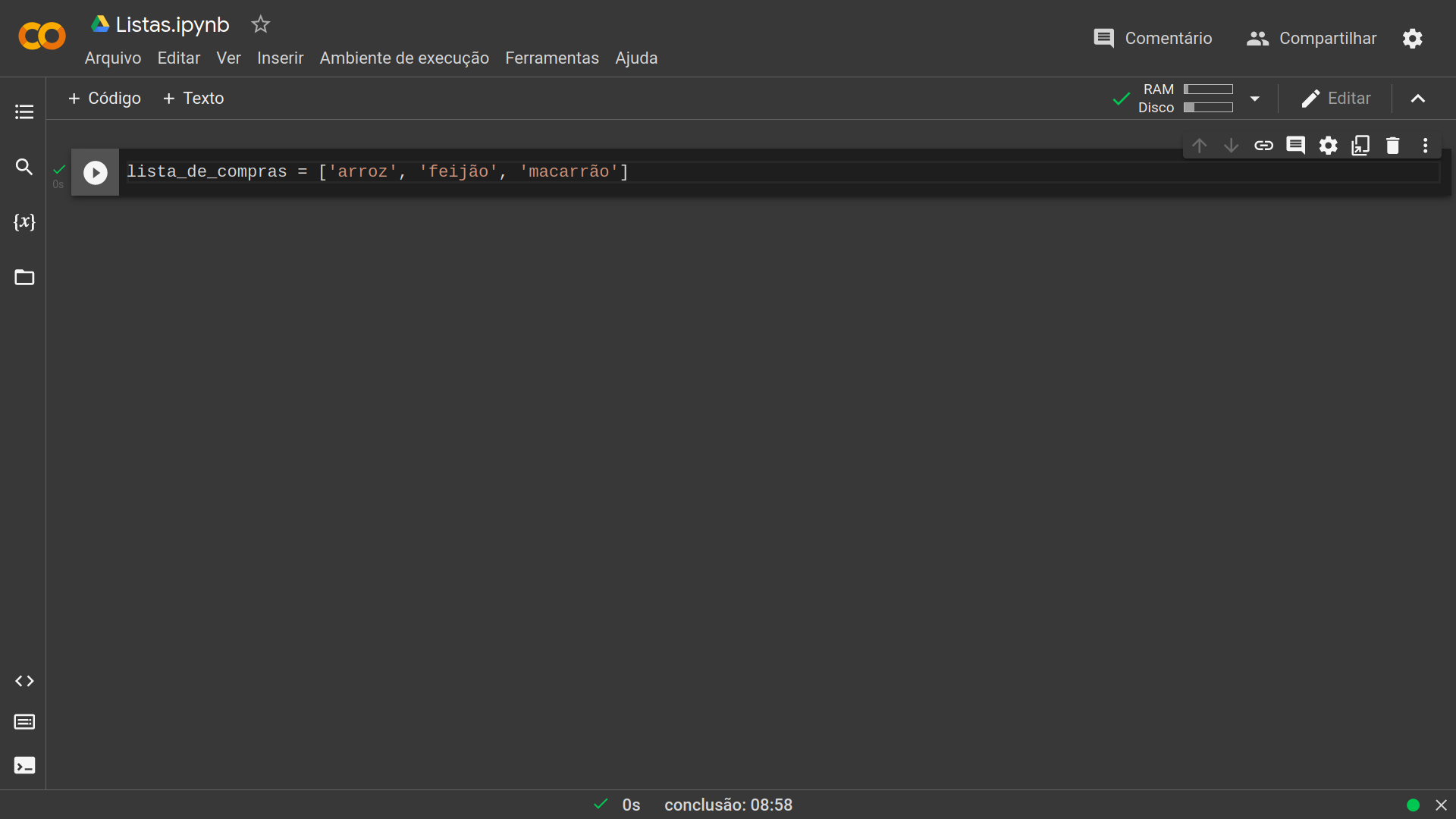
**Aula 30 - Estrutura de dados - listas, tuplas e dicionários**

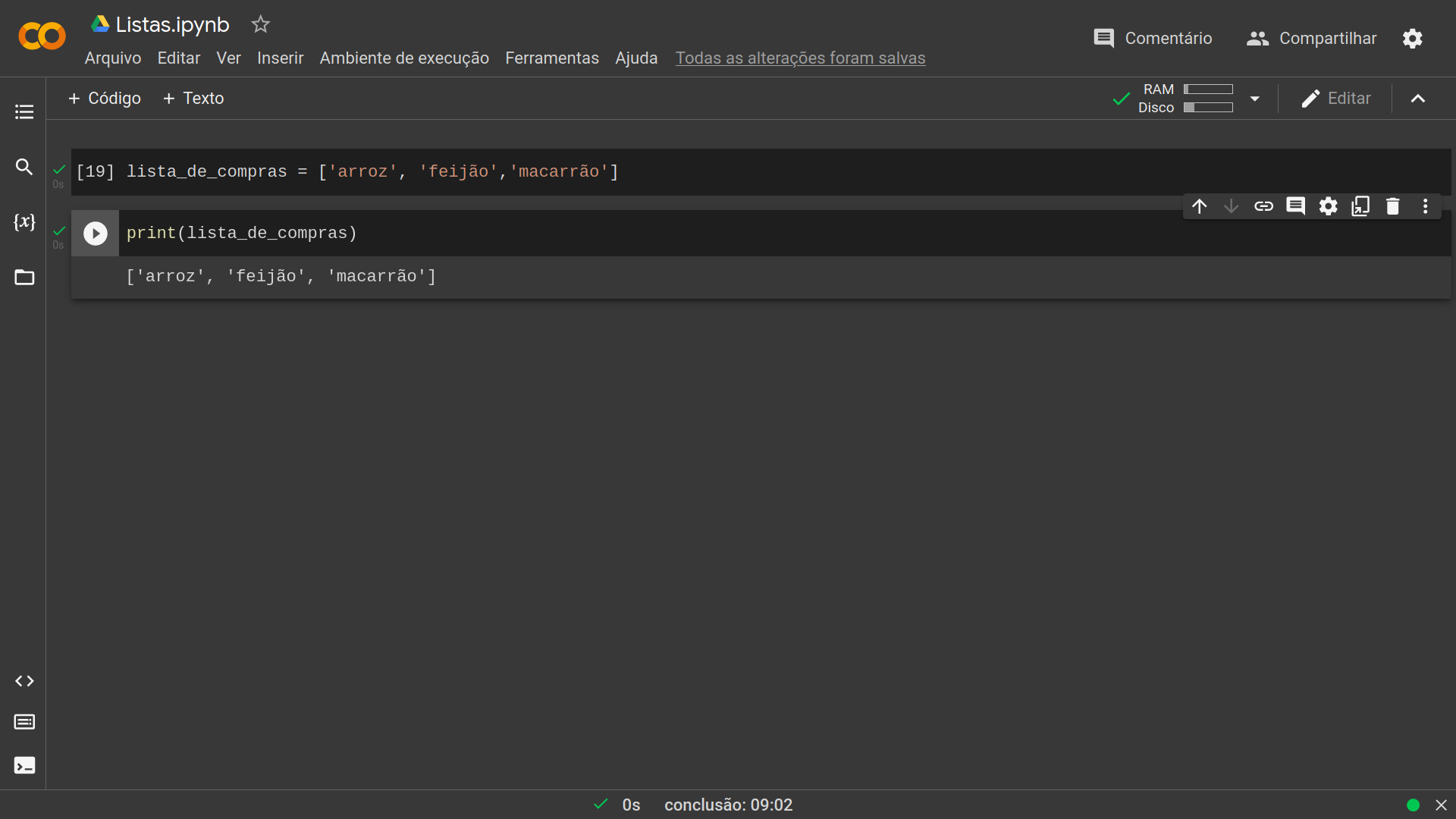
Olá, sejam bem-vindos a mais uma aula, bem nas duas aulas anteriores nós vimos um pouco mais a fundo sobre variáveis. Eu não cheguei a falar, mas uma variável também representa uma estrutura de dados. Mas o que é uma estrutura de dados.

Estrutura de dados em programação, são responsáveis por armazenar dados, igual o que fazíamos com as variáveis, por exemplo, onde a gente guardava um dado, um nome ou um número, ou até mesmo um valor booleano. E existem algumas outras estruturas de dados além dessa, como listas, pilhas, filas, árvores, dicionários e tuplas. No python, a estrutura mais simples dessas é a lista.

Uma lista é uma estrutura de dados que é utilizada para armazenar uma sequência de dados. Por exemplo, uma lista de supermercado, é onde você armazena os dados do que vai comprar. Cada produto que você vai comprar, possuí uma posição na lista e em python para representar listas não é muito diferente. Cada posição é denominada índice, vamos ver como se cria uma lista em python:

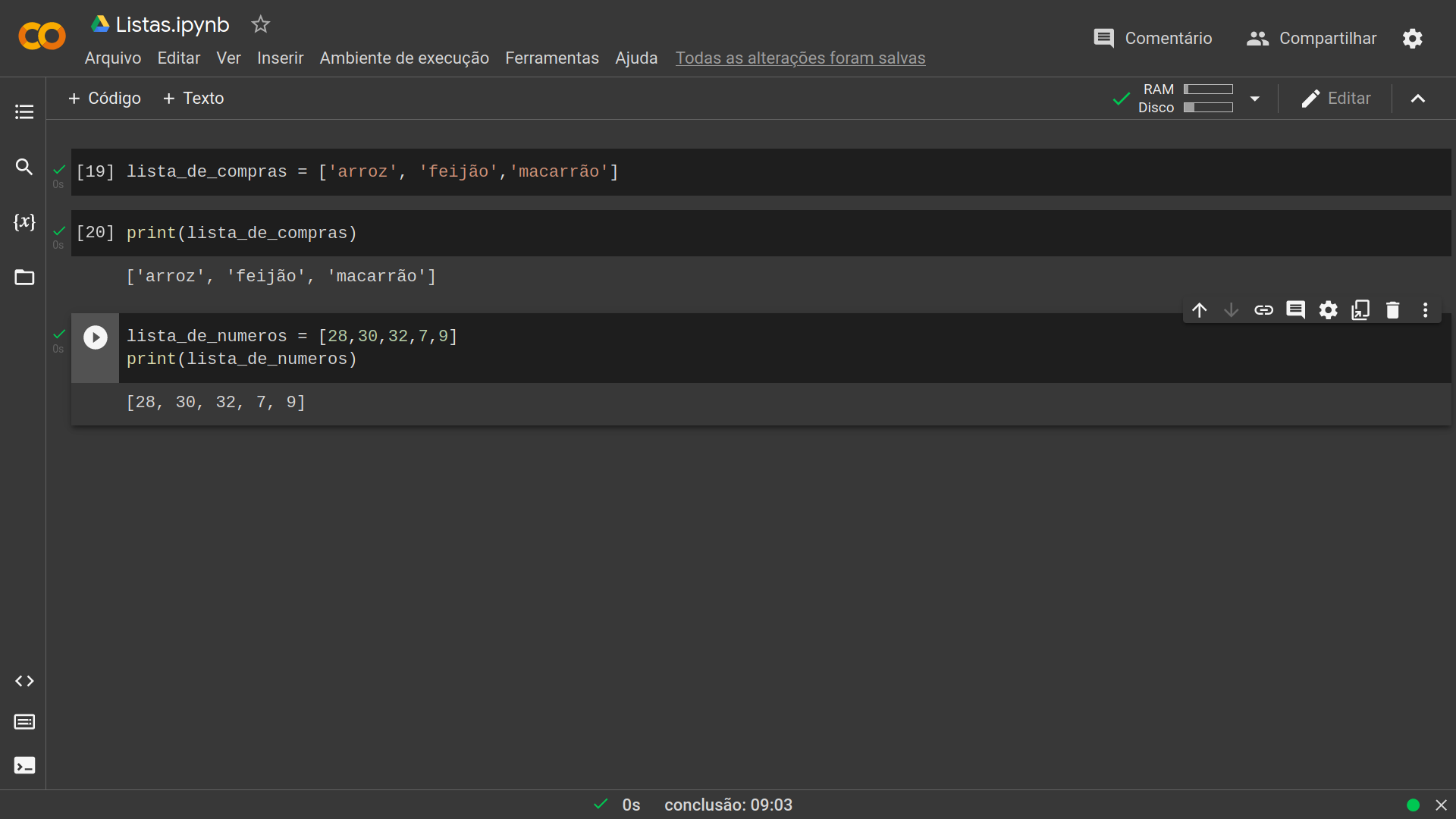


Para declarar uma lista é bem simples, basta escolher o nome da variável e dentro de colchetes, passar as variáveis que desejamos.



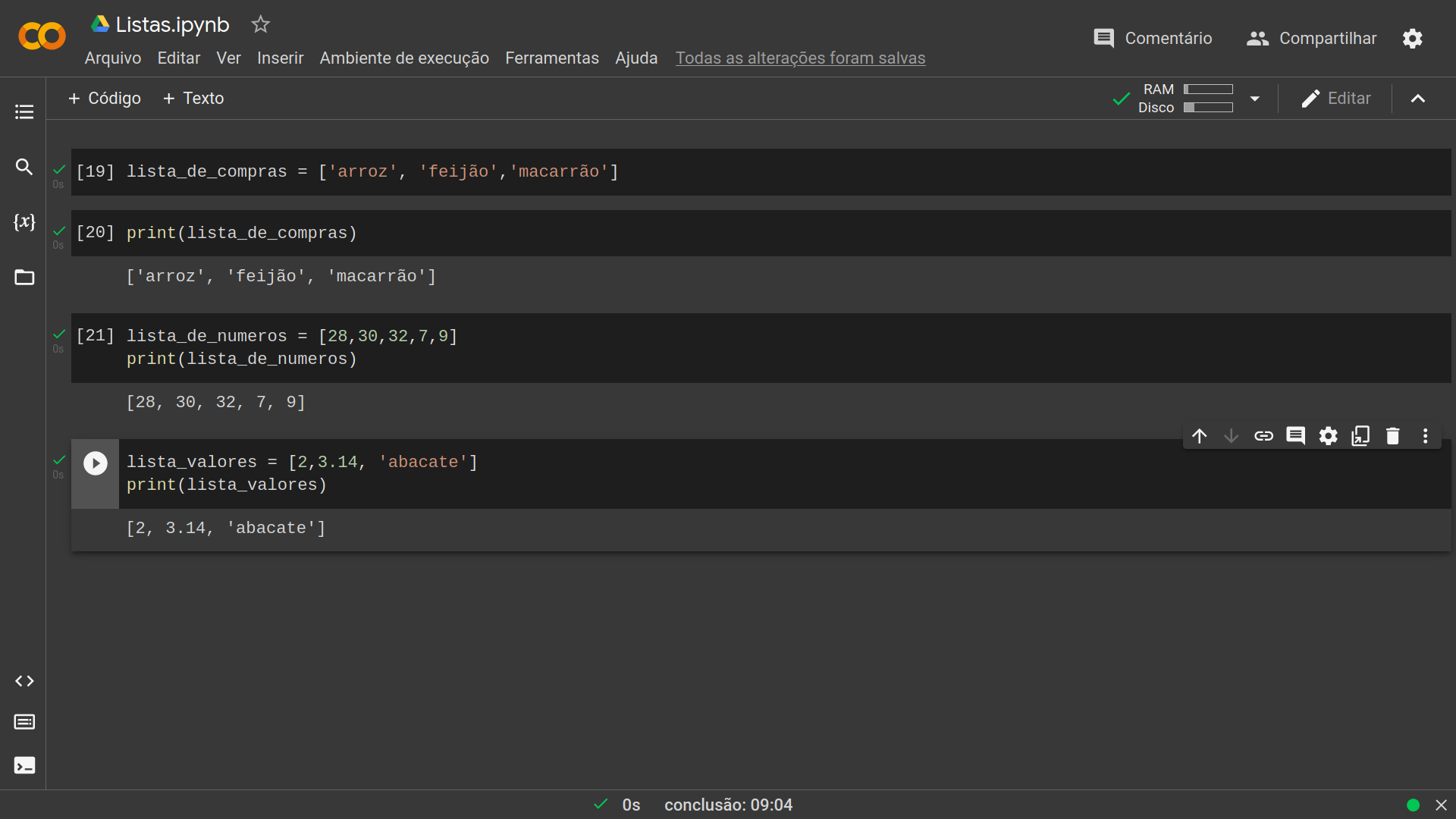
Podemos verificar que a lista está salva.

Podemos fazer o mesmo para uma lista de números:



Dependendo do tipo de dado inserido dentro da lista esse será o tipo de dado que a nossa lista irá armazenar.

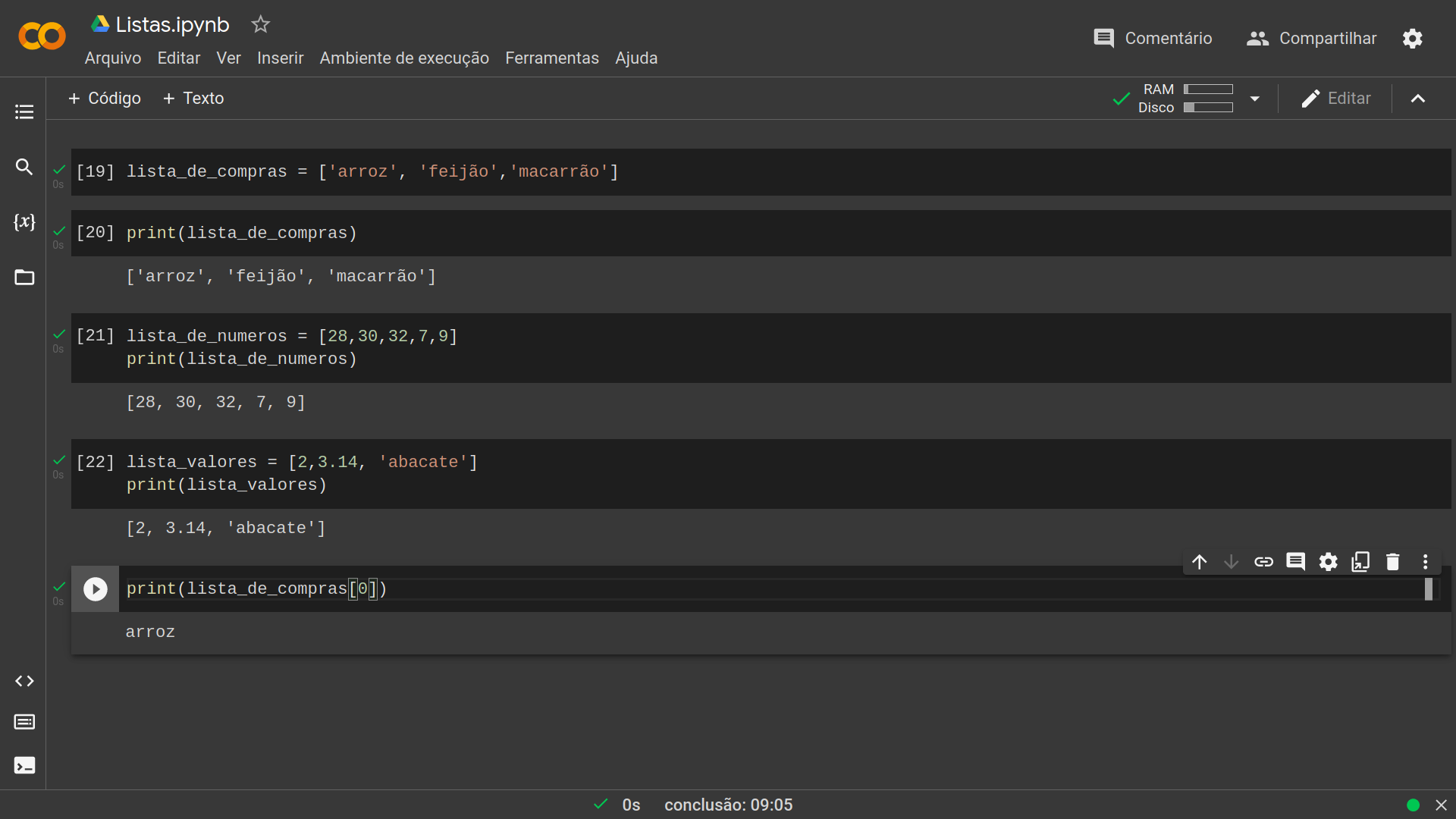
No caso de lista\_de\_compras o tipo de dado que estamos adicionando é String e no outro caso int. O python também nos permite inserir valores de diferentes tipos em uma mesma lista.



Note que nessa lista temos três tipos de dados, float, string e int.

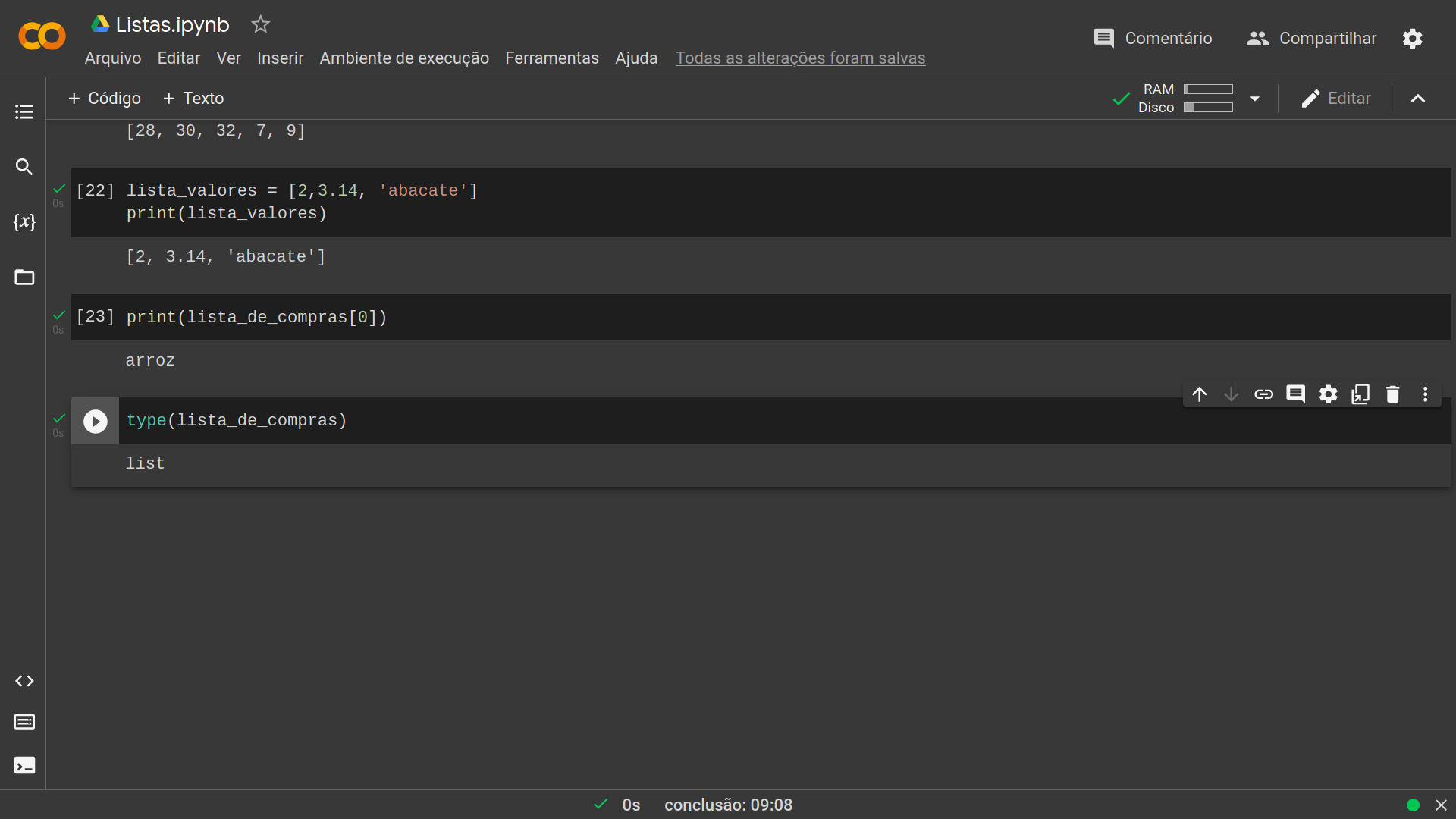
Mas se quisermos acessar algum valor da lista, como podemos fazer?

Cada item da lista é representado por um índice iniciando de 0. Ou seja, o primeiro elemento da lista está na lista e o índice 0. Podemos acessá lo da seguinte maneira:

****

Basta que coloquemos entre colchetes o índice do item que queremos acessar.

Podemos também verificar o tipo de dados.

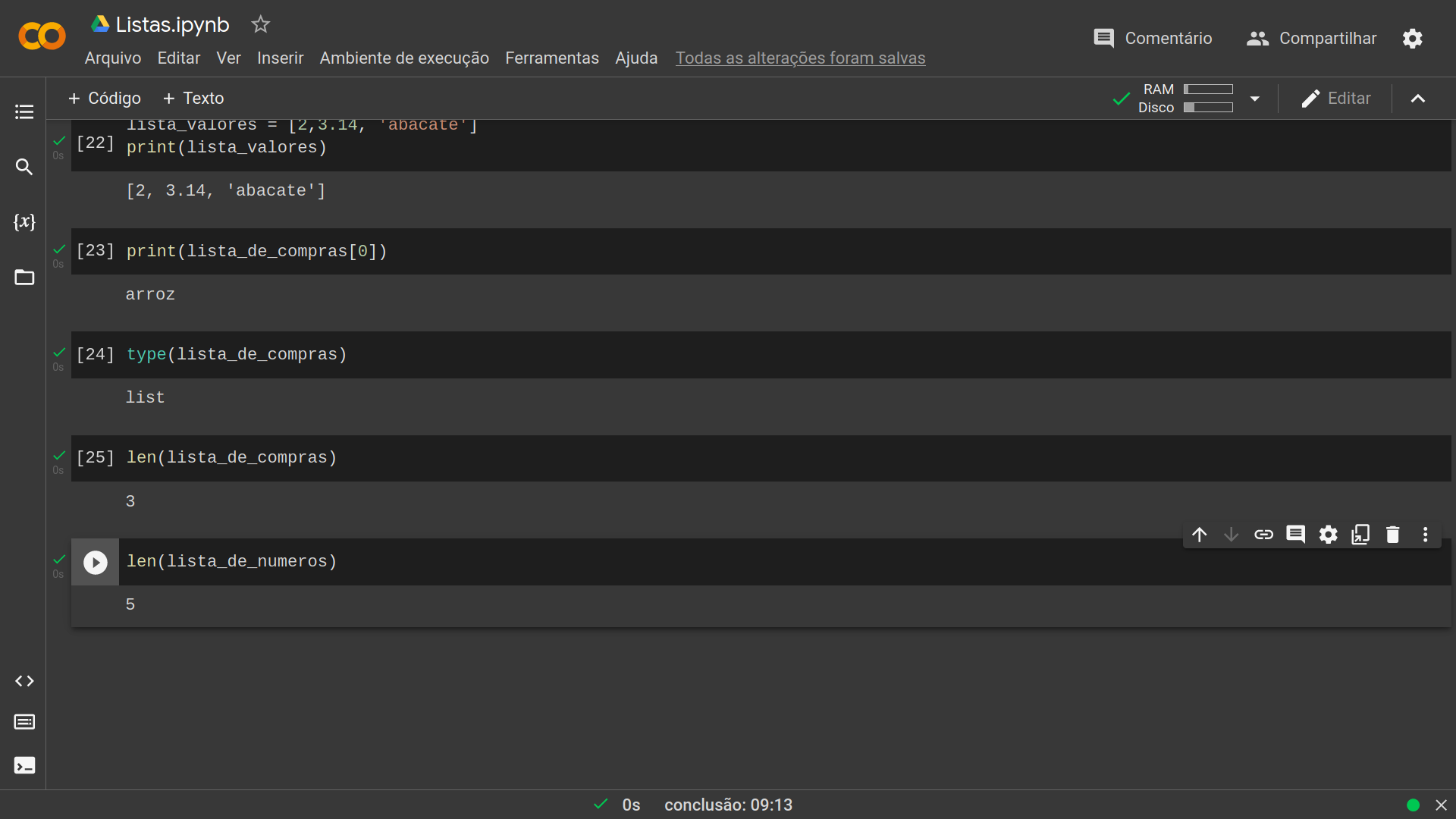


Podemos ver que o tipo da lista é list. Que é um tipo de dados do próprio python. Mas uma vantagem do python, em algumas outras linguagens não podemos armazenar mais de um tipo de variável dentro de uma lista.

O próprio python já possuí algumas funções que podemos usar para manipular as listas:

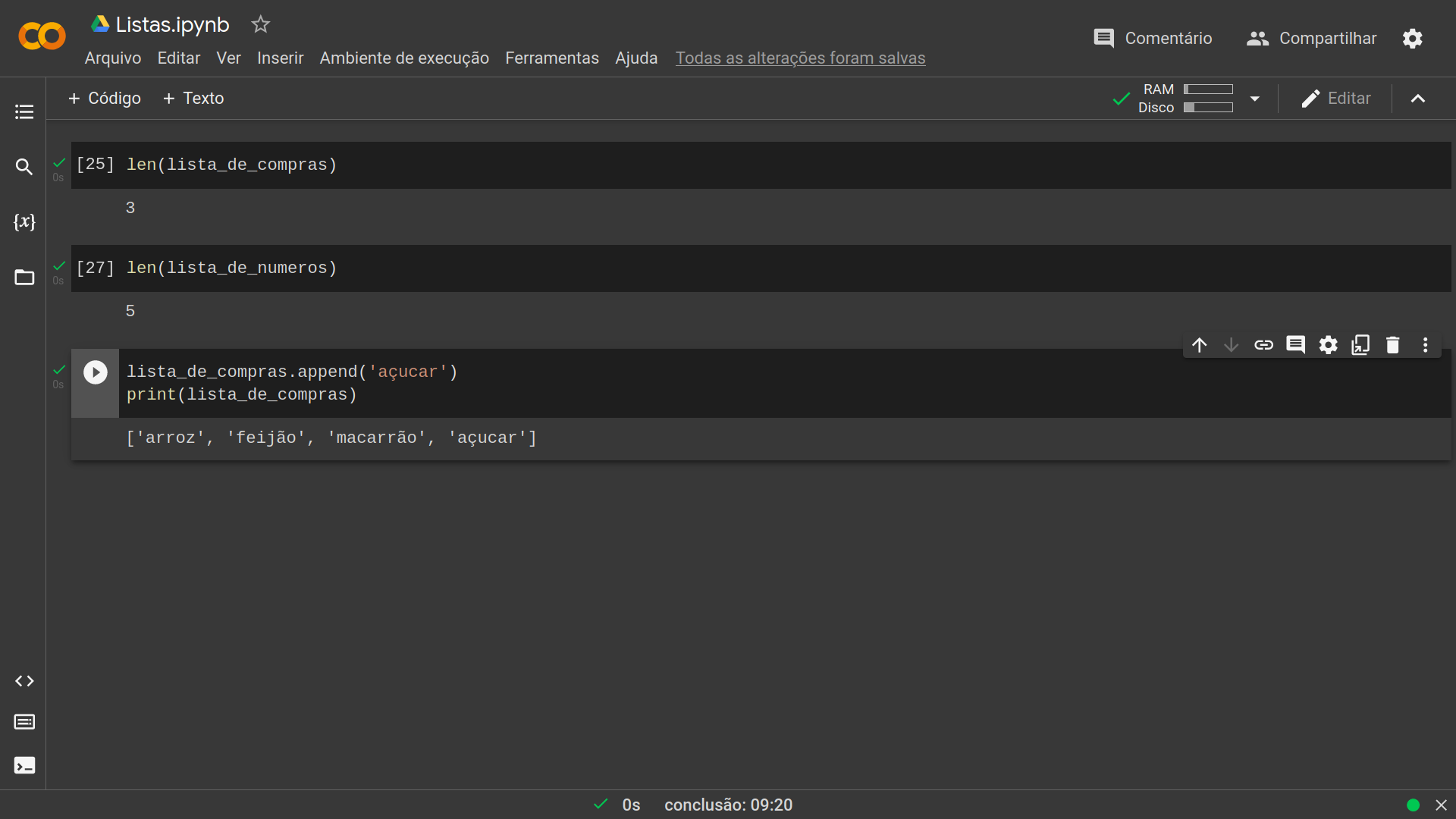
Podemos verificar o tamanho de uma lista com a função:

len(nome\_da\_lista):



Podemos adicionar um novo elemento na lista com a seguinte função:

lista.append(nome\_da\_lista):



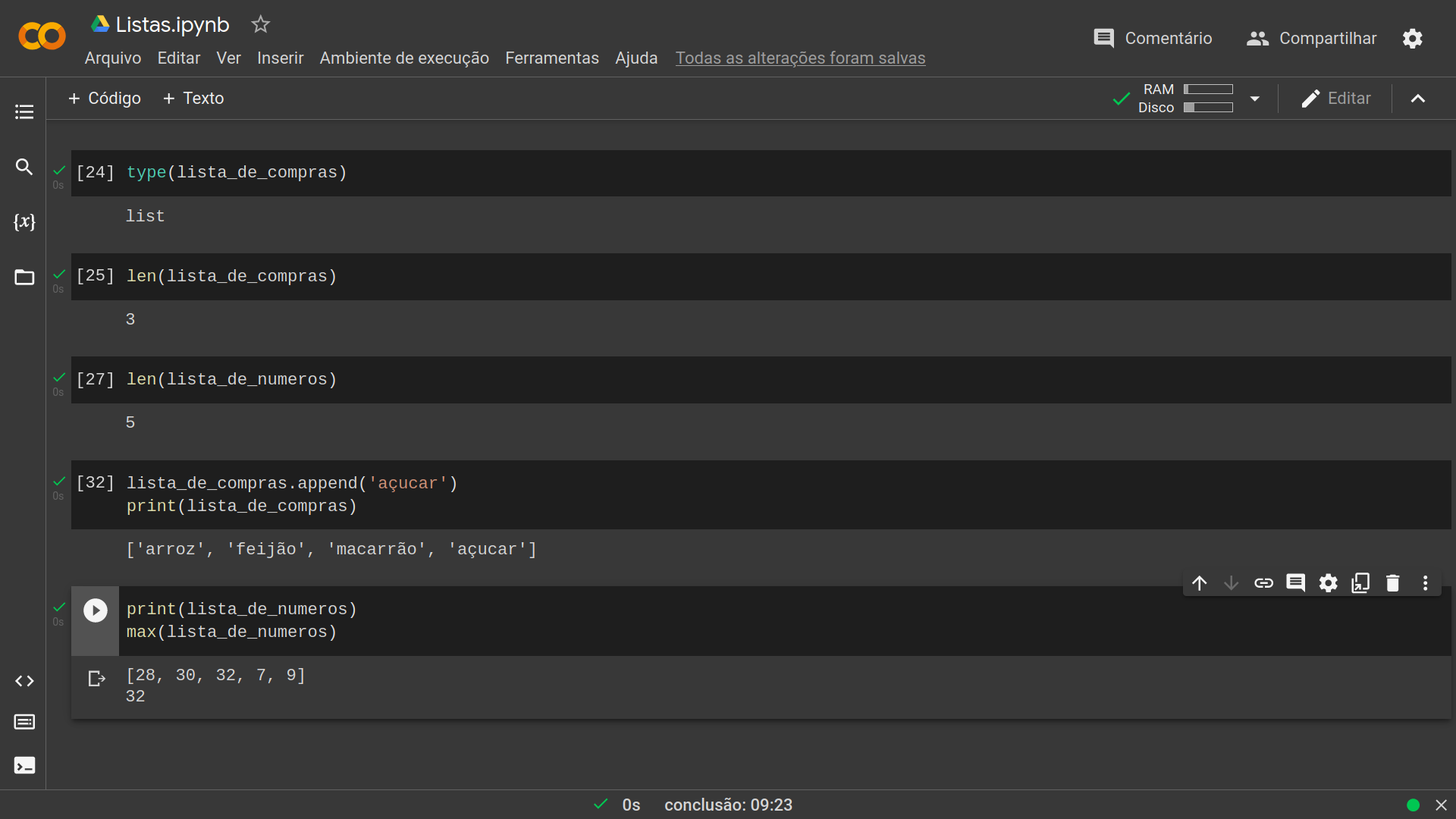
agora a nossa lista possui mais um item. Podemos notar que esse comando insere sempre um item ao final da lista.

Temos também os comandos:

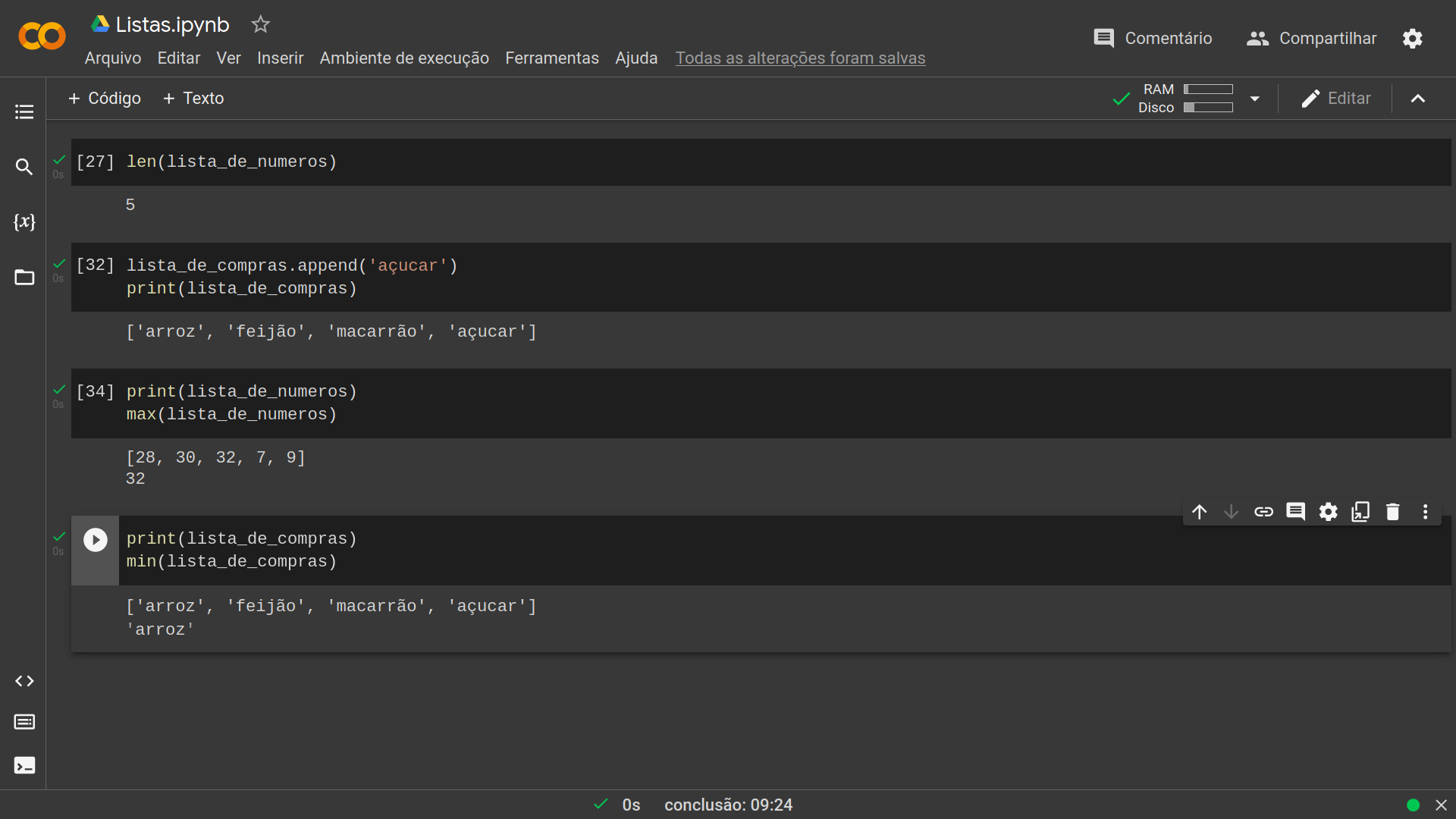
max(nome\_da\_lista)

min(nome\_da\_lista)

que nos retorna respectivamente o maior e o menor elemento da lista.



podemos ver que funciona para strings também, já que palavras podem ser organizadas em ordem alfabética:



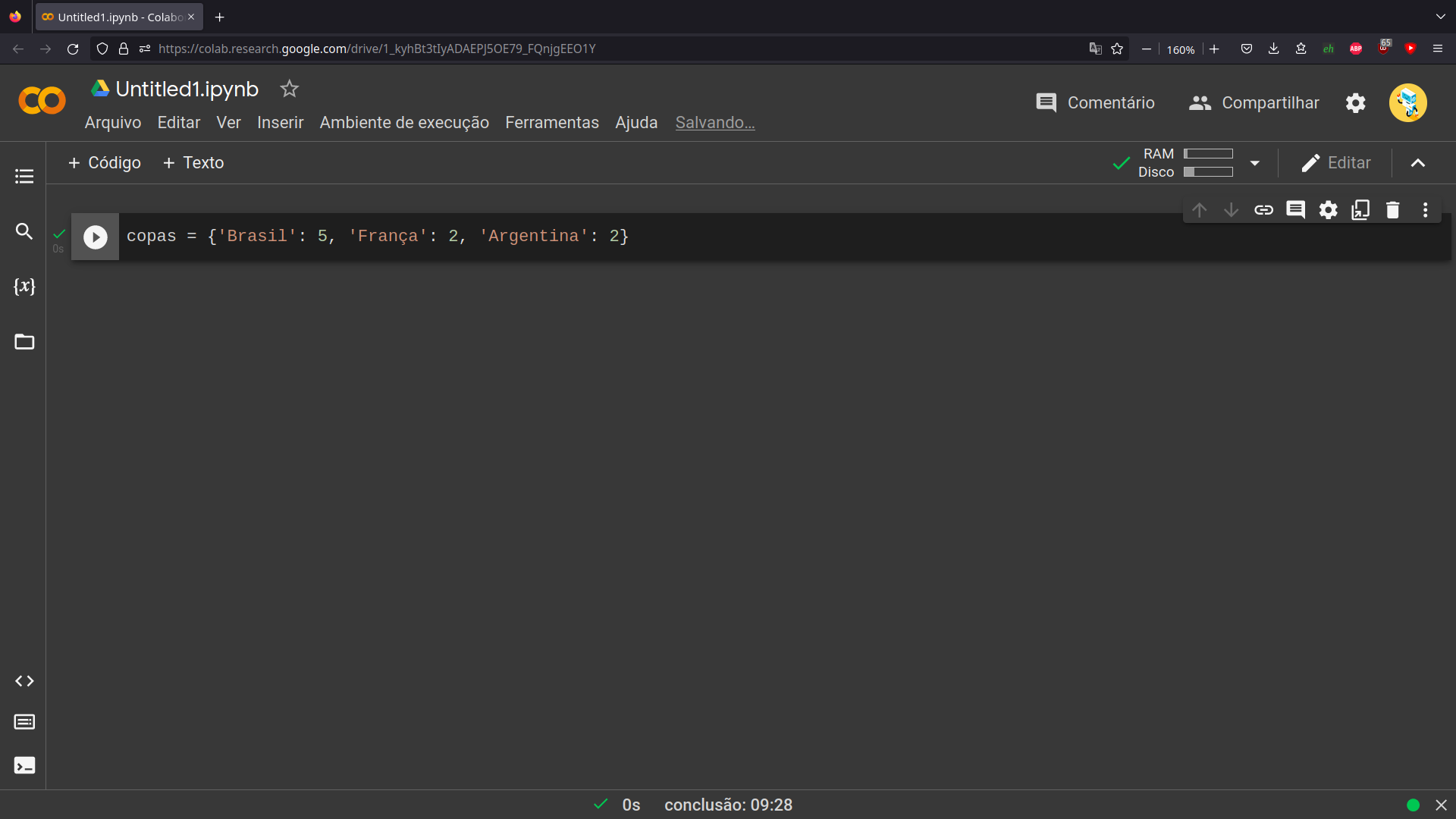
Uma outra estrutura de dados que vamos ter no Python é chamado de dicionário:

**Dicionário.**

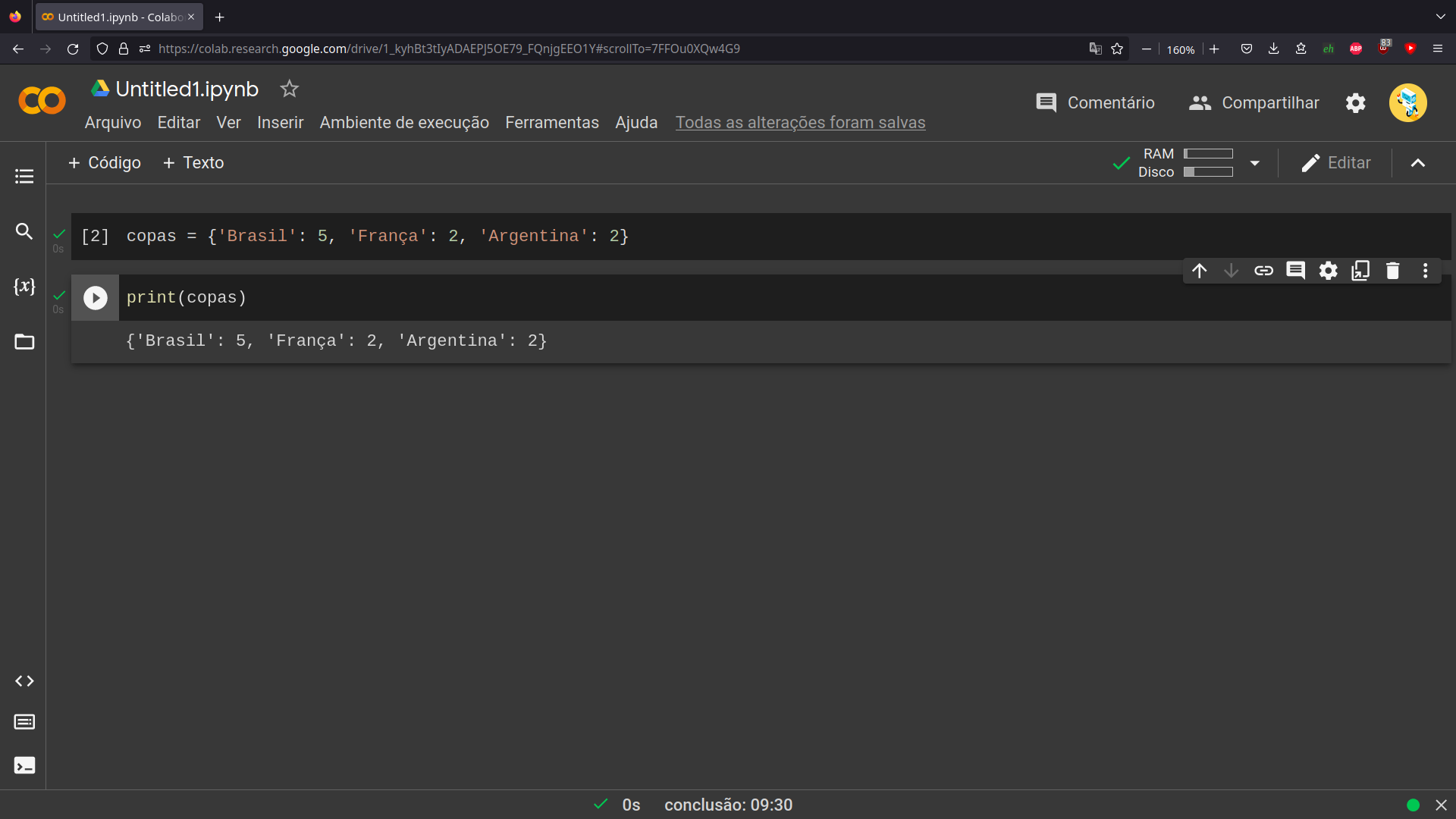
No python um dicionário é uma estrutura de dados que representa uma coleção de dados. Ou seja, é um objeto que possui diversos valores, a diferença entre um dicionário e uma lista é que para acessar um item de um dicionário, fazemos acesso através das chaves, e não do índice.

Vamos ver como funciona na prática:

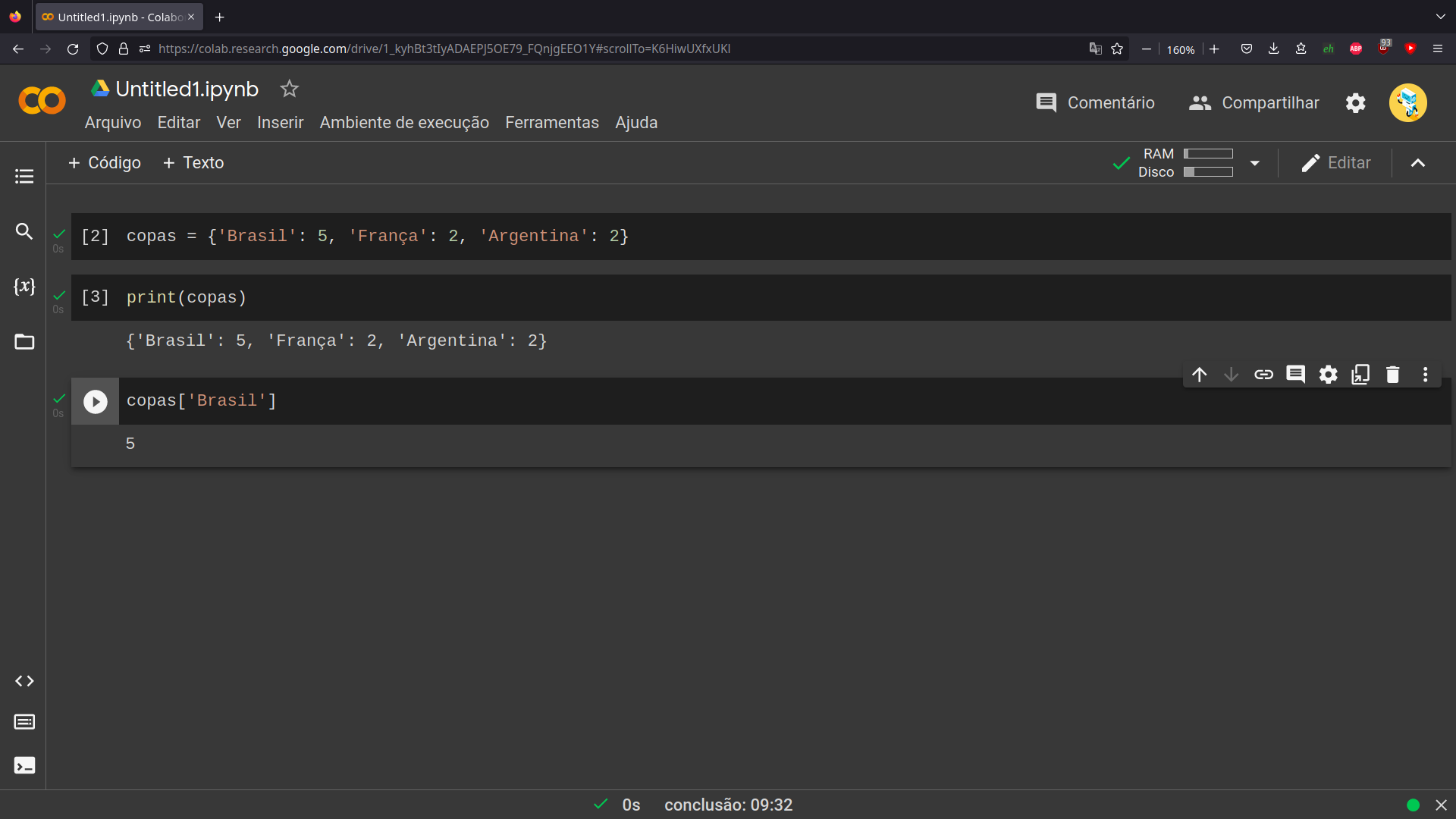
Para declarar um dicionário fazemos o uso de chaves ao invés de parênteses como nas listas.



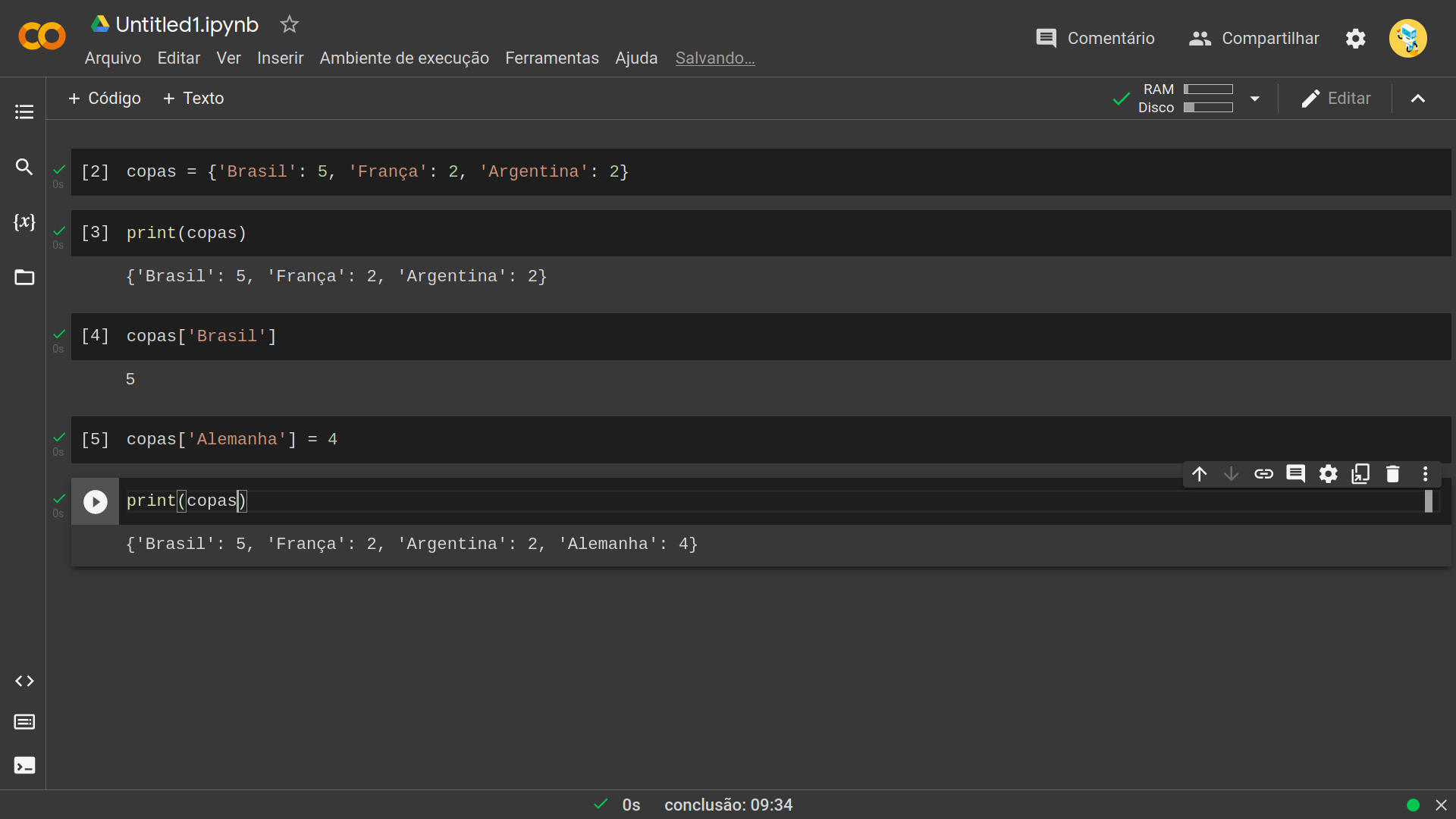
Aqui temos uma variável copas, que representa os países e a quantidade de títulos que possuem. O que está entre aspas representa as chaves, e o que segue após os dois pontos são os valores.



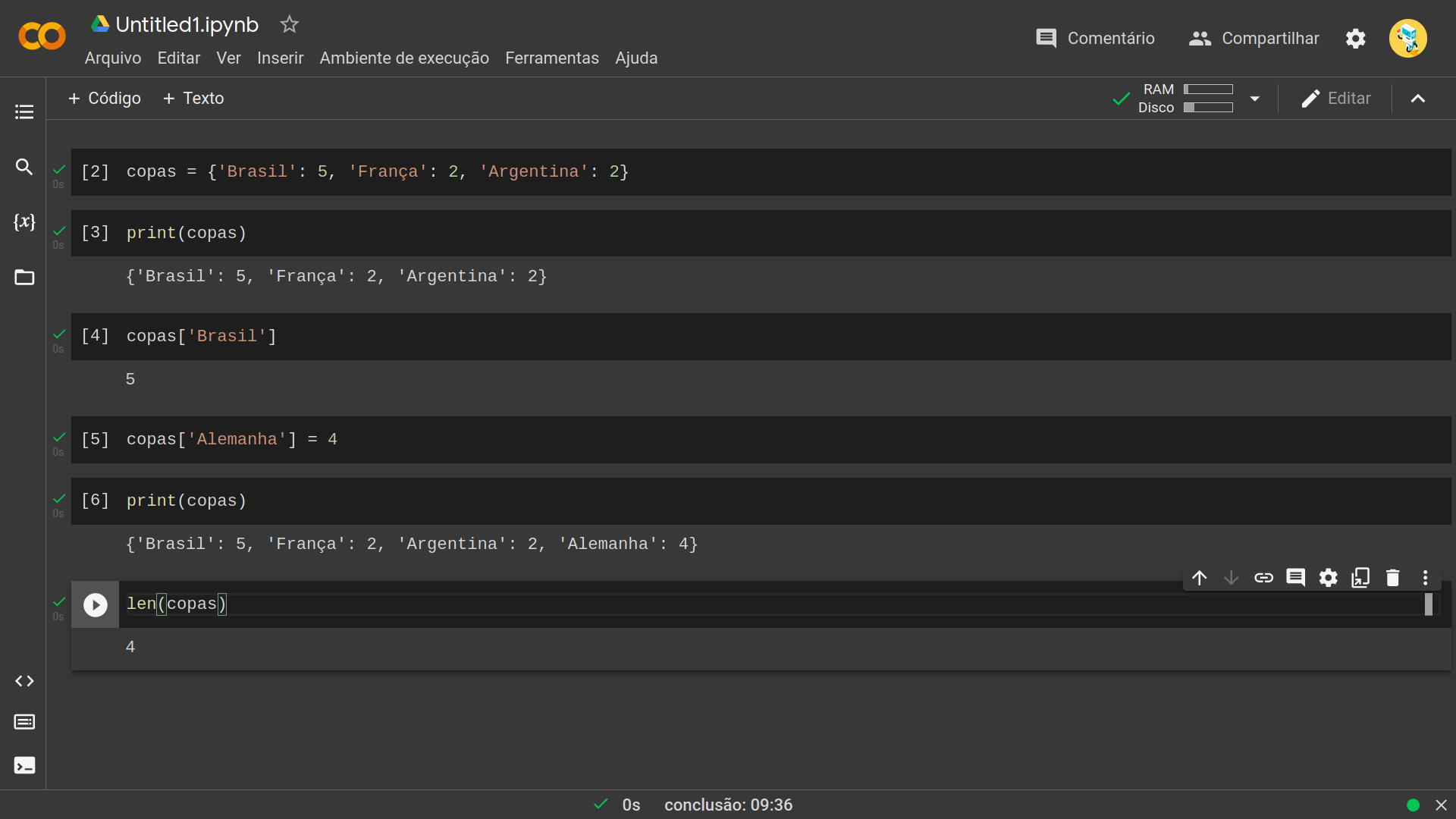
Para acessar algum valor utilizamos a chave:



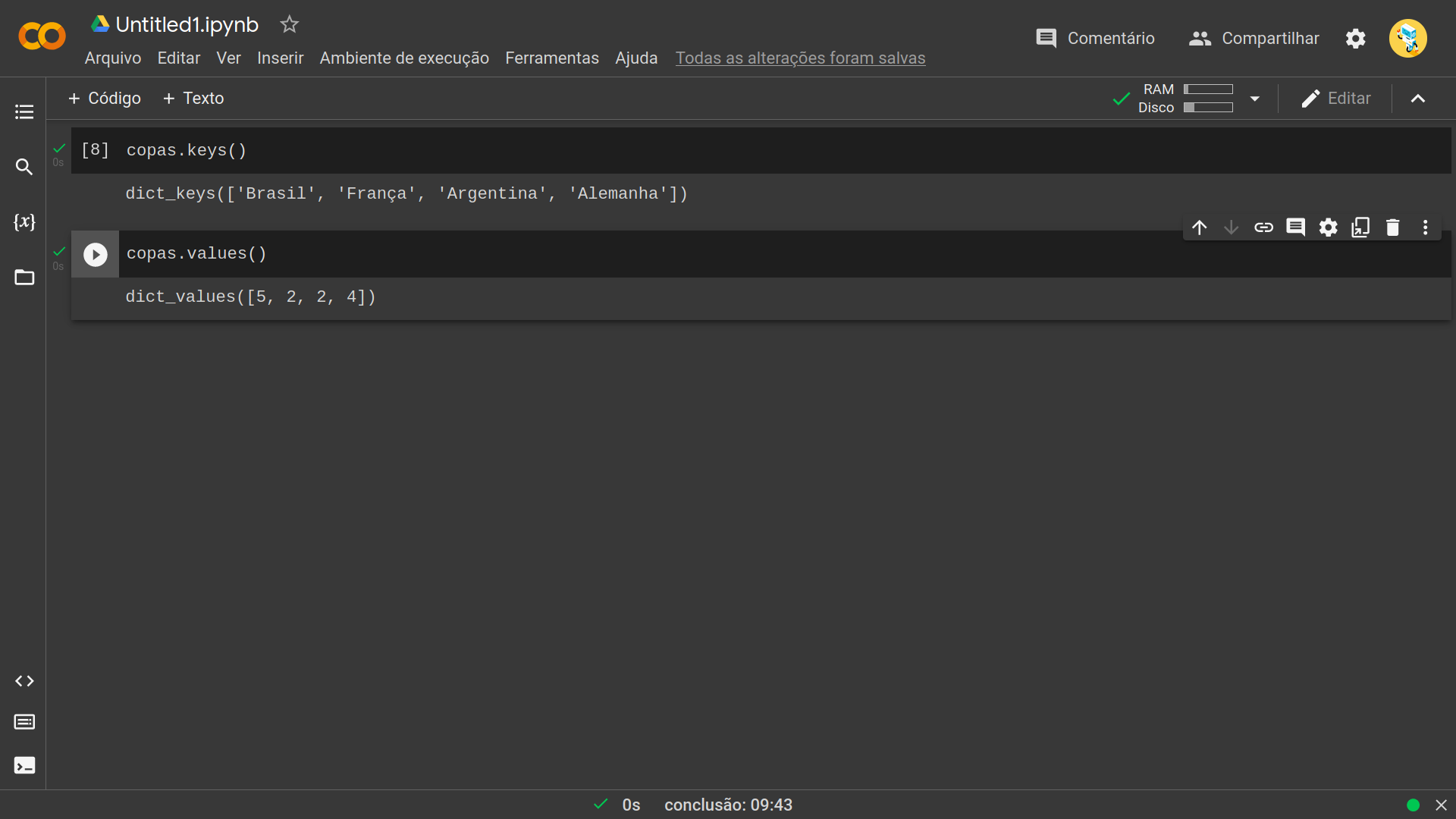
podemos também adicionar outros valores ao dicionário:



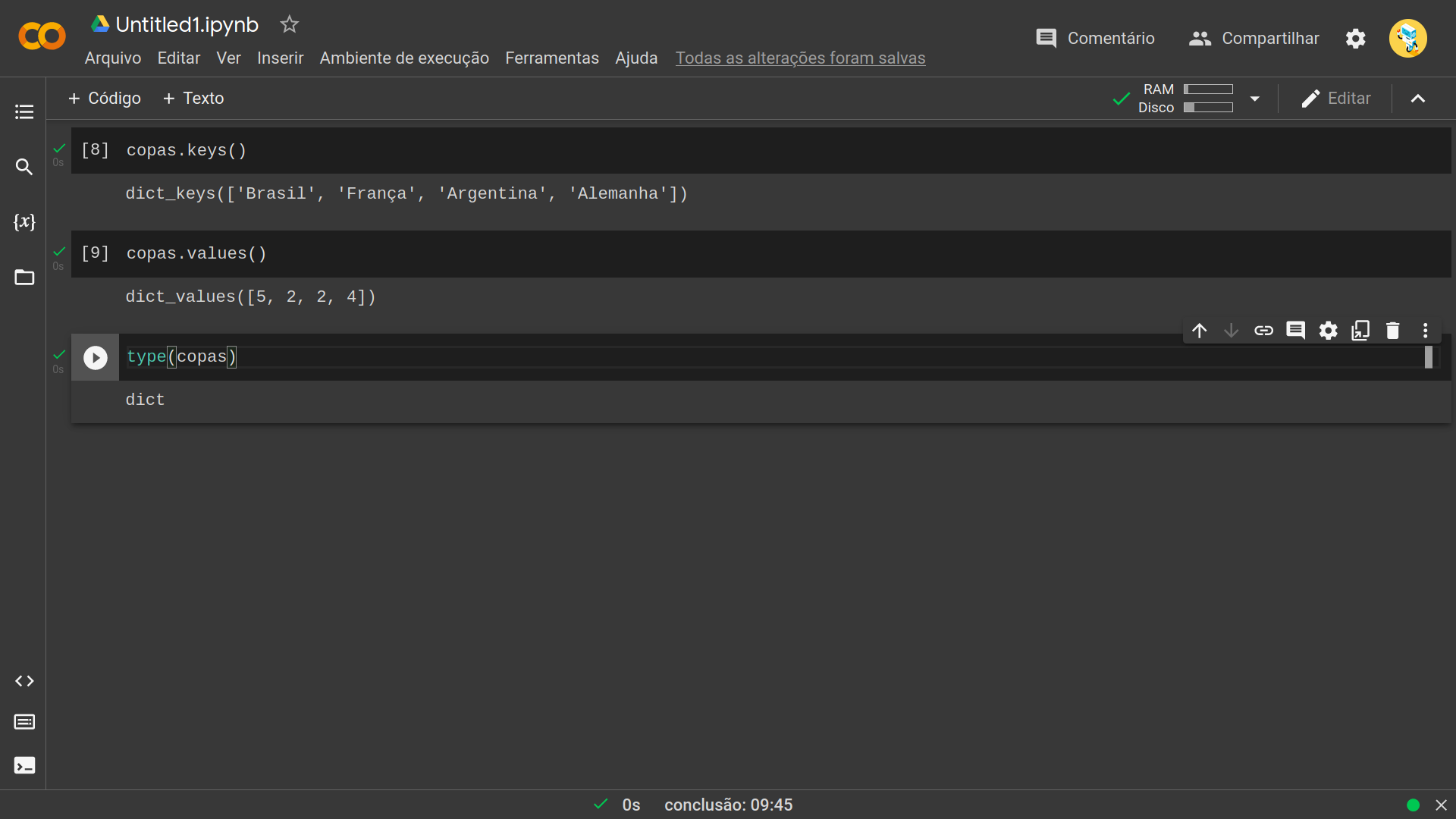
E com a mesma função que usamos para ver o tamanho de uma lista podemos verificar o tamanho de um dicionário:



Podemos usar também as funções keys() e value() para verificar respectivamente as chaves e os valores dos dicionários.



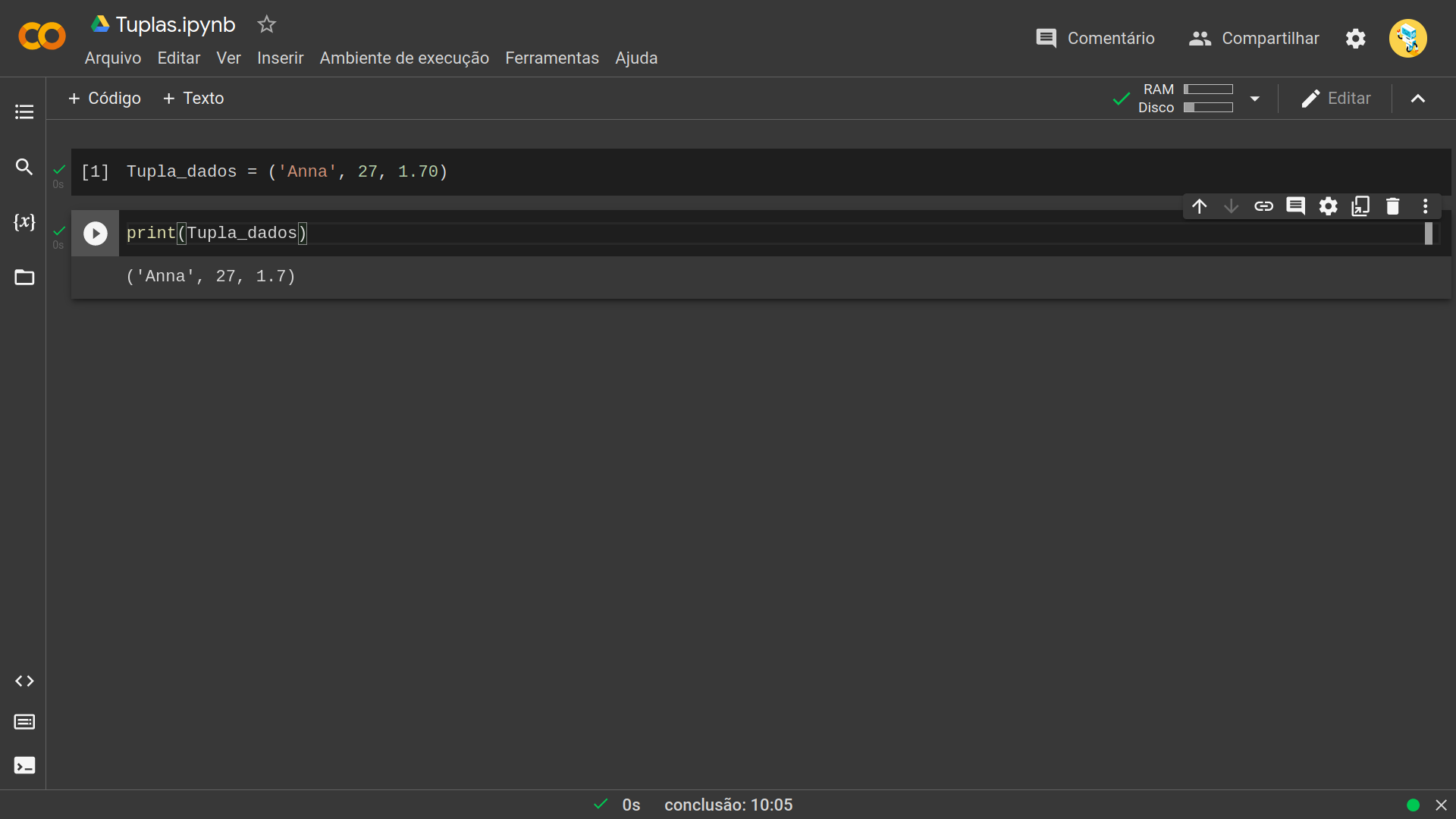
E além disso, se usarmos a função type() podemos verificar o tipo de variável dict, que é dicionário.



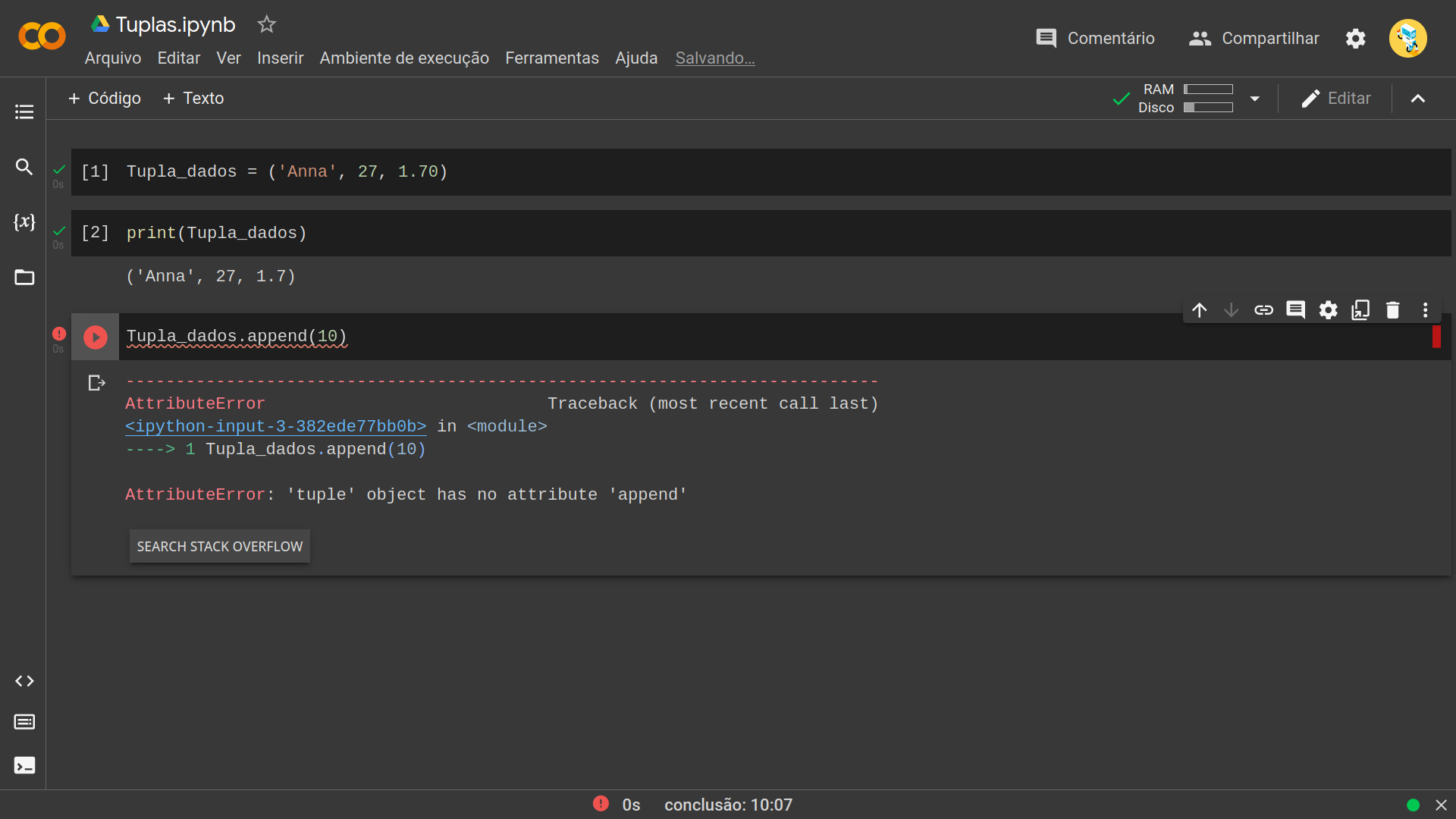
Outra estrutura de dados muito utilizada no python são as tuplas.

**Tuplas:**

Uma tupla no python é uma estrutura de dados que é parecida com uma lista, no entanto, a principal característica de uma tupla é ser imutável, ou seja, uma vez que declaramos uma tupla e seus valores, não podemos mais adicionar novos valores a esta, assim como nós fazemos na lista através da função append, e nem mesmo remover algum elemento. Esse tipo de variável é geralmente utilizado quando queremos garantir que os dados inseridos não poderão ser alterados posteriormente. É interessante dizer que por mais que uma tupla não possa ser alterada, alguns de seus elementos internos podem ser alterados, caso por exemplo um de seus elementos seja uma lista, esta lista pode ser alterada: Vamos ver um exemplo de como declarar uma tupla. Em lista usamos, [], em dicionários usamos {} e nas tuplas vamos usar ():

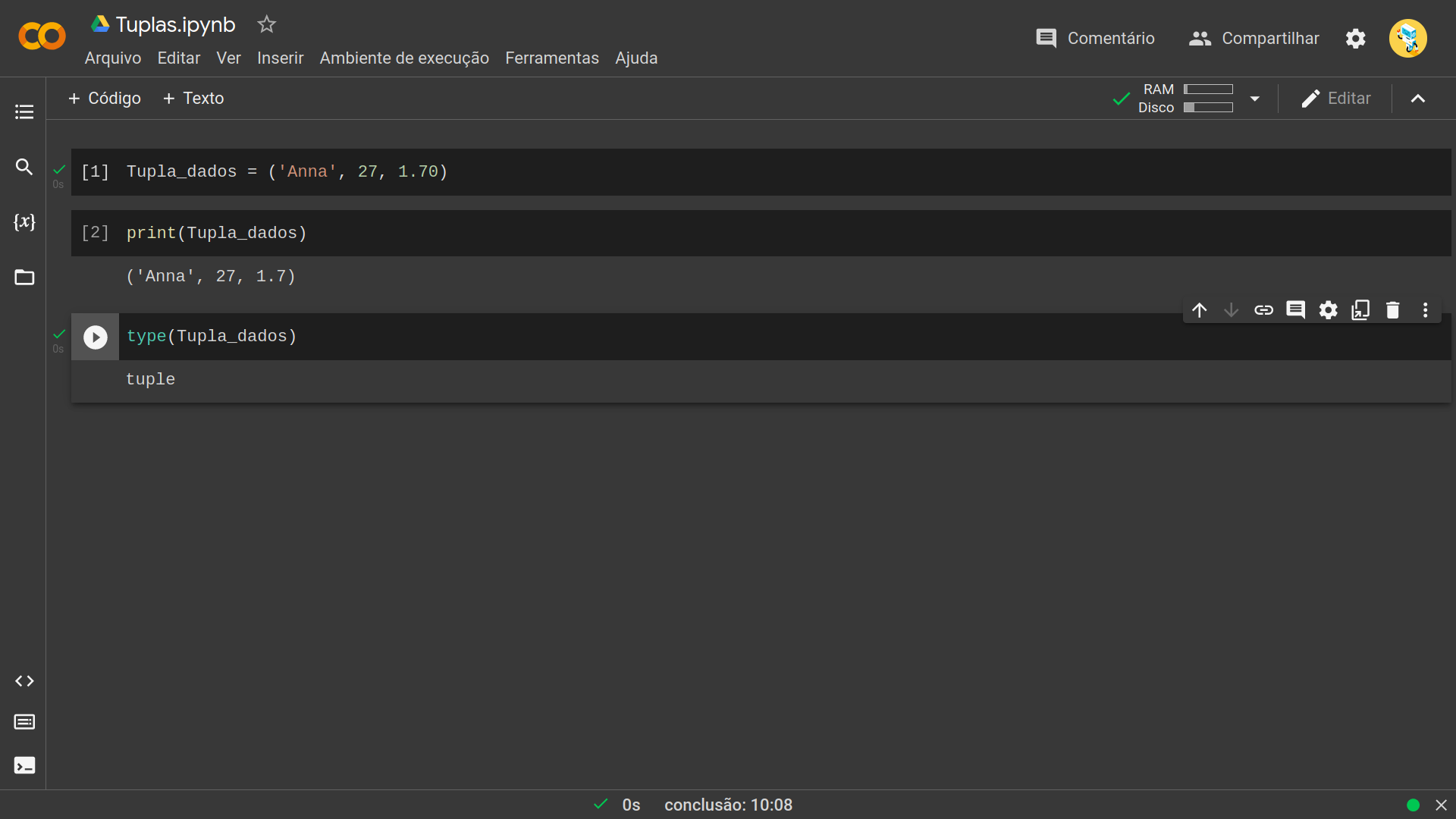


Mas se tentarmos adicionar algum valor na tupla como fazemos em listas:

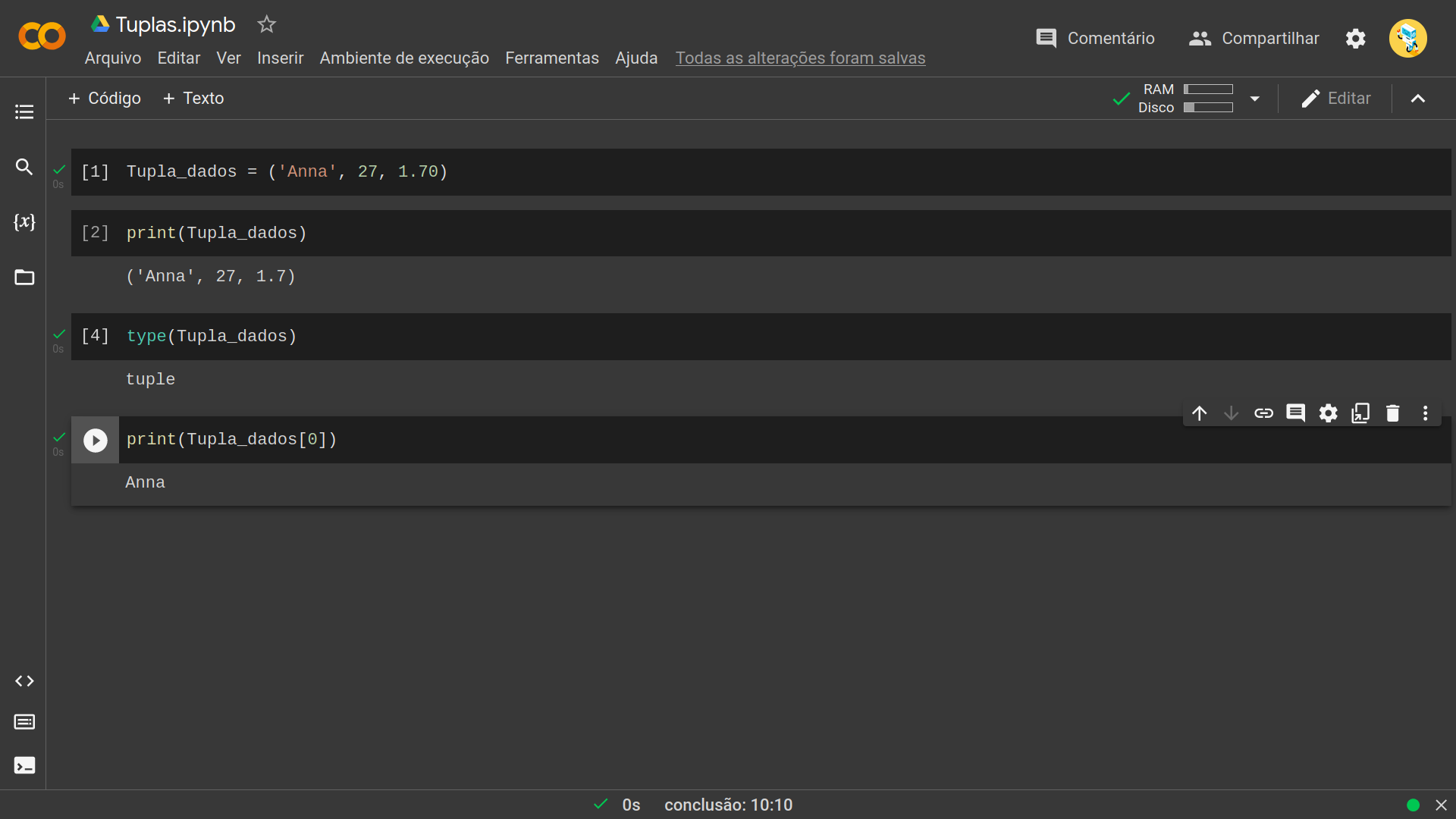


Vamos obter um erro dizendo que append não é um atributo.

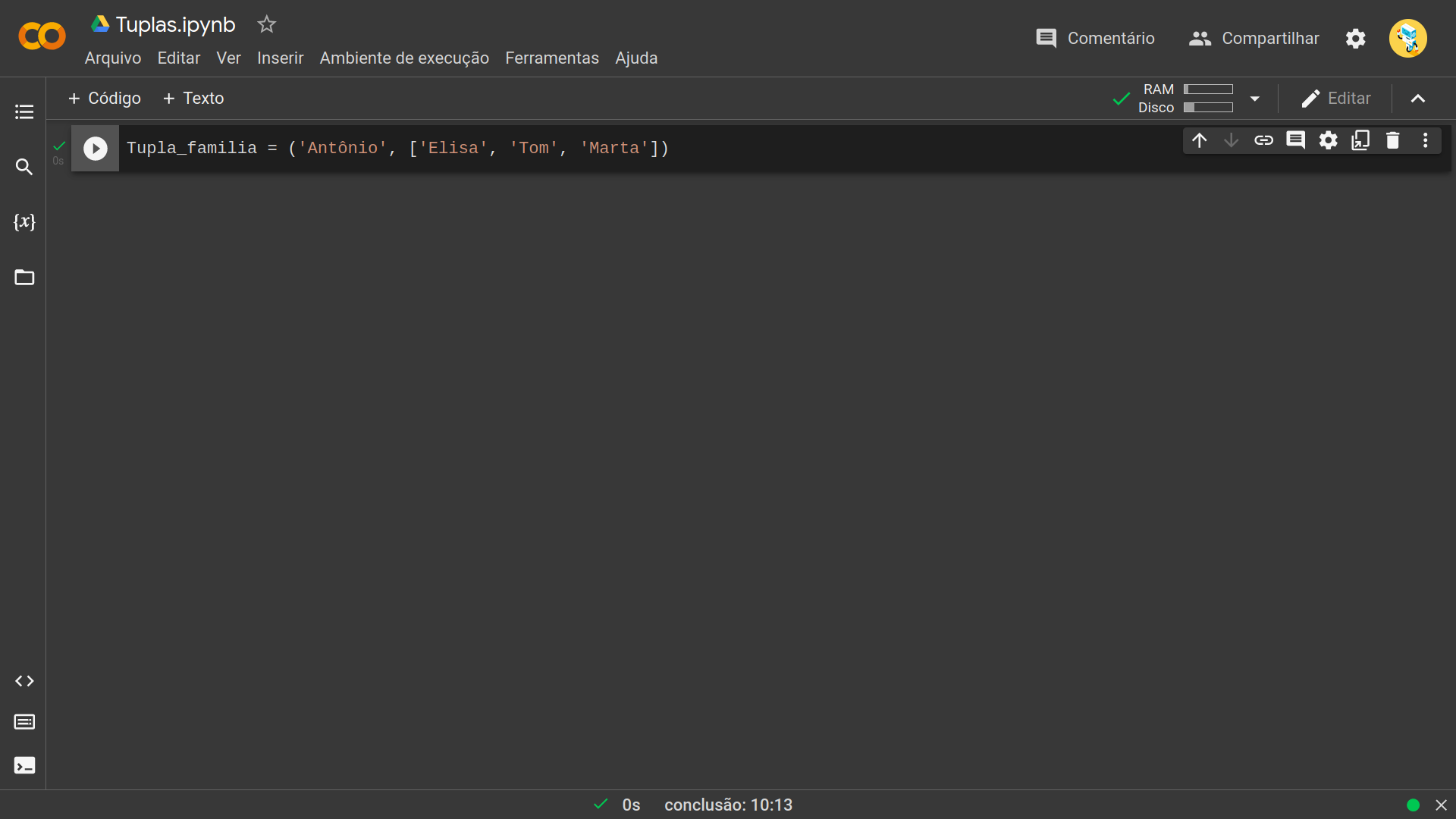
Podemos verificar o tipo da variável tupla com o type():



Para acessar uma posição de uma tupla fazemos da mesma maneira que fazemos em listas, através do índice:

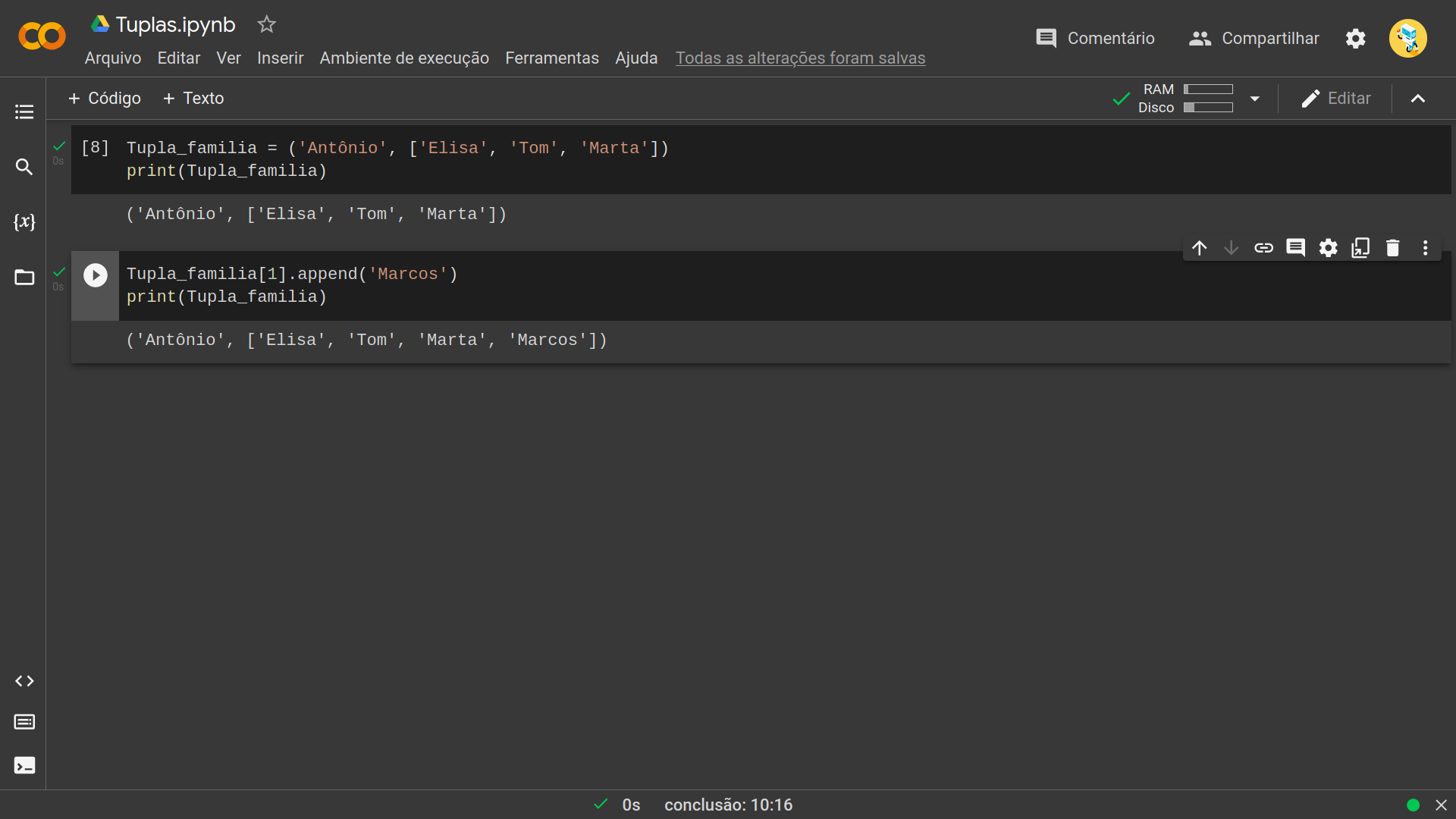


Mas como eu falei antes, dependendo do tipo de dado que uma tupla contém, se for uma lista, podemos alterar: Vamos criar uma tupla para guardar as informações de uma pessoa e os nomes de seu ou seus filhos:



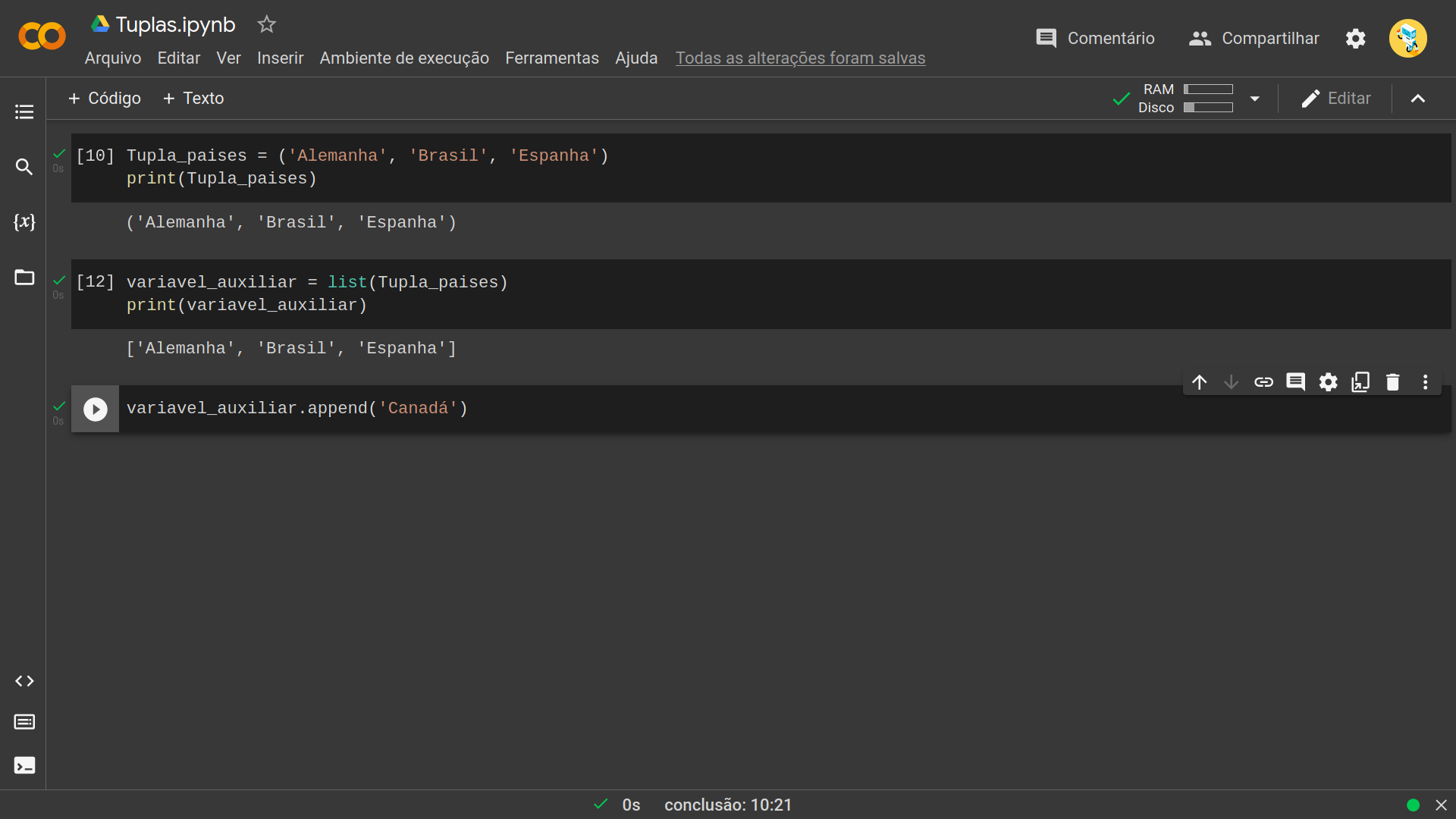
Antônio têm três filhos, mas vamos supor que ele tenha mais um, será que podemos adicionar este filho?

Vamos fazer o seguinte, vamos acessar a posição 1 da tupla, que representa uma lista com os nomes dos filhos de Antônio, e então vamos adicionar o Marcos como filho.

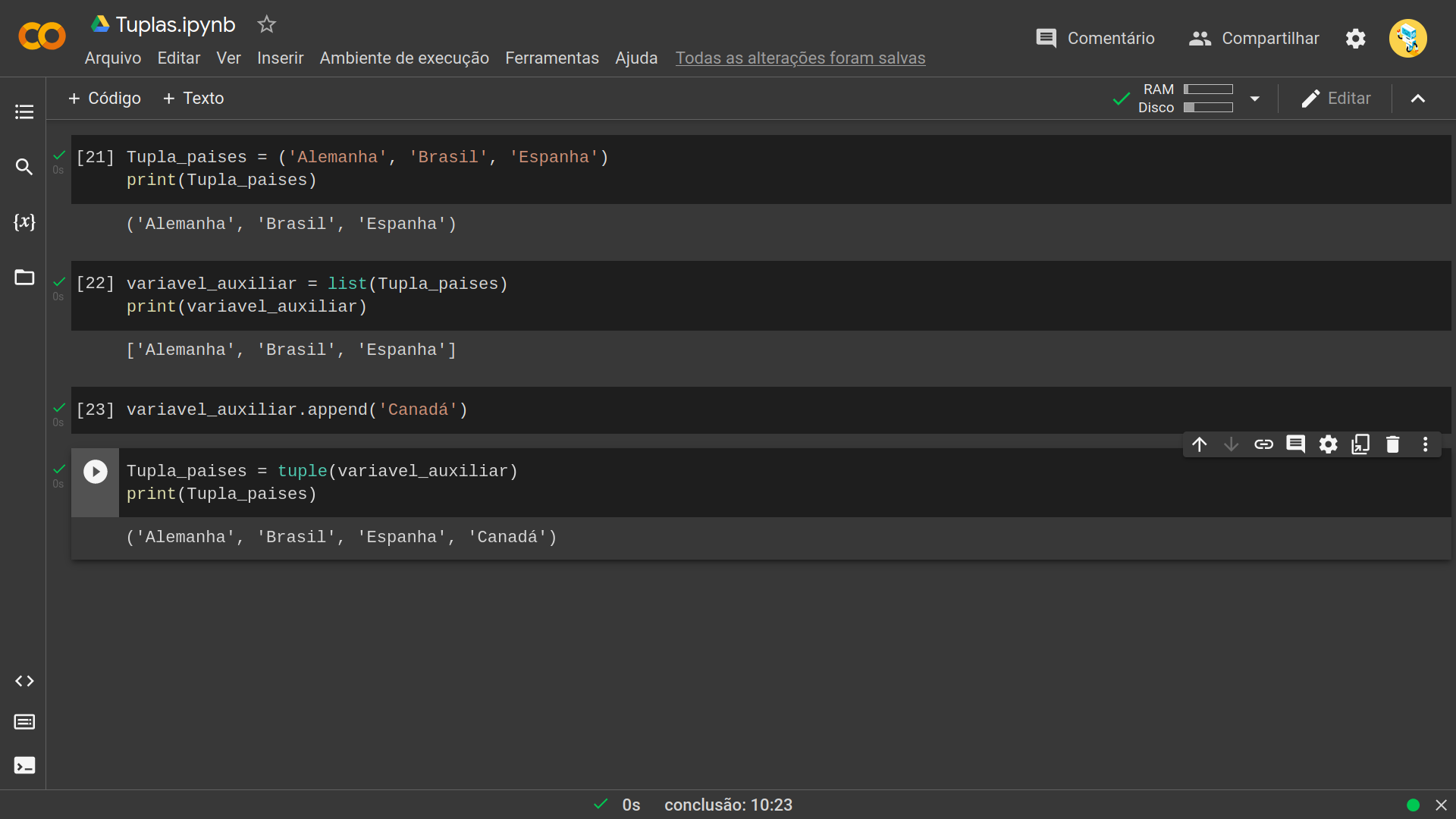


Na tupla, não conseguimos inserir novos elementos, mas em um elemento da tupla, que no caso é uma lista, podemos inserir.

Mas é possível alterar uma tupla, caso queiramos. Mas nesse caso, vamos precisar de uma variável auxiliar que seja uma lista, e através do comando list, vamos fazer essa lista receber os valores da tupla. Vejamos como fazer:



E agora podemos usar o comando tuple() para transformar uma lista numa tupla:



Dessa forma, conseguimos manipular as informações dentro de uma tupla.

Bom, por esta aula é isto, nas próximas aulas vamos começar a ver sobre estruturas condicionais, muito obrigado por assistir até aqui, até a próxima aula.