Tutorial III

Python Programming
with Jupyter Notebooks

Google Colab
&
Python for Data Science

2023 © dr. daniel benninger

Google Colab

- Google is quite aggressive in AI research. Over many years, Google developed AI framework called **TensorFlow** and a development tool called **Colaboratory**.

- **TensorFlow** is an *open-sourced* end-to-end platform that makes it easy for you to build and deploy Machine Learning models.
- It is an *entire ecosystem* to help you solve challenging, real-world problems with machine learning.

jupyter colab

2023 © dr. daniel benninger

Google Colab

- Colaboratory is now known as **Google Colab** or simply **Colab**.
- The introduction of Colab has eased the learning and development of machine learning applications.

- Another feature that Google offers to the developers is the use of GPU (graphical processing unit). Colab supports GPU and is free.
- The reasons for making it free for public was to make its software *a standard in the academics for teaching machine learning and data science*.

2023 © dr. daniel benninger

Google Colab

- Colab is a free **Jupyter notebook** environment that runs entirely in the cloud.
- Most importantly, *it does not require a setup* and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs.
- Colab *supports many popular machine learning libraries* which can be easily loaded in your notebook.

https://jupyter.org/

2023 © dr. daniel benninger

What Colab Offers You?

- As a programmer, you can perform the following using Google Colab
  - **Write and execute code in Python**
  - **Document your code** that supports mathematical equations
  - Create/Upload/**Share** notebooks
  - *Import/Save notebooks from/to Google Drive*
  - **Import**/Publish **notebooks from GitHub**
  - **Import external datasets** e.g. from Kaggle
  - *Integrate Machine Learning Packages* like PyTorch, TensorFlow, Keras, OpenCV ...
  - **Free Cloud** service with free GPU

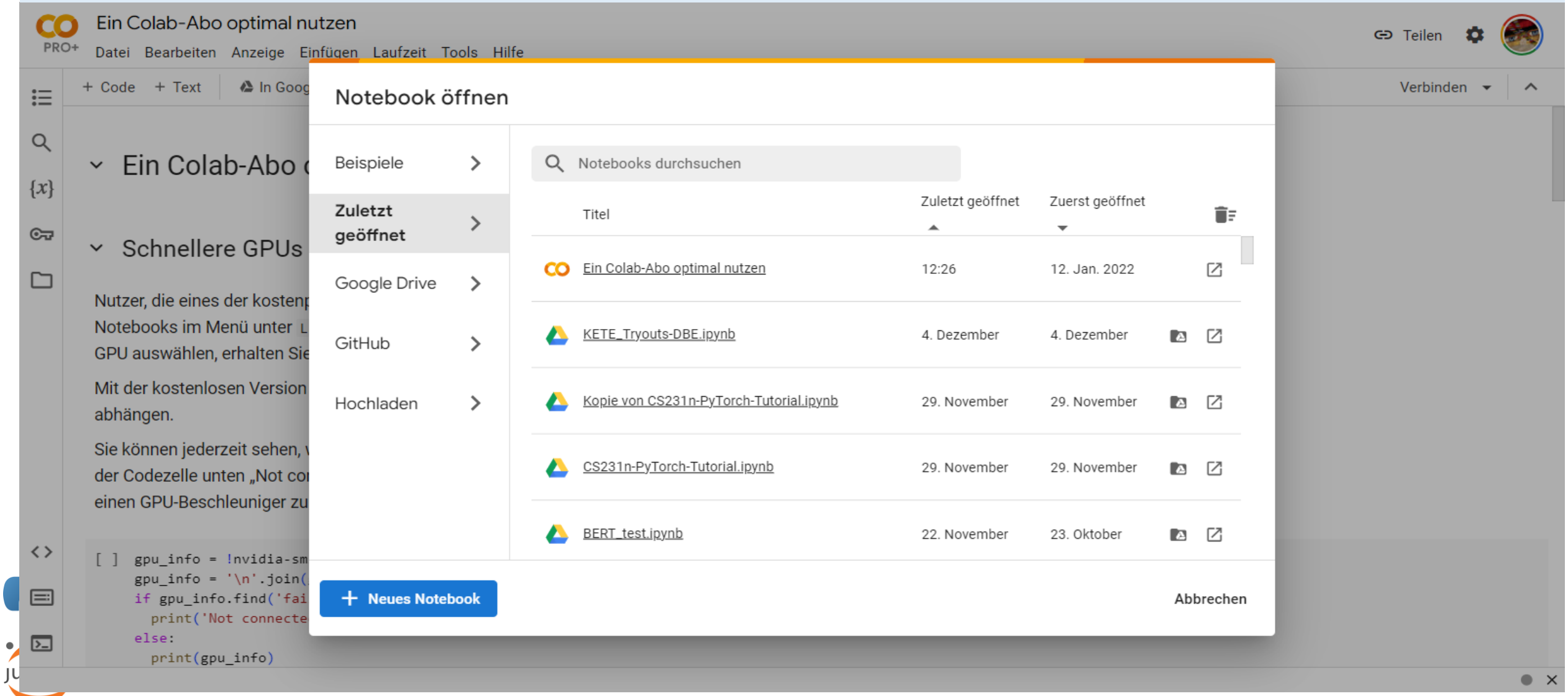
2023 © dr. daniel benninger

How Google Colab Works?

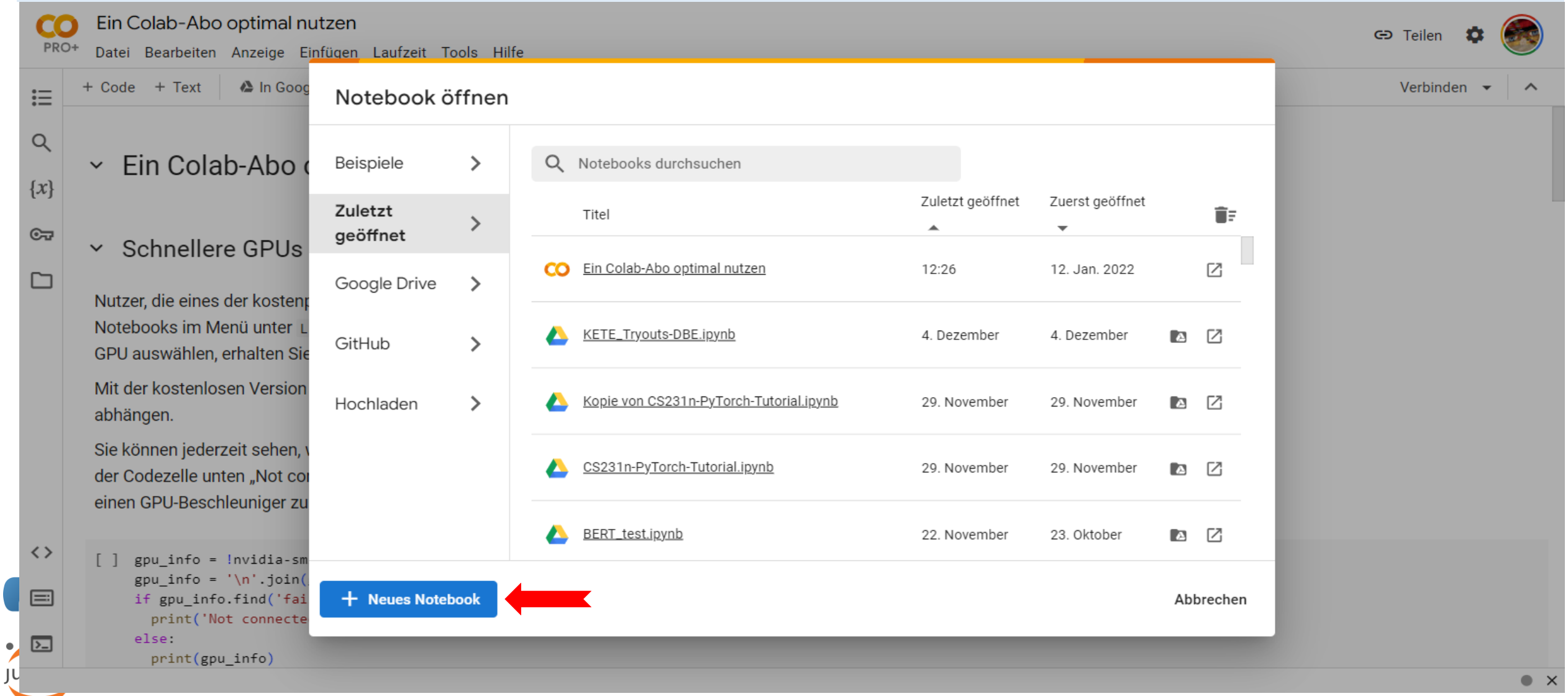
As Colab **implicitly uses Google Drive** for storing your notebooks, ensure that you are logged in to your **Google** Drive **account** before proceeding further.

2023 © dr. daniel benninger

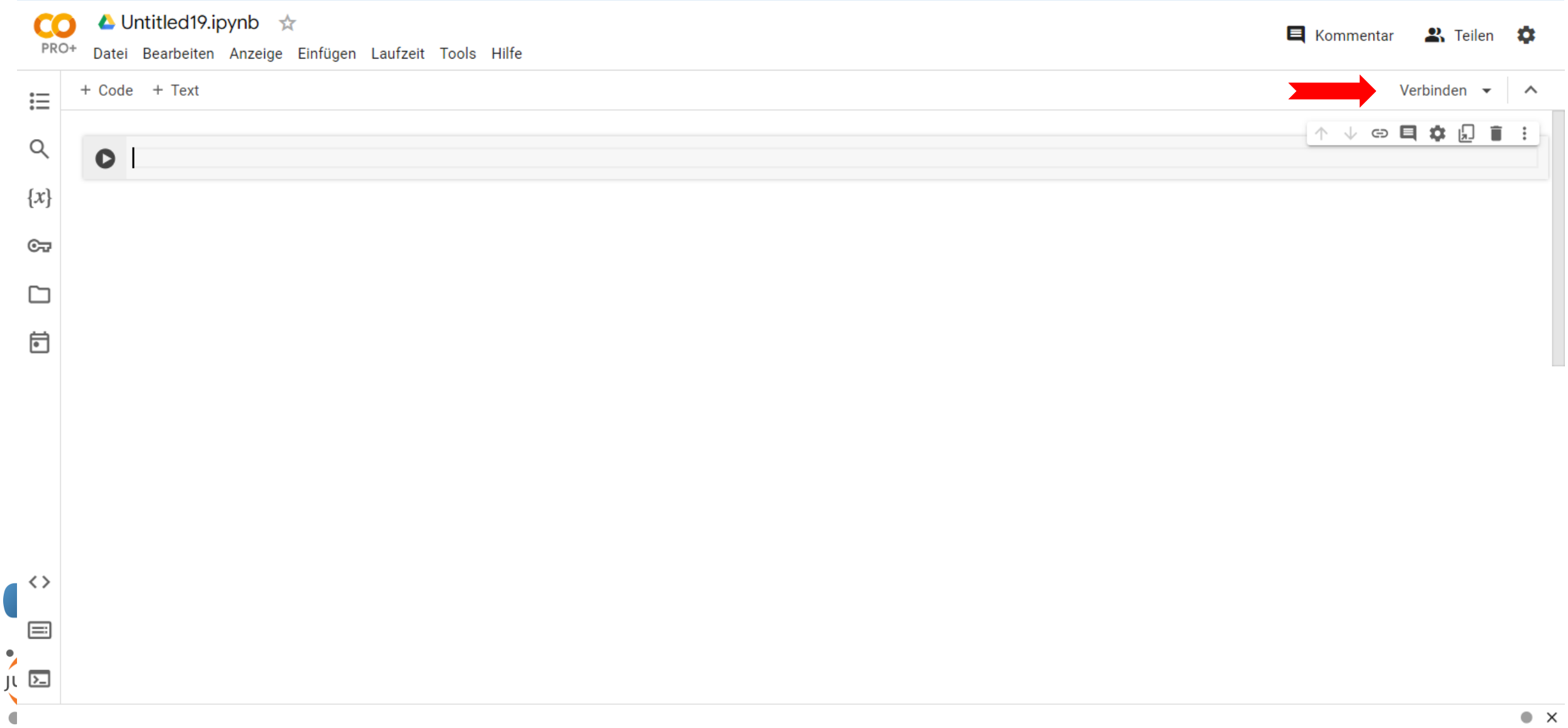
Step 0: Open the following URL in your browser:
https://colab.research.google.com



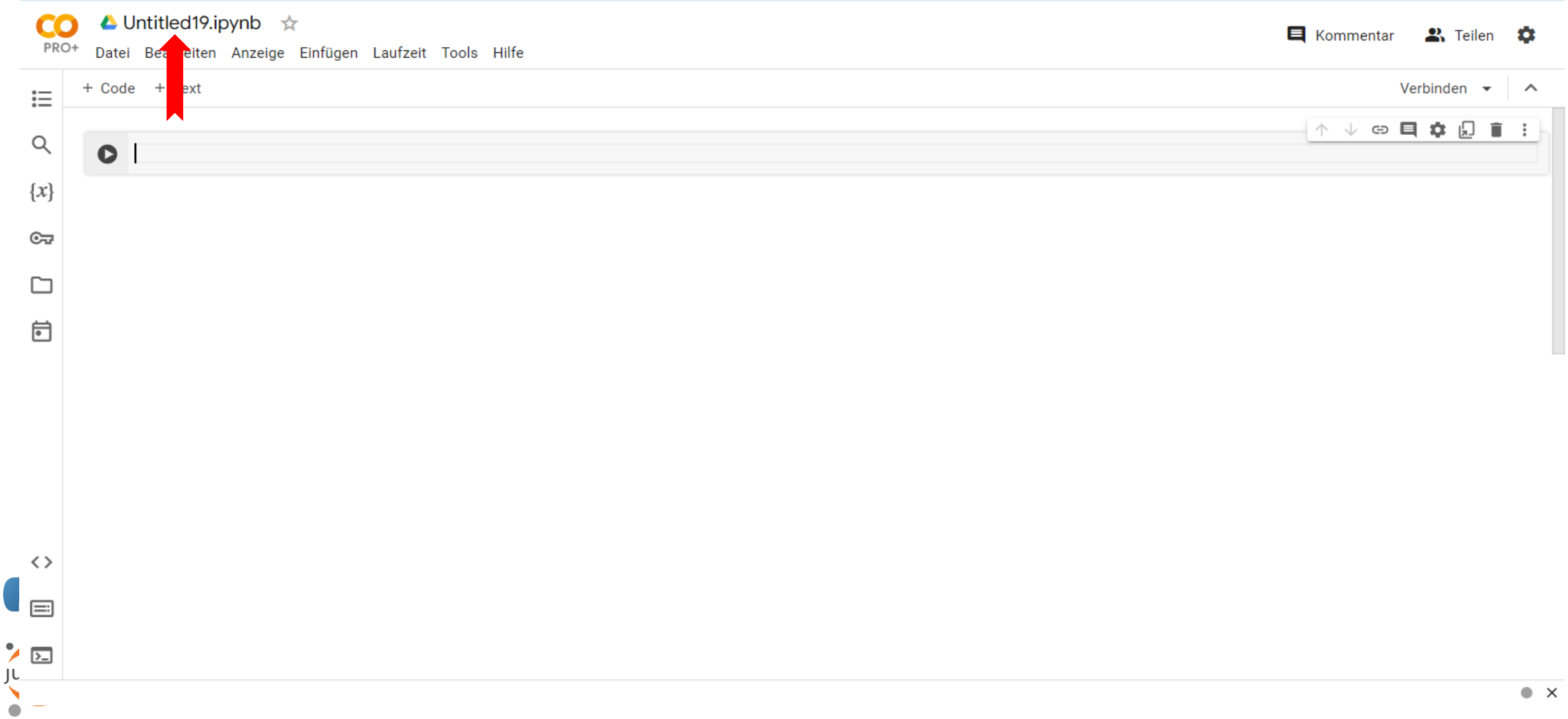
Step 1: Click on the **NEW NOTEBOOK** link at the bottom left of the opened window



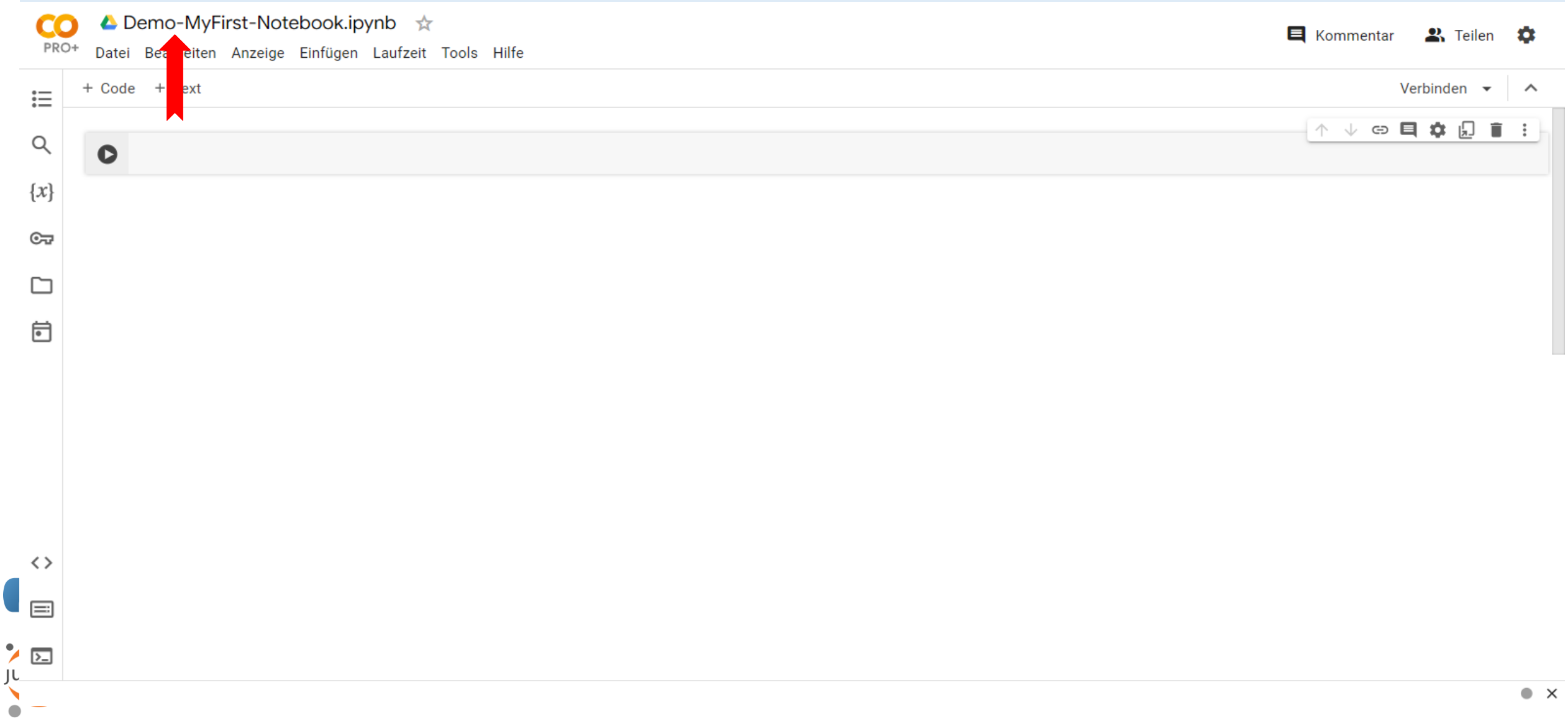
Step 3: Click on the **connect** at the top (right) of the screen



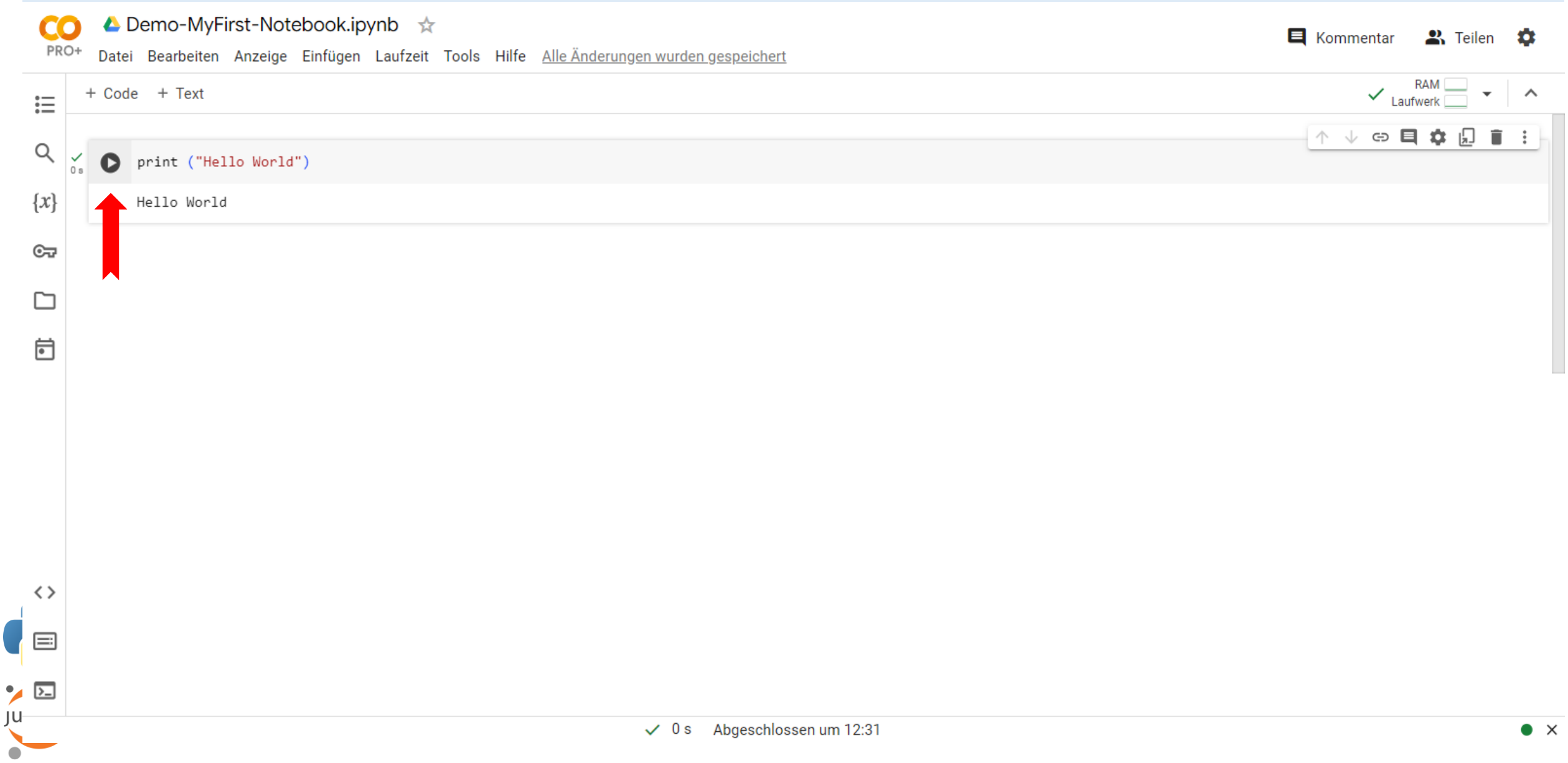
Step 4a: Click on the **untitledxx.ipynb** at the top (left) of the screen



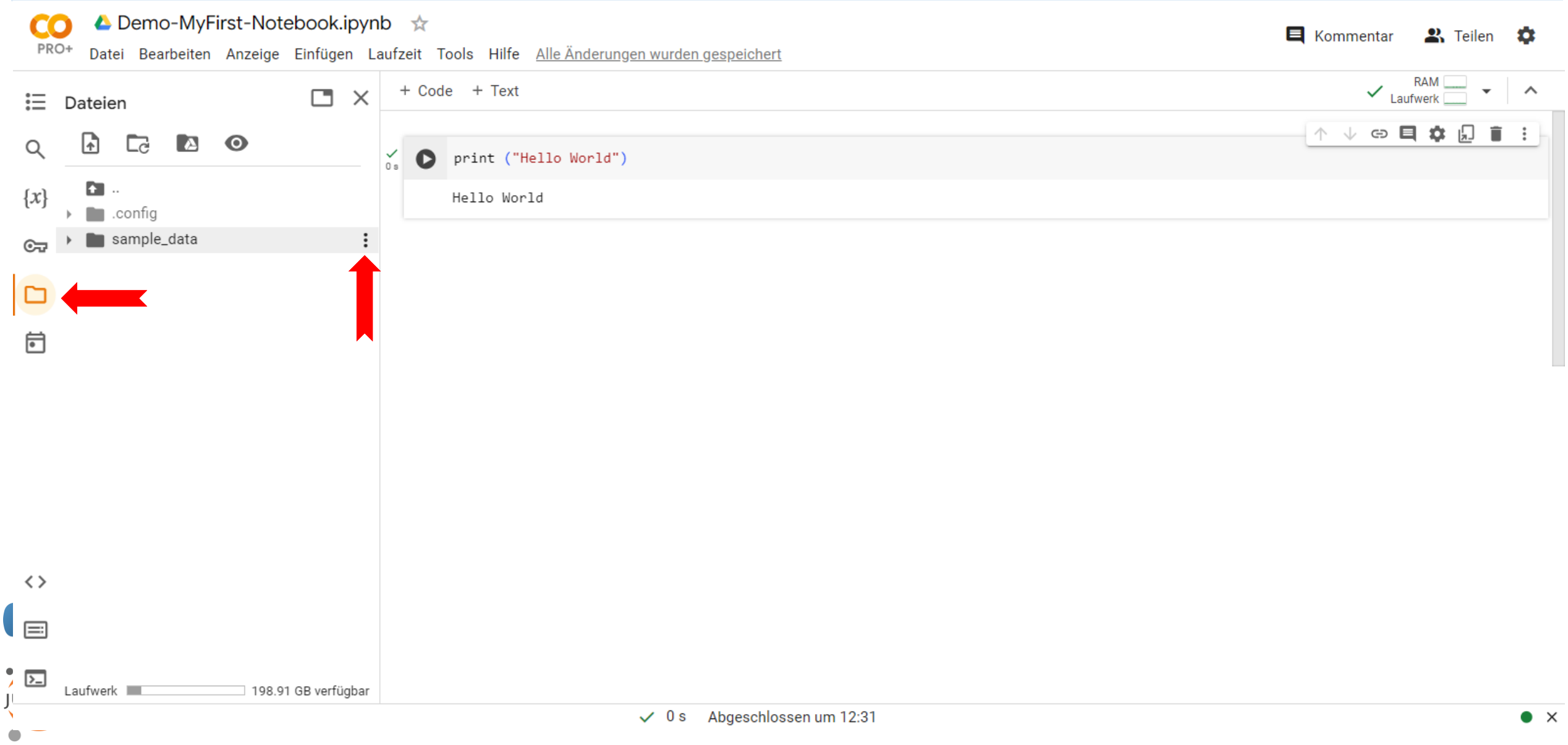
Step 4b: Replace **untitledxx** with an appropriate notebook **name**



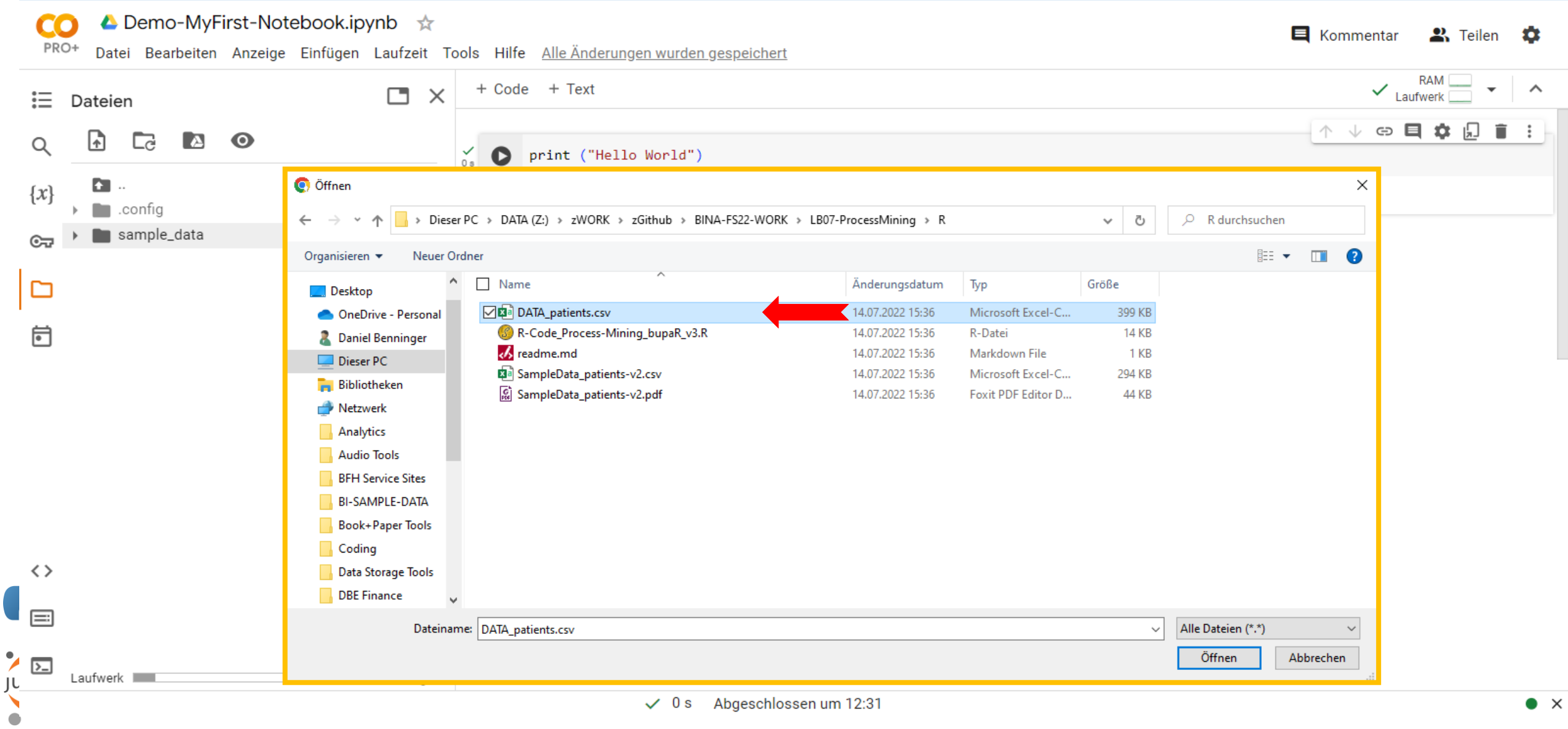
Step 5: Write and execute a single program statement



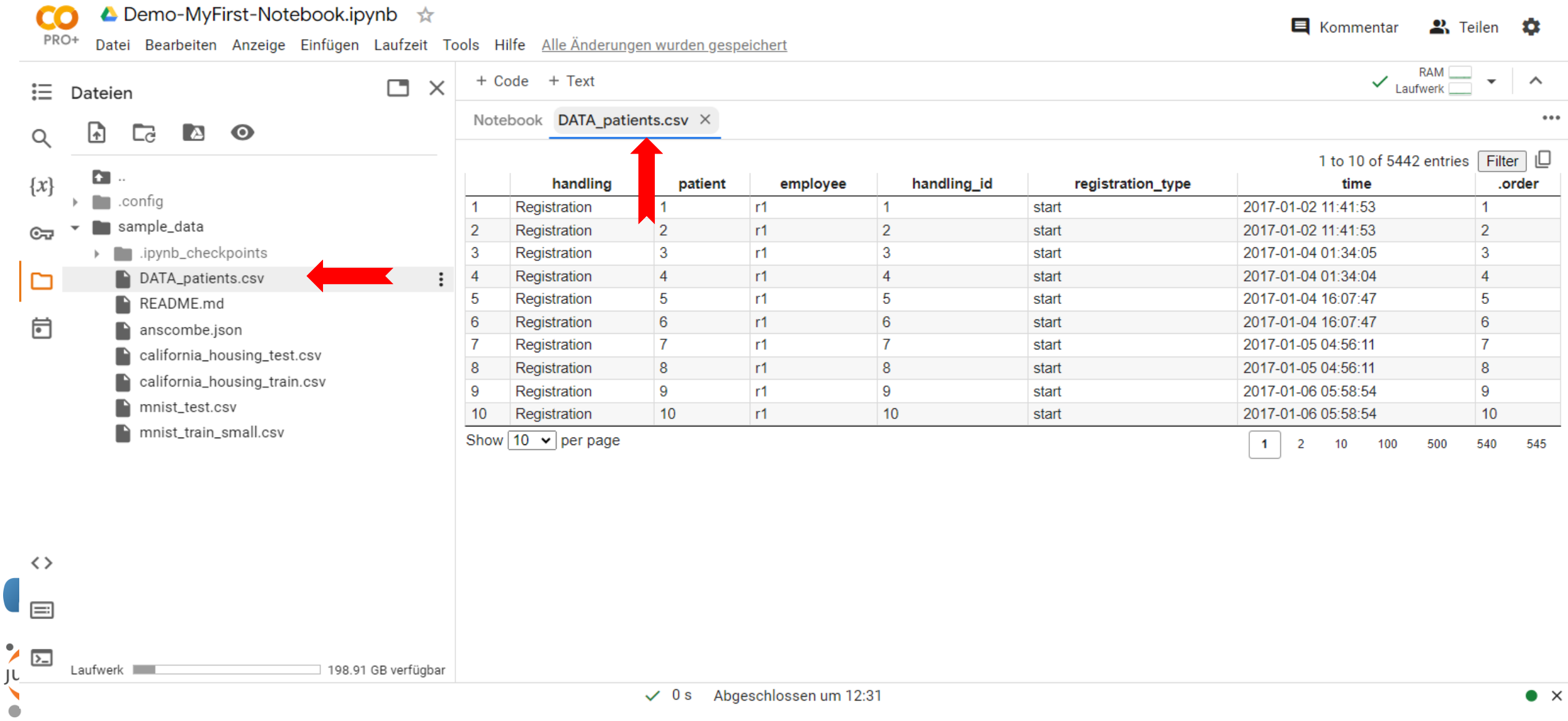
Step 6: Click on box at left side to import (upload) files



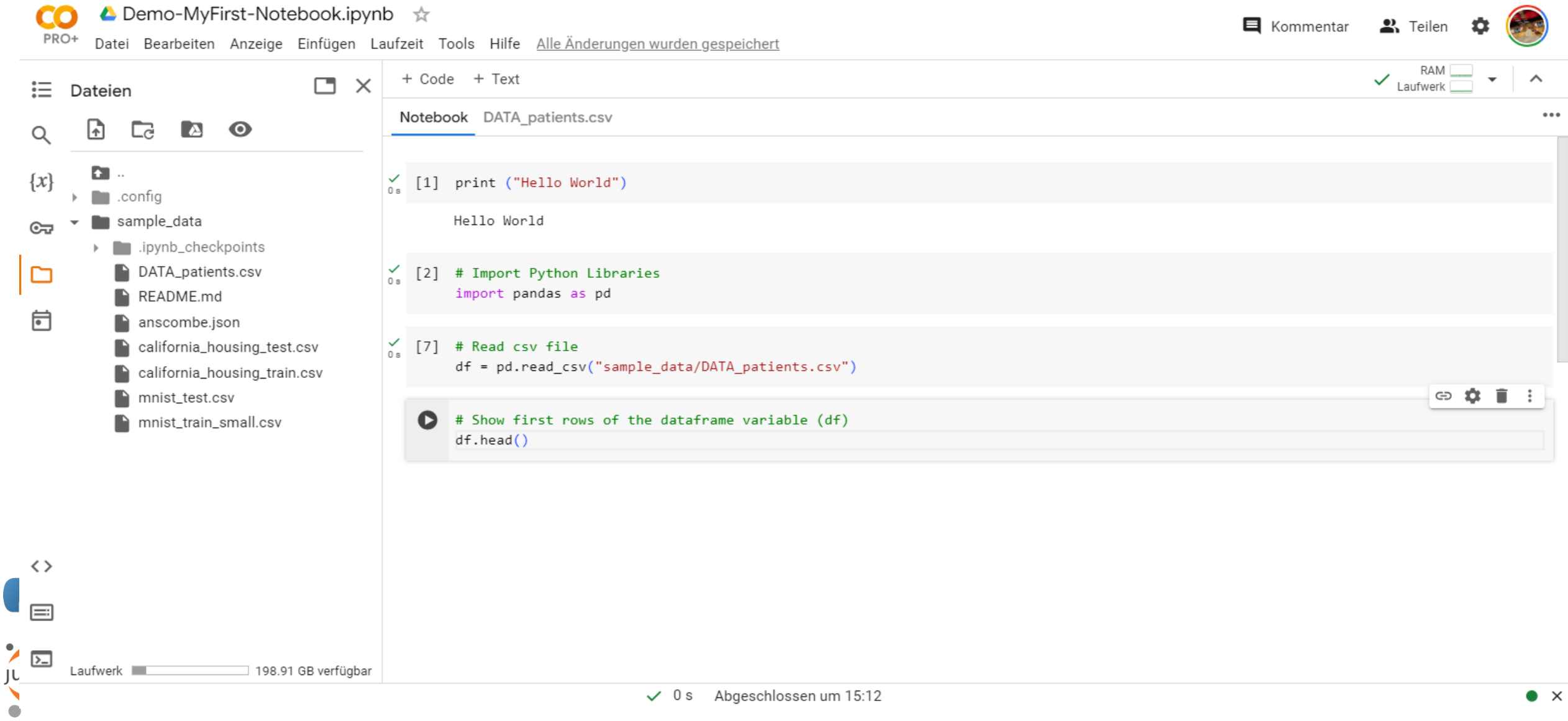
Step 7a: Select (local) file to upload



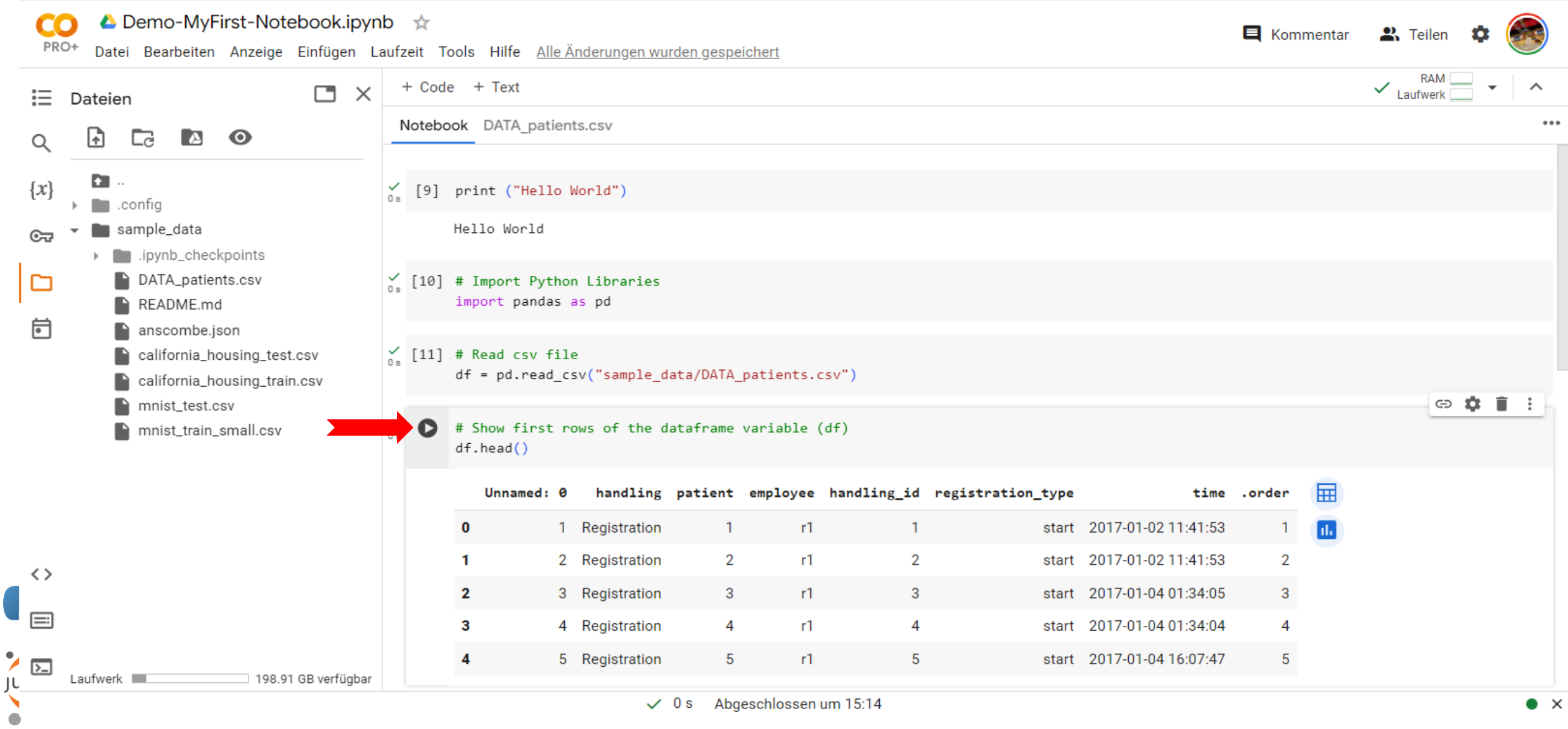
Step 7b: Show/Inspect uploaded file



Step 8a: **Write** program to load/read data from a csv file



Step 8b: **Execute** program to load/read data from a csv file



What is Python?

- **Python** is an *interpreted*, high level and general-purpose programming language.
- **Python** is a *general-purpose* coding language—which means that, unlike HTML, CSS, and JavaScript, it can be used for other types of programming and software development besides web development.

2023 © dr. daniel benninger

Why Python?

- Python works on **different platforms** (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a **simple syntax** similar to the English language with influence from mathematics.
- Python has **syntax** that allows developers to *write programs with fewer lines* than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. *Thus prototyping can be very quick.*
- Python can be treated in a procedural way, an object-oriented way or a functional way.

2023 © dr. daniel benninger

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

jupyter colab

2023 © dr. daniel benninger

Multi Words Variable Names

- Variable names with more than one word can be difficult to read. There are several techniques you can use to make them more readable:
- **Pascal Case**
  - Each word starts with a capital letter:
    e.g. MyVariableName = "John"
- **Camel Case**
  - Each word, except the first, starts with a capital letter:
    e.g. myVariableName = "John"
- **Snake Case**
  - Each word is separated by an underscore character**:**
    e.g. my_variable_name = "John"

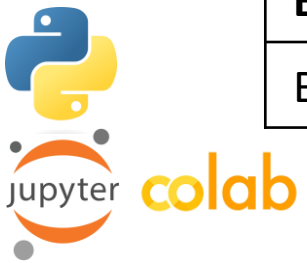
2023 © dr. daniel benninger

Data Type in Python

Built-in Data Types

- In programming, data type is an important concept.
- Variables can store data of different types,
  and different types can do different things.
- Python has the following data types built-in by default, in these categories:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview



2023 © dr. daniel benninger

Data Type in Python

- You can get the data type of any object by using the **type()** function

```
x = 5
print(type(x))
```

2023 © dr. daniel benninger

Setting the (implicit) Data Type

- In Python, the data type is set when you assign a value to a variable:

Example	Data Type
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set
`x = frozenset({"apple", "banana", "cherry"})`	frozenset
`x = True`	bool
`x = b"Hello"`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

2023 © dr. daniel benninger

Setting the Specific Data Type

If you want to specify the data type, you can use the following **constructor functions**:

Example	Data Type
`x = str("Hello World")`	str
`x = int(20)`	int
`x = float(20.5)`	float
`x = complex(1j)`	complex
`x = list(("apple", "banana", "cherry"))`	list
`x = tuple(("apple", "banana", "cherry"))`	tuple
`x = range(6)`	range
`x = dict(name="John", age=36)`	dict
`x = set(("apple", "banana", "cherry"))`	set
`x = frozenset(("apple", "banana", "cherry"))`	frozenset
`x = bool(5)`	bool
`x = bytes(5)`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

2023 © dr. daniel benninger

Python Libraries for Data Science

Some popular Python toolboxes/libraries:

- Data Processing and Modeling
  - **NumPy**
  - SciPy
  - **Pandas**
  - SciKit-Learn
  - TensorFlow
  - Keras
- Data Mining
  - scrapy

Visualization libraries

- **matplotlib**
- Seaborn
- plotly
- pydot

….. and many more …..

2023 © dr. daniel benninger

Python Libraries for Data Science

Loading Python Libraries

In [ ]:
```
#Import Python Libraries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
import seaborn as sns
```

Press `Shift+Enter` to execute the *jupyter* cell

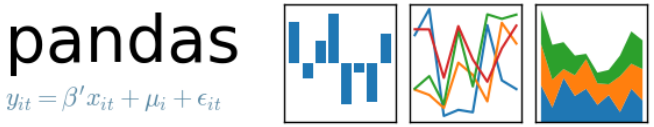
2023 © dr. daniel benninger

Python Libraries for Data Science

NumPy

- NumPy (*Numerical Python*) is a perfect tool for scientific computing and performing basic and advanced *array operations*.
- introduces objects for *multidimensional arrays and matrices*, as well as functions that allow to easily perform advanced *mathematical and statistical operations* on those objects
- provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance
- many other python libraries are built on NumPy

Link: http://www.numpy.org/



Python Libraries for Data Science

Pandas

- Pandas is a library created to help developers work with "*labeled" and "relational" data.*
- adds data structures and tools designed *to work with table-like data* (similar to Series and Data Frames in R)
- provides tools for *data manipulation*: reshaping, merging, sorting, slicing, aggregation etc.
- allows *handling missing data*

Link: http://pandas.pydata.org/

2023 © dr. daniel benninger

Python Libraries for Data Science

Reading data using pandas

```
In [ ]: #Read csv file
        df = pd.read_csv("Salaries.csv")
```

Note: The data import process.

There is a number of pandas commands to read other data formats:

```
pd.read_excel('myfile.xlsx',sheet_name='Sheet1', index_col=None, na_values=['NA'])
pd.read_stata('myfile.dta')
pd.read_sas('myfile.sas7bdat')
pd.read_hdf('myfile.h5','df')
```

2023 © dr. daniel benninger

Python Libraries for Data Science

matplotlib

- A standard data science library that helps to *generate data visualizations* such as two-dimensional diagrams and graphs (histograms, line plots, pie charts, scatterplots).
- Matplotlib is one of those plotting libraries that are very useful in data science projects — it provides an object-oriented API for embedding plots into applications.
- relatively low-level; *some effort needed to create advanced visualization*

Link: https://matplotlib.org/

2023 © dr. daniel benninger

Python Libraries for Data Science

Seaborn

- Seaborn is *based on Matplotlib* and serves as a useful Python machine learning *tool for visualizing statistical models* – heatmaps and other types of visualizations that summarize data and depict the overall distributions.
- When using this library, you get to benefit from *an extensive gallery of visualizations* (including complex ones like time series, joint plots, and violin diagrams).
- provides *high level interface* for drawing attractive statistical graphics

Link: https://seaborn.pydata.org/

Python Libraries for Data Science

Plotly

- A web-based tool for data visualization that offers many useful out-of-box graphics – you can find them on the plotly webseite
- The library works very well in interactive web applications.
- Its creators are busy expanding the library with new graphics and features for supporting multiple linked views, animation, and crosstalk integration.

Link: https://plotly.com/

2023 © dr. daniel benninger

Python Libraries for Data Science

Scrapy

- One of the most popular Python data science libraries, Scrapy helps to *build crawling programs* (spider bots) that can retrieve structured data from the web – for example, URLs or contact info.
- It's a great tool for scraping data used in, for example, Python machine learning models.
- Developers use it for gathering data from APIs. This full-fledged framework follows the Don't Repeat Yourself principle in the design of its interface. As a result, the tool inspires users to write universal code that can be reused for building and scaling large crawlers.

Link: https://scrapy.org/

Python Libraries for Data Science

SciPy*:*

- A *large collection of algorithms* for linear algebra, differential equations, numerical integration, optimization, statistics and more
- part of SciPy Stack
- built on NumPy

Link: https://www.scipy.org/scipylib/

Python Libraries for Data Science

SciKit-Learn*:*

- This is an *industry-standard for data science projects* based in Python.
- Scikits is *a group of packages* in the SciPy Stack that were created for specific functionalities – for example, image processing. Scikit-learn uses the math operations of SciPy to expose a concise interface to the most common machine learning algorithms.
- Data scientists use it for handling *standard machine learning and data mining tasks* such as classification, clustering, regression, model selection, model validation, dimensionality reduction, and classification.
- built on NumPy, SciPy and matplotlib

Link: http://scikit-learn.org/

Python Libraries for Data Science

TensorFlow

- TensorFlow is a popular Python framework for machine learning and deep learning, which was developed at Google Brain.
- It's the best tool for tasks like object identification, speech recognition, and many others. It helps in working with artificial neural networks that need to handle multiple data sets.
- The library includes various layer-helpers (tflearn, tf-slim, skflow), which make it even more functional.
- TensorFlow is constantly expanded with its new releases – including fixes in potential security vulnerabilities or improvements in the integration of TensorFlow and GPU.

Link: https://www.tensorflow.org/

Python Libraries for Data Science

Keras

- Keras is a *great library for building neural networks and modeling*.
- It's very *straightforward to use* and provides developers with a *good degree of extensibility.*
- The library takes advantage of other packages, (Theano or TensorFlow) as its backends. Moreover, Microsoft integrated CNTK (Microsoft Cognitive Toolkit) to serve as another backend.
- It's a *great pick if you want to experiment quickly* using compact systems – the minimalist approach to design really pays off!

Link: https://keras.io/