

Aula 26 - Google Colab

Olá, sejam muito bem-vindos a mais uma aula do curso. Nas aulas anteriores nós vimos como funciona um algoritmo e vimos algumas maneiras de representá-lo, e uma delas foi o português. Bem, como eu havia dito, o português não é propriamente uma linguagem de programação. E para conseguir fazer os nossos algoritmos voltados para aprendizado de máquina, vamos utilizar uma linguagem de programação que cresceu muito nestes últimos anos, que é a linguagem de programação python. O python é uma linguagem muito querida pela comunidade muito por conta da sua imensa biblioteca e também por conta da sua facilidade de aprendizado. Mas para poder usar a linguagem python, vamos precisar de um programa que nos permita trabalhar com a linguagem. Existem diversos programas voltados para este fim, até mesmo para dispositivos móveis. Mas a fim de tornar o curso acessível a todos, vamos utilizar uma plataforma que tem crescido nos últimos anos que nos vai permitir programar com a linguagem sem mesmo sequer ter nenhum software dedicado no computador, apenas o seu próprio navegador. O nome dessa ferramenta é Google Colab.

O que é o Google Colab?

O Google Colab é uma plataforma do Google que permite que qualquer pessoa seja capaz de executar um código em python pelo navegador. Quem já utilizou o Google Docs para fazer o seu documento? O Google Colab funciona de maneira semelhante onde permite que os códigos criados sejam rodados na nuvem do Google. E ainda, de maneira semelhante ao Google Docs, o Google Colab nos permite compartilhar o código com quem nós quisermos e ainda cada um pode editar o código em tempo real.

Vantagens:

A vantagem de se usar essa ferramenta é que o usuário não fica restrito ao desempenho do seu próprio computador, uma vez que este serviço irá rodar nos servidores do Google, nos permitindo até uma grande capacidade de processamento, o desempenho do computador é fundamental quando estamos tentando trabalhar com um grande volume de dados. Além disso, para se trabalhar com a linguagem python no nosso próprio computador com aprendizado de máquina. Precisariamos instalar, além do ambiente de desenvolvimento python, as bibliotecas necessárias. No caso do Google Colab isso não é necessário as bibliotecas já estão pré-instaladas e tudo já fica integrado dentro da plataforma. Além de todos estes benefícios, o python não é a única linguagem que é disponível na plataforma. Sendo possível programar em R, Swift dentre outras.

Como usar?

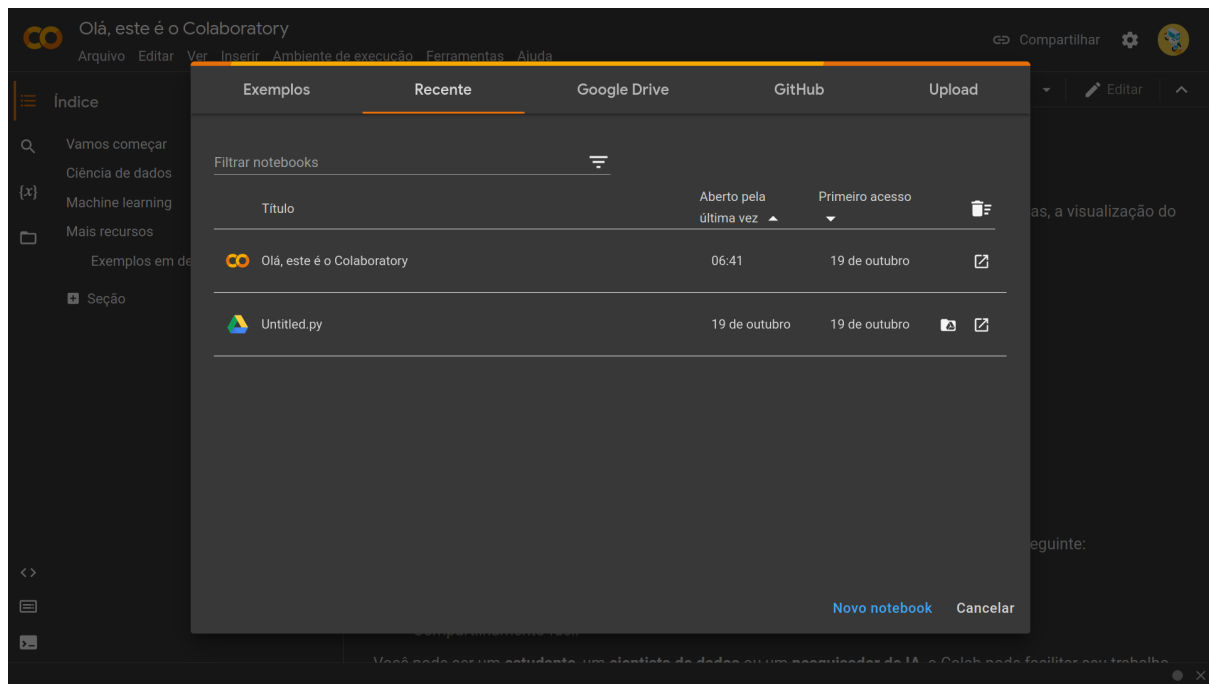
Para ter acesso à plataforma basta apenas que você tenha um navegador com conexão com a internet e uma conta do Google.

Pesquisando no Google por Google Colab você encontrará o site:

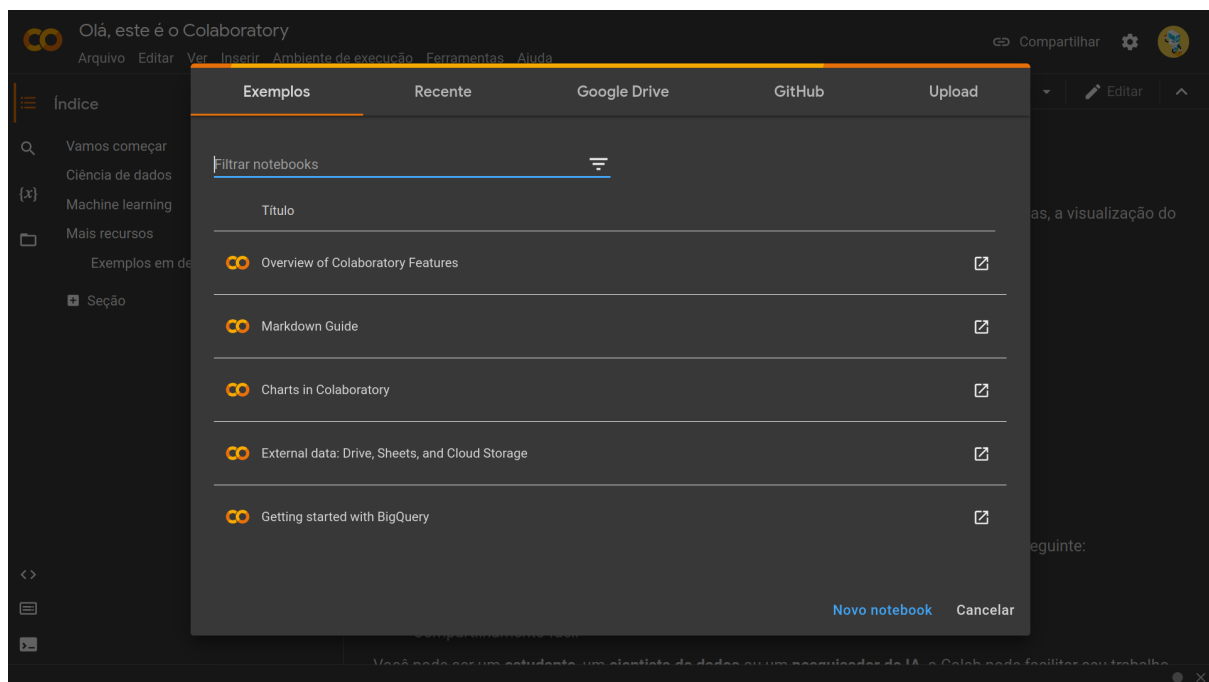
<https://colab.research.google.com/>

Basta que você faça login com a sua conta do Google.

Ao entrar na conta você irá se deparar com esta tela:



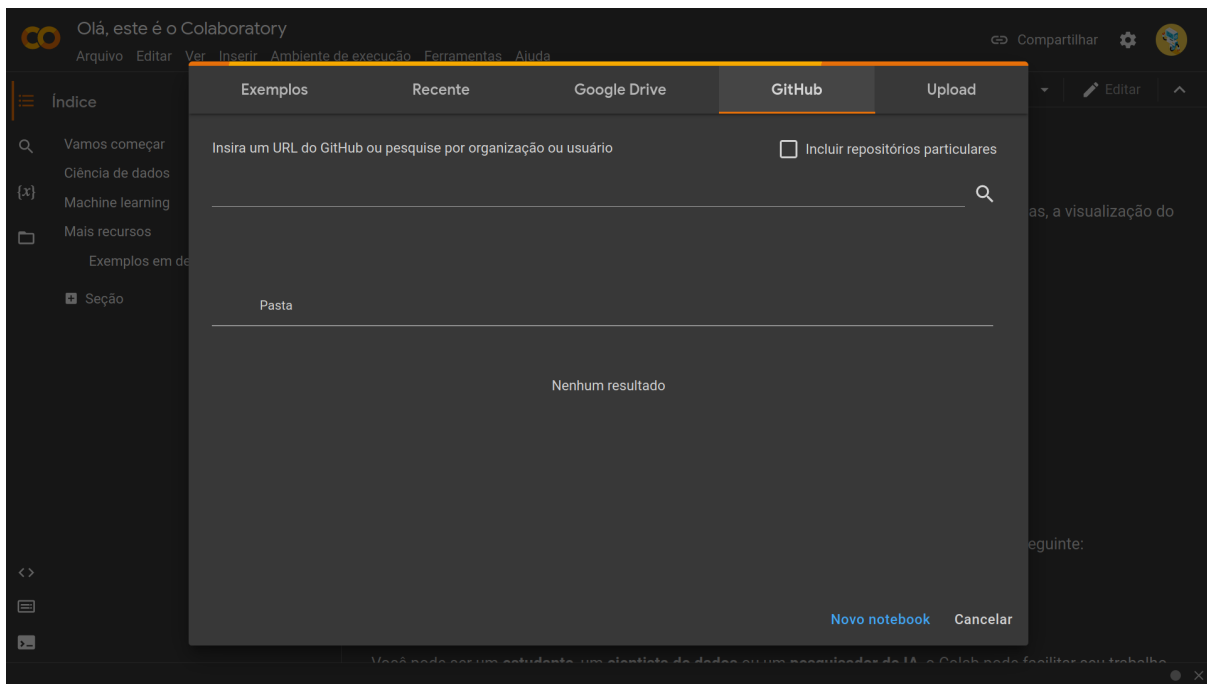
Dentro da própria plataforma existem tutoriais ensinando como usar. Na aba de exemplos você pode encontrar alguns códigos que são para mostrar funcionalidades para aprender a usar a ferramenta:



Os projetos feitos nas ferramentas são salvos no formato notebook e são armazenados na sua própria conta do google, no google drive para ser mais específico.

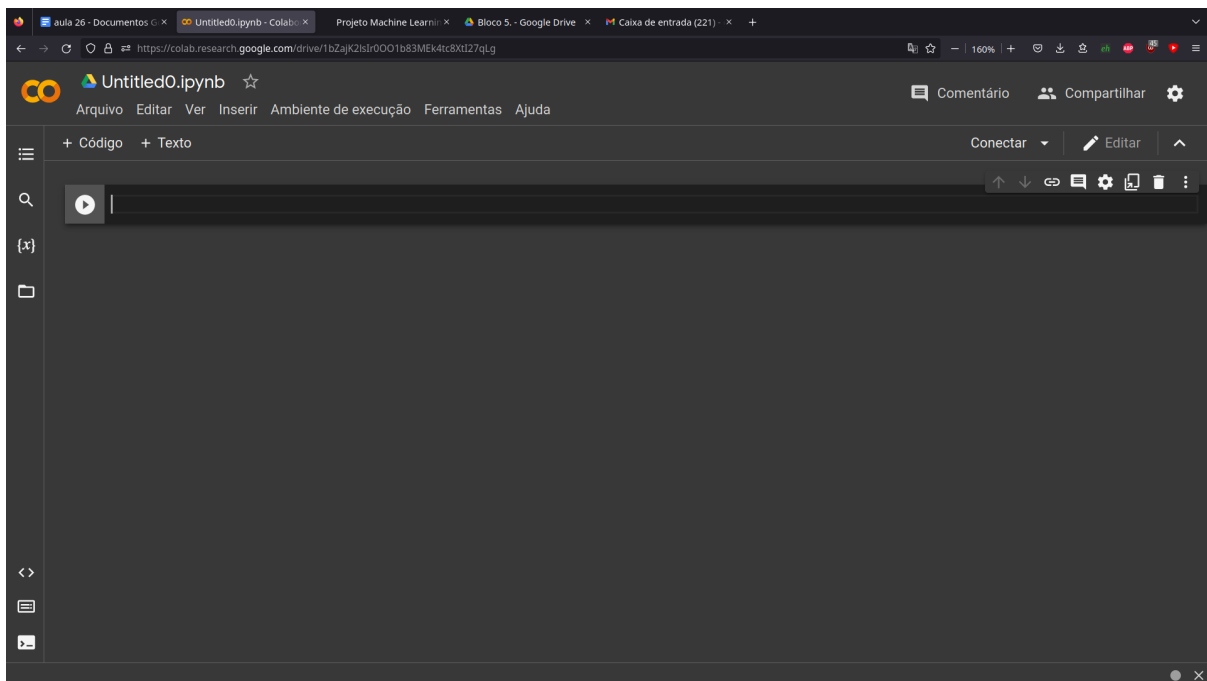
A aba google drive permite a você ter acesso aos seus arquivos do drive, a aba upload serve para que você envie um arquivo que está no seu computador local.

Além disso, uma aba interessante a se falar aqui é a aba GitHub que permite que você importe arquivos diretamente de um repositório do GitHub.

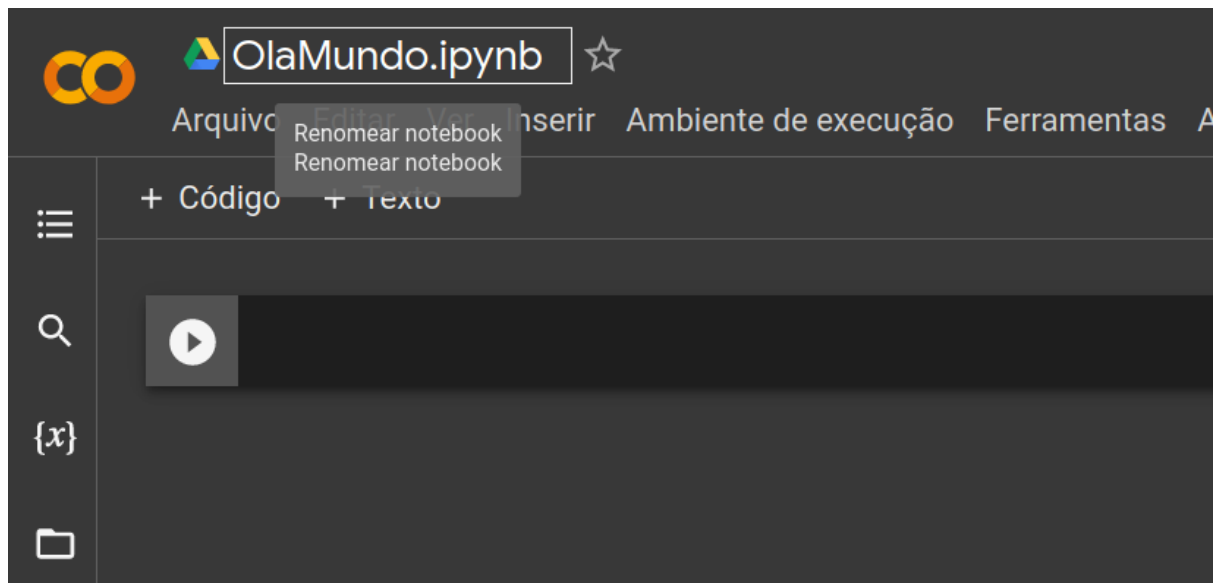


Para quem não conhece o GitHub, esta é uma ferramenta muito utilizada pelos programadores que permite que códigos sejam compartilhados com a comunidade além de servir como uma maneira de guardar nossos códigos.

Para criar um arquivo do zero, vamos clicar em novo notebook. Após isso você irá se deparar com esta tela:



E pode renomear o seu projeto clicando nesta parte:



Vamos criar então a nossa primeira linha de código usando a plataforma, e para isso vamos fazer o computador nos mostrar a seguinte mensagem: Olá, mundo!

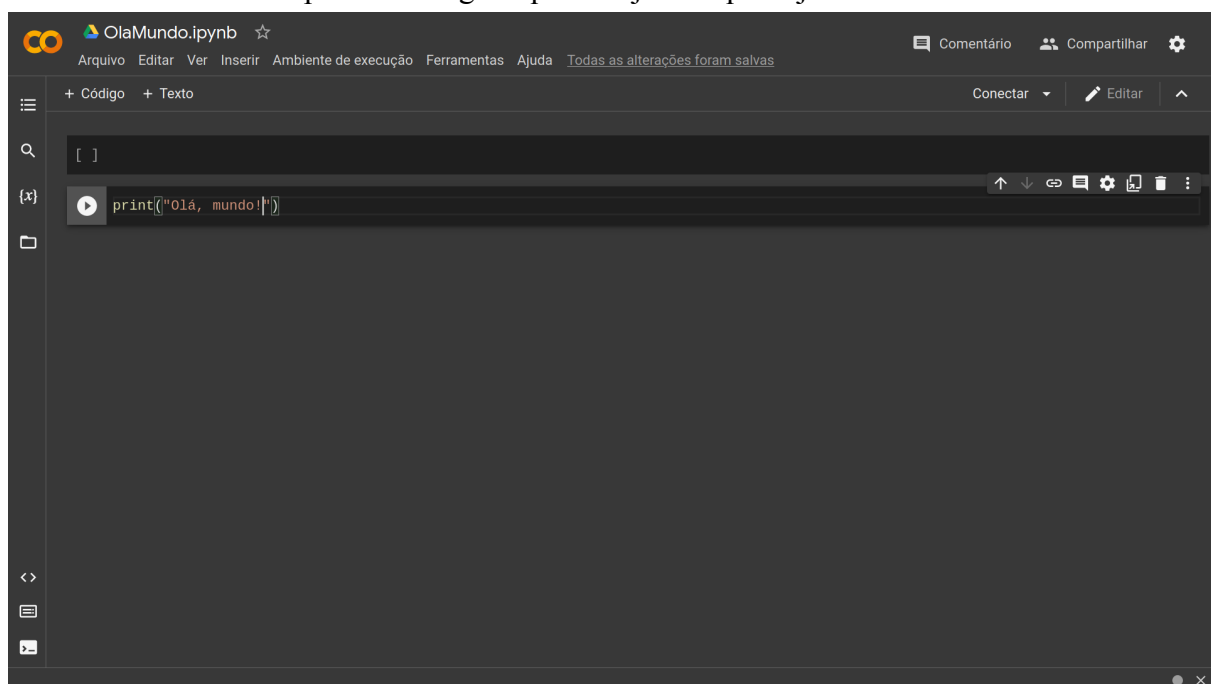
Em português estaremos dizendo:

escreva("Olá, mundo!").

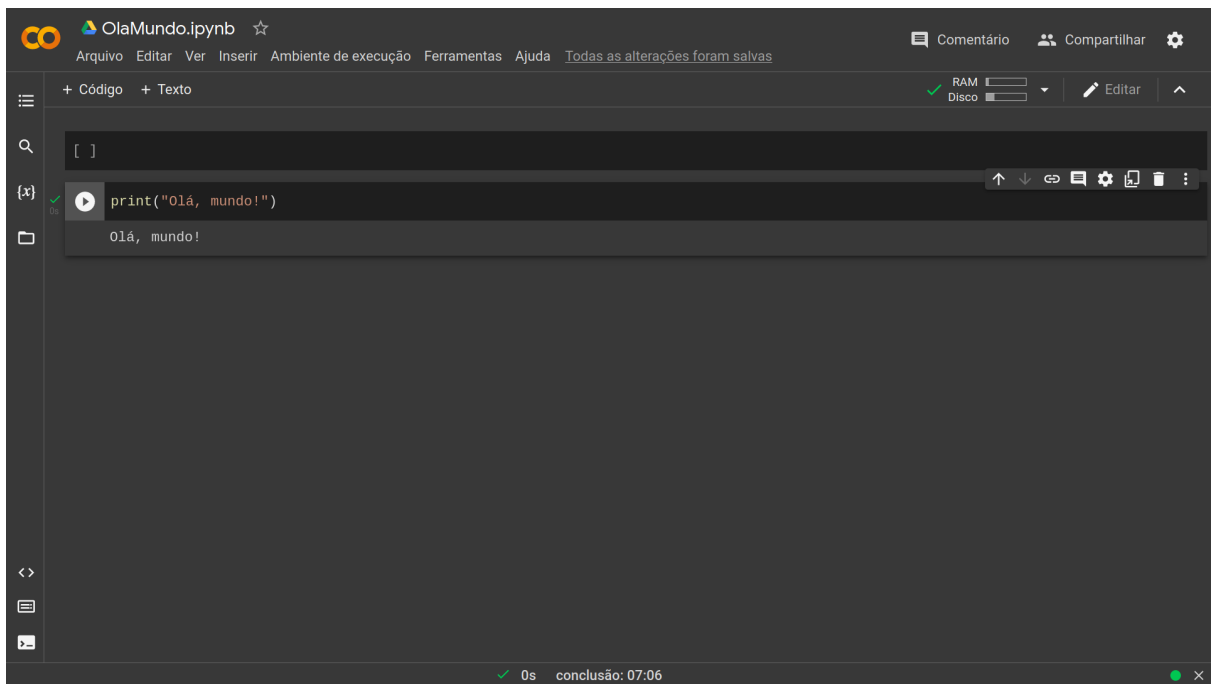
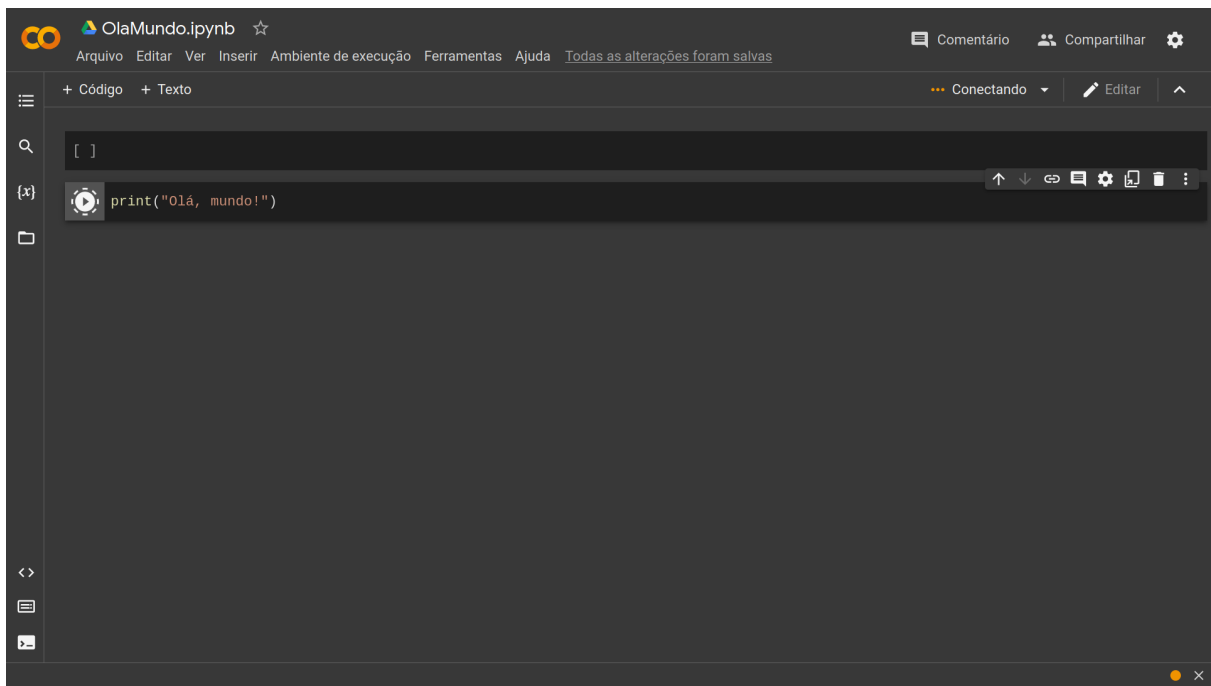
em python essa instrução para escrever na tela será feita da seguinte forma:

escrevemos `print("")`

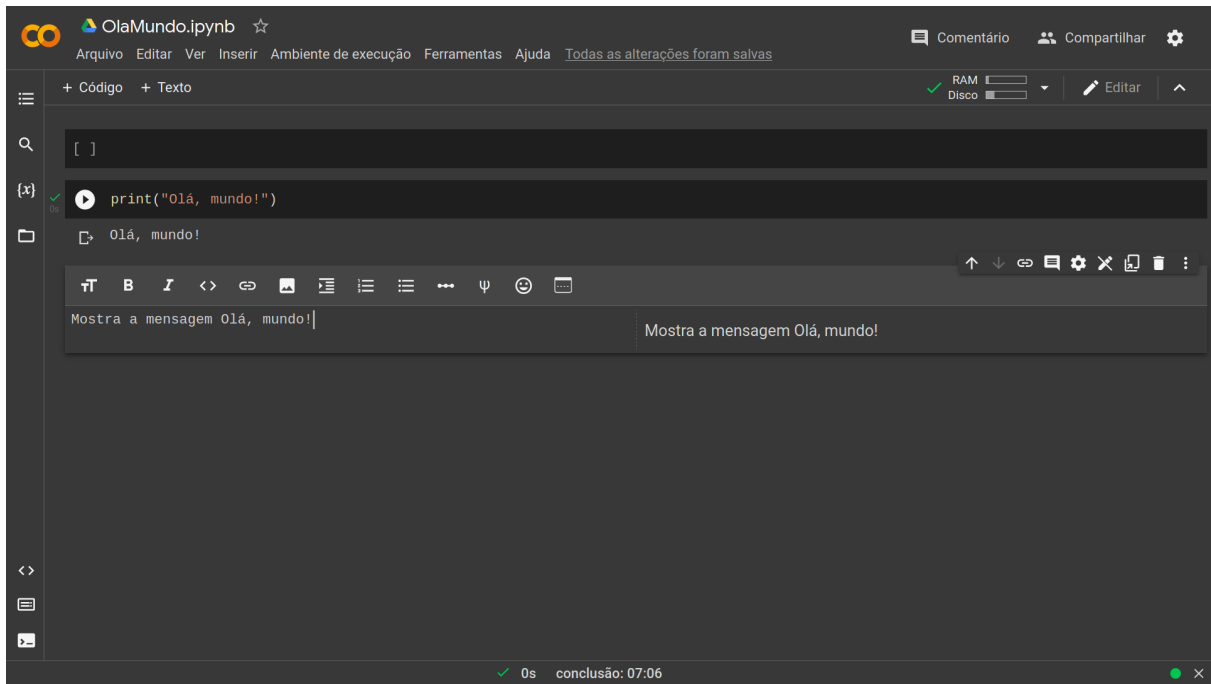
e colocamos entre as aspas a mensagem que desejamos que seja mostrada:



E para executar o código basta clicar no símbolo ao lado da linha de código ou basta utilizar o atalho no teclado `Ctrl + enter`:



Na parte do texto é possível adicionar comentários no seu código:



OlaMundo.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

✓ RAM Disco

[]

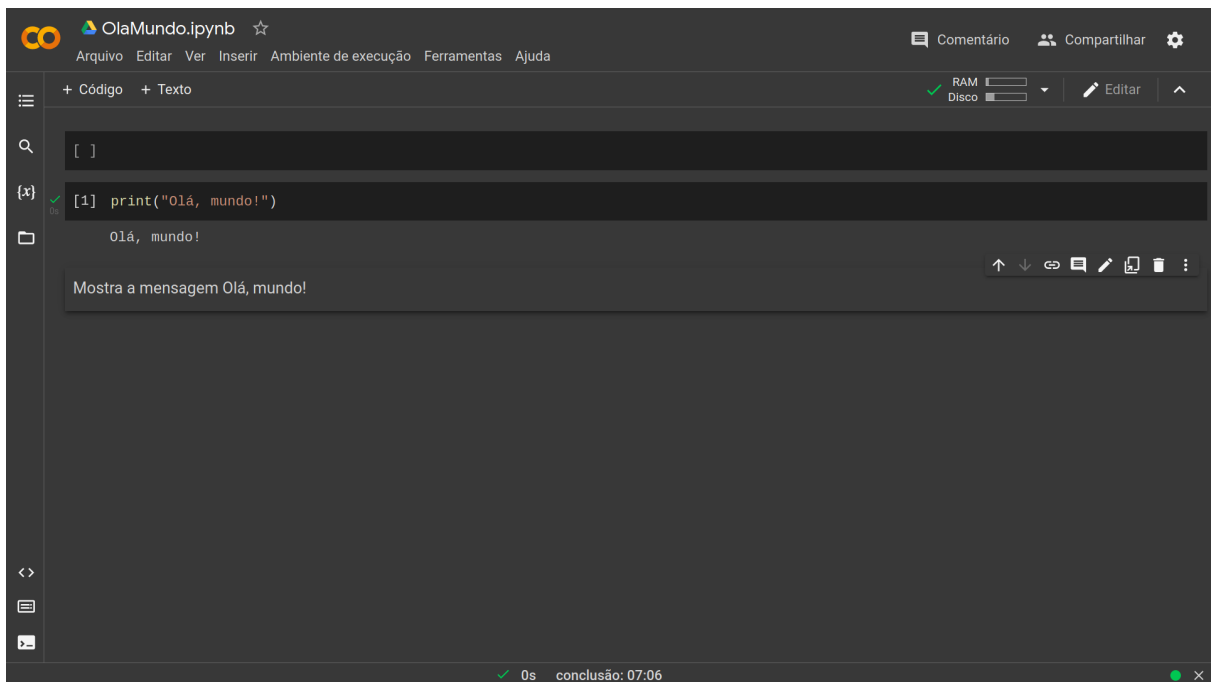
{x} ✓ 0s `print("Olá, mundo!")`

Olá, mundo!

Mostra a mensagem Olá, mundo!

Mostra a mensagem Olá, mundo!

✓ 0s conclusão: 07:06



OlaMundo.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

+ Código + Texto

✓ RAM Disco

[]

{x} ✓ 0s `[1] print("Olá, mundo!")`

Olá, mundo!

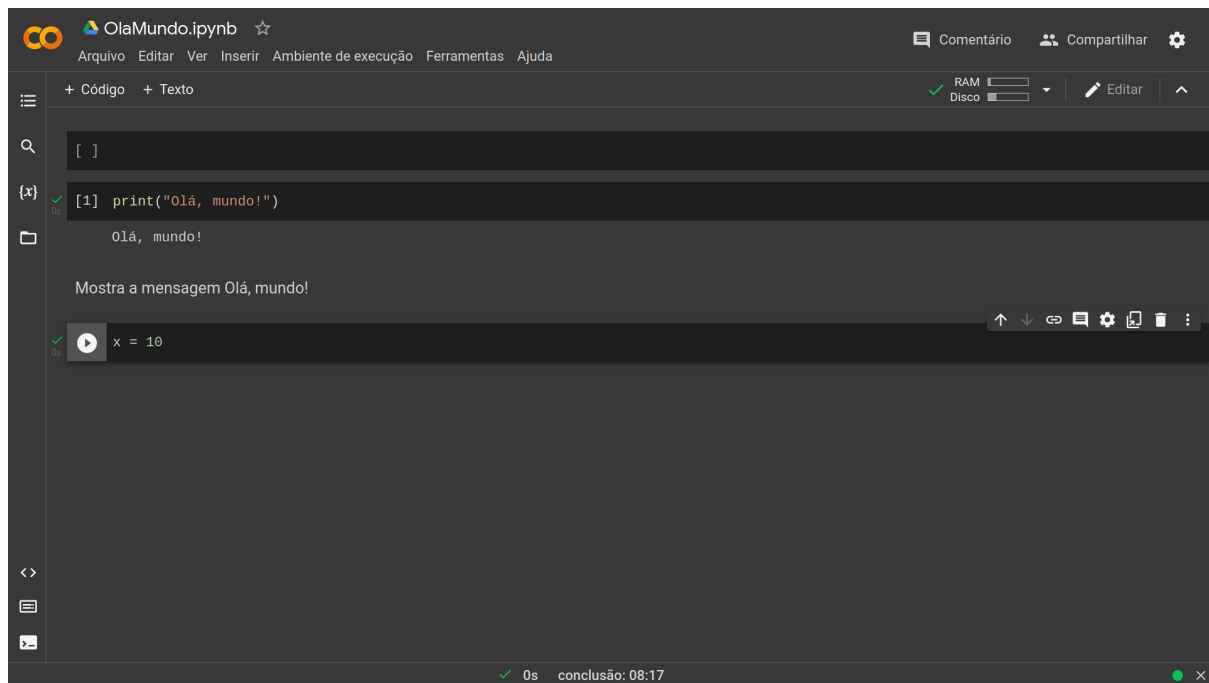
Mostra a mensagem Olá, mundo!

✓ 0s conclusão: 07:06

Agora, por que não tentar fazer um código bem simples:

Vamos fazer uma variável guardar uma informação:

Para declarar uma variável que guarda um valor de um número fazemos o seguinte: `x = 10`.



The screenshot shows a Jupyter Notebook window titled 'OlaMundo.ipynb'. The interface includes a top menu bar with options like 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. Below the menu, there are tabs for '+ Código' and '+ Texto'. The main area contains a code cell with the following content:

```
[ ]
```

```
[1] print("Olá, mundo!")
```

Below the code cell, the output is displayed:

```
Olá, mundo!
```

Mostra a mensagem Olá, mundo!

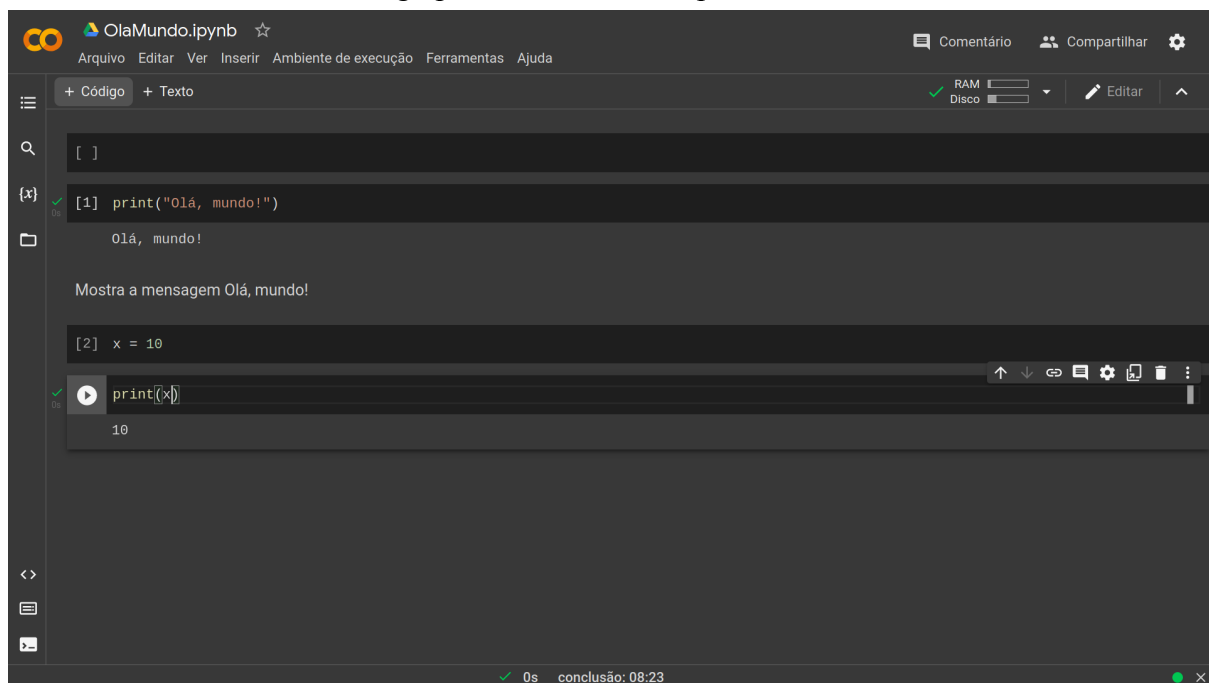
```
x = 10
```

The status bar at the bottom indicates '0s' and 'conclusão: 08:17'.

Se apenas fizermos isso e executarmos o código veremos que nada irá acontecer, isso por que nós precisamos mostrar o que tá salvo na variável.

A verdade é que sim, aconteceu algo, internamente a nossa variável `x` está na memória do nosso computador, que no caso são os servidores do google. Mas para que a gente possa ver o que aconteceu é necessário que a gente mostre o valor da nossa variável.

Usando a mesma linha de código para mostrar na tela podemos fazer:



The screenshot shows the same Jupyter Notebook window. The code cell now contains:

```
[2] x = 10
```

```
print(x)
```

The output of the code cell is:

```
10
```

The status bar at the bottom indicates '0s' and 'conclusão: 08:23'.

Note agora que o dentro do parênteses não tem mais a presença da aspas. Só usamos aspas quando queremos mostrar um texto. Quando queremos mostrar uma variável apenas colocamos ela entre os parênteses.

Se atribuirmos outro valor para nossa variável e executarmos o código, podemos perceber que o valor da variável será mudado.

The screenshot shows a Jupyter Notebook titled 'OlaMundo.ipynb'. The interface includes a top menu bar with options like 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. Below the menu, there's a toolbar with icons for adding code or text, a search icon, and a status bar showing 'RAM' and 'Disco' usage. The notebook content consists of three cells: 1. A code cell with `print("Olá, mundo!")` that has been executed, showing the output 'Olá, mundo!'. 2. A code cell with `x = 10` that has been executed, showing the output '10'. 3. A code cell with `x = 12` and `print(x)` that has been executed, showing the output '12'. The status bar at the bottom indicates '0s' and 'conclusão: 08:26'.

Além disso, podemos ainda efetuar operações.

Vamos criar outra variável e atribuir a ela o valor 100. E então vamos pedir para o nosso código nos mostrar o valor da soma das duas variáveis.

The screenshot shows the same Jupyter Notebook interface, but with additional code cells. The notebook now contains six cells: 1. `print("Olá, mundo!")` (output: 'Olá, mundo!'). 2. `x = 10` (output: '10'). 3. `x = 12` (output: '12'). 4. `y = 100` (output: '100'). 5. `print(x)` (output: '12'). 6. `print(x + y)` (output: '112'). The status bar at the bottom indicates '0s' and 'conclusão: 08:29'.

Sempre lembrando, que temos de executar cada linha do código.

Mas por essa aula é isto, nas próximas aulas vamos entender o básico da linguagem de programação python.