

**COLAB**

# INTRODUÇÃO

# Introdução

# Introdução

- Google tem investido bastante em pesquisas na área de Inteligência Artificial (IA).

# Introdução

- Google tem investido bastante em pesquisas na área de Inteligência Artificial (IA).
- Ferramentas:

# Introdução

- Google tem investido bastante em pesquisas na área de Inteligência Artificial (IA).
- Ferramentas:
  - TensorFlow.

# Introdução

- Google tem investido bastante em pesquisas na área de Inteligência Artificial (IA).
- Ferramentas:
  - TensorFlow.
  - Colaboratory.

# Introdução

- Google tem investido bastante em pesquisas na área de Inteligência Artificial (IA).
- Ferramentas:
  - TensorFlow.
  - Colaboratory.
  - Ambas são ferramentas código-aberto (*open source*).



O que é o COLAB?

# O que é o COLAB?

- É um ambiente aberto de desenvolvimento de notebook Jupyter que roda totalmente em nuvem.

# O que é o COLAB?

- É um ambiente aberto de desenvolvimento de notebook Jupyter que roda totalmente em nuvem.
- Não demanda uma configuração.

# O que é o COLAB?

- É um ambiente aberto de desenvolvimento de notebook Jupyter que roda totalmente em nuvem.
- Não demanda uma configuração.
- Os notebooks criados podem ser simultaneamente editados por todos os membros da equipe.

# O que é o COLAB?

- É um ambiente aberto de desenvolvimento de notebook Jupyter que roda totalmente em nuvem.
- Não demanda uma configuração.
- Os notebooks criados podem ser simultaneamente editados por todos os membros da equipe.
- Suporta as principais bibliotecas de aprendizagem de máquina (*machine learning*).

# Principais Recursos

# Principais Recursos

- Escrever e executar código em Python.

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.



# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.
- Importar e salvar notebooks do/para o Google Drive.

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.
- Importar e salvar notebooks do/para o Google Drive.
- Importar e publicar notebooks do GitHub.

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.
- Importar e salvar notebooks do/para o Google Drive.
- Importar e publicar notebooks do GitHub.
- Importar external datasets (ex: Kaggle).

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.
- Importar e salvar notebooks do/para o Google Drive.
- Importar e publicar notebooks do GitHub.
- Importar external datasets (ex: Kaggle).
- Integrar código do PyTorch, TensorFlow, Keras e OpenCV.

# Principais Recursos

- Escrever e executar código em Python.
- Documentar o seu código para suportar equações matemáticas.
- Criar, enviar e compartilhar notebooks.
- Importar e salvar notebooks do/para o Google Drive.
- Importar e publicar notebooks do GitHub.
- Importar external datasets (ex: Kaggle).
- Integrar código do PyTorch, TensorFlow, Keras e OpenCV.
- Serviço de nuvem gratuito com GPU.

# PRIMEIROS PASSOS

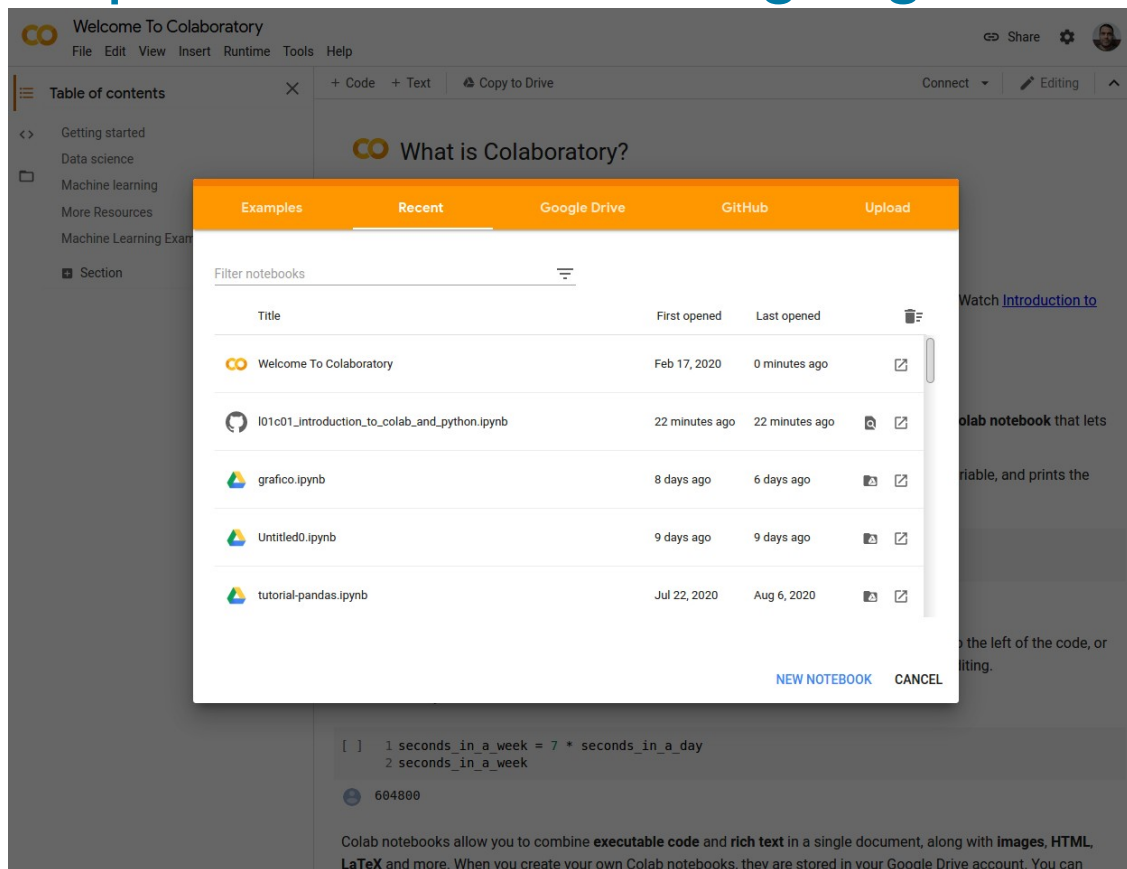
# Primeiro Notebook

- O usuário precisa estar conectado ao Google.



# Primeiro Notebook

- **Passo 1:** o usuário deve abrir a URL abaixo no navegador.
  - <https://colab.research.google.com>

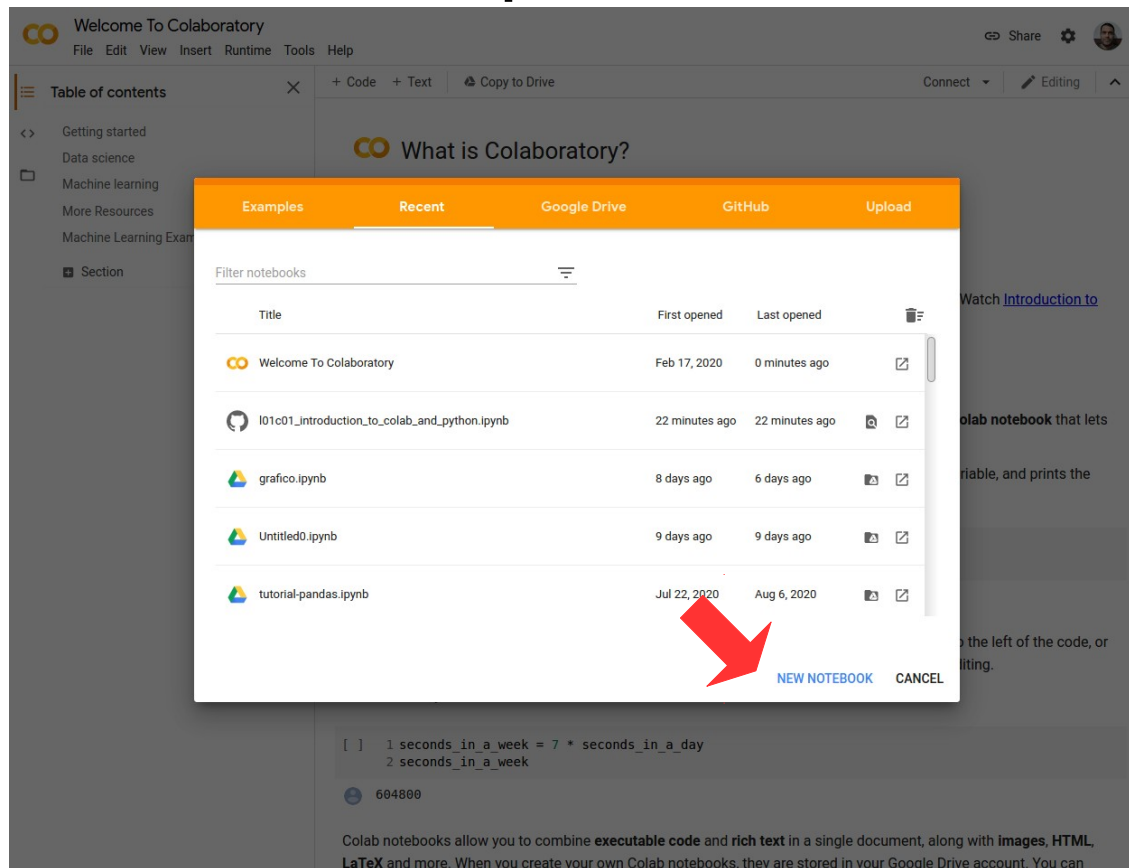


The screenshot shows the Google Colaboratory interface. At the top, there is a 'Welcome To Colaboratory' header with a menu (File, Edit, View, Insert, Runtime, Tools, Help) and a 'Share' button. A red arrow points to the user profile icon in the top right corner. Below the header, there is a 'Table of contents' sidebar on the left and a main content area displaying 'What is Colaboratory?'. A modal window titled 'Recent' is open, showing a list of notebooks with columns for 'Title', 'First opened', 'Last opened', and 'Actions'. The notebooks listed are: 'Welcome To Colaboratory', 'I01c01\_introduction\_to\_colab\_and\_python.ipynb', 'grafico.ipynb', 'Untitled0.ipynb', and 'tutorial-pandas.ipynb'. At the bottom of the modal, there are 'NEW NOTEBOOK' and 'CANCEL' buttons. The background shows a code editor with Python code and a terminal output.

Examples	Recent	Google Drive	GitHub	Upload
Filter notebooks				
Title		First opened	Last opened	
🔗 Welcome To Colaboratory		Feb 17, 2020	0 minutes ago	🗑️
🔗 I01c01_introduction_to_colab_and_python.ipynb		22 minutes ago	22 minutes ago	🗑️
🔗 grafico.ipynb		8 days ago	6 days ago	🗑️
🔗 Untitled0.ipynb		9 days ago	9 days ago	🗑️
🔗 tutorial-pandas.ipynb		Jul 22, 2020	Aug 6, 2020	🗑️

# Primeiro Notebook

- **Passo 2:** Clique em New Notebook.



# Primeiro Notebook

- **Passo 3:** A interface estará pronta para codificação.



# Primeiro Notebook

- **Passo 4:**

# Primeiro Notebook

- **Passo 4:**
  - Por convenção, o notebook terá o nome de `Untitled0.ipynb`.

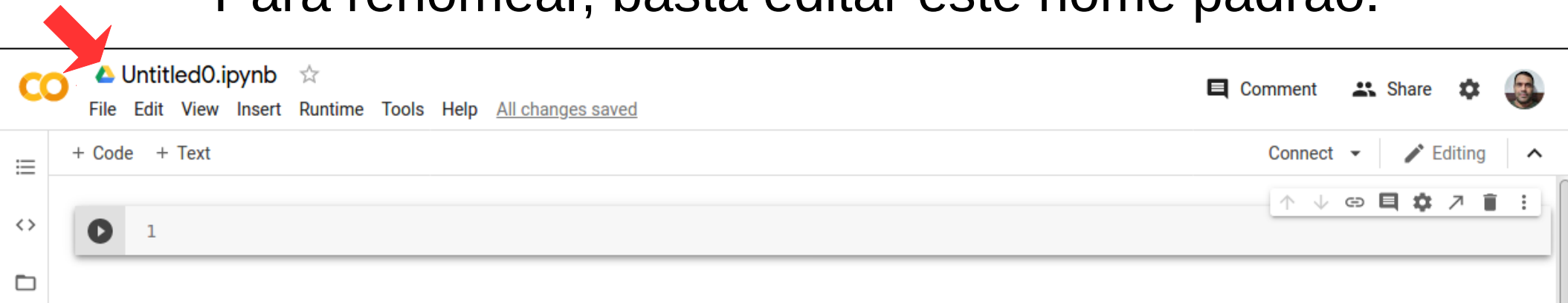
# Primeiro Notebook

- **Passo 4:**
  - Por convenção, o notebook terá o nome de Untitled0.ipynb.



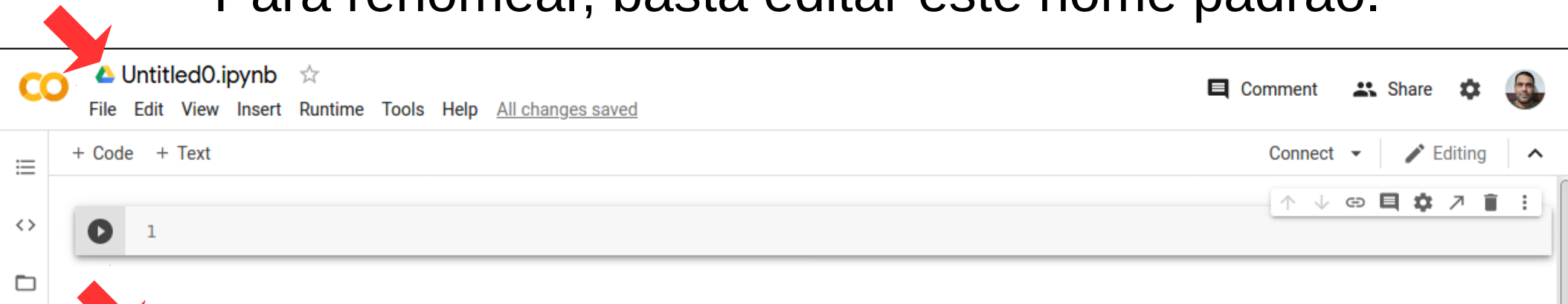
# Primeiro Notebook

- **Passo 4:**
  - Por convenção, o notebook terá o nome de Untitled0.ipynb.
  - Para renomear, basta editar este nome padrão.

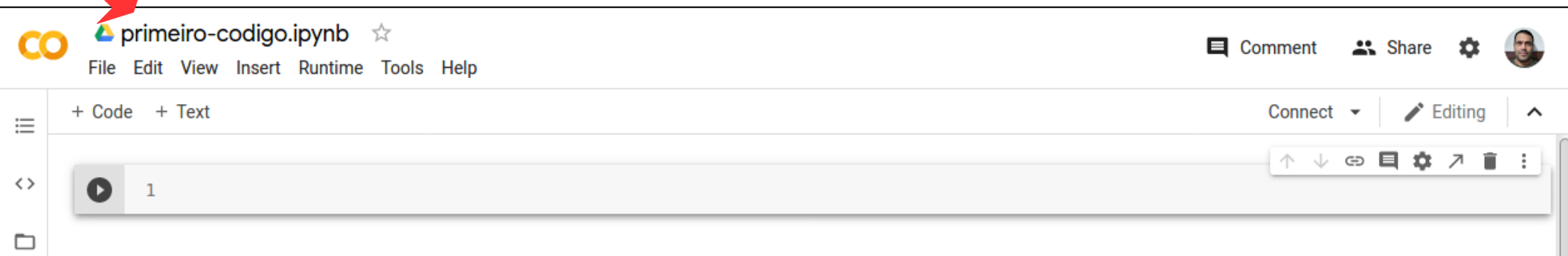


# Primeiro Notebook

- **Passo 4:**
  - Por convenção, o notebook terá o nome de Untitled0.ipynb.
  - Para renomear, basta editar este nome padrão.



A screenshot of a Jupyter Notebook interface. The top bar shows the Colab logo, the notebook name "Untitled0.ipynb", and a star icon. Below the name is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", followed by the text "All changes saved". On the right side of the top bar are icons for "Comment", "Share", a settings gear, and a user profile picture. Below the top bar is a toolbar with "+ Code" and "+ Text" buttons. The main area shows a code cell with a play button icon and the number "1". A red arrow points to the notebook name "Untitled0.ipynb".



A screenshot of a Jupyter Notebook interface, identical in layout to the previous one, but with the notebook name changed to "primeiro-codigo.ipynb". A red arrow points to the new name.



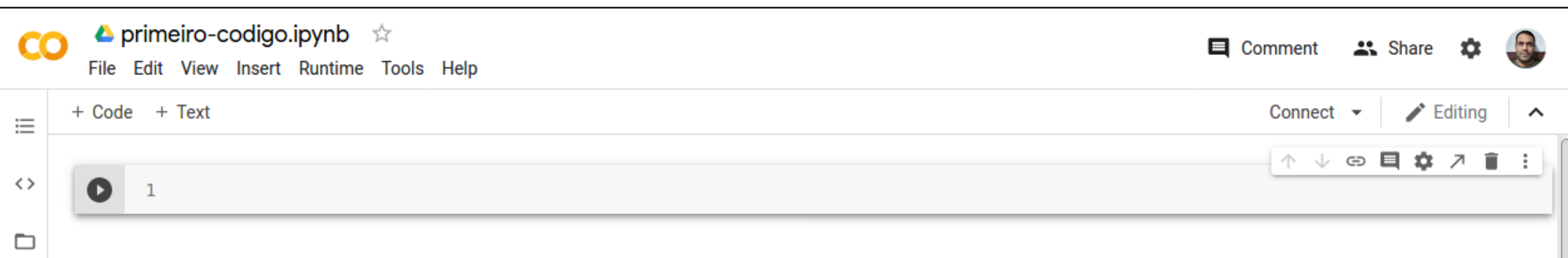
# Codificação

# Codificação

- Segue abaixo um código trivial em Python.

# Codificação

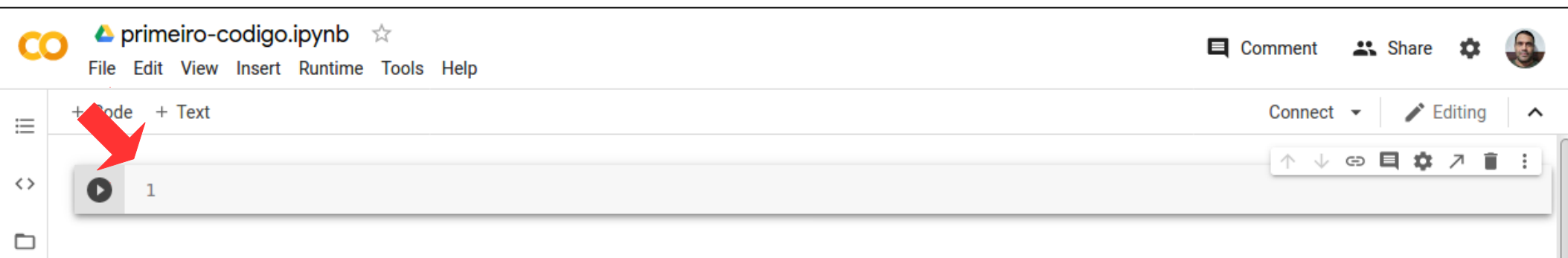
- Segue abaixo um código trivial em Python.



The image shows a screenshot of a Jupyter Notebook interface. At the top left, there is a logo for 'CO' and the text 'primeiro-codigo.ipynb' with a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the top right, there are icons for 'Comment', 'Share', a settings gear, and a user profile picture. Below the menu bar, there are buttons for '+ Code' and '+ Text'. On the right side of the notebook area, there are buttons for 'Connect', 'Editing', and an upward arrow. The main area of the notebook contains a single code cell with a play button icon and the number '1'.

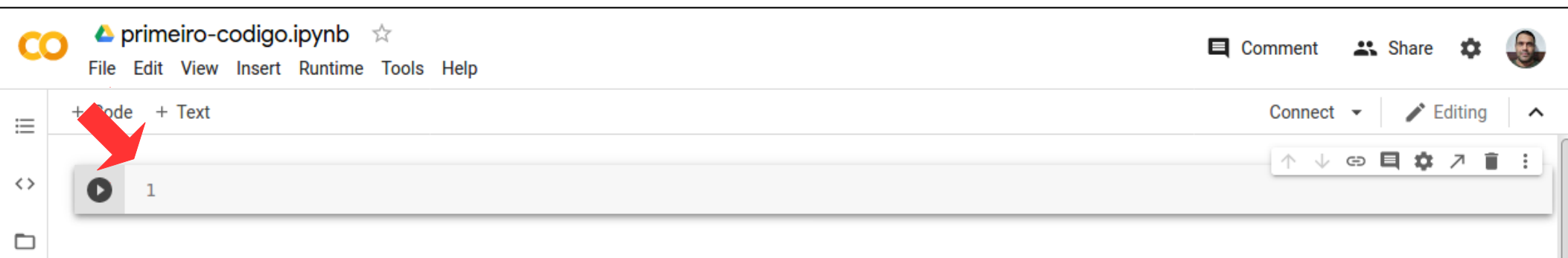
# Codificação

- Segue abaixo um código trivial em Python.

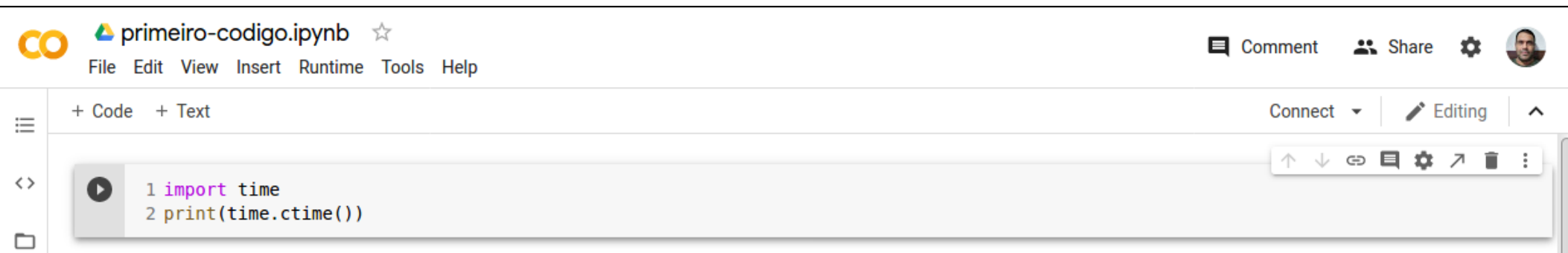


# Codificação

- Segue abaixo um código trivial em Python.



The screenshot shows the top part of a Jupyter Notebook interface. The title bar reads "primeiro-codigo.ipynb" with a star icon. Below the title bar is a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side, there are icons for "Comment", "Share", and a user profile. The main area shows a code cell with a play button icon on the left and the number "1" in the center. A red arrow points to the play button. To the right of the code cell, there are icons for "Connect", "Editing", and a list of actions (up, down, link, comment, settings, refresh, trash, and more).



The screenshot shows the same Jupyter Notebook interface as above, but with two lines of Python code in the code cell:

```
1 import time
2 print(time.ctime())
```

The rest of the interface, including the title bar, menu, and right-side controls, is identical to the previous screenshot.

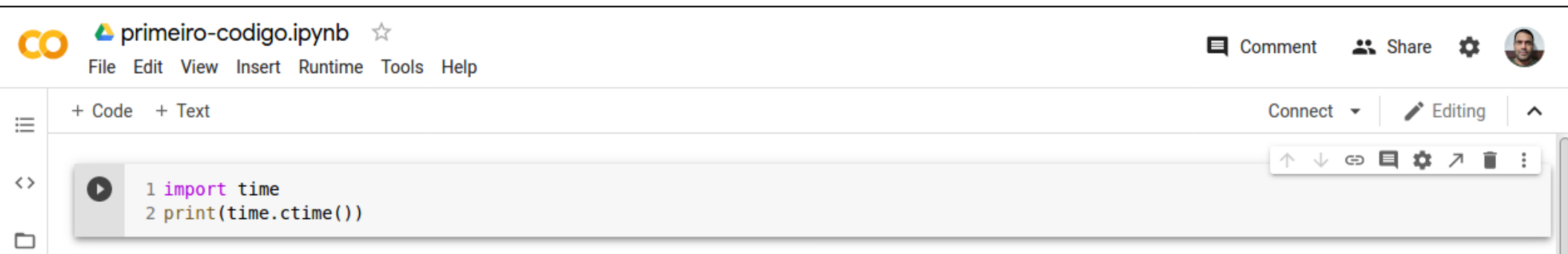
# Execução

# Execução

- Para executar o código, clique na seta à esquerda da janela do código.

# Execução

- Para executar o código, clique na seta à esquerda da janela do código.

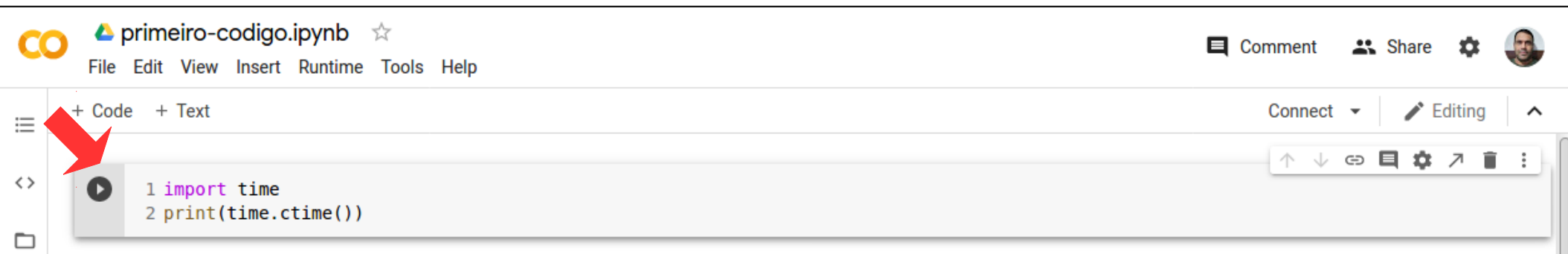


The screenshot shows the top part of a Jupyter Notebook interface. At the top left is the Colab logo and the file name "primeiro-codigo.ipynb" with a star icon. Below it is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the top right, there are icons for "Comment", "Share", a settings gear, and a user profile picture. Below the menu bar, there are tabs for "+ Code" and "+ Text". On the right side of this bar, there are "Connect" and "Editing" options. The main area shows a code cell with a play button icon on the left and two lines of Python code: `1 import time` and `2 print(time.ctime())`. To the right of the code cell, there is a toolbar with icons for up, down, link, comment, settings, refresh, and delete.



# Execução

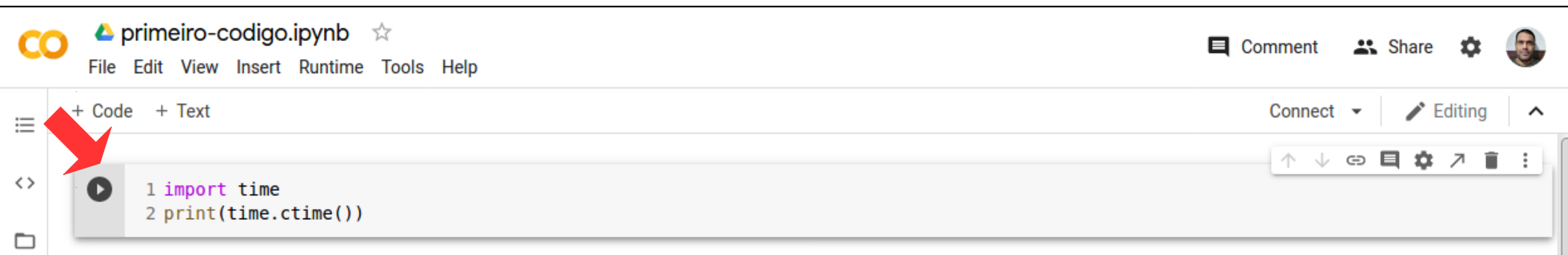
- Para executar o código, clique na seta à esquerda da janela do código.



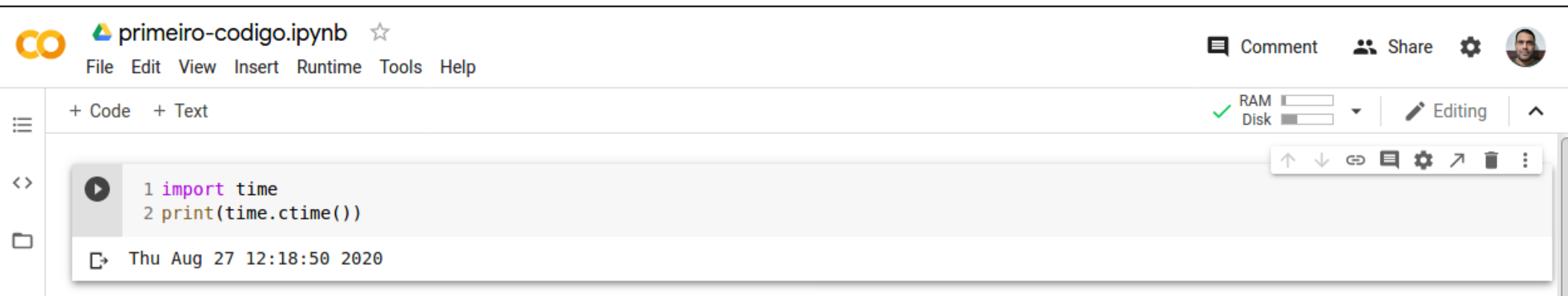
The screenshot shows the top part of a Jupyter Notebook interface. At the top left is the Colab logo and the file name "primeiro-codigo.ipynb" with a star icon. Below that is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side, there are icons for "Comment", "Share", a settings gear, and a user profile picture. Below the menu bar, there are buttons for "+ Code" and "+ Text". A red arrow points to a play button (a right-pointing triangle inside a circle) located on the left side of a code cell. The code cell contains two lines of Python code: "1 import time" and "2 print(time.ctime())". To the right of the code cell, there is a toolbar with icons for "Connect", "Editing", and other actions.

# Execução

- Para executar o código, clique na seta à esquerda da janela do código.



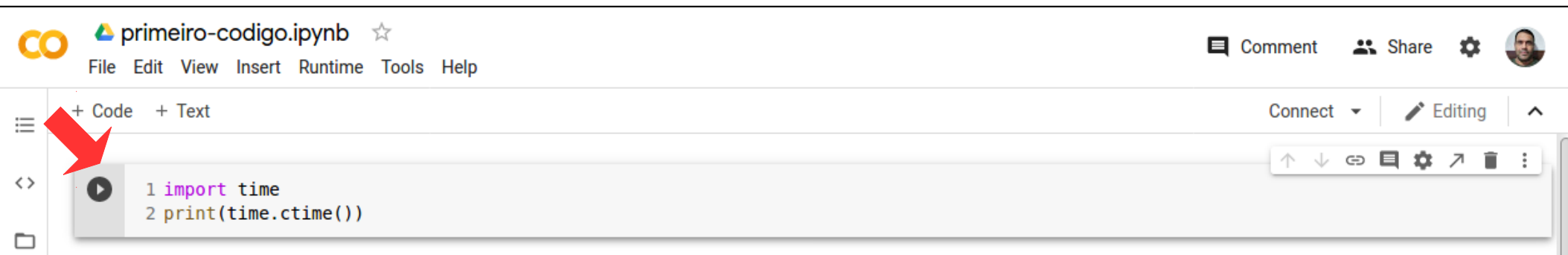
The screenshot shows the top part of a Jupyter Notebook interface. The title bar reads "primeiro-codigo.ipynb" with a star icon. Below it is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right, there are icons for "Comment", "Share", and a user profile. The main area shows a code cell with two lines of Python code: `1 import time` and `2 print(time.ctime())`. A red arrow points to a play button icon on the left side of the code cell. To the right of the code cell, there are icons for "Connect", "Editing", and a list of actions (up, down, link, comment, settings, refresh, trash, and a vertical ellipsis).



The screenshot shows the same Jupyter Notebook interface as above, but now the code cell has been executed. The code is still visible: `1 import time` and `2 print(time.ctime())`. Below the code, a timestamp is displayed: "Thu Aug 27 12:18:50 2020". The play button icon on the left is now greyed out. The right side of the interface shows "RAM" and "Disk" usage indicators, and the "Editing" mode is active.

# Execução

- Para executar o código, clique na seta à esquerda da janela do código.



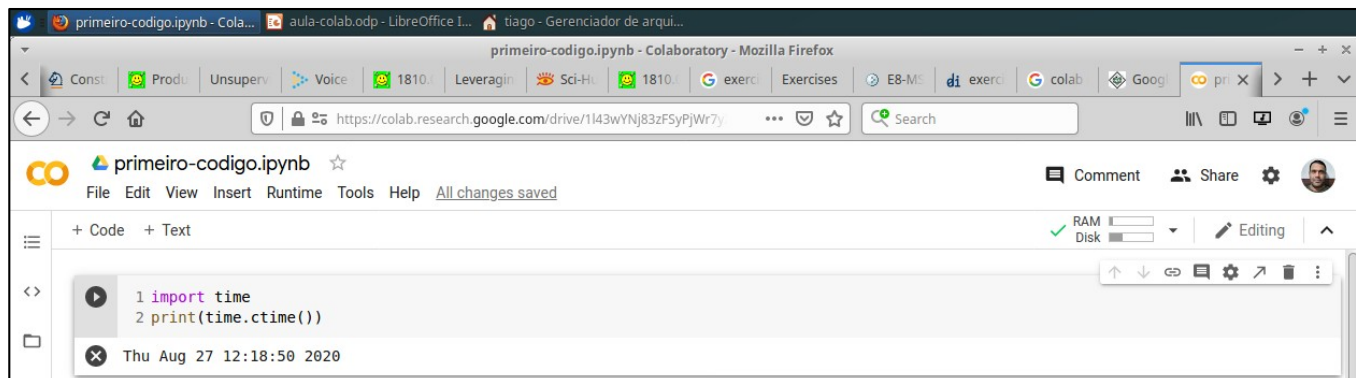
# Execução

# Execução

- Para limpar a saída da execução do programa, basta clicar.

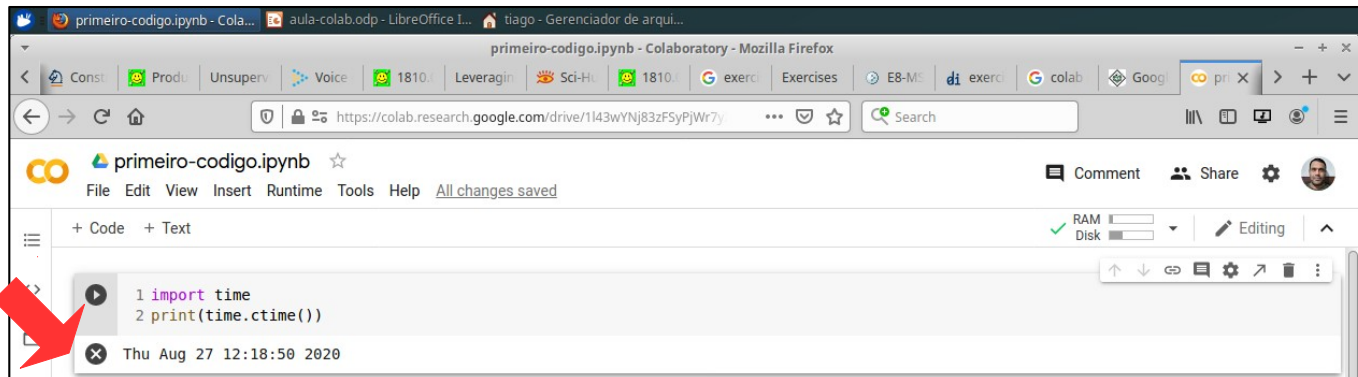
# Execução

- Para limpar a saída da execução do programa, basta clicar.



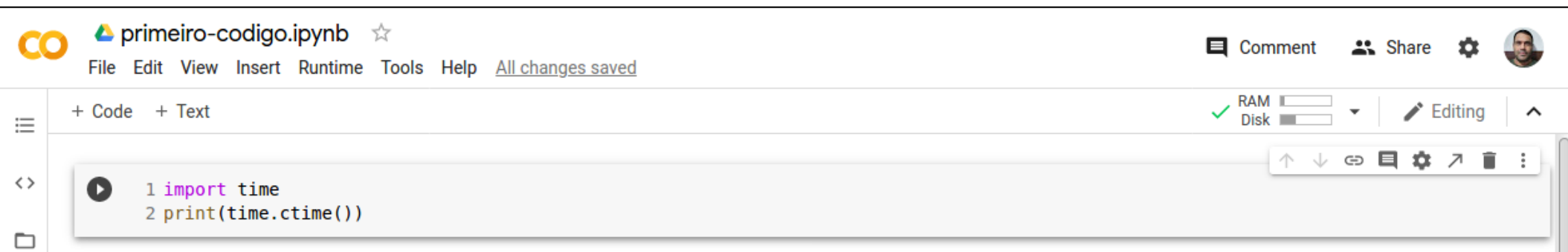
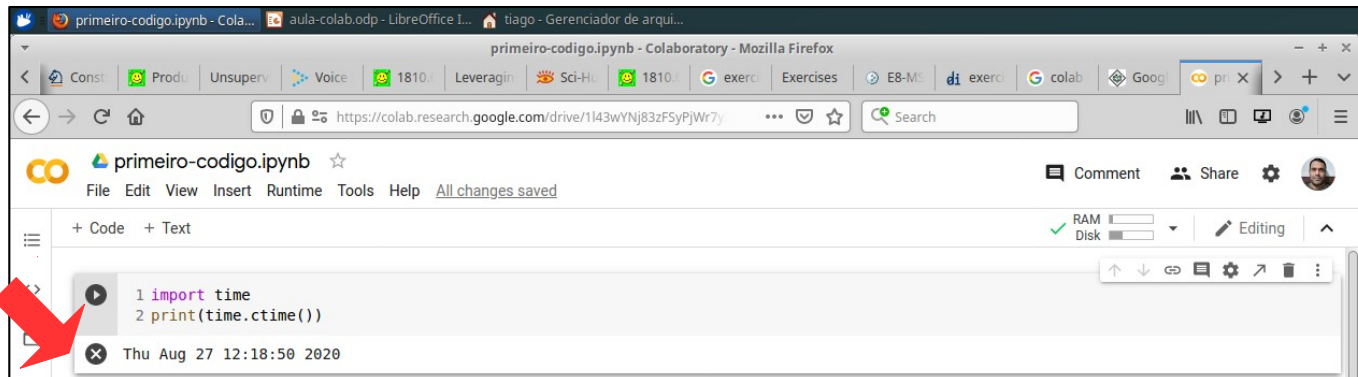
# Execução

- Para limpar a saída da execução do programa, basta clicar.



# Execução

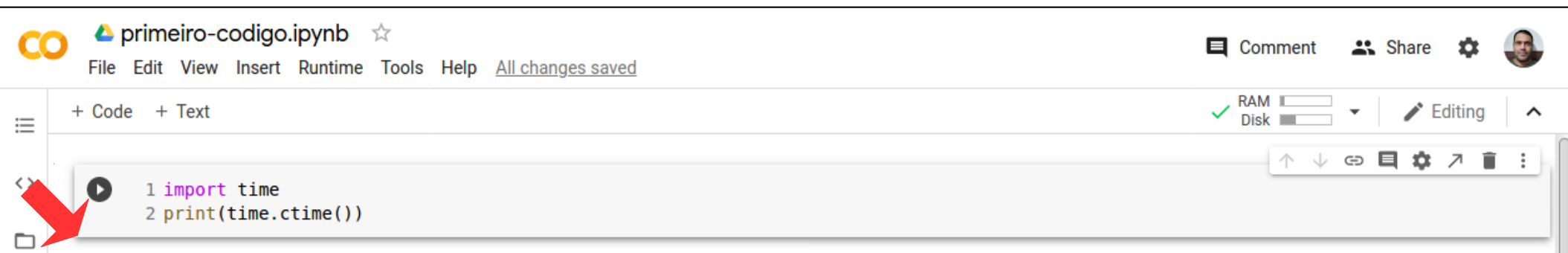
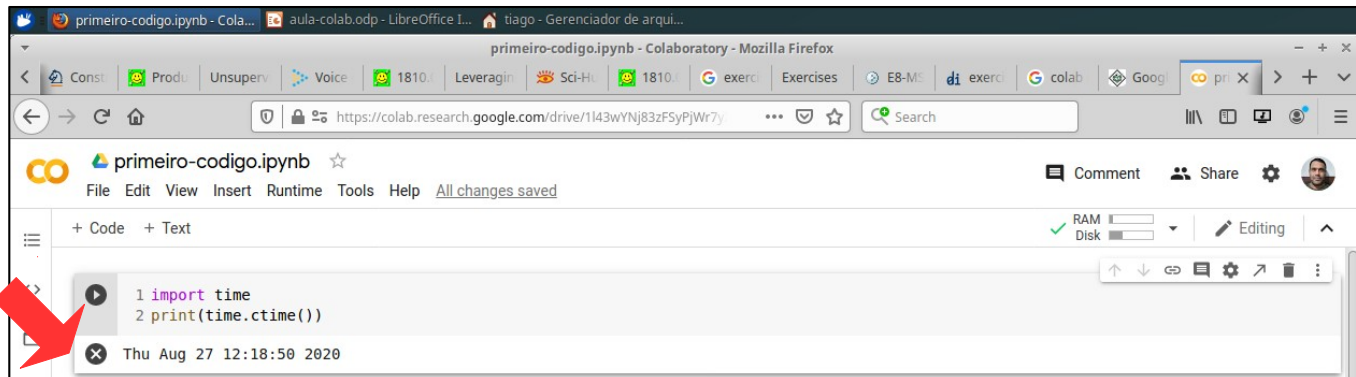
- Para limpar a saída da execução do programa, basta clicar.





# Execução

- Para limpar a saída da execução do programa, basta clicar.



# Adicionando Células de Código

# Adicionando Células de Código

- Para inserir uma célula de código:

# Adicionando Células de Código

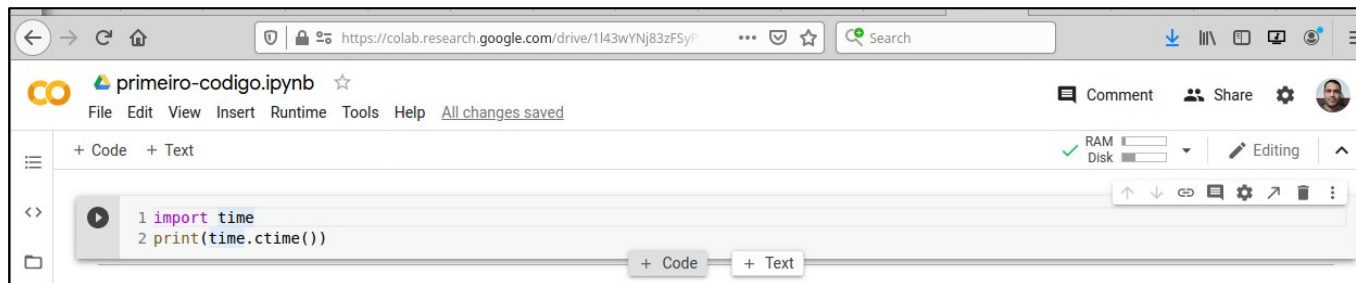
- Para inserir uma célula de código:
  - Insert → Code Cell

# Adicionando Células de Código

- Para inserir uma célula de código:
  - Insert → Code Cell
- Uma outra alternativa:

# Adicionando Células de Código

- Para inserir uma célula de código:
  - Insert → Code Cell
- Uma outra alternativa:



# Adicionando Células de Código

# Adicionando Células de Código

- Digitando o código na nova célula:



# Adicionando Células de Código

- Digitando o código na nova célula:



The screenshot shows a Jupyter Notebook interface. At the top, there is a header with the Colab logo, the file name "primeiro-codigo.ipynb", and a star icon. Below the header is a menu bar with options: File, Edit, View, Insert, Runtime, Tools, Help, and a link for "All changes saved". On the right side of the header, there are icons for "Comment", "Share", a settings gear, and a user profile picture.

Below the header, there is a toolbar with "+ Code" and "+ Text" buttons. To the right of the toolbar, there are RAM and Disk usage indicators, an "Editing" mode indicator, and an upward arrow.

The main area of the notebook contains two code cells. The first cell is labeled "[2]" and contains the following code:

```
1 import time
2 print(time.ctime())
```

Below the code, there is a timestamp: "Fri Aug 28 00:59:02 2020".

The second cell is a code cell with a play button icon on the left and contains the following code:

```
1 time.sleep(5)
2 print (time.ctime())
```

Below the code, there is a timestamp: "Fri Aug 28 00:59:08 2020".

On the right side of the second cell, there is a context menu with icons for up, down, link, comment, settings, share, delete, and a vertical ellipsis.

# Executando Todo Código

# Executando Todo Código

- Para executar o código inteiro (todas as células) sem interrupção, basta usar a opção do menu:

# Executando Todo Código

- Para executar o código inteiro (todas as células) sem interrupção, basta usar a opção do menu:
  - Runtime → Reset and run all

# Mudando Ordem Células

# Mudando Ordem Células

- Quando o notebook tem muitas células, é possível que o programador precise alterar a ordem de execução.

# Mudando Ordem Células

- Quando o notebook tem muitas células, é possível que o programador precise alterar a ordem de execução.
- Isto pode ser feito selecionando as células para CIMA ou para BAIXO.

# Mudando Ordem Células

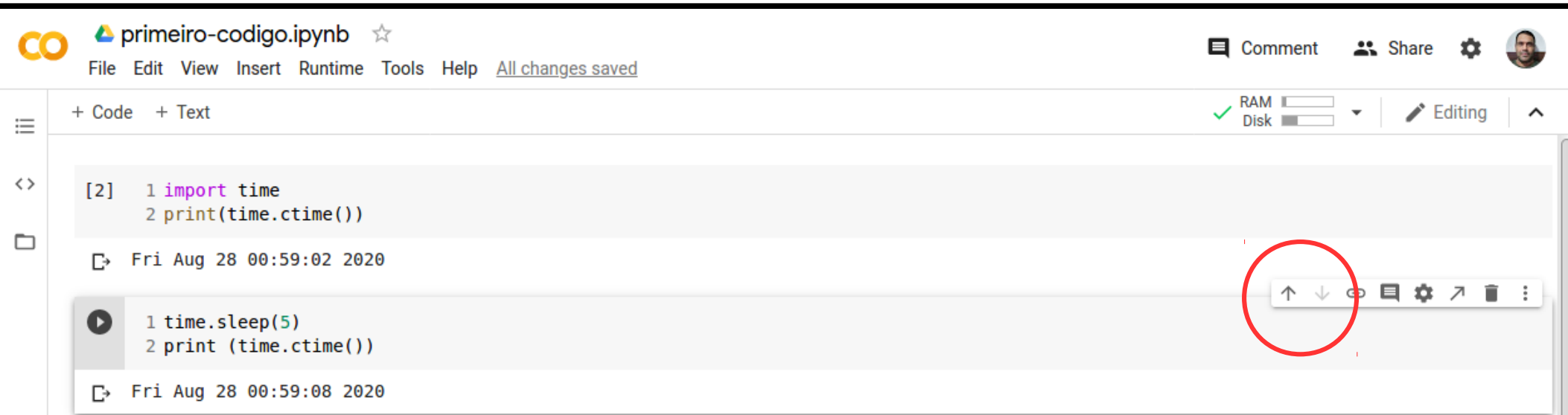
- Quando o notebook tem muitas células, é possível que o programador precise alterar a ordem de execução.
- Isto pode ser feito selecionando as células para CIMA ou para BAIXO.

The screenshot shows a Jupyter Notebook interface. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A status bar indicates 'All changes saved'. On the right, there are buttons for 'Comment', 'Share', and a user profile icon. Below the menu bar, there are '+ Code' and '+ Text' buttons. A toolbar shows 'RAM' and 'Disk' usage indicators, and an 'Editing' mode button. The notebook content consists of two code cells. The first cell, labeled '[2]', contains the code: `1 import time` and `2 print(time.ctime())`. Below it is a timestamp: 'Fri Aug 28 00:59:02 2020'. The second cell contains the code: `1 time.sleep(5)` and `2 print (time.ctime())`. Below it is a timestamp: 'Fri Aug 28 00:59:08 2020'. A cell toolbar at the bottom right of the second cell includes 'up', 'down', 'link', 'comment', 'settings', 'refresh', 'trash', and 'more' icons.



# Mudando Ordem Células

- Quando o notebook tem muitas células, é possível que o programador precise alterar a ordem de execução.
- Isto pode ser feito selecionando as células para CIMA ou para BAIXO.



The screenshot shows a Jupyter Notebook interface with the following elements:

- Top bar: "primeiro-codigo.ipynb" with a star icon, and navigation links for File, Edit, View, Insert, Runtime, Tools, and Help. A status message "All changes saved" is visible.
- Right side: "Comment", "Share", and a user profile icon.
- RAM and Disk usage indicators are shown in the top right.
- Left sidebar: "+ Code" and "+ Text" buttons.
- Main area: Two code cells. The first cell contains:

```
[2] 1 import time
     2 print(time.ctime())
```

with a timestamp "Fri Aug 28 00:59:02 2020". The second cell contains:

```
1 time.sleep(5)
2 print (time.ctime())
```

with a timestamp "Fri Aug 28 00:59:08 2020".
- Bottom right of the selected cell: A toolbar with icons for up, down, copy, comment, settings, refresh, delete, and a menu. This toolbar is circled in red.

# Mudando Ordem Células

- Quando o notebook tem muitas células, é possível que o programador precise alterar a ordem de execução.
- Isto pode ser feito selecionando as células para CIMA ou para BAIXO.



The screenshot shows a Jupyter Notebook interface for a file named 'primeiro-codigo.ipynb'. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', with a status message 'All changes saved'. On the right, there are options for 'Comment', 'Share', and a user profile icon. Below the navigation bar, there are indicators for RAM and Disk usage, and a status 'Editing'. The notebook content consists of two code cells. The first cell contains the code: 

```
[2] 1 import time
     2 print(time.ctime())
```

 and has a timestamp of 'Fri Aug 28 00:59:02 2020'. The second cell contains the code: 

```
1 time.sleep(5)
  2 print (time.ctime())
```

 and has a timestamp of 'Fri Aug 28 00:59:08 2020'. A red arrow points to a red circle around the 'up' and 'down' arrow icons in the toolbar of the second cell, which are used to move the cell up or down in the notebook's execution order.

# Apagando Células

# Apagando Células

- Em muitas situações, o programador pode precisar apagar as células de código.

# Apagando Células

- Em muitas situações, o programador pode precisar apagar as células de código.



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** "primeiro-codigo.ipynb" with a star icon, and a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and "All changes saved".
- Right Panel:** "Comment", "Share", "RAM" (with a progress bar), "Disk" (with a progress bar), "Editing" (with a pencil icon), and a user profile picture.
- Code Cells:**
  - Cell 1: [2] `1 import time`  
`2 print(time.ctime())`  
Output: `Fri Aug 28 00:59:02 2020`
  - Cell 2: `1 time.sleep(5)`  
`2 print (time.ctime())`  
Output: `Fri Aug 28 00:59:08 2020`
- Context Menu:** Open over the second cell, showing icons for Run, Copy, Paste, Delete, Clear Output, and a menu.

# Apagando Células

- Em muitas situações, o programador pode precisar apagar as células de código.



The screenshot shows a Jupyter Notebook interface for a file named "primeiro-codigo.ipynb". The top navigation bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", with a status message "All changes saved". On the right, there are buttons for "Comment", "Share", and a settings gear, along with a user profile picture. Below the navigation bar, there are tabs for "+ Code" and "+ Text", and a status bar showing "RAM" and "Disk" usage, a pencil icon for "Editing", and an upward arrow.

The notebook contains two code cells. The first cell is executed and shows the output "Fri Aug 28 00:59:02 2020". The second cell is selected and its context menu is open, showing options for "Up", "Down", "Copy", "Paste", "Run", "Clear", "Delete", and "More". The "Delete" icon (a trash can) is circled in red, indicating the action being demonstrated.

```
[2] 1 import time
     2 print(time.ctime())

Fri Aug 28 00:59:02 2020

1 time.sleep(5)
2 print (time.ctime())

Fri Aug 28 00:59:08 2020
```

# Apagando Células

- Em muitas situações, o programador pode precisar apagar as células de código.



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** "primeiro-codigo.ipynb" with a star icon, and a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A status bar indicates "All changes saved".
- Right Side:** "Comment", "Share", and a user profile icon. Below these are RAM and Disk usage indicators and an "Editing" mode indicator.
- Code Cells:**
  - Cell 1: `[2] 1 import time`  
`2 print(time.ctime())`  
Execution time: Fri Aug 28 00:59:02 2020
  - Cell 2: `1 time.sleep(5)`  
`2 print (time.ctime())`  
Execution time: Fri Aug 28 00:59:08 2020
- Cell Toolbar:** A toolbar for the selected cell containing icons for up, down, refresh, comment, settings, and delete. A red arrow points to the delete icon (trash can), which is circled in red.

# DOCUMENTAÇÃO DO CÓDIGO



# Documentação de Código

# Documentação de Código

- Python possui instruções para deixar código comentado.

# Documentação de Código

- Python possui instruções para deixar código comentado.
- Porém, em muitas situações, o programador precisa mais do que texto simples para documentar os algoritmos de Aprendizagem de Máquinas (AM).

# Documentação de Código

- Python possui instruções para deixar código comentado.
- Porém, em muitas situações, o programador precisa mais do que texto simples para documentar os algoritmos de Aprendizagem de Máquinas (AM).
- As células de texto usam uma linguagem de marcação (*markdown*).

# Exemplos de Marcação

# Exemplos de Marcação

- Algumas maneiras de mudar a formatação do texto:

# Exemplos de Marcação

- Algumas maneiras de mudar a formatação do texto:
  - **\*\*negrito\*\***

# Exemplos de Marcação

- Algumas maneiras de mudar a formatação do texto:
  - **\*\*negrito\*\***
  - *\*italico\**

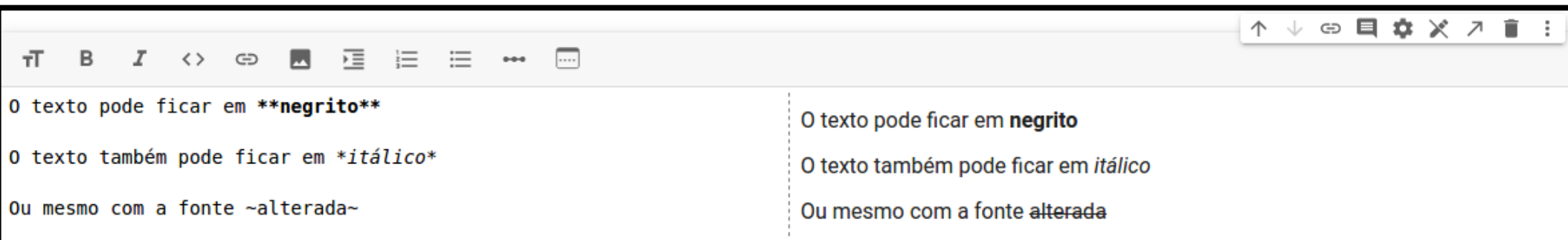


# Exemplos de Marcação

- Algumas maneiras de mudar a formatação do texto:
  - **\*\*negrito\*\***
  - *\*italico\**
  - ~riscada~

# Exemplos de Marcação

- Algumas maneiras de mudar a formatação do texto:
  - **\*\*negrito\*\***
  - *\*italico\**
  - ~riscada~



The screenshot shows a rich text editor interface with a toolbar at the top containing icons for text color, bold, italic, code, link, image, list, and other functions. Below the toolbar, the editor is split into two columns by a vertical dashed line. The left column shows the raw text with formatting codes: "O texto pode ficar em **\*\*negrito\*\***", "O texto também pode ficar em *\*italico\**", and "Ou mesmo com a fonte ~alterada~". The right column shows the rendered output: "O texto pode ficar em **negrito**", "O texto também pode ficar em *italico*", and "Ou mesmo com a fonte alterada".

# Equações Matemáticas

# Equações Matemáticas

- As equações matemáticas são baseadas na sintaxe de Latex.

# Equações Matemáticas

- As equações matemáticas são baseadas na sintaxe de Latex.
- Exemplo:

# Equações Matemáticas

- As equações matemáticas são baseadas na sintaxe de Latex.
- Exemplo:
  - Adicione uma célula de texto.

# Equações Matemáticas

- As equações matemáticas são baseadas na sintaxe de Latex.
- Exemplo:
  - Adicione uma célula de texto.
  - Adicione o seguinte texto:

# Equações Matemáticas

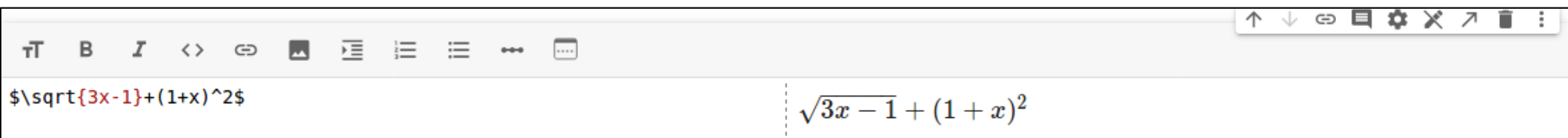
- As equações matemáticas são baseadas na sintaxe de Latex.
- Exemplo:
  - Adicione uma célula de texto.
  - Adicione o seguinte texto:  
$$\sqrt{3x-1}+(1+x)^2$$



# Equações Matemáticas

- As equações matemáticas são baseadas na sintaxe de Latex.
- Exemplo:
  - Adicione uma célula de texto.
  - Adicione o seguinte texto:

$\sqrt{3x-1}+(1+x)^2$



The screenshot shows a LaTeX editor interface. At the top, there is a toolbar with various icons for text formatting and editing. Below the toolbar, the LaTeX code  $\sqrt{3x-1}+(1+x)^2$  is entered in a text field. To the right of the code field, the rendered mathematical expression  $\sqrt{3x-1}+(1+x)^2$  is displayed.

# Equações Matemáticas

# Equações Matemáticas

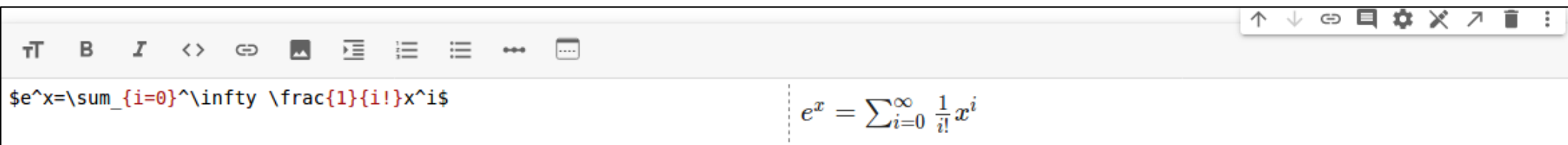
- Outro exemplo:

# Equações Matemáticas

- Outro exemplo:
  - $e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$

# Equações Matemáticas

- Outro exemplo:
  - $e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$



The screenshot shows a LaTeX editor interface. The top toolbar includes icons for undo, redo, bold, italic, link, unlink, insert image, list, table, and other editing tools. The main editing area is split into two sections. The left section contains the LaTeX source code:  $e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$ . The right section shows the rendered output of this code, which is the mathematical expression  $e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$ .

# Equações Matemáticas

# Equações Matemáticas

- Existem sites que facilitam a criação dessas fórmulas.

# Equações Matemáticas

- Existem sites que facilitam a criação dessas fórmulas.
- CODECOGS



# Equações Matemáticas

- Existem sites que facilitam a criação dessas fórmulas.
- CODECOGS
  - <https://latex.codecogs.com>

GRAVANDO SEU TRABALHO

# Onde Gravar?

- COLAB permite gravar o trabalho no Google Drive ou no GitHub.

# Google Drive

# Google Drive

- COLAB permite gravar o notebook no Google Drive.

# Google Drive

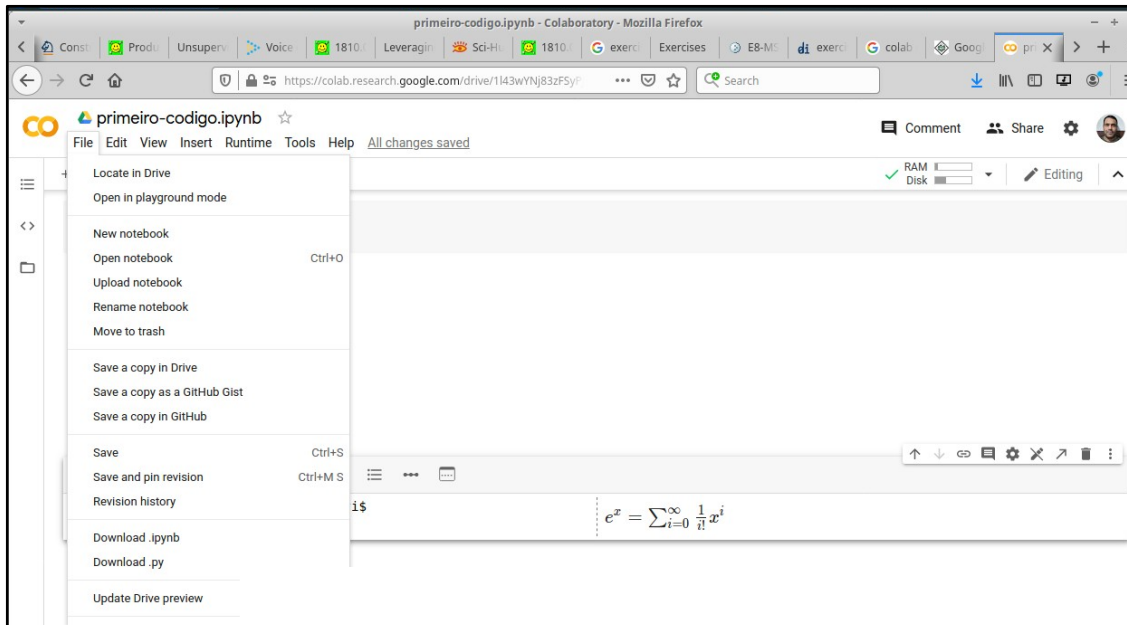
- COLAB permite gravar o notebook no Google Drive.
- Opção no menu:

# Google Drive

- COLAB permite gravar o notebook no Google Drive.
- Opção no menu:
  - File → Save a copy in Drive

# Google Drive

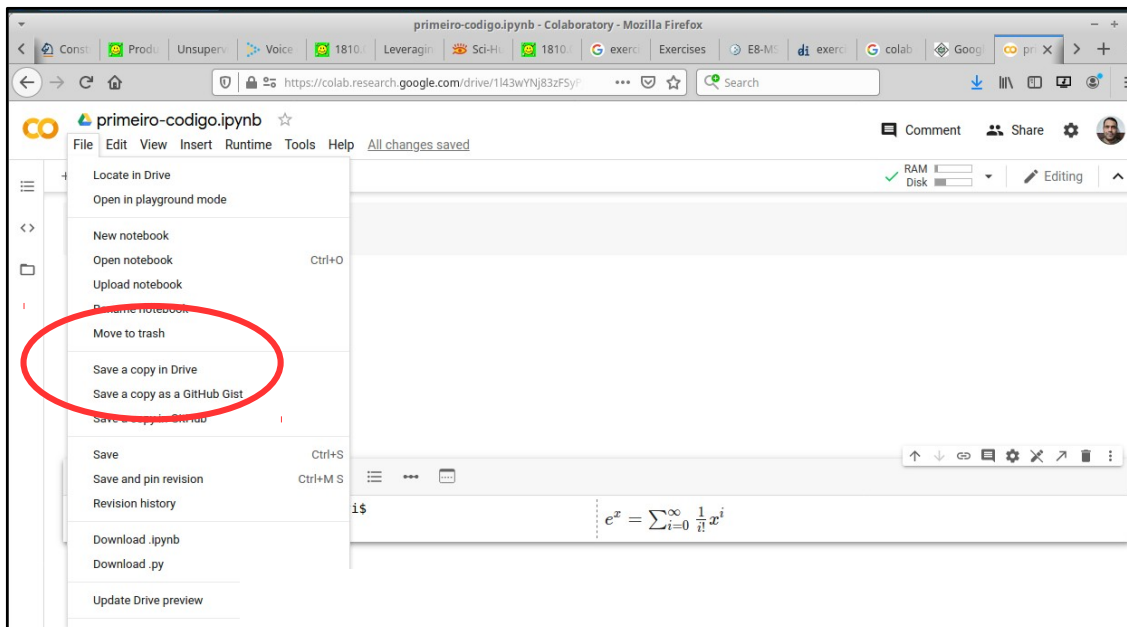
- COLAB permite gravar o notebook no Google Drive.
- Opção no menu:
  - File → Save a copy in Drive





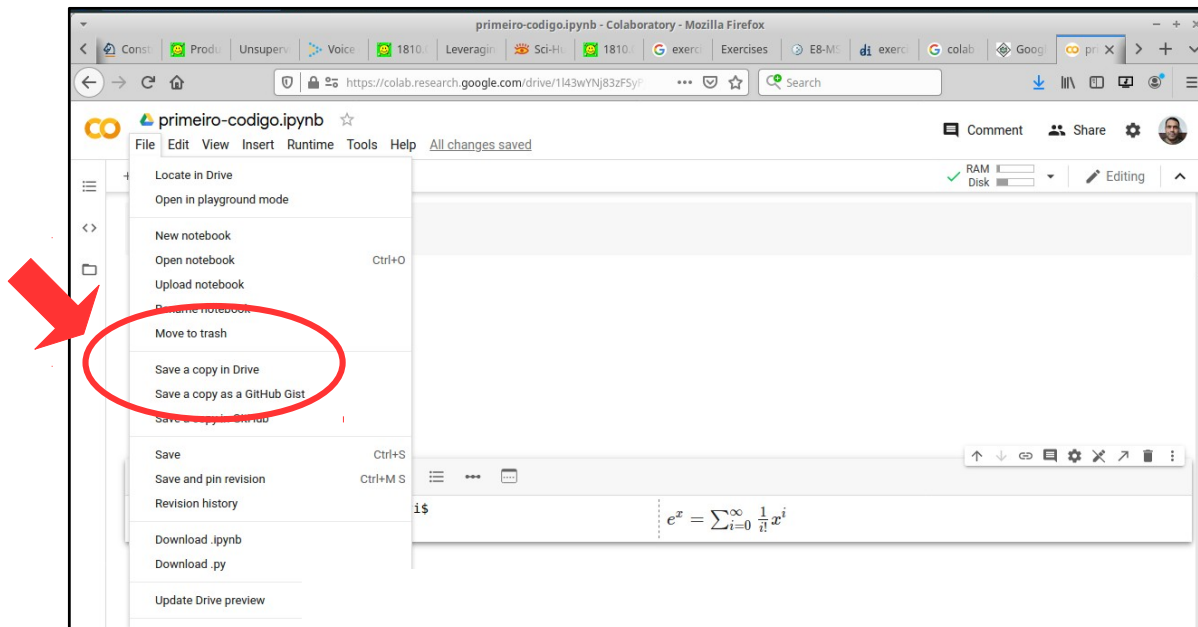
# Google Drive

- COLAB permite gravar o notebook no Google Drive.
- Opção no menu:
  - File → Save a copy in Drive



# Google Drive

- COLAB permite gravar o notebook no Google Drive.
- Opção no menu:
  - File → Save a copy in Drive



# Google Drive

- Salvar no Google Drive permite que você acesse e/ou compartilhe com outras pessoas posteriormente.

# GitHub

# GitHub

- É possível salvar todo o seu trabalho no GitHub.

# GitHub

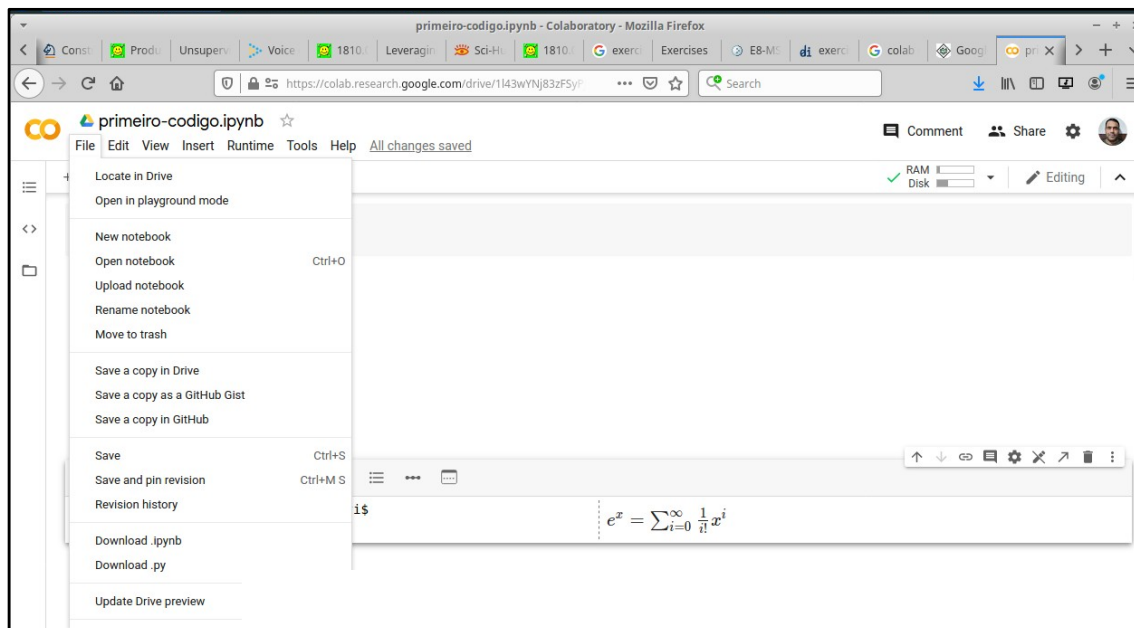
- É possível salvar todo o seu trabalho no GitHub.
- Opção no menu:

# GitHub

- É possível salvar todo o seu trabalho no GitHub.
- Opção no menu:
  - File → Save a copy in GitHub

# GitHub

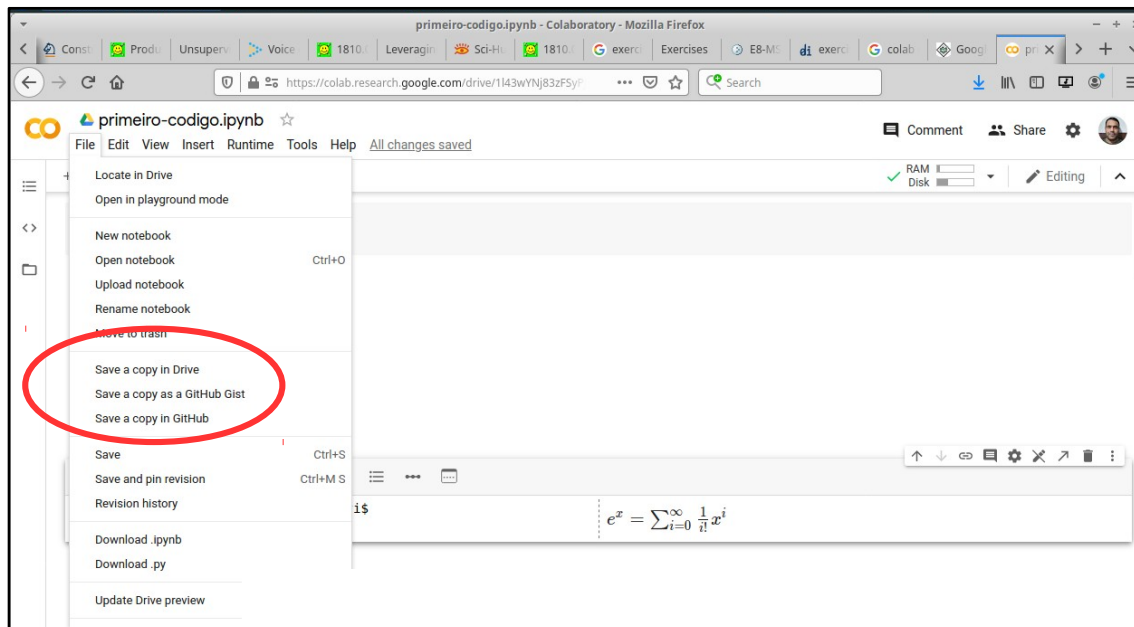
- É possível salvar todo o seu trabalho no GitHub.
- Opção no menu:
  - File → Save a copy in GitHub





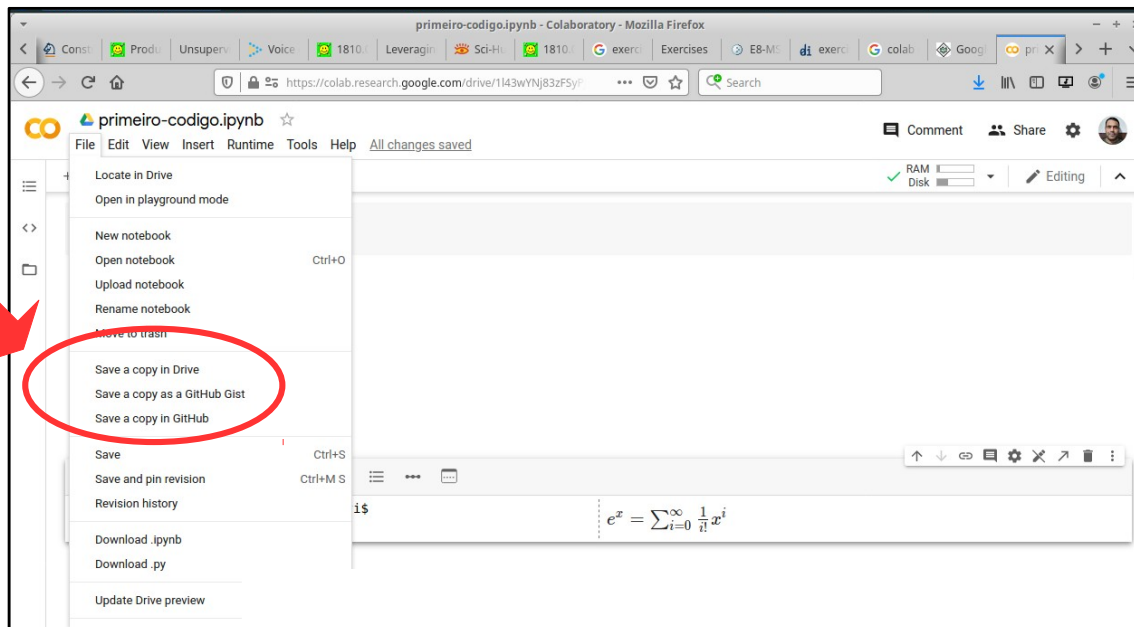
# GitHub

- É possível salvar todo o seu trabalho no GitHub.
- Opção no menu:
  - File → Save a copy in GitHub



# GitHub

- É possível salvar todo o seu trabalho no GitHub.
- Opção no menu:
  - File → Save a copy in GitHub

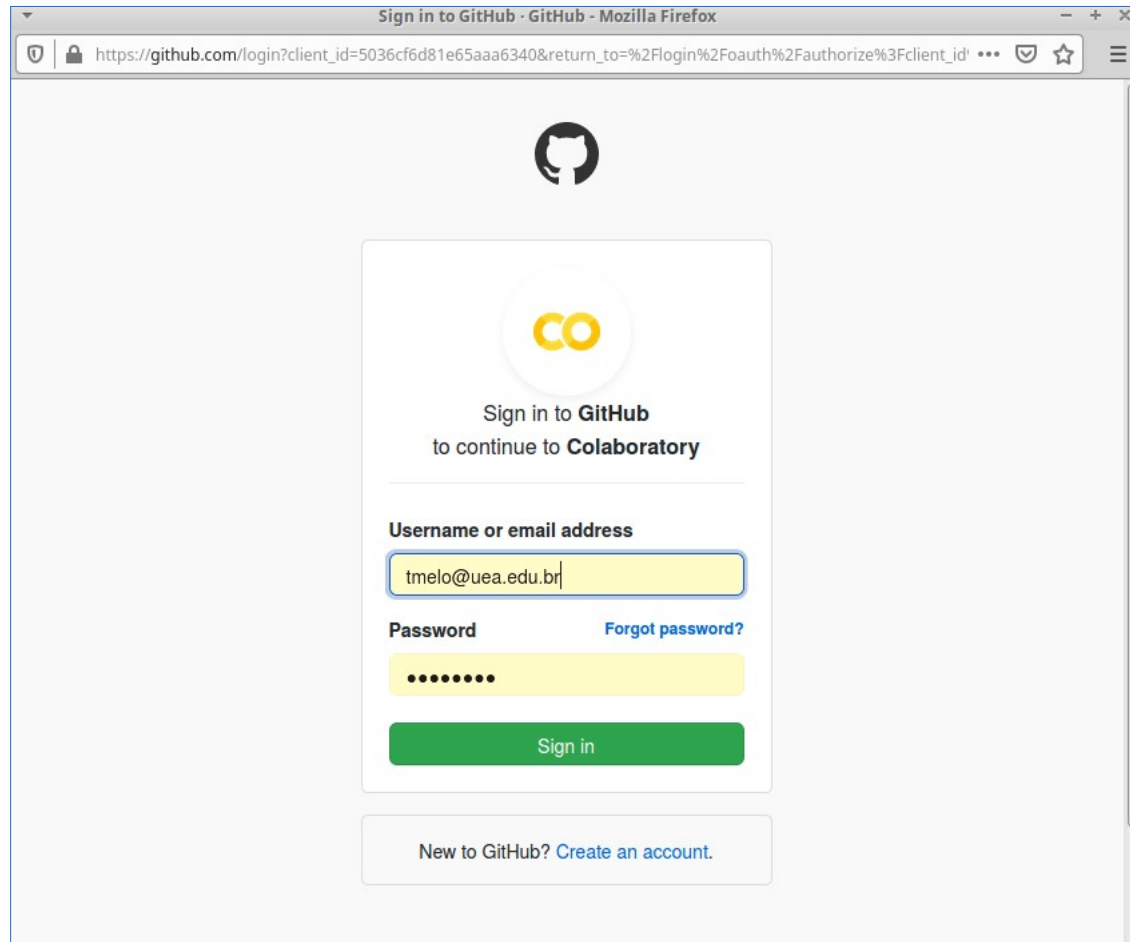


# GitHub

- É necessário ter uma autorização (login) no GitHub.

# GitHub

- É necessário ter uma autorização (login) no GitHub.



# GitHub

- É necessário criar um repositório no GitHub.

# GitHub

- É necessário criar um repositório no GitHub.

### Copy to GitHub

Repository: [🔗](#)  
tmelo-uea/covid19

Branch: [🔗](#)  
master

File path  
primeiro\_codigo.ipynb

Commit message  
Created using Colaboratory

Include a link to Colaboratory

# COMPARTILHAMENTO DE NOTEBOOK

# Compartilhamento



# Compartilhamento

- O compartilhamento do notebook pode ser feito através do Google Drive.

# Compartilhamento

- O compartilhamento do notebook pode ser feito através do Google Drive.
- Para publicar o notebook para o público em geral, você pode compartilhá-lo através do repositório do GitHub.

# Compartilhamento

- O compartilhamento é feito através do botão (canto superior à direita).

# Compartilhamento

- O compartilhamento é feito através do botão (canto superior à direita).



# Compartilhamento

- O compartilhamento é feito através do botão (canto superior à direita).



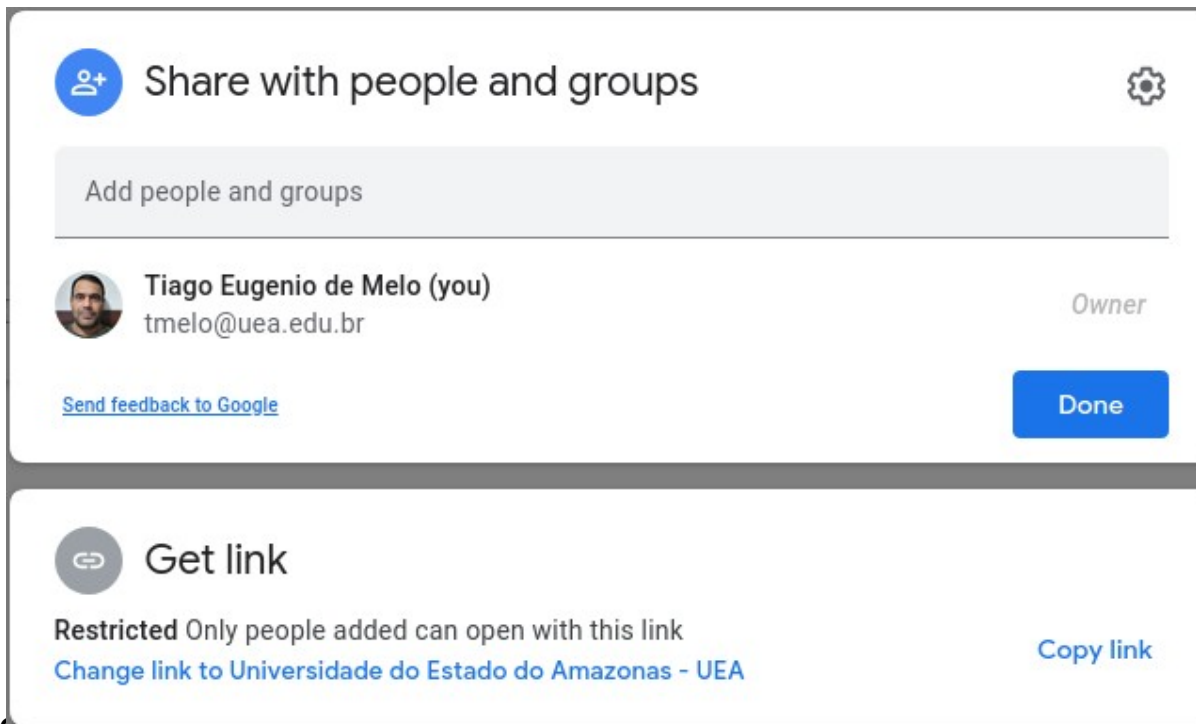
# Compartilhamento

- O compartilhamento é feito através do botão (canto superior à direita).



# Compartilhamento

- O compartilhamento é feito através do botão (canto superior à direita).



# Compartilhamento



# Compartilhamento

- Existem várias opções de compartilhamento:

# Compartilhamento

- Existem várias opções de compartilhamento:
  - Para um grupo específico de pessoas.

# Compartilhamento

- Existem várias opções de compartilhamento:
  - Para um grupo específico de pessoas.
  - Colegas da sua empresa (equipe).

# Compartilhamento

- Existem várias opções de compartilhamento:
  - Para um grupo específico de pessoas.
  - Colegas da sua empresa (equipe).
  - Qualquer pessoa que possua o *link*.

# Compartilhamento

- Existem várias opções de compartilhamento:
  - Para um grupo específico de pessoas.
  - Colegas da sua empresa (equipe).
  - Qualquer pessoa que possua o *link*.
  - O público em geral.

# COMANDOS DE SISTEMA

# Comandos de Sistema

# Comandos de Sistema

- Jupyter permite a execução de vários comandos disponíveis em sistemas operacionais.



# Comandos de Sistema

- Jupyter permite a execução de vários comandos disponíveis em sistemas operacionais.
- O COLAB dá suporte a esse recurso.

# Comandos de Sistema

- Jupyter permite a execução de vários comandos disponíveis em sistemas operacionais.
- O COLAB dá suporte a esse recurso.
- Esses comandos são iniciados com o sinal de exclamação (!).

# Baixando Dados Remotos

# Baixando Dados Remotos

- O comando `WGET` permite baixar (*download*) de arquivos remotos.

# Baixando Dados Remotos

- O comando `WGET` permite baixar (*download*) de arquivos remotos.
- Executar o comando:

# Baixando Dados Remotos

- O comando `WGET` permite baixar (*download*) de arquivos remotos.
- Executar o comando:  
`!wget http://tiagodemelo.info/datasets/dados-curso.csv`

# Baixando Dados Remotos

- O comando WGET permite baixar (*download*) de arquivos remotos.
- Executar o comando:

```
!wget http://tiagodemelo.info/datasets/dados-curso.csv
```

```
[6] 1 !wget http://tiagodemelo.info/datasets/dados-curso.csv

↳ --2020-08-28 02:07:35-- http://tiagodemelo.info/datasets/dados-curso.csv
Resolving tiagodemelo.info (tiagodemelo.info)... 108.167.188.189
Connecting to tiagodemelo.info (tiagodemelo.info)|108.167.188.189|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 102696 (100K) [text/csv]
Saving to: 'dados-curso.csv'

dados-curso.csv      100%[=====>] 100.29K  ---KB/s   in 0.1s

2020-08-28 02:07:35 (833 KB/s) - 'dados-curso.csv' saved [102696/102696]
```

# Baixando Dados Remotos

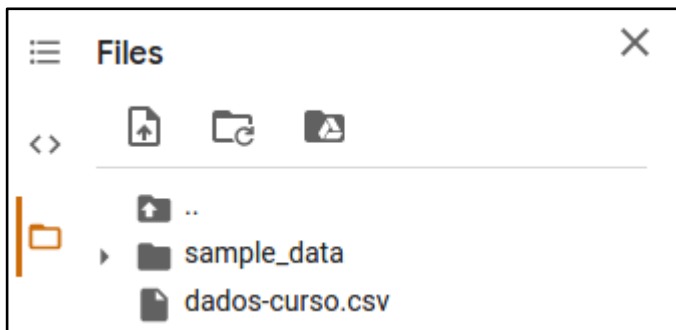


# Baixando Dados Remotos

- O arquivo ficará disponível no Google Drive.

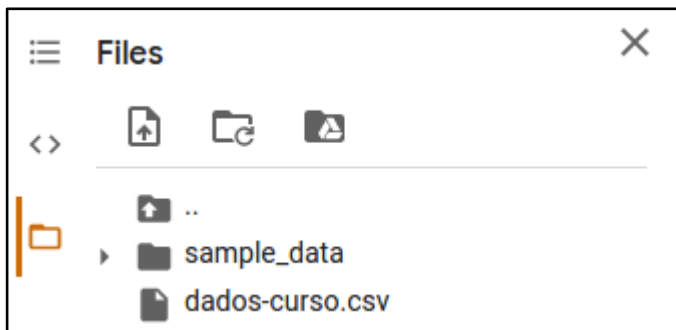
# Baixando Dados Remotos

- O arquivo ficará disponível no Google Drive.



# Baixando Dados Remotos

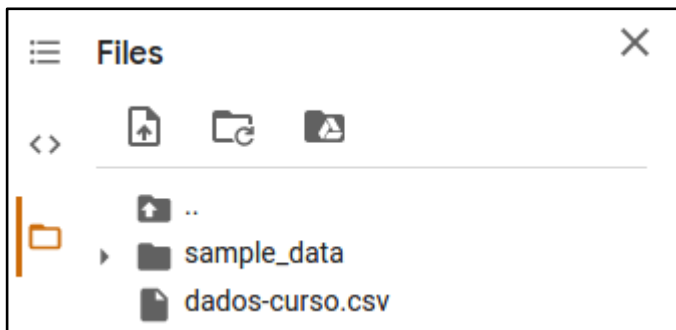
- O arquivo ficará disponível no Google Drive.



- Esse arquivo pode ser lido (usado) através da biblioteca Pandas:

# Baixando Dados Remotos

- O arquivo ficará disponível no Google Drive.



- Esse arquivo pode ser lido (usado) através da biblioteca Pandas:

```
1 import pandas as pd
2
3 data = pd.read_csv("/content/dados-curso.csv")
4
5 data.head()
```

# Baixando Dados Remotos

- Resultado:

# Baixando Dados Remotos

- Resultado:

	data	texto	retweet	idioma	lugar	pais	sigla	latitude	longitude
0	2020-05-23 00:21:14	Para voltar tudo ao normal, você precisa fazer...	0	pt	Rio das Ostras	Brazil	BR	-41.937900	-22.522600
1	2020-03-22 22:57:51	14.245 - O que é a hidroxiclороquina? <a href="https://...">https://...</a>	0	pt	Sao Paulo	Brazil	BR	-46.674739	-23.606067
2	2020-04-14 00:11:33	Quarta morte em Lar de Estarreja associada à C...	1	pt	Lisbon	Portugal	PT	-9.099043	38.747518
3	2020-05-25 20:45:43	COVID-19   Hospital municipal Tide Setúbal, n...	0	pt	Sao Paulo	Brazil	BR	-46.633300	-23.550000
4	2020-04-15 10:34:39	#cenasdocotidiano #santos #distanciamentosocia...	0	pt	Santos	Brazil	BR	-46.293700	-23.975947

# Baixando Dados Remotos

- Uma outra maneira de baixar os arquivos é usar os recursos da biblioteca Pandas para baixar o arquivo diretamente.

# Baixando Dados Remotos

- Uma outra maneira de baixar os arquivos é usar os recursos da biblioteca Pandas para baixar o arquivo diretamente.

```
1 import pandas as pd
2
3 data = pd.read_csv("http://tiagodemelo.info/datasets/dados-curso.csv")
4
5 data.head()
```



# Baixando Dados Remotos

- Uma outra maneira de baixar os arquivos é usar os recursos da biblioteca Pandas para baixar o arquivo diretamente.

```
1 import pandas as pd
2
3 data = pd.read_csv("http://tiagodemelo.info/datasets/dados-curso.csv")
4
5 data.head()
```

# Baixando Dados Remotos

- Uma outra maneira de baixar os arquivos é usar os recursos da biblioteca Pandas para baixar o arquivo diretamente.

```
1 import pandas as pd
2
3 data = pd.read_csv("http://tiagodemelo.info/datasets/dados-curso.csv")
4
5 data.head()
```

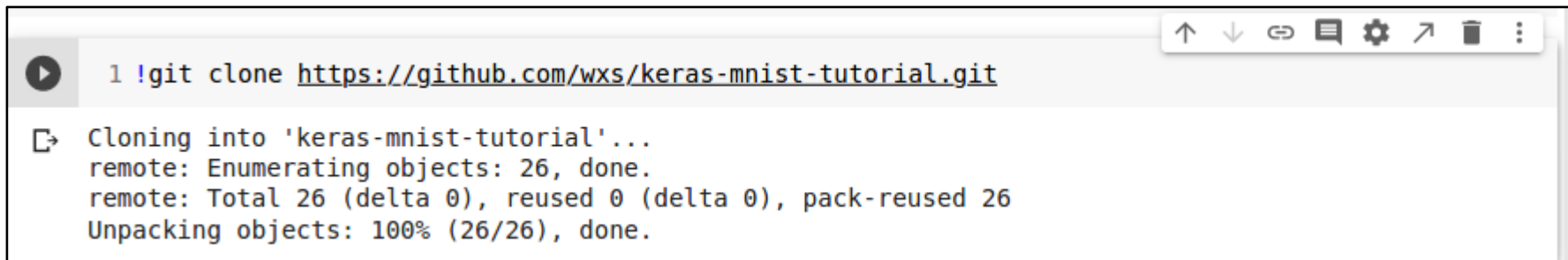


# Clonando Repositório Git

- É possível clonar o repositório inteiro do GitHub utilizando o comando `git`.

# Clonando Repositório Git

- É possível clonar o repositório inteiro do GitHub utilizando o comando `git`.



```
1 !git clone https://github.com/wxs/keras-mnist-tutorial.git
Cloning into 'keras-mnist-tutorial'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Unpacking objects: 100% (26/26), done.
```

# EXECUTANDO ARQUIVOS EXTERNOS DE PYTHON

# Arquivos Externos (Python)

# Arquivos Externos (Python)

- Suponha que você já tenha desenvolvido alguns programas em Python e que estejam no Google Drive.

# Arquivos Externos (Python)

- Suponha que você já tenha desenvolvido alguns programas em Python e que estejam no Google Drive.
- É possível executá-los no COLAB.



# Montar Drive

# Montar Drive

- O primeiro passo é montar o Google Drive.

# Montar Drive

- O primeiro passo é montar o Google Drive.
- Opção no menu:

# Montar Drive

- O primeiro passo é montar o Google Drive.
- Opção no menu:
  - Tools → Command palette

# Montar Drive

- O primeiro passo é montar o Google Drive.
- Opção no menu:
  - Tools → Command palette

The screenshot shows a Jupyter Notebook interface with the following elements:

- File Explorer (Left):** Shows a directory structure with folders 'keras-mnist-tutorial' and 'sample\_data', and a file 'dados-curso.csv'.
- Code Cell (Center):** Contains a code cell with the following content:

```
[6] 1 !wget http://tiagodemelo.info/datasets/dados-curso.csv
2
3
4
5
```

Below the code cell, the output shows the file being downloaded and saved:

```
dados-curso.csv 100% [833 KB/s] - 'dados-curso.csv' saved [102696/102696]
```
- Command Palette (Right):** A menu is open, listing various actions. The 'Mount Drive' option is highlighted.
- Code Cell (Bottom):** Contains a code cell with the following content:

```
[8] 1 !git clone https://github.com/wxs/keras-mnist-tutorial.git
2
3
4
5
```

Below the code cell, the output shows the git clone command being executed:

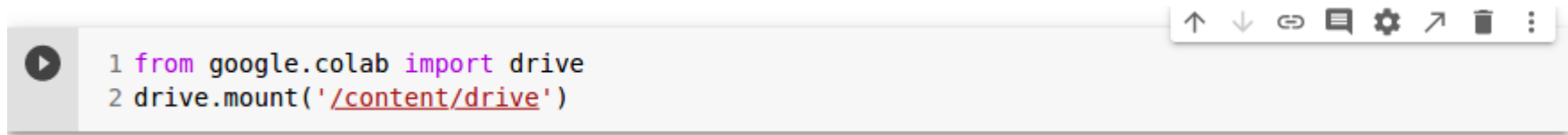
```
Cloning into 'keras-mnist-tutorial'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Unpacking objects: 100% (26/26), done.
```

# Montar Drive

- Uma célula de código será automaticamente criada.

# Montar Drive

- Uma célula de código será automaticamente criada.

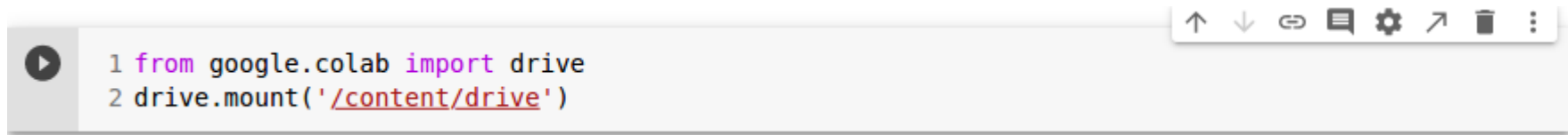


```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

The image shows a code cell from Google Colab. On the left, there is a play button icon. The code consists of two lines: the first line imports the 'drive' module from 'google.colab', and the second line calls 'drive.mount()' with the path '/content/drive'. On the right side of the code cell, there is a toolbar with icons for undo, redo, copy, paste, settings, share, delete, and a menu.

# Montar Drive

- Uma célula de código será automaticamente criada.



```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

The image shows a code cell from Google Colab. It contains two lines of Python code: `1 from google.colab import drive` and `2 drive.mount('/content/drive')`. The code is displayed in a light gray box with a play button icon on the left and a toolbar with various icons (up, down, refresh, comment, settings, link, trash, and menu) on the right.

- Se você executar o código acima, será solicitada uma autenticação.



# Montar Drive

- Uma célula de código será automaticamente criada.

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

- Se você executar o código acima, será solicitada uma autenticação.

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6c](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6c)

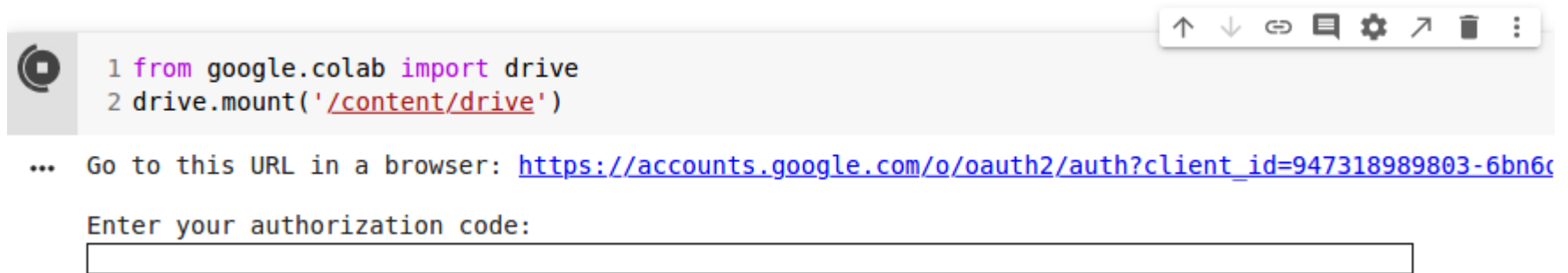
Enter your authorization code:

# Montar Drive

- Basta clicar no link e fazer a autenticação:

# Montar Drive

- Basta clicar no link e fazer a autenticação:



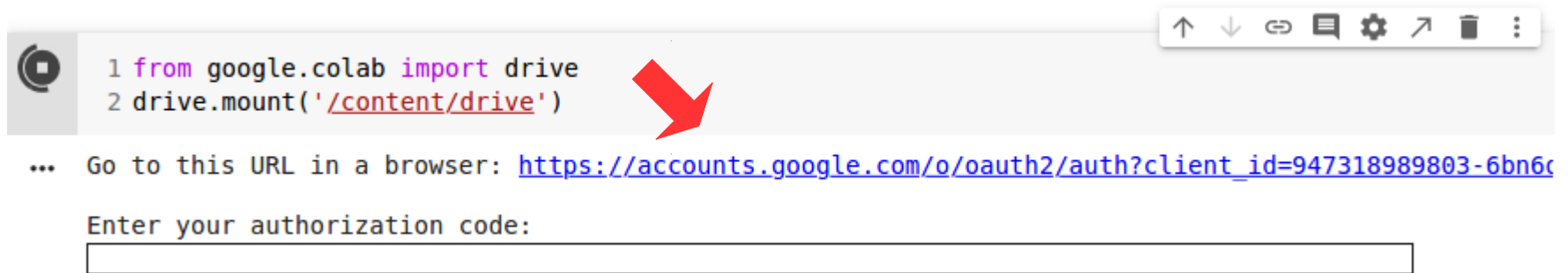
```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6c](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6c)

Enter your authorization code:

# Montar Drive

- Basta clicar no link e fazer a autenticação:




The screenshot shows a terminal window in Google Colab. At the top right, there is a toolbar with icons for undo, redo, copy, paste, settings, refresh, and delete. The terminal contains two lines of Python code: `1 from google.colab import drive` and `2 drive.mount('/content/drive')`. A red arrow points to the second line of code. Below the code, there is a message: `... Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client\_id=947318989803-6bn6c`. Underneath this message, it says "Enter your authorization code:" followed by an empty text input field.

# Montar Drive


- Autenticação:

# Montar Drive


- Autenticação:

 Fazer login com o Google


---




Escolha uma conta  
para prosseguir para [Google Drive File Stream](#)

 **Tiago Melo**  
tiago.melo@oceanbrasil.com


---

 **Tiago Eugenio de Melo**  
tmelo@uea.edu.br


---

 **Tiago Melo**  
tiagodemelo@gmail.com


---

 **Tiago Melo**  
tiagoeugeniodemelo@gmail.com


---

 **Tiago Eugênio de Melo** Desconectado  
tiago.melo@icomp.ufam.edu.br

---

 **Tiago Melo**  
tiagodemelo.phd@gmail.com

---

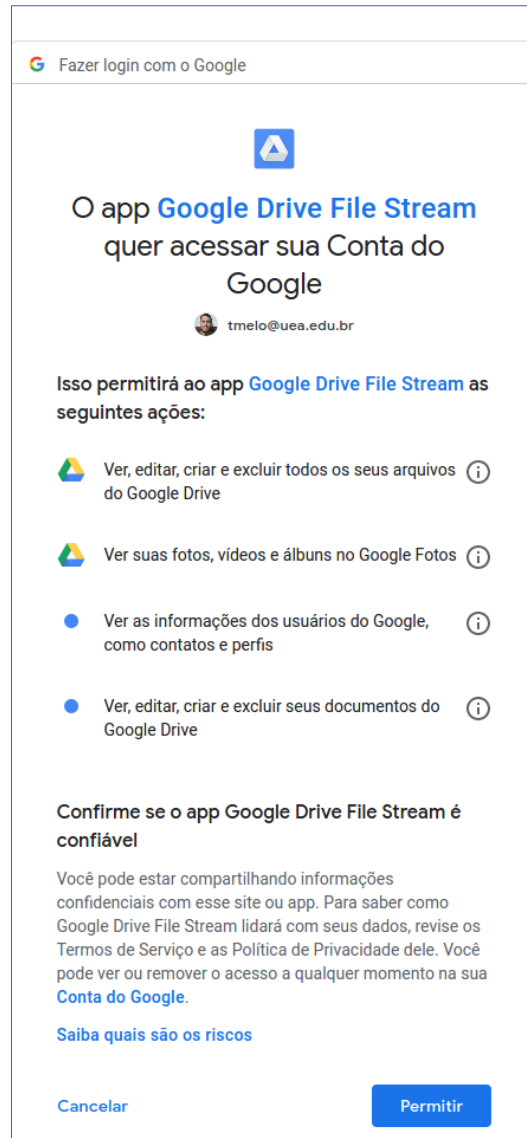
 Usar outra conta

# Montar Drive

- Autenticação:

# Montar Drive

- Autenticação:



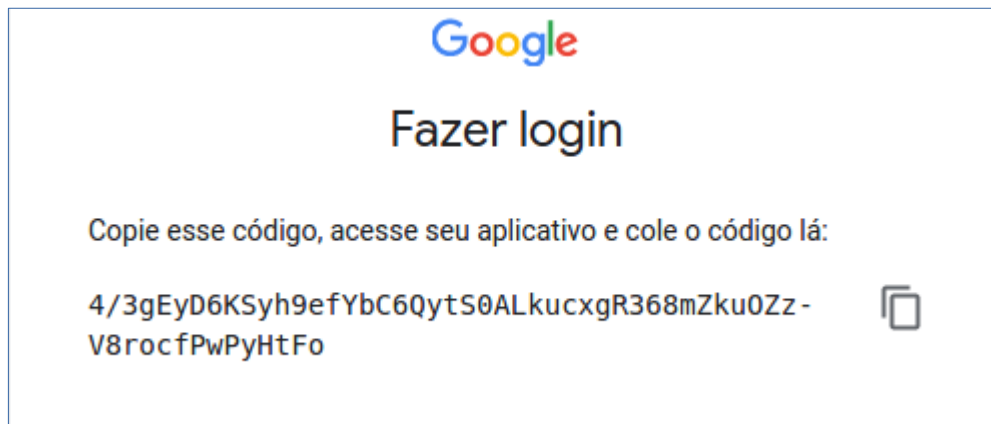


# Montar Drive

- Será gerado um código automaticamente:

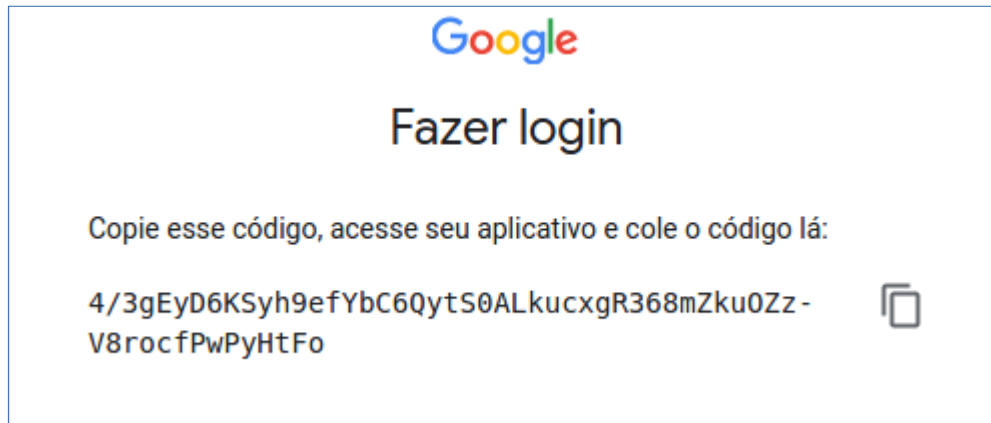
# Montar Drive

- Será gerado um código automaticamente:



# Montar Drive

- Será gerado um código automaticamente:



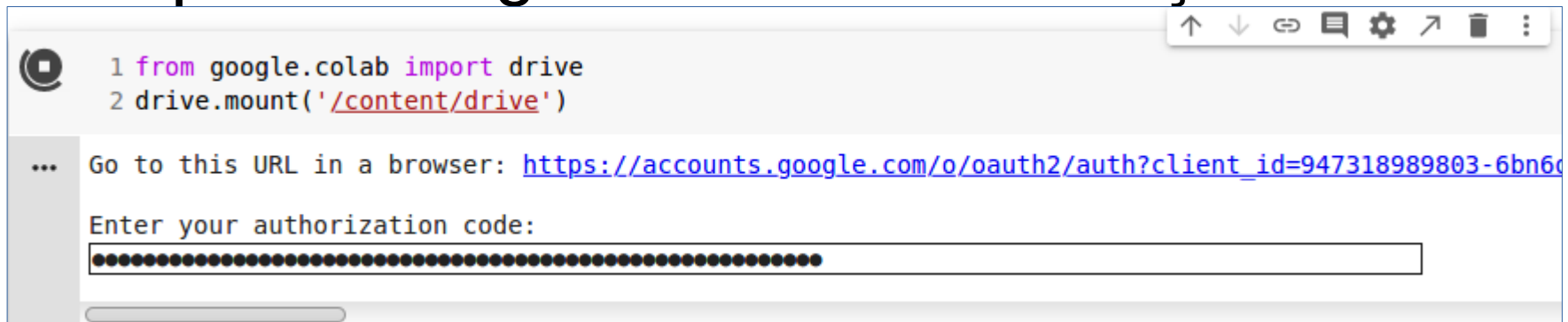
- Copia o código e cola na solicitação:

# Montar Drive

- Será gerado um código automaticamente:



- Copia o código e cola na solicitação:

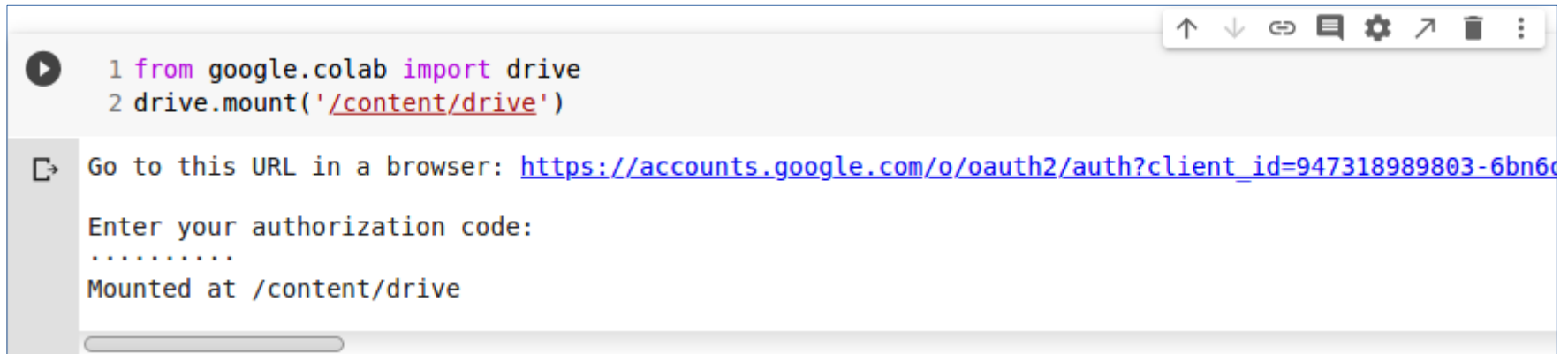


# Montar Drive

- Confirmada a autenticação:

# Montar Drive

- Confirmada a autenticação:



```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6c](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6c)

Enter your authorization code:  
.....

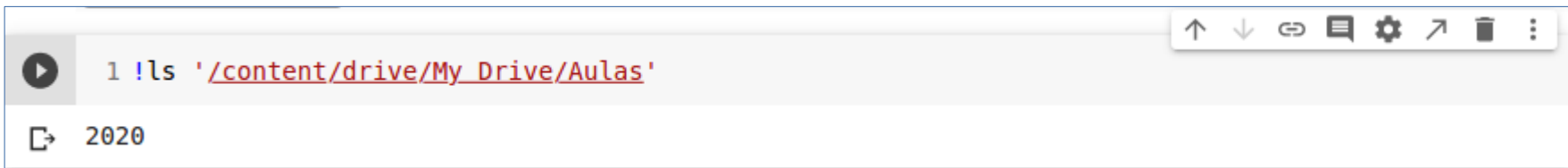
Mounted at /content/drive

# Execução do Código Python

- É possível listar os arquivos em um determinado diretório:

# Execução do Código Python

- É possível listar os arquivos em um determinado diretório:



```
1 !ls '/content/drive/My_Drive/Aulas'
```

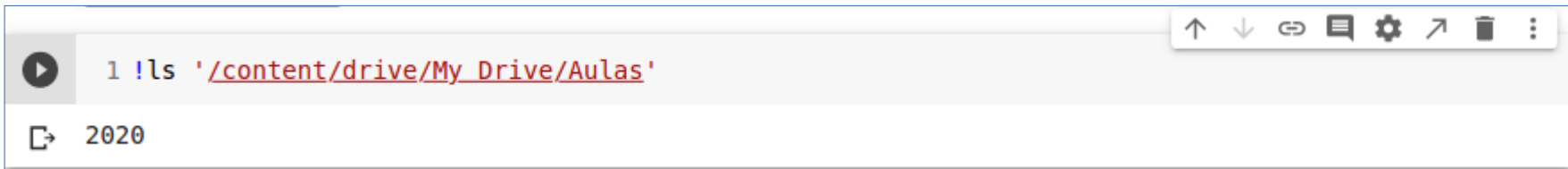
↳ 2020

The image shows a terminal window with a light gray background. At the top right, there is a toolbar with icons for back, forward, search, settings, and other functions. The terminal prompt is a play button icon followed by the command `!ls '/content/drive/My_Drive/Aulas'`. Below the command, the output is `↳ 2020`.



# Execução do Código Python

- É possível listar os arquivos em um determinado diretório:

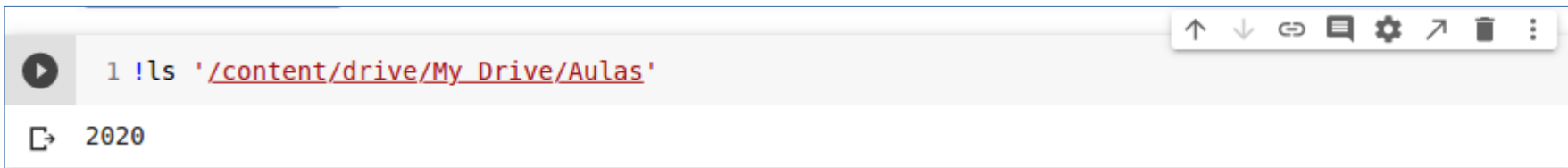


A screenshot of a terminal window. The command prompt shows a play button icon followed by the command `1 !ls '/content/drive/My_Drive/Aulas'`. Below the command, the output shows a folder icon followed by the year `2020`. The terminal has a standard toolbar at the top right with icons for back, forward, search, settings, and other functions.

- A execução do programa:

# Execução do Código Python

- É possível listar os arquivos em um determinado diretório:

A screenshot of a terminal window. The command prompt shows a play button icon followed by the command `!ls '/content/drive/My_Drive/Aulas'`. Below the command, the output shows the year `2020`. The terminal window has a standard toolbar with icons for back, forward, search, settings, and other navigation functions.

```
1 !ls '/content/drive/My_Drive/Aulas'  
2020
```

- A execução do programa:

```
!python3 "/content/drive/My Drive/Colab Notebooks/hello.py"
```

# GRÁFICOS

# Gráficos

# Gráficos

- COLAB dá suporte a sofisticados tipos de gráficos.

# Gráficos

- COLAB dá suporte a sofisticados tipos de gráficos.
- Exemplo:

# Gráficos

- COLAB dá suporte a sofisticados tipos de gráficos.
- Exemplo:

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 y = np.random.randn(100)
5
6 x = [x for x in range(len(y))]
7
8 plt.plot(x, y, '-')
9
10 plt.fill_between(x, y, 200, where=(y > 195), facecolor='g', alpha=0.6)
11
12 plt.title("Sample Plot")
13
14 plt.show()
```

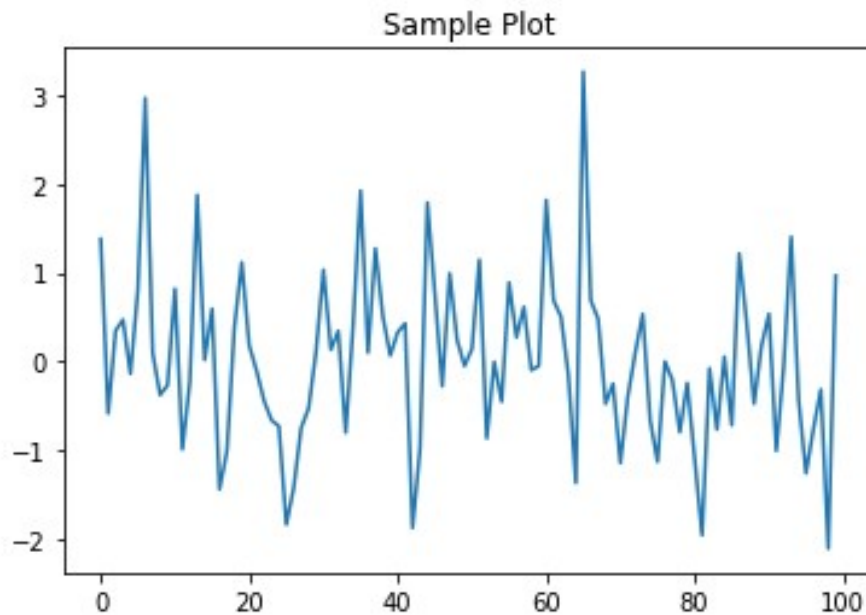
# Gráficos

- Saída do exemplo (código) anterior:



# Gráficos

- Saída do exemplo (código) anterior:



AJUDA (HELP)

# Help

# Help

- As ferramentas de IDE costumam usar ajuda sensível ao contexto.

# Help

- As ferramentas de IDE costumam usar ajuda sensível ao contexto.
- Também recursos como auto-completar.

# Help

- As ferramentas de IDE costumam usar ajuda sensível ao contexto.
- Também recursos como auto-completar.
- COLAB possui todos esses recursos que facilitam a atividade de programação.

# Lista de Função

# Lista de Função

- **Passo 1**



# Lista de Função

- **Passo 1**

- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

# Lista de Função

- **Passo 1**

- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

```
import torch
```

# Lista de Função

- **Passo 1**

- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

```
import torch
```

- **Passo 2**

# Lista de Função

- **Passo 1**

- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

```
import torch
```

- **Passo 2**

- Digite o código abaixo:

# Lista de Função

- **Passo 1**

- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

```
import torch
```

- **Passo 2**

- Digite o código abaixo:

```
Tensor = torch.
```

# Lista de Função

- **Passo 1**

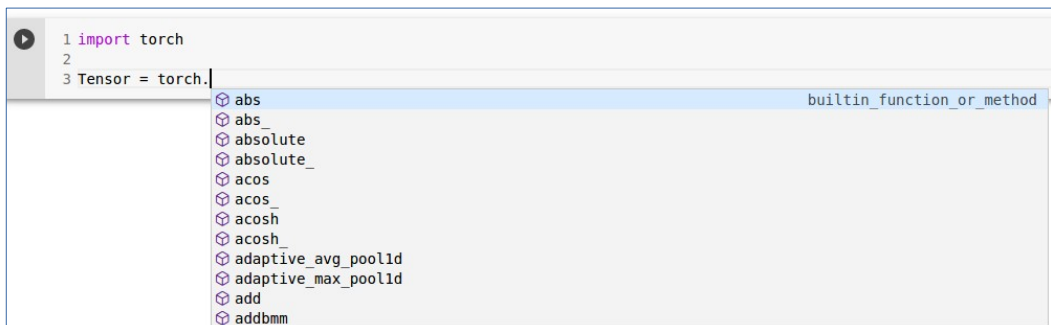
- Abra um novo notebook ou uma nova célula de código e digite o código abaixo:

```
import torch
```

- **Passo 2**

- Digite o código abaixo:

```
Tensor = torch.
```



```
1 import torch
2
3 Tensor = torch.
```

The screenshot shows a Jupyter Notebook cell with three lines of code. The first line is 'import torch', the second is blank, and the third is 'Tensor = torch.'. A dropdown menu is open below the third line, showing a list of functions from the torch module. The first item is 'abs' with the label 'builtin\_function\_or\_method' to its right. Other visible items include 'abs\_', 'absolute', 'absolute\_', 'acos', 'acos\_', 'acosh', 'acosh\_', 'adaptive\_avg\_pool1d', 'adaptive\_max\_pool1d', 'add', and 'addbmm'.

# Função de Documentação

# Função de Documentação

- COLAB fornece documentação sobre qualquer **função** ou **classe** como ajuda ao contexto.



# Função de Documentação

- COLAB fornece documentação sobre qualquer **função** ou **classe** como ajuda ao contexto.
- Digite o código abaixo:

# Função de Documentação

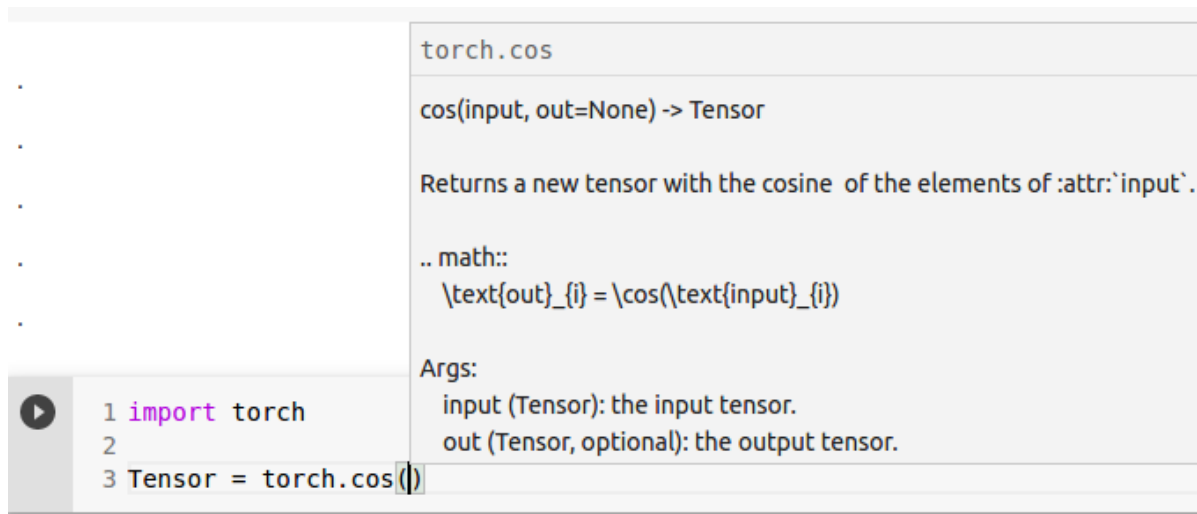
- COLAB fornece documentação sobre qualquer **função** ou **classe** como ajuda ao contexto.
- Digite o código abaixo:

```
Tensor = torch.cos (
```

# Função de Documentação

- COLAB fornece documentação sobre qualquer **função** ou **classe** como ajuda ao contexto.
- Digite o código abaixo:

```
Tensor = torch.cos (
```



```
torch.cos
cos(input, out=None) -> Tensor

Returns a new tensor with the cosine of the elements of :attr:`input`.

.. math::
    \text{out}_{i} = \cos(\text{input}_{i})

Args:
    input (Tensor): the input tensor.
    out (Tensor, optional): the output tensor.
```

```
1 import torch
2
3 Tensor = torch.cos()
```

# COMANDOS MÁGICOS

# Comandos Mágicos

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).



# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:
  - Única linha de comando.

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:
  - Única linha de comando.
  - Código usa (%) no início do comando.

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:
  - Única linha de comando.
  - Código usa (%) no início do comando.
- Células mágicas:

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:
  - Única linha de comando.
  - Código usa (%) no início do comando.
- Células mágicas:
  - Atua para toda a célula de código.

# Comandos Mágicos

- COLAB possui um conjunto de comandos de sistema que ampliam as funcionalidades.
- Tipos:
  - Linhas mágicas (*line magics*).
  - Células mágicas (*cell magics*).
- Linhas mágicas:
  - Única linha de comando.
  - Código usa (%) no início do comando.
- Células mágicas:
  - Atua para toda a célula de código.
  - Código usa (%%) no início do comando.

# Linhas Mágicas



# Linhas Mágicas

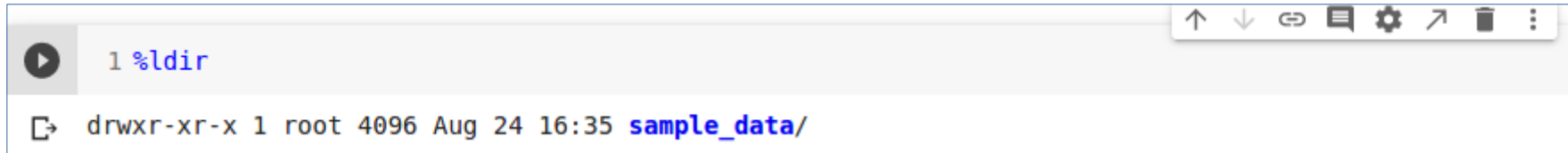
- Exemplo 1:

**%ldir**

# Linhas Mágicas

- Exemplo 1:

**%ldir**



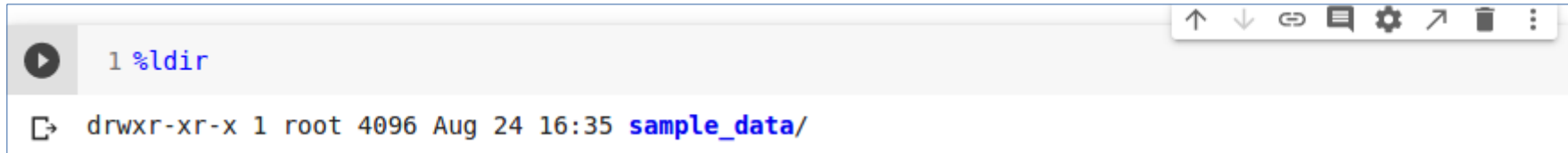
```
1 %ldir
drwxr-xr-x 1 root 4096 Aug 24 16:35 sample_data/
```

The image shows a terminal window with a light gray background. At the top right, there is a toolbar with icons for up, down, search, chat, settings, and refresh. The terminal prompt is '1 %ldir' in blue text. Below the prompt, the output is 'drwxr-xr-x 1 root 4096 Aug 24 16:35 sample\_data/' in black text, where 'sample\_data/' is highlighted in blue.

# Linhas Mágicas

- Exemplo 1:

**%!dir**



A terminal window showing the execution of the magic command `%!dir`. The prompt is `1 %!dir`. The output is `drwxr-xr-x 1 root 4096 Aug 24 16:35 sample_data/`. The terminal window has a toolbar with icons for back, forward, search, settings, and other functions.

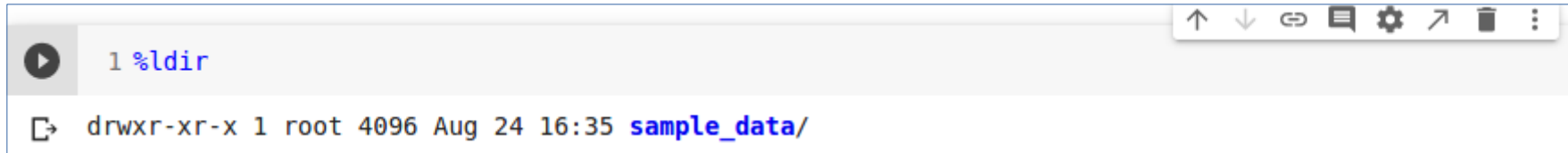
- Exemplo 2:

**%!history**

# Linhas Mágicas

- Exemplo 1:

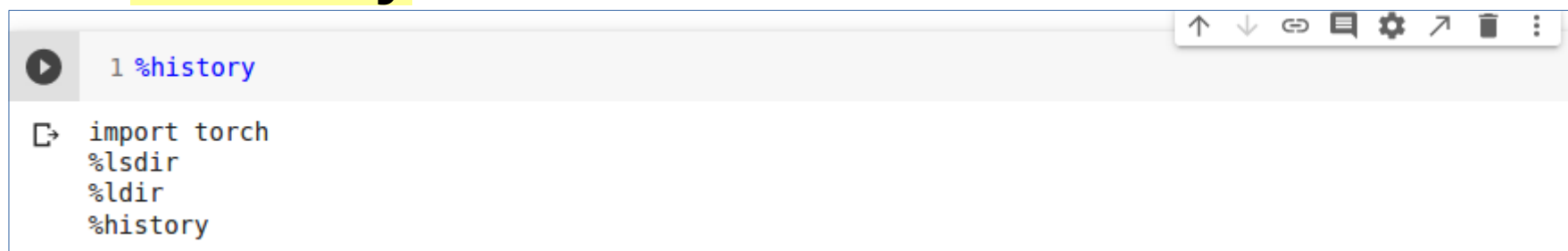
**%ldir**



```
1 %ldir
drwxr-xr-x 1 root 4096 Aug 24 16:35 sample_data/
```

- Exemplo 2:

**%history**



```
1 %history
import torch
%lsdir
%ldir
%history
```

# Células Mágicas

# Células Mágicas

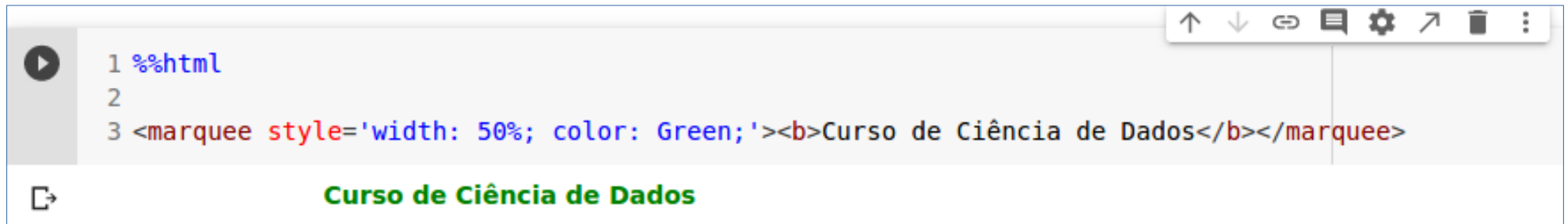
- Uso de recursos de HTML.

# Células Mágicas

- Uso de recursos de HTML.
- Exemplo:

# Células Mágicas

- Uso de recursos de HTML.
- Exemplo:



```
1 %%html
2
3 <marquee style='width: 50%; color: Green;'><b>Curso de Ciência de Dados</b></marquee>
```

**Curso de Ciência de Dados**



# Lista Mágica

# Lista Mágica

- Comando para verificar a lista de comandos mágicos:

# Lista Mágica

- Comando para verificar a lista de comandos mágicos:

**%lsmagic**

# FORMULÁRIOS

# Introdução

# Introdução

- COLAB provê um recurso bastante interessante de formulários.

# Introdução

- COLAB provê um recurso bastante interessante de formulários.
- Os formulários podem ser usados para receber a entrada (*input*) de dados de usuários em tempo de execução.

# Inserção de Formulários



# Inserção de Formulários

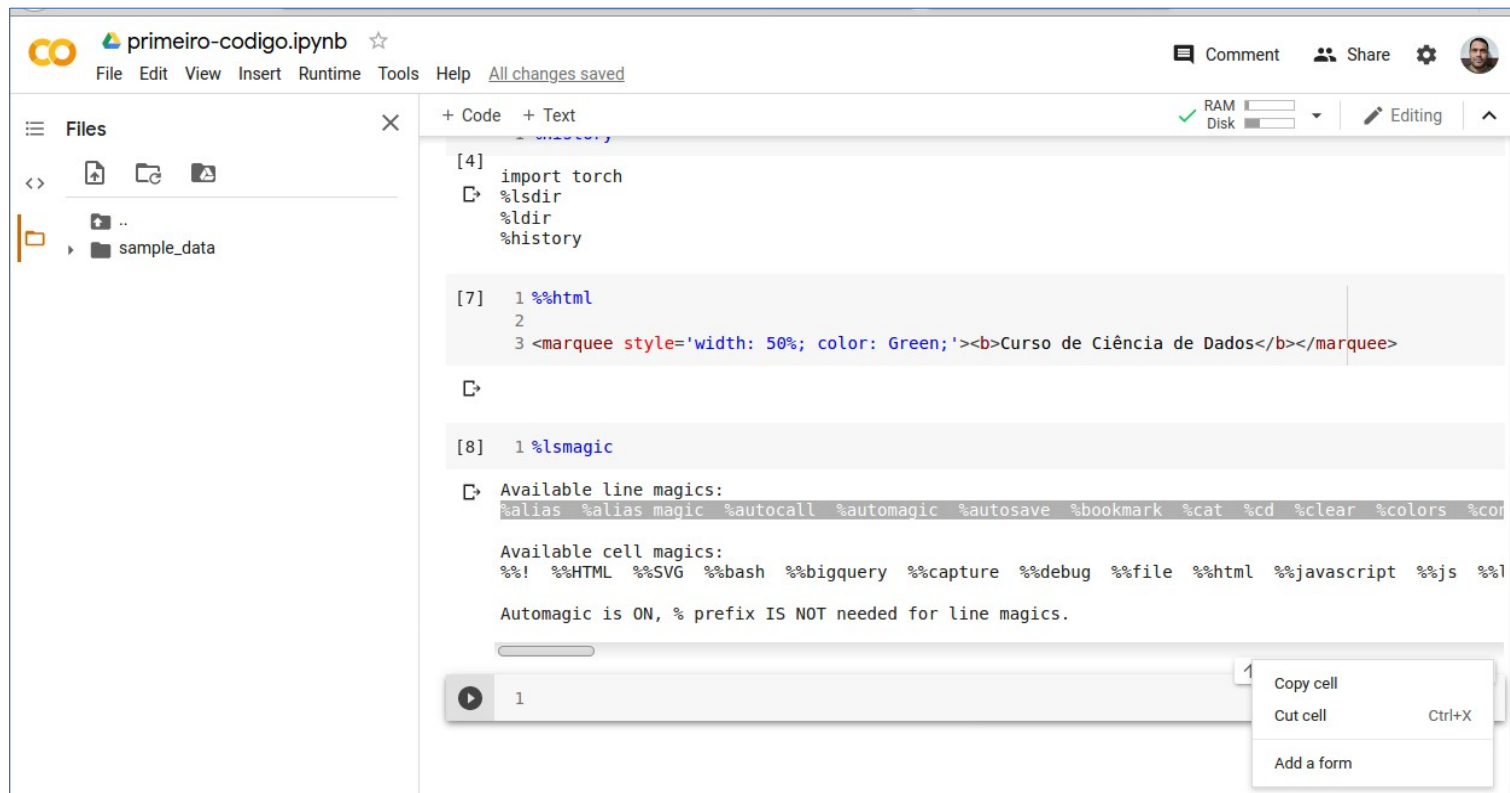
- **Passo 1:**

# Inserção de Formulários

- **Passo 1:**
  - Adiciona o formulário.

# Inserção de Formulários

- **Passo 1:**
  - Adiciona o formulário.



The screenshot shows a Jupyter Notebook window titled "primeiro-codigo.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a file browser on the left showing a "sample\_data" folder, and a main code editor area. The code editor contains three cells:

- Cell [4]:

```
import torch
%lsdir
%ldir
%history
```
- Cell [7]:

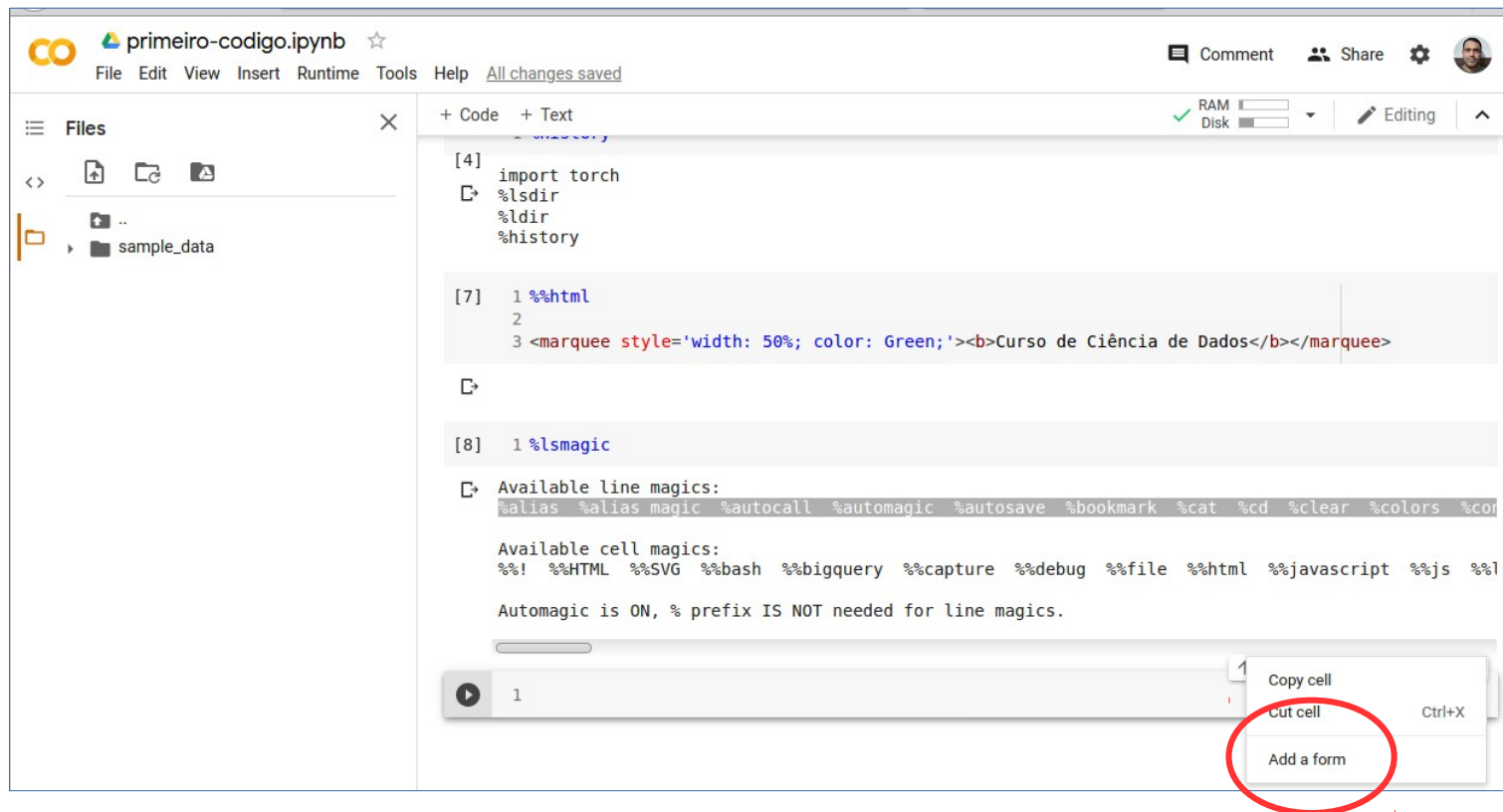
```
1 %%html
2
3 <marquee style='width: 50%; color: Green;'><b>Curso de Ciência de Dados</b></marquee>
```
- Cell [8]:

```
1 %%lsmagic
```

The output of cell [8] shows the available line and cell magics. A context menu is open over the first line of cell [8], with options: "Copy cell", "Cut cell" (with a "Ctrl+X" shortcut), and "Add a form".

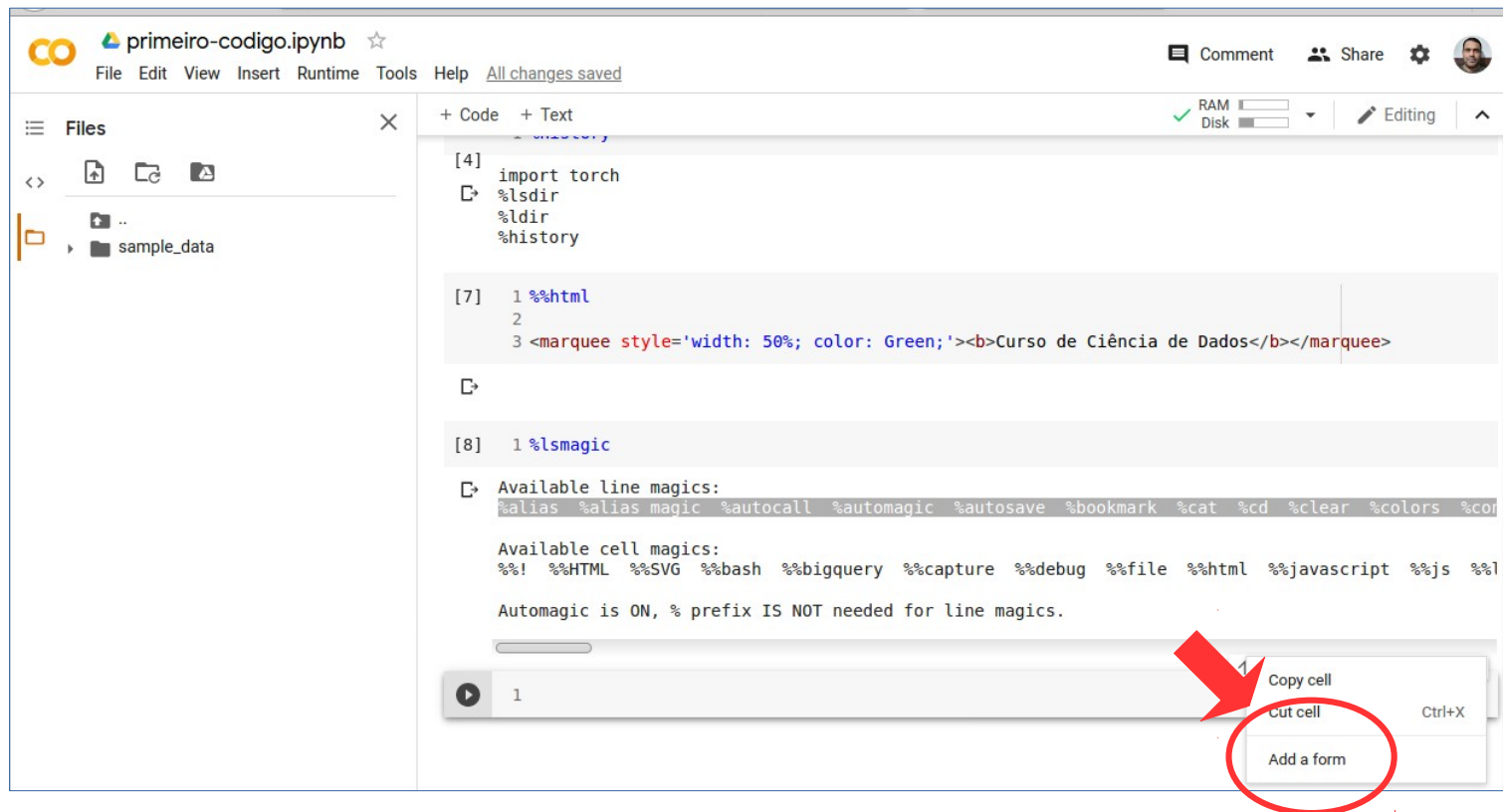
# Inserção de Formulários

- **Passo 1:**
  - Adiciona o formulário.



# Inserção de Formulários

- **Passo 1:**
  - Adiciona o formulário.



# Inserção de Formulários

# Inserção de Formulários

- **Passo 2:**

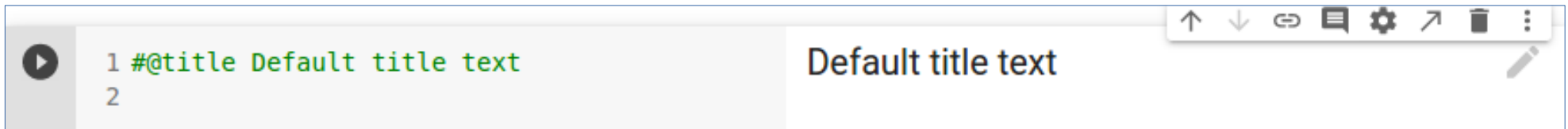
# Inserção de Formulários

- **Passo 2:**
  - Botão de configuração (ícone de caneta).



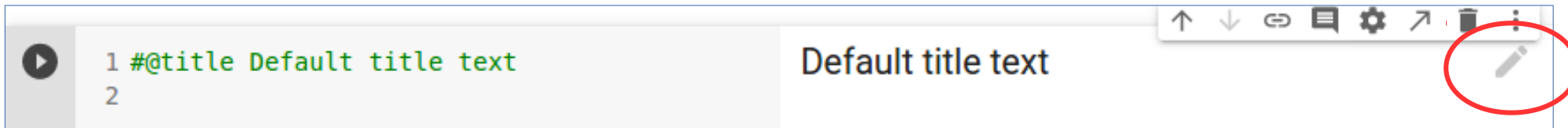
# Inserção de Formulários

- **Passo 2:**
  - Botão de configuração (ícone de caneta).



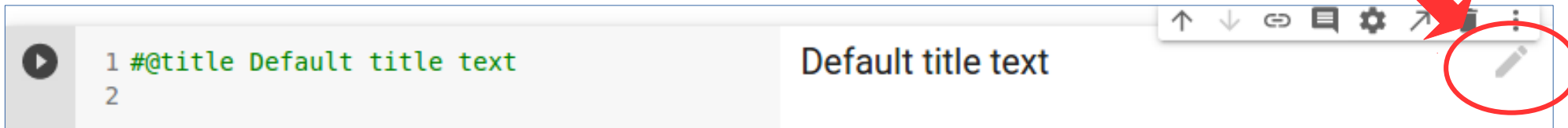
# Inserção de Formulários

- **Passo 2:**
  - Botão de configuração (ícone de caneta).



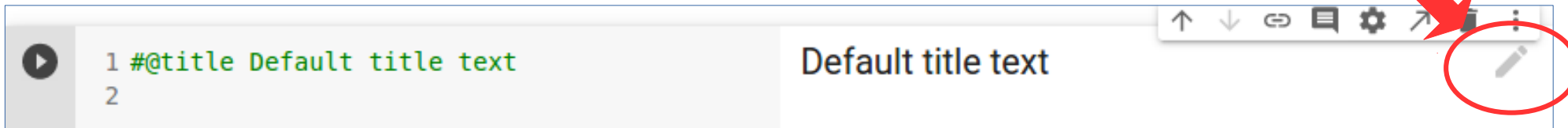
# Inserção de Formulários

- **Passo 2:**
  - Botão de configuração (ícone de caneta).



# Inserção de Formulários

- **Passo 2:**
  - Botão de configuração (ícone de caneta).



### Edit form attributes

Title text  
Default title text

---

Initial form visibility  
last shown (default) ▼

Output height \_\_\_\_\_ px

Form width \_\_\_\_\_ px ▼

Auto-execute cell when fields change

Display output below form

CANCEL SAVE

# Inserção de Formulários

# Inserção de Formulários

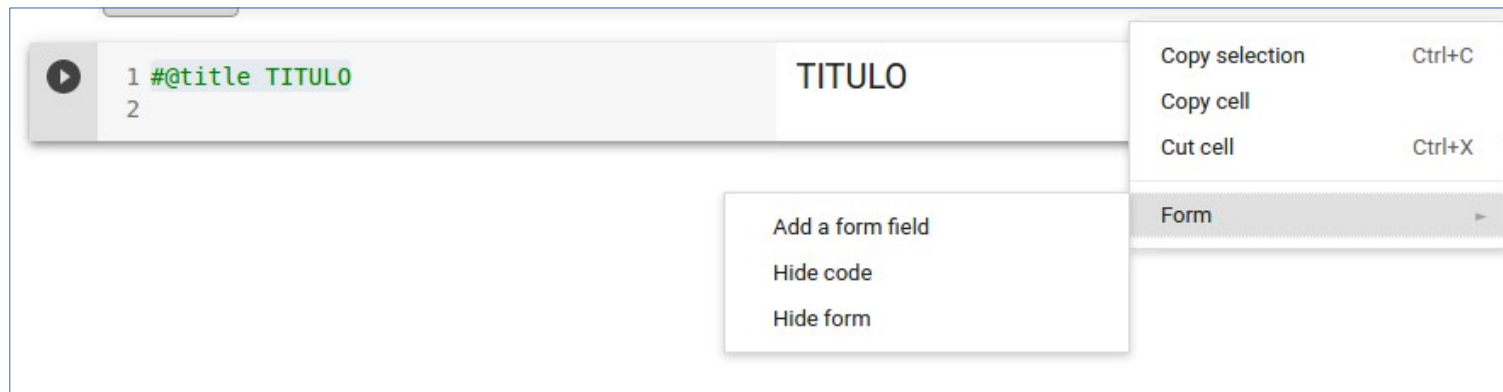
- Passo 3:

# Inserção de Formulários

- Passo 3:
  - Adicionar formulário:

# Inserção de Formulários

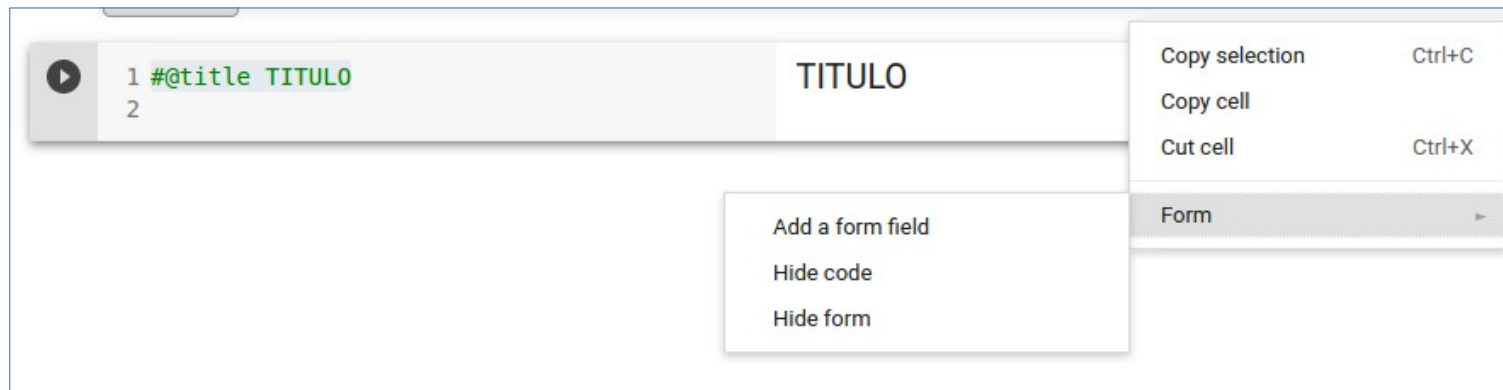
- Passo 3:
  - Adicionar formulário:





# Inserção de Formulários

- Passo 3:
  - Adicionar formulário:



### Add new form field

Form field type

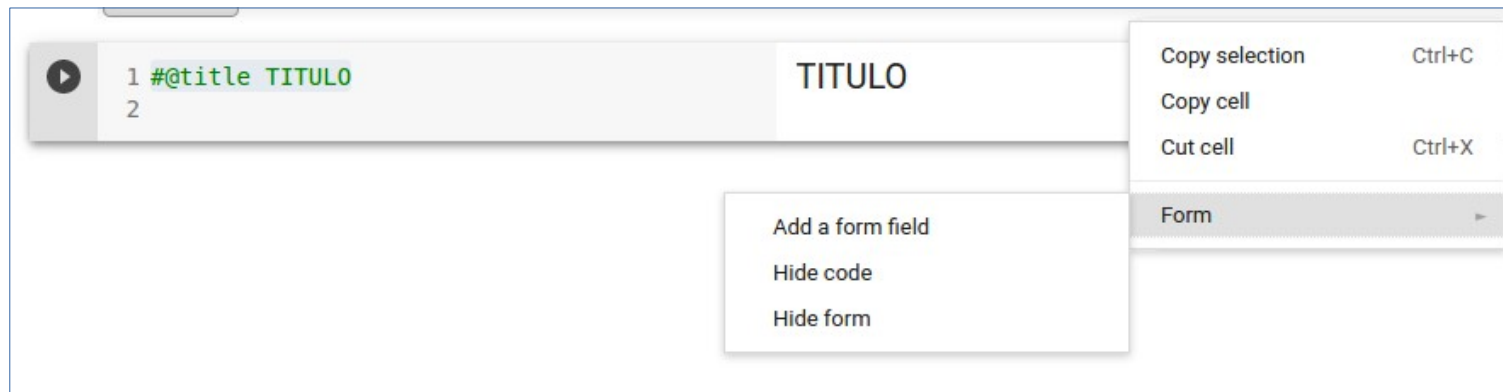
Variable type

Variable name

CANCEL SAVE

# Inserção de Formulários

- Passo 3:
  - Adicionar formulário:



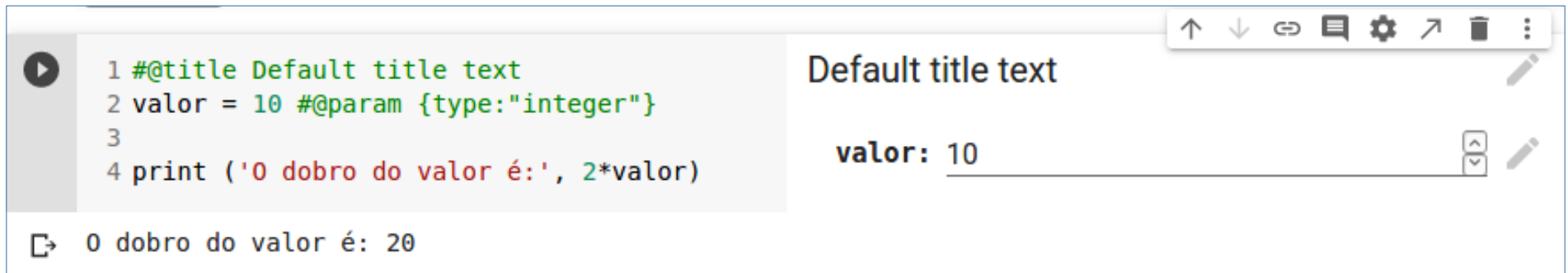
The "Add new form field" dialog box is shown. It has a title "Add new form field" and two dropdown menus. The first dropdown menu is labeled "Form field type" and has "input" selected. The second dropdown menu is labeled "Variable type" and has "string" selected. Below the dropdowns is a text input field labeled "Variable name" with the text "variable\_name" entered. At the bottom right of the dialog are "CANCEL" and "SAVE" buttons. Red arrows point to the "Form field type" and "Variable type" dropdowns, and red circles highlight the selected options "input" and "string".

# Inserção de Formulários

- Uso do formulário:

# Inserção de Formulários

- Uso do formulário:



The image shows a code editor window with a light gray background. On the left, there is a code editor with a play button icon and a list of four lines of Python code:

```
1 #@title Default title text
2 valor = 10 #@param {type:"integer"}
3
4 print ('0 dobro do valor é:', 2*valor)
```

Below the code editor, there is a terminal output showing the result of the execution:

```
0 dobro do valor é: 20
```

On the right side of the editor, there is a preview of the rendered HTML. It shows the title "Default title text" and a form input field labeled "valor: 10". The input field has a small spinner icon on the right side, indicating it is an integer input field. The preview also includes a toolbar with various icons like up, down, link, chat, settings, share, and delete.

# Entrada de Texto

# Entrada de Texto

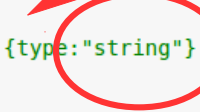
- Para aceitar a entrada de texto como parâmetro, basta alterar o tipo de dados.

# Entrada de Texto

- Para aceitar a entrada de texto como parâmetro, basta alterar o tipo de dados.
- Exemplo:

# Entrada de Texto

- Para aceitar a entrada de texto como parâmetro, basta alterar o tipo de dados.
- Exemplo:



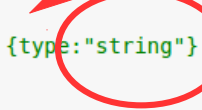
```
1 #@title Default title text
2 nome = "Curso de Programa\xE7\xE3o" #@param {type:"string"}
3
4 print (nome)
```

↳ Curso de Programação



# Entrada de Texto

- Para aceitar a entrada de texto como parâmetro, basta alterar o tipo de dados.
- Exemplo:



```
1 #@title Default title text
2 nome = "Curso de Programa\xE7\xE3o" #@param {type:"string"}
3
4 print (nome)
```

Curso de Programação

Default title text

nome: "Curso de Programa\xE7\xE3o"

# Lista de Dados

# Lista de Dados

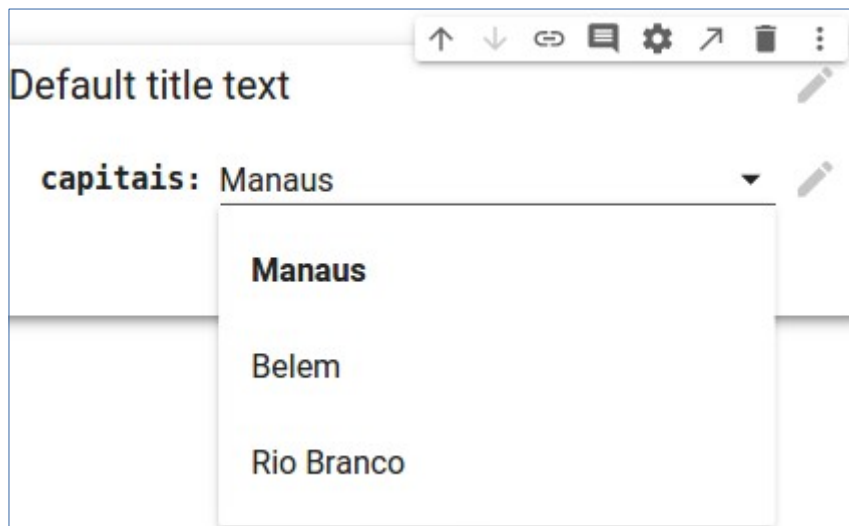
- É possível usar o formulário com lista de dados.

# Lista de Dados

- É possível usar o formulário com lista de dados.
- Exemplo:

# Lista de Dados

- É possível usar o formulário com lista de dados.
- Exemplo:



# Entrada de Datas

# Entrada de Datas

- É possível usar a entrada de datas.

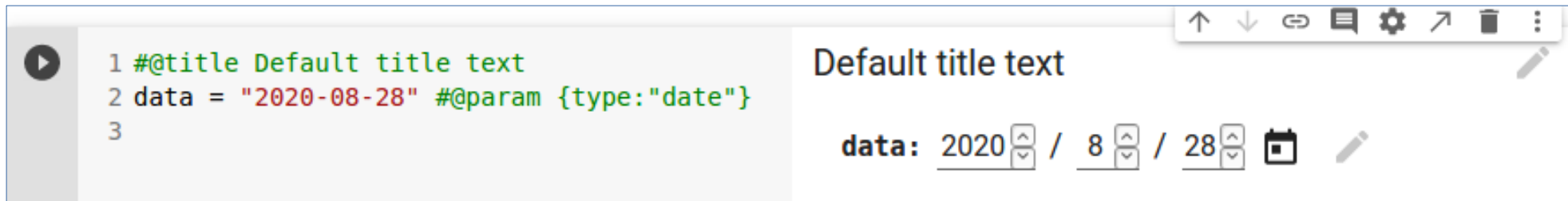
# Entrada de Datas

- É possível usar a entrada de datas.
- Exemplo:



# Entrada de Datas

- É possível usar a entrada de datas.
- Exemplo:



The screenshot shows a document editor interface. On the left, a code editor displays three lines of text: `1 #@title Default title text`, `2 data = "2020-08-28" #@param {type:"date"}`, and `3`. On the right, the rendered output is shown. The title is "Default title text". Below it, the parameter "data" is rendered as a date input field with the value "2020 / 8 / 28" and a calendar icon. The interface includes a toolbar with icons for navigation, search, settings, and editing.

# INSTALAÇÃO DE PACOTES AVANÇADOS

# Introdução

# Introdução

- COLAB suporta a maioria das bibliotecas de *machine learning* disponíveis.

# Introdução

- COLAB suporta a maioria das bibliotecas de *machine learning* disponíveis.
- A instalação é similar ao ambiente Linux.

# Introdução

- COLAB suporta a maioria das bibliotecas de *machine learning* disponíveis.
- A instalação é similar ao ambiente Linux.

**!pip install**

# Introdução

- COLAB suporta a maioria das bibliotecas de *machine learning* disponíveis.
- A instalação é similar ao ambiente Linux.

**!pip install**

**!apt-get install**

# Keras



# Keras

- Escrita em Python e executar usando *frameworks* como TensorFlow ou Theano.

# Keras

- Escrita em Python e executar usando *frameworks* como TensorFlow ou Theano.
- Permite rápida prototipação de aplicações usando redes neurais.

# Keras

- Escrita em Python e executar usando *frameworks* como TensorFlow ou Theano.
- Permite rápida prototipação de aplicações usando redes neurais.
- Suporta uso de GPU.

# Keras

- Escrita em Python e executar usando *frameworks* como TensorFlow ou Theano.
- Permite rápida prototipação de aplicações usando redes neurais.
- Suporta uso de GPU.

```
!pip install -q keras
```

# PyTorch

# PyTorch

- Pacote bastante usado no desenvolvimento de aplicações com redes neurais.

# PyTorch

- Pacote bastante usado no desenvolvimento de aplicações com redes neurais.

```
!pip3 install torch torchvision
```

GPU



# Introdução

- Google fornece o uso gratuito de GPUs.

# Configuração GPUs

# Configuração GPUs

- Para usar o recurso de GPU:



# Configuração GPUs

- Para usar o recurso de GPU:
  - Runtime → Change runtime type

# Configuração GPUs

- Para usar o recurso de GPU:
  - Runtime → Change runtime type

## Notebook settings

Hardware accelerator  
GPU  

To get the most out of Colab, avoid using a GPU unless you need one. [Learn more](#)

Omit code cell output when saving this notebook

CANCEL SAVE

# Listando Dispositivos

# Listando Dispositivos

- É possível descobrir (listar) os dispositivos usados pelo seu notebook na nuvem.

# Listando Dispositivos

- É possível descobrir (listar) os dispositivos usados pelo seu notebook na nuvem.
- Exemplo (CPU):



# Listando Dispositivos

- É possível descobrir (listar) os dispositivos usados pelo seu notebook na nuvem.
- Exemplo (CPU):

```
1 from tensorflow.python.client import device_lib
2 device_lib.list_local_devices()

[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 13740433489619651683, name: "/device:XLA_CPU:0"
 device_type: "XLA_CPU"
 memory_limit: 17179869184
 locality {
 }
 incarnation: 9018186359868751899
 physical_device_desc: "device: XLA_CPU device"]
```

# Listando Dispositivos

- Exemplo (GPU):

# Listando Dispositivos

- Exemplo (GPU):

```
1 from tensorflow.python.client import device_lib
2 device_lib.list_local_devices()

[{"name": "/device:CPU:0",
  device_type: "CPU",
  memory_limit: 268435456,
  locality {
  },
  incarnation: 2684393887369763698, name: "/device:XLA_CPU:0",
  device_type: "XLA_CPU",
  memory_limit: 17179869184,
  locality {
  },
  incarnation: 15107021630547681460,
  physical_device_desc: "device: XLA_CPU device", name: "/device:XLA_GPU:0",
  device_type: "XLA_GPU",
  memory_limit: 17179869184,
  locality {
  },
  incarnation: 16755329250648886052,
  physical_device_desc: "device: XLA_GPU device", name: "/device:GPU:0",
  device_type: "GPU",
  memory_limit: 15695549568,
  locality {
    bus_id: 1
    links {
    }
  },
  incarnation: 16200912918321920207,
  physical_device_desc: "device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:00:04.0, compute capability: 6.0"]
```

# Memória

# Memória

- É possível checar os recursos de memória através do comando abaixo:

# Memória

- É possível checar os recursos de memória através do comando abaixo:

```
!cat /proc/meminfo
```