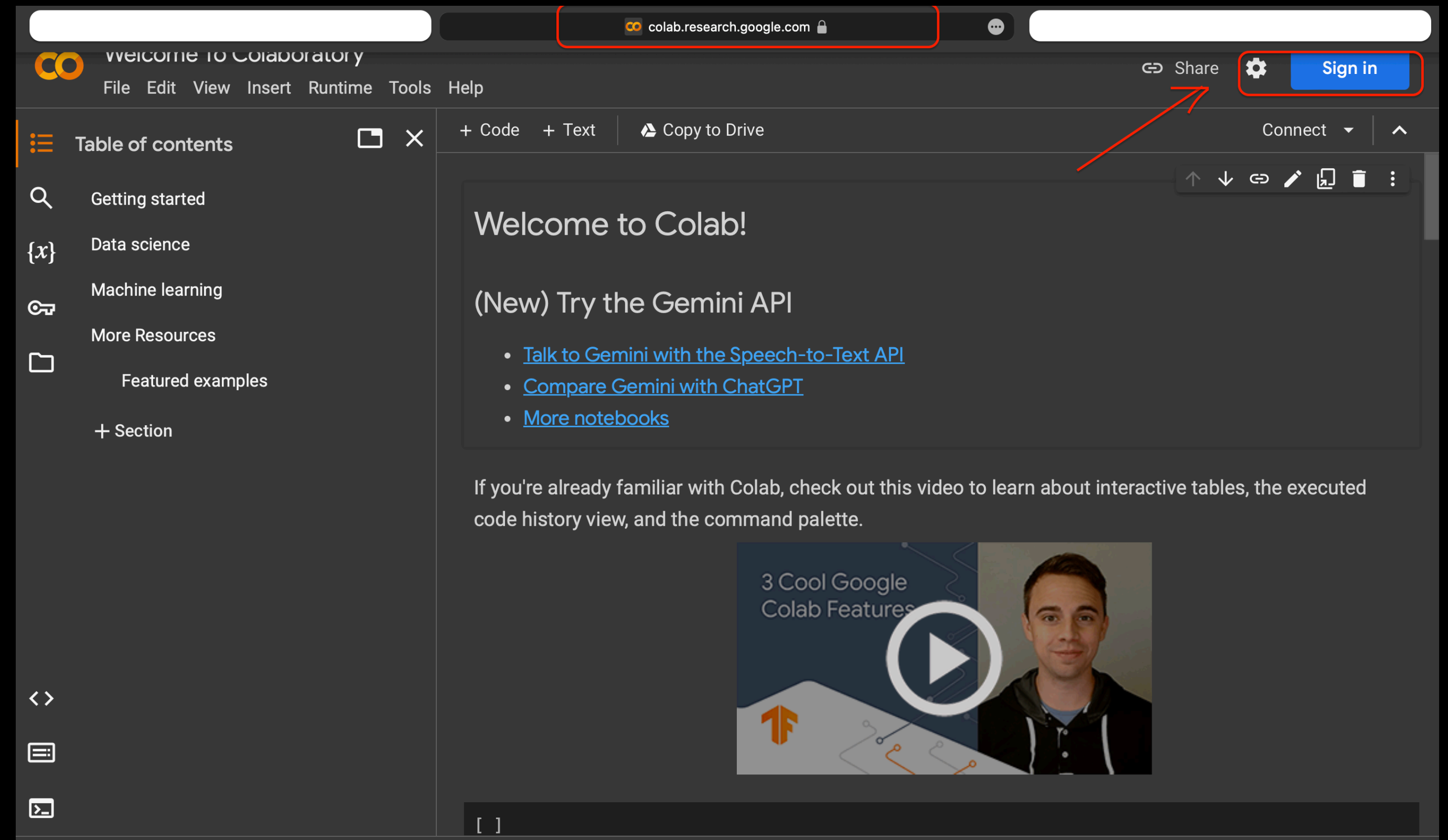


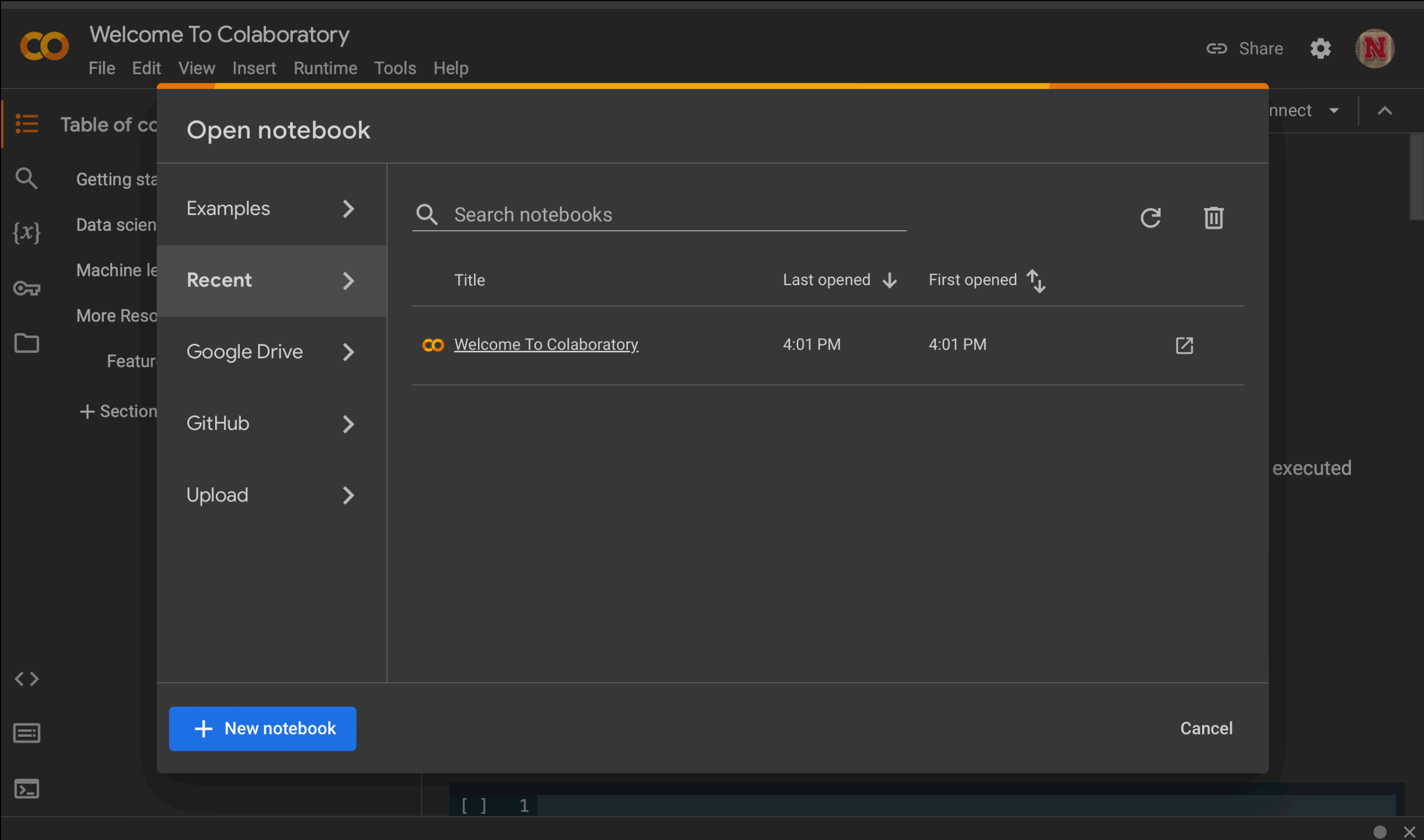
Using Google Colaboratory

Go to colab.research.google.com

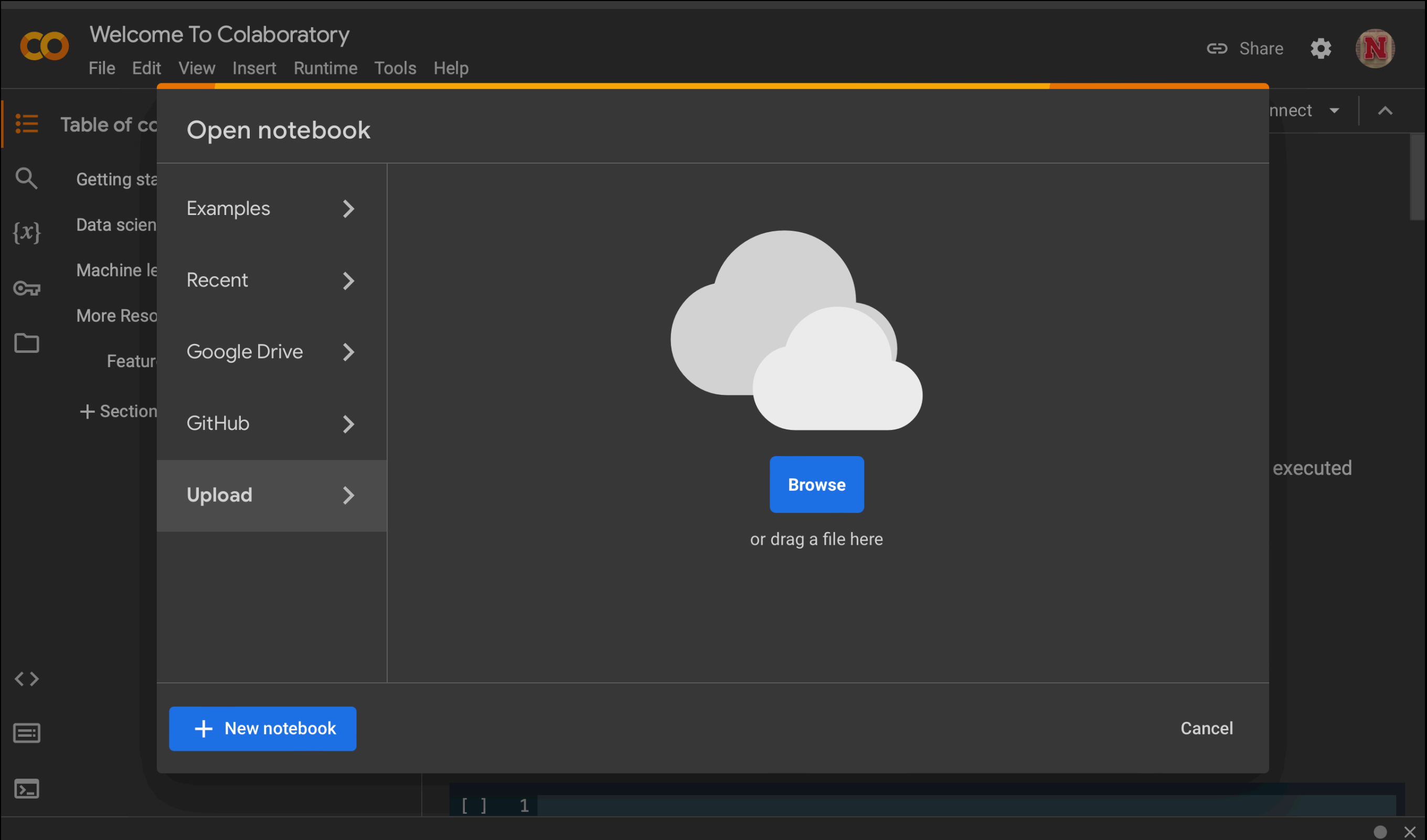
Sign in with your google account



A pop up box like this might
Appear. Go to “Upload”

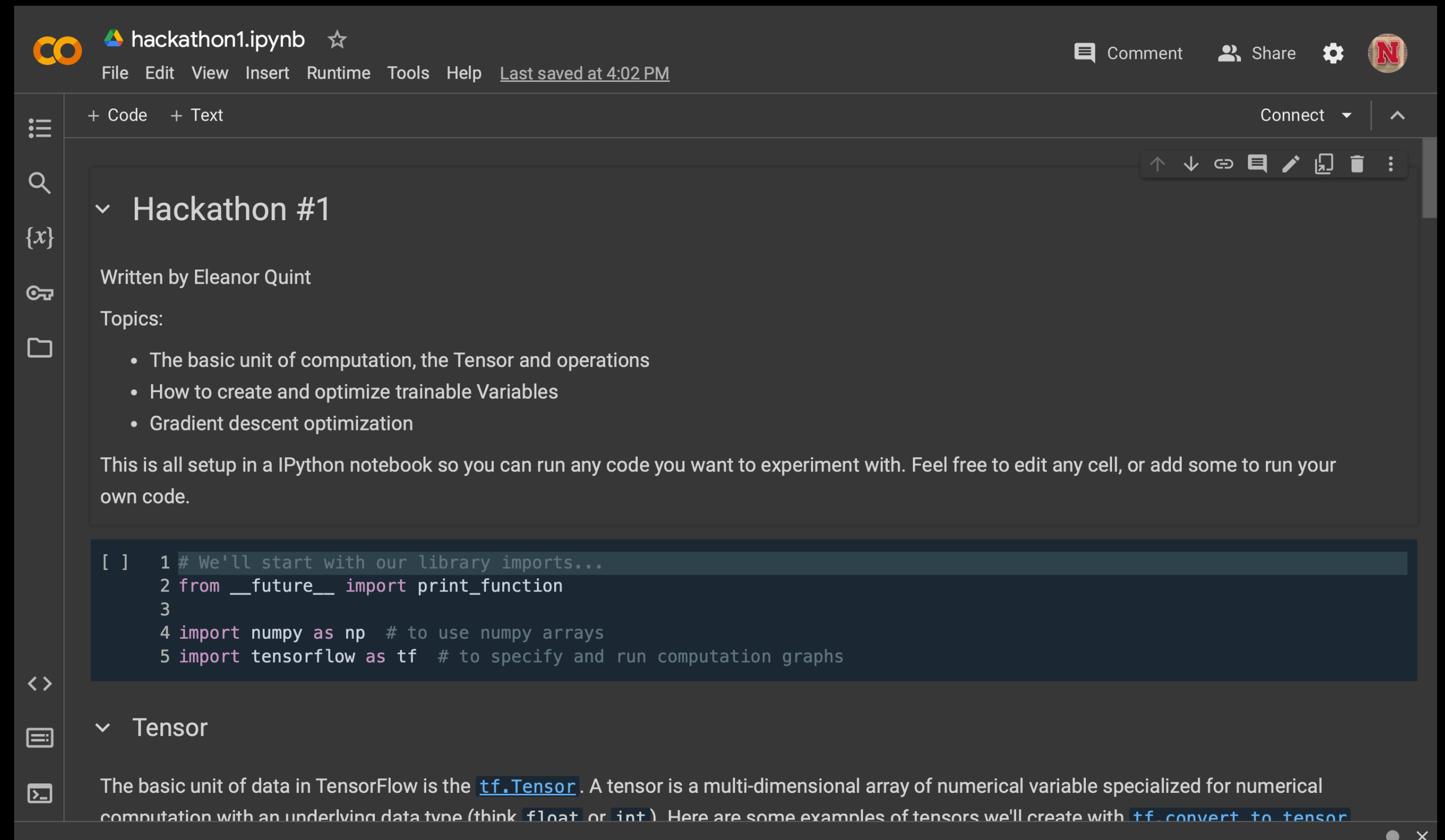


Upload Hackathon file that you'd have downloaded from GitHub



Once uploaded, it will look something like this.

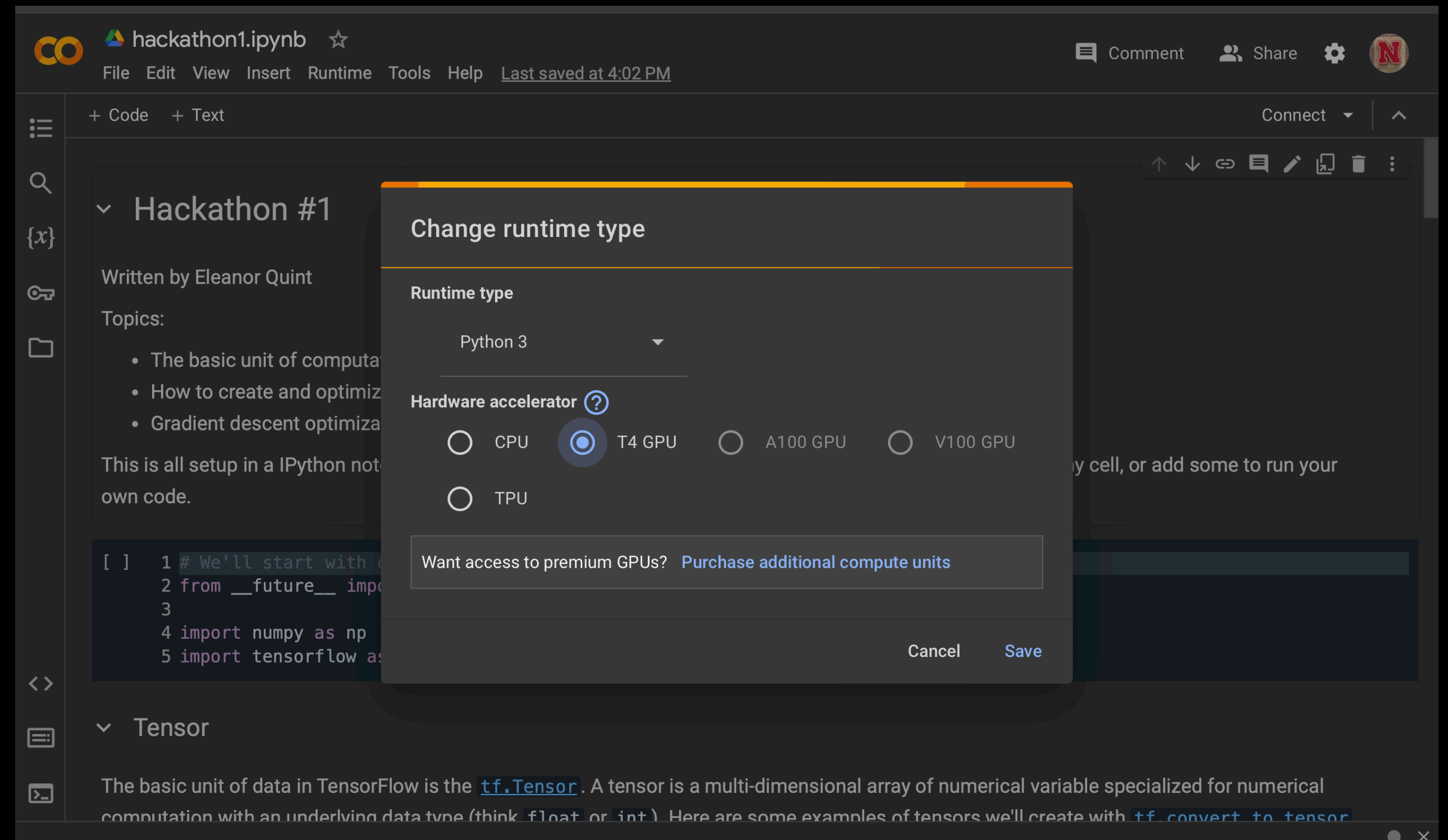
Go to “Connect” on the top right corner of the screen.



Select the option “Change runtime type”.

It brings out this pop-up window.

Select the GPU option.



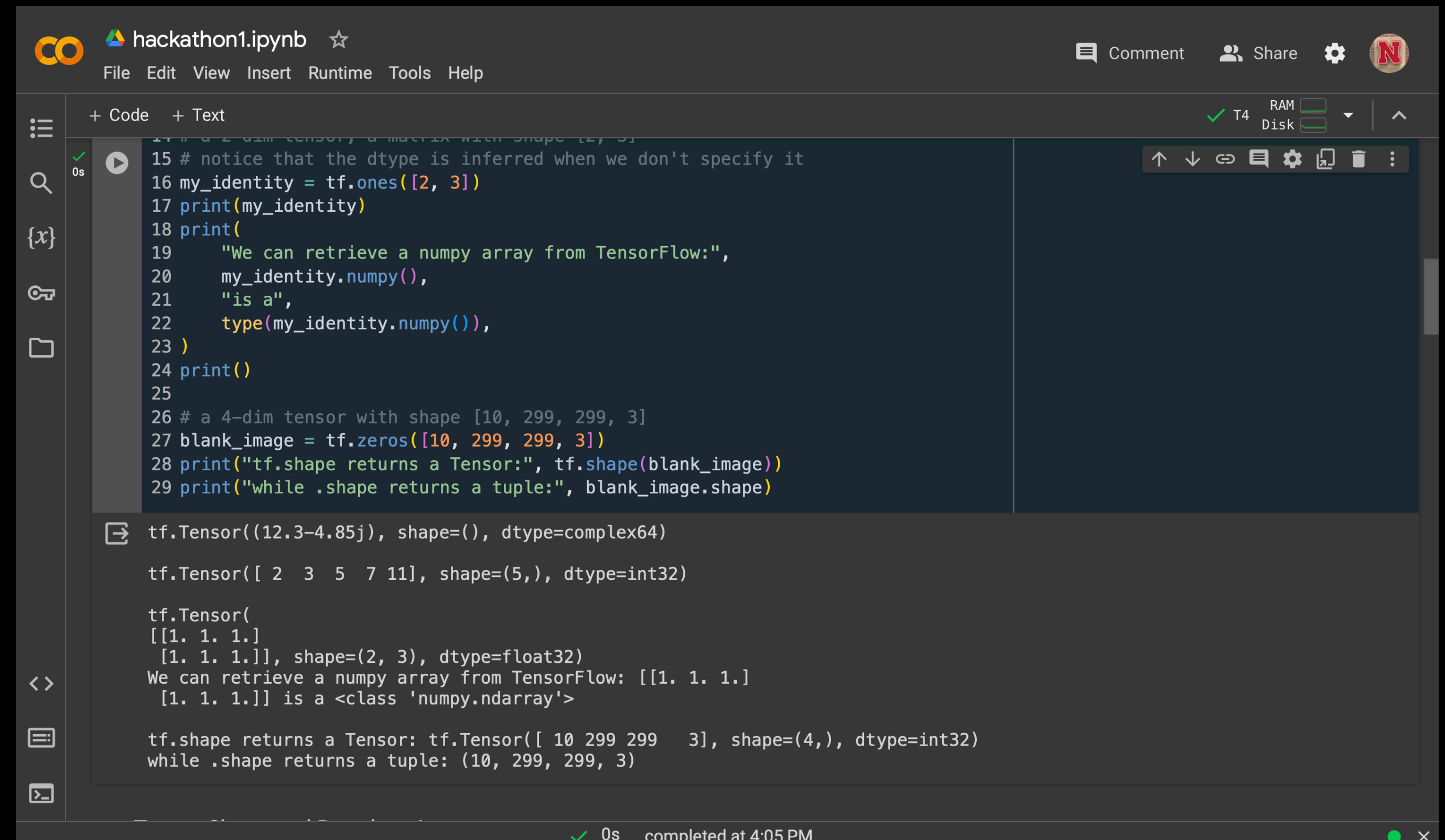
That's it. You can run the code blocks and observe the output.

The screenshot shows a Google Colab notebook interface. At the top, the title bar reads "hackathon1.ipynb" with a star icon. Below it is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", followed by a status message "All changes saved". On the right side of the title bar are icons for "Comment", "Share", a settings gear, and a profile picture. Below the title bar, there's a toolbar with "+ Code" and "+ Text" buttons. To the right of the toolbar are status indicators for "T4" (with a green checkmark), "RAM" (with a progress bar), and "Disk" (with a progress bar). The main content area is divided into sections. The first section is titled "Hackathon #1" and contains the text "Written by Eleanor Quint". Below this, under the heading "Topics:", there is a bulleted list: "The basic unit of computation, the Tensor and operations", "How to create and optimize trainable Variables", and "Gradient descent optimization". A paragraph follows: "This is all setup in a IPython notebook so you can run any code you want to experiment with. Feel free to edit any cell, or add some to run your own code." Below this is a code cell with the following Python code:

```
[ ] 1 # We'll start with our library imports...
    2 from __future__ import print_function
    3
    4 import numpy as np # to use numpy arrays
    5 import tensorflow as tf # to specify and run computation graphs
```

The code cell is highlighted with a blue background. Below the code cell is a section titled "Tensor" which contains the text: "The basic unit of data in TensorFlow is the [tf.Tensor](#). A tensor is a multi-dimensional array of numerical variable specialized for numerical computation with an underlying data type (think [float](#) or [int](#)). Here are some examples of tensors we'll create with [tf.convert_to_tensor](#)". At the bottom of the notebook, there is a status bar that reads "Connected to Python 3 Google Compute Engine backend (GPU)" with a green checkmark and a close button.

This is how it looks when you run
A code block



The screenshot shows a Jupyter Notebook interface with a code block being executed. The code defines a 2D identity matrix and a 4D blank image tensor, then prints their shapes and dtypes. The output shows the tensors and their respective shapes and dtypes.

```
14 # a 2-dim tensor, a matrix with shape [2, 3]
15 # notice that the dtype is inferred when we don't specify it
16 my_identity = tf.ones([2, 3])
17 print(my_identity)
18 print(
19     "We can retrieve a numpy array from TensorFlow:",
20     my_identity.numpy(),
21     "is a",
22     type(my_identity.numpy()),
23 )
24 print()
25
26 # a 4-dim tensor with shape [10, 299, 299, 3]
27 blank_image = tf.zeros([10, 299, 299, 3])
28 print("tf.shape returns a Tensor:", tf.shape(blank_image))
29 print("while .shape returns a tuple:", blank_image.shape)
```

Output:

```
tf.Tensor((12.3-4.85j), shape=(), dtype=complex64)

tf.Tensor([ 2  3  5  7 11], shape=(5,), dtype=int32)

tf.Tensor(
[[1.  1.  1.]
 [1.  1.  1.]], shape=(2, 3), dtype=float32)
We can retrieve a numpy array from TensorFlow: [[1.  1.  1.]
 [1.  1.  1.]] is a <class 'numpy.ndarray'>

tf.shape returns a Tensor: tf.Tensor([ 10 299 299   3], shape=(4,), dtype=int32)
while .shape returns a tuple: (10, 299, 299, 3)
```


Go to Files> Download

To download, you have two options.

