
```

Colab ti dà accesso a una GPU o a un TPU. Fino a 15 GB di spazio su disco per archiviare i tuoi set di dati. Le sessioni si interromperanno dopo 60 minuti di inattività, anche se possono durare fino a 12 ore.

Attualmente forniscono VM a 2 core con 12 Gb di RAM e 50 Gb (kernel non GPU) o 350 Gb (kernel GPU) di spazio su disco.



Earth Engine's public data archive includes more than forty years of historical imagery and scientific datasets, updated daily and available for online analysis.

ClimaCell - CBAM India Weather Forecasts Climacell High-resolution weather forecast model data serving India	Cloud-to-Ground Lightning Strikes NOAA Aggregated lightning strike data from 1987 to 2018	CMIP 6 NOAA Climate Model Data from the World Climate Research Programme	GHCN Daily NOAA Global Historical Climatology Network Daily Weather Data
GHCN Monthly NOAA Global Historical Climatology Network Monthly Weather Data	Global Forecast System NOAA The National Weather Service's Primary Prediction Model	Global Historical Tsunami Database NOAA Tsunami and source event data as far back as 2000 BC	Global Hurricane Tracks (IBTrACS) NOAA Tropical Cyclone/Hurricane position and intensity
NOAA Rapid Refresh (RAP) NOAA Rapid Refresh operational weather prediction system data.	OnPoint Weather - Past Weather and Climatology Data... Weather Source Past, Present and Forecast Global Weather Data	OnPoint Weather - Temperature History &... Weather Source Past, Present and Forecast Global Weather Data	Preliminary Storm Reports NOAA Preliminary storm reports from NWS's Storm Prediction Center
Project Sunroof Google Project Sunroof Find out how much sunlight hits your roof in a year	Real-time Air Quality OpenAQ Real-time air quality data from around the world	Sentinel-2 ESA Comprehensive land, ocean, and atmosphere observation data	Severe Storm Event Details NOAA Detailed data about severe weather between 1950 and this year

# Cloud-based software | Google Colab

```
# Load the Drive helper and mount
from google.colab import drive
# This will prompt for authorization.
drive.mount('/content/drive')

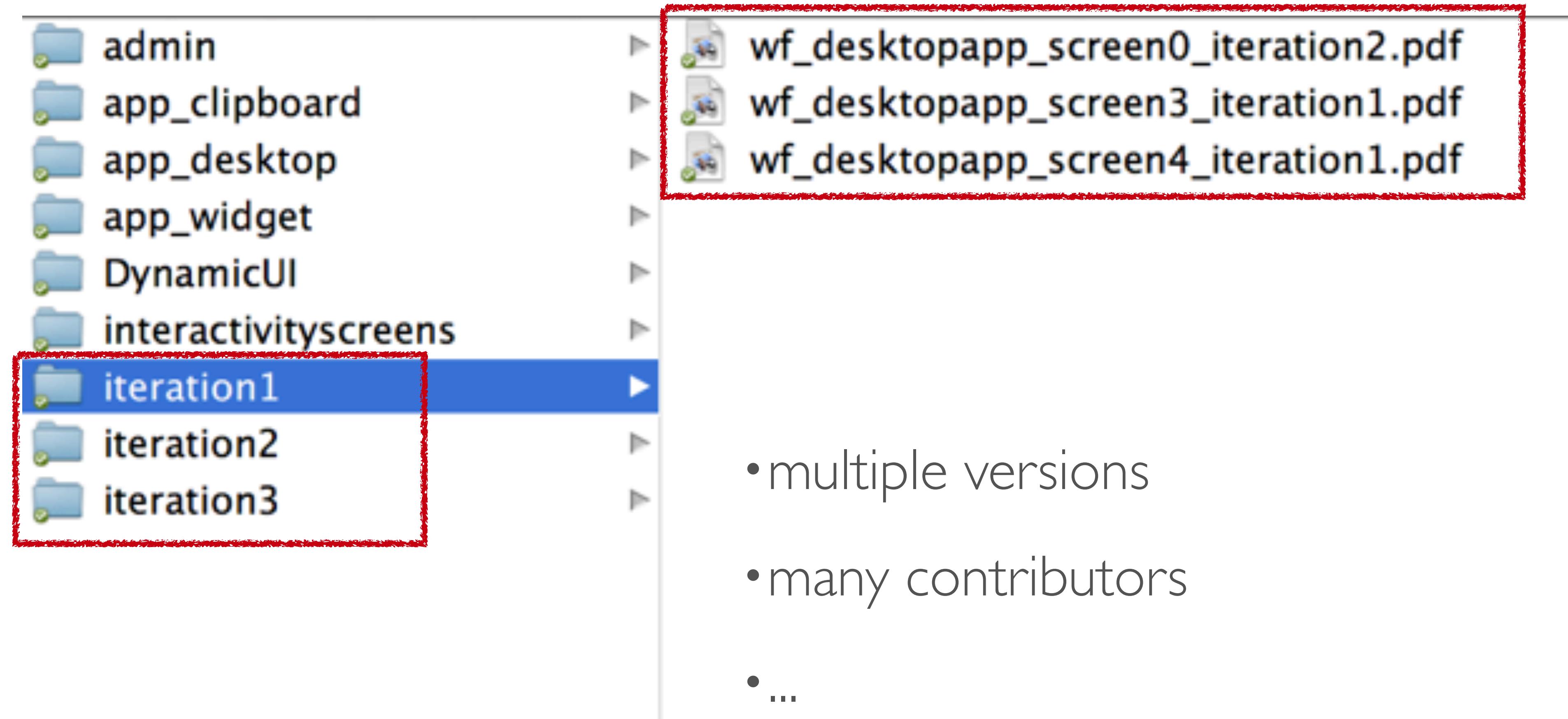
# After executing the cell above, Drive
# files will be present in "/content/drive/My Drive".
!ls "/content/drive/My Drive"
```

## Earth Engine

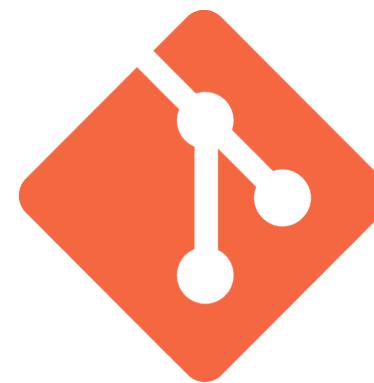
Earth Engine's public data archive includes more than forty years of historical imagery and scientific datasets, updated daily and available for online analysis.

 climacell ClimaCell - CBAM India Weather Forecasts Climacell High-resolution weather forecast model data serving India	 Cloud-to-Ground Lightning Strikes NOAA Aggregated lightning strike data from 1987 to 2018	 WCRP CMIP 6 NOAA Climate Model Data from the World Climate Research Programme	 GHCN Daily NOAA Global Historical Climatology Network Daily Weather Data
 GHCN Monthly NOAA Global Historical Climatology Network Monthly Weather Data	 Global Forecast System Database NOAA The National Weather Service's Primary Prediction Model	 Global Historical Tsunami Database NOAA Tsunami and source event data as far back as 2000 BC	 Global Hurricane Tracks (IBTrACS) NOAA Tropical Cyclone/Hurricane position and intensity
 NOAA Rapid Refresh (RAP) NOAA Rapid Refresh operational weather prediction system data.	 OnPoint Weather - Past Weather and Climatology Data... Weather Source Past, Present and Forecast Global Weather Data	 OnPoint Weather - Temperature History &... Weather Source Past, Present and Forecast Global Weather Data	 Preliminary Storm Reports NOAA Preliminary storm reports from NWS's Storm Prediction Center
 Project Sunroof Google Project Sunroof Find out how much sunlight hits your roof in a year	 Real-time Air Quality OpenAQ Real-time air quality data from around the world	 Sentinel-2 ESA Comprehensive land, ocean, and atmosphere observation data	 Severe Storm Event Details NOAA Detailed data about severe weather between 1950 and this year

# Git e Github



# Git e Github

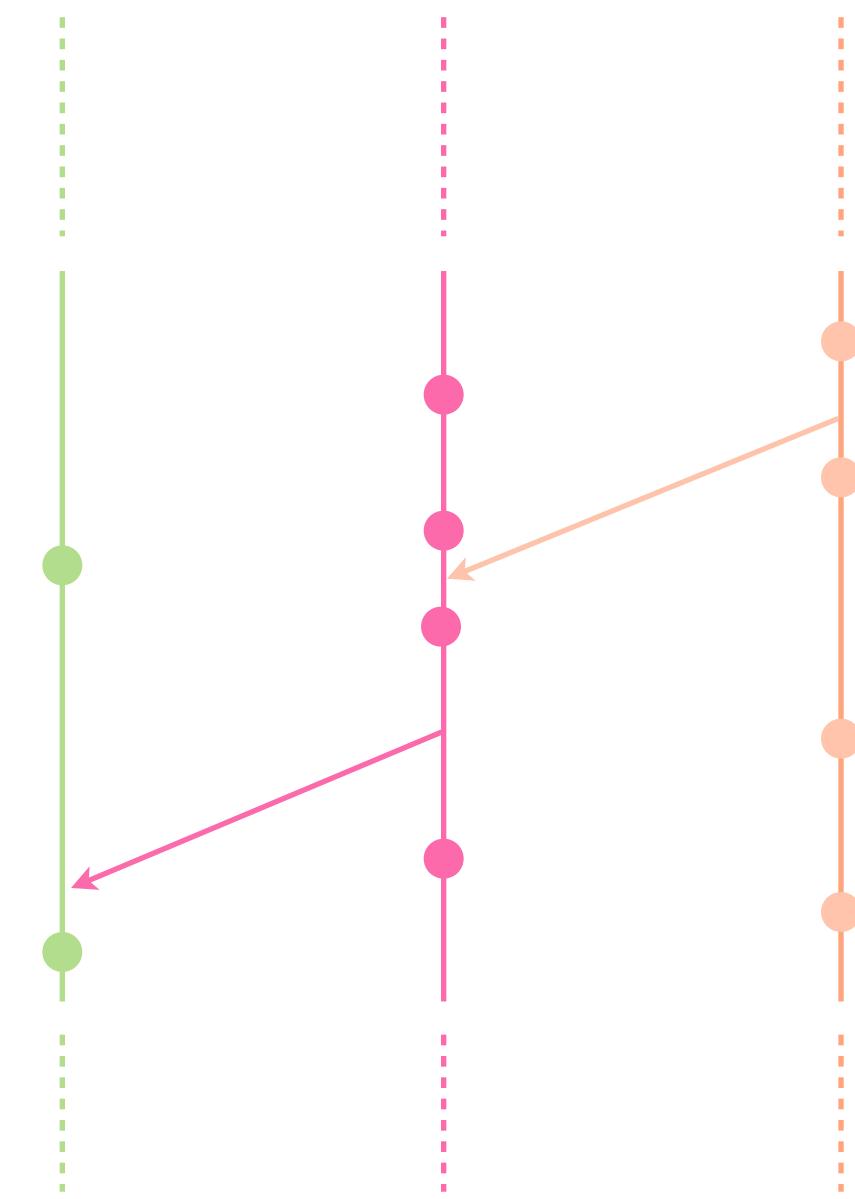


git is an open source, distributed **version control system** designed for speed and efficiency

Designed to make it easier to have multiple versions of a code base, sometimes across multiple developers or teams

It allows you to see changes you make to your code and easily revert them.

- Performing a diff
- Viewing file history
- Committing changes
- Merging branches
- Obtaining any other revision of a file
- Switching branches





\$ git clone <url> <name>

andreall / python\_ARPAL

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

python\_ARPAL Private

Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags

andreal Update README.md b2ecf5a 26 minutes ago 2 commits

README.md Update README.md 26 minutes ago

README.md

Corso di formazione per l'ottimizzazione e la gestione di modelli idrodinamici procedure per l'analisi automatizzata di dati di qualità delle acque marine - Ott. 2023

ARPAL - Ottobre 2023

Andrea Lira Loarca  
MeteOcean Research Group - Dipartimento di Ingegneria Chimica, Civile e Ambientale  
Università degli Studi di Genova

Introduzione a Python e Numpy (4 ore)  
Venerdì 13 ottobre 2023 | 09:00-13:00

- Introduzione a Python e librerie scientifiche
- Ambiente di sviluppo (IDE): Spyder, Visual Studio Code, Jupyter notebook
- Nozioni di base sulla programmazione Python
- I principali tipi di dato di Python e i loro metodi
- Introduzione a Numpy
- Array (1D, 2D, N-D) in Numpy

About

Corso di formazione per l'ottimizzazione e la gestione di modelli idrodinamici procedure per l'analisi automatizzata di dati di qualità delle acque marine - Ott. 2023

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

\$ git pull

= git fetch + git merge

\$ git push

# Git GUI

- GitHub Desktop
- GitKraken
- Sourcetree
- SmartGit
- Fork
- Tower
- ...



- <https://www.gitkraken.com/blog/best-git-gui-client>

# Github corso

---

- [https://github.com/andreall/python\\_ARPAL.git](https://github.com/andreall/python_ARPAL.git)

## Risorse addizionali

- <https://projectpythia.org>
- <https://dabeaz-course.github.io/practical-python/Notes/Contents.html>