

Google Colab



Ramiro Logares

Intro

- Google Colab, short for Google Colaboratory, is a cloud-based Jupyter Notebook environment provided by Google for free for small tasks
- It allows users to create, edit, and share Python notebooks without the need to install any software on their local machines (runs bash and R too)
- Colab is designed to facilitate collaboration, making it easy for multiple users to work on a project simultaneously
- It also provides access to more powerful resources, such as GPUs and TPUs, for accelerated computing

Key Features

- Real-time collaboration: work with your colleagues simultaneously
- Version control: track changes and revert to previous versions
- Rich text and markdown support: create well-documented notebooks
- Integration with Google Drive: store and share notebooks easily

Getting Started

- Visit <https://colab.research.google.com>
- Sign in with your Google account
- Create a new Python notebook or open an existing one
- Start coding and running cells

☰

Table of contents

🗑️

✕

🔍

Getting started

⏏

Data science

{x}

Machine learning

🔑

More Resources

📁

Featured examples

+

Section

Welcome to Colab!

Explore the Gemini API

The Gemini API gives you access to Gemini models created by Google DeepMind. Gemini models are built from the ground up to be multimodal, so you can reason seamlessly across text, images, code, and audio.

How to get started?

1. Go to [Google AI Studio](#) and log in with your Google account.
2. [Create an API key](#).
3. Use a quickstart for [Python](#), or call the REST API using [curl](#).

Discover Gemini's advanced capabilities

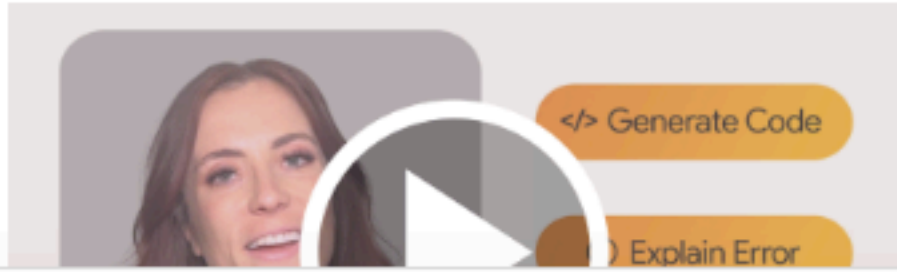
- Discover the [multimodal Live API](#) (demo [here](#)).
- Learn how to [analyze images and detect items in your pictures](#) using Gemini (bonus, there's a [3D version](#) as well!).
- Unlock the power of [Gemini thinking model](#), capable of solving complex task with its inner thoughts.

Explore complex use cases

- Use [Gemini grounding capabilities](#) to create a report on a company based on what the model can find on internet.
- Extract [invoices and form data from PDF](#) in a structured way.
- Create [illustration based on a whole book](#) using Gemini large context window and Imagen.

To learn more, check out the [Gemini cookbook](#) or visit the [Gemini API documentation](#).

Colab now has AI features powered by [Gemini](#). The video below provides information on how to use these features, whether you're new to Python, or a seasoned veteran.



Commands + Code + Text Add chunks of code or text (Markdown)

Start coding or generate with AI.

Analyze files with Gemini

Executable cells (expecting Python by default)

Upload files from your computer into Colab

Files

Analyze your files with code written by Gemini Upload

..

sample_data

OTU_table.tsv

Resources

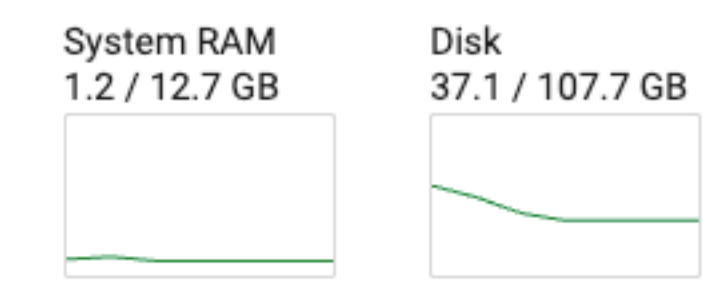
Connected to Python 3 Google Compute Engine backend
RAM: 1.17 GB/12.67 GB
Disk: 37.06 GB/107.72 GB

You are not subscribed. Learn more
You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units here.
At your current usage level, this runtime may last up to 85 hours 40 minutes.

Manage sessions

Want more memory and disk space? Upgrade to Colab Pro

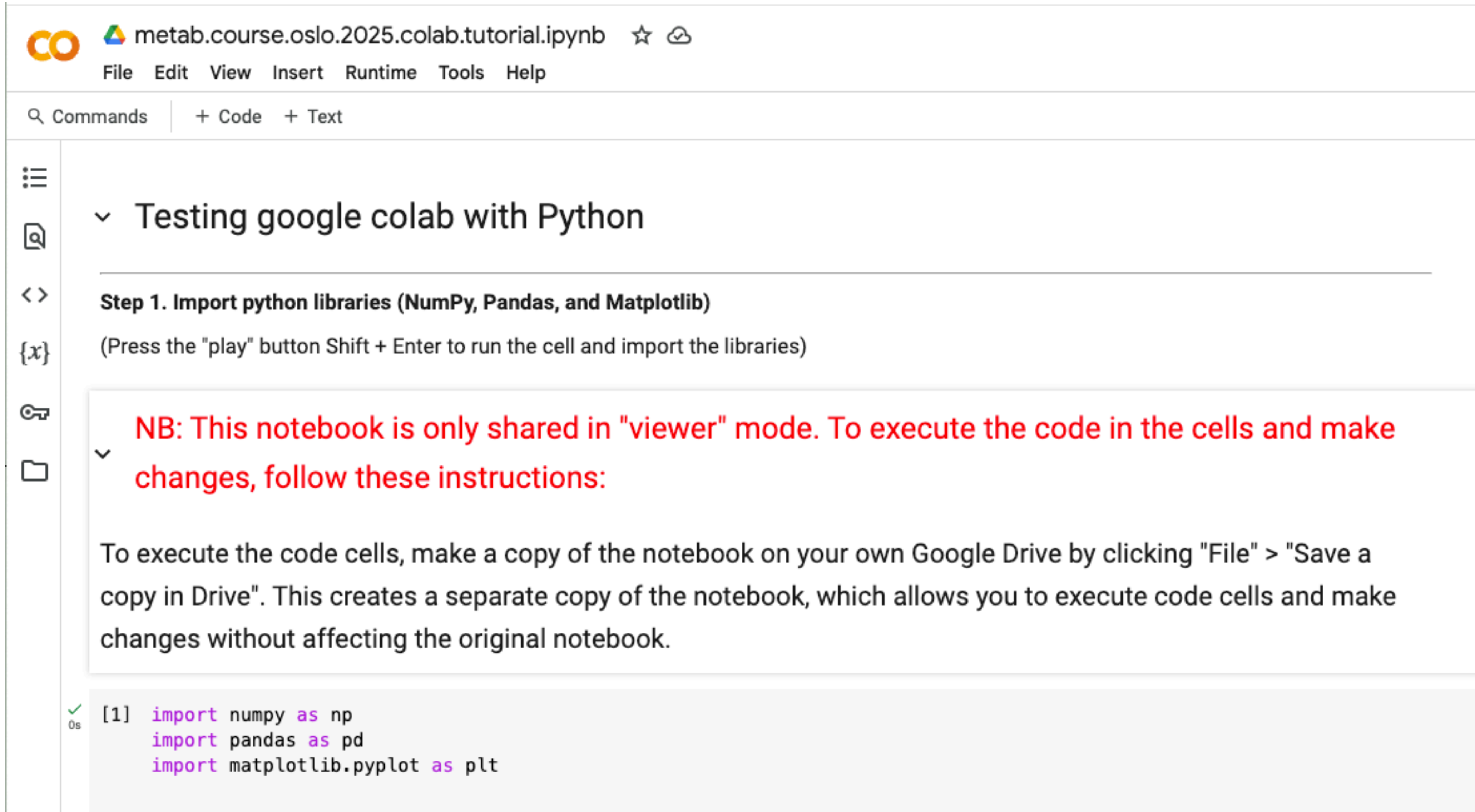
Python 3 Google Compute Engine backend
Showing resources from 10:06 PM to 10:07 PM



Resources

Let's run one example in Python




1. Load libraries



The screenshot shows a Google Colab notebook titled "metab.course.oslo.2025.colab.tutorial.ipynb". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". Below the menu is a search bar and buttons for "Commands", "+ Code", and "+ Text". On the left sidebar, there are icons for file management and a list of cells. The main content area displays a section titled "Testing google colab with Python". Under this section, it says "Step 1. Import python libraries (NumPy, Pandas, and Matplotlib)" and provides instructions: "(Press the 'play' button Shift + Enter to run the cell and import the libraries)". A red warning message states: "NB: This notebook is only shared in 'viewer' mode. To execute the code in the cells and make changes, follow these instructions:". Below the warning, it explains: "To execute the code cells, make a copy of the notebook on your own Google Drive by clicking 'File' > 'Save a copy in Drive'. This creates a separate copy of the notebook, which allows you to execute code cells and make changes without affecting the original notebook." At the bottom, a code cell is shown with the following Python code:







```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```


2. Generate random data

 metab.course.oslo.2025.colab.tutorial.ipynb  

File Edit View Insert Runtime Tools Help

🔍 Commands | + Code + Text




Step 2. Generate sample data



```
[2] # Set the seed for reproducibility
np.random.seed(42)

# Generate a Pandas DataFrame with random numbers
data = pd.DataFrame(np.random.randn(100, 4), columns=['A', 'B', 'C', 'D'])

# Display the first 10 rows of the DataFrame
data.head(10)
```






	A	B	C	D
0	0.496714	-0.138264	0.647689	1.523030
1	-0.234153	-0.234137	1.579213	0.767435
2	-0.469474	0.542560	-0.463418	-0.465730
3	0.241962	-1.913280	-1.724918	-0.562288
4	-1.012831	0.314247	-0.908024	-1.412304
5	1.465649	-0.225776	0.067528	-1.424748
6	-0.544383	0.110923	-1.150994	0.375698
7	-0.600639	-0.291694	-0.601707	1.852278
8	-0.013497	-1.057711	0.822545	-1.220844
9	0.208864	-1.959670	-1.328186	0.196861









Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

3. Summary stats

 metab.course.oslo.2025.colab.tutorial.ipynb  

File Edit View Insert Runtime Tools Help

🔍 Commands | + Code + Text






Step 3: Calculate summary statistics

```
[ ] # Calculate summary statistics for the dataset
summary = data.describe()

# Display the summary statistics
summary
```

	A	B	C	D
count	100.000000	100.000000	100.000000	100.000000
mean	-0.009811	0.033746	0.022496	0.043764
std	0.868065	0.952234	1.044014	0.982240
min	-2.025143	-1.959670	-3.241267	-1.987569
25%	-0.716089	-0.564362	-0.616727	-0.727600
50%	-0.000248	-0.024646	0.068665	0.075219
75%	0.528231	0.547116	0.701519	0.778891
max	2.314659	3.852731	2.189803	2.720169






Next steps: [Generate code with summary](#) [View recommended plots](#) [New interactive sheet](#)

4. Data visualization










Running Bash

Prepend command with !

 metab.course.oslo.2025.colab.tutorial.ipynb  


File Edit View Insert Runtime Tools Help

 Commands | + Code + Text




To execute a Bash command in a code cell, prepend the command with an exclamation mark (!)


Step 1. list the files and directories in the current working directory

 0s


```
[5] !ls
```

 sample_data


Let's check what remote machine we are running

 0s

```
[6] !uname -a
```


 Linux 9e24de877f96 6.1.85+ #1 SMP PREEMPT_DYNAMIC Thu Jun 27 21:05:47 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux

Step 2. Generate a sample directory


 0s

```
[7] !mkdir sample_directory
```

Step 3. Change de current working directory




 0s

```
[8] %cd sample_directory
```

 /content/sample_directory







Note: We use %cd instead of !cd in this case because %cd changes the working directory for the entire notebook, while !cd would only change it for the specific cell.

Running Bash

 metab.course.oslo.2025.colab.tutorial.ipynb  

File Edit View Insert Runtime Tools Help

🔍 Commands | + Code + Text

Step 4. Let's check the working directory

```
[10] !pwd
```

```
/content/sample_directory
```

Step 5. We generate an empty file

```
[11] !touch sample_file.txt
!ls
```

```
sample_file.txt
```

Step 6: Write content to the file

We use the echo command along with the > operator to write content to "sample_file.txt":

```
[12] !echo "This is the metab course of Oslo v2025!" > sample_file.txt
```




Step 7: Read the content of the file

Use the cat command to read and display the content of "sample_file.txt":


```
[13] !cat sample_file.txt
```







```
This is the metab course of Oslo v2025!
```


Running Bash

 metab.course.oslo.2025.colab.tutorial.ipynb  

File Edit View Insert Runtime Tools Help

 Commands | + Code + Text

Step 8: Remove the file and directory

First, change the working directory to the parent directory:

```
[14] %cd ..
```




/content

Now, use the rm command to remove the "sample_file.txt" and "sample_directory":


```
!rm -r sample_directory  
!ls
```







sample_data

Uploading files from your computer into Colab

 metab.course.oslo.2025.colab.tutorial.ipynb  

File Edit View Insert Runtime Tools Help

 Commands | + Code + Text




✓ Uploading files from your computer

Step 1. let's import the necessary module from the google.colab package:

[] from google.colab import files

Step 2: Use the upload() function to upload files

[] uploaded = files.upload()


 Choose Files OTU_table.tsv

- OTU_table.tsv(text/tab-separated-values) - 1082385 bytes, last modified: 4/23/2021 - 100% done

Saving OTU_table.tsv to OTU_table.tsv

Click "Choose Files" and select one or more files from your computer to upload. The uploaded files will be saved in the current working directory

Step 2: Use the upload() function to upload files

 uploaded = files.upload()

... Choose Files No file chosen Cancel upload

Uploading files from your computer into Colab

[illegible]

Tutorial

<https://colab.research.google.com/drive/1ilsdx2b--mEQbgQQqfTPBDpvVB4BAQ0L?usp=sharing>