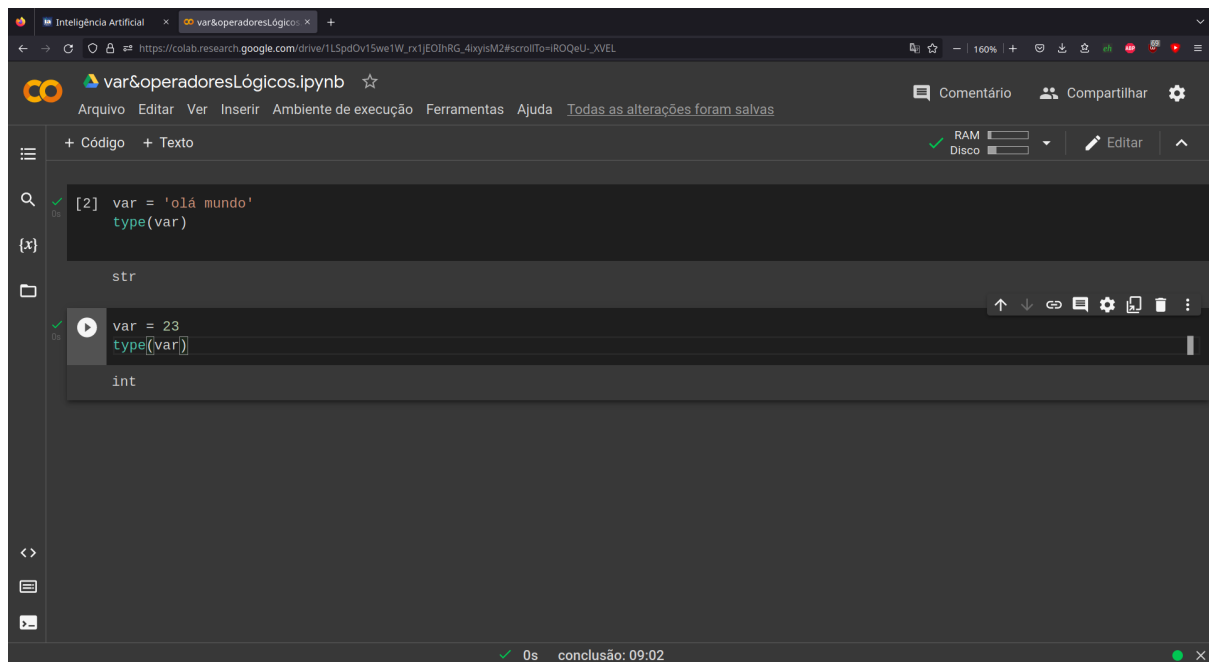


## Aula 28 - Operadores e variáveis em Python

Olá, sejam todos bem-vindos a mais uma aula de python, nessa aula vamos nos aprofundar um pouco mais nos conceitos de variáveis, e vamos também entender o que são operadores e para que vamos usá-los. Bem eu já falei um pouco sobre os tipos de variáveis no python na aula anterior. E disse também que, diferente de outras linguagens de programação, no python, não precisamos declarar o tipo de uma variável, além disso, por ser o python uma linguagem não tipada, uma variável pode mudar também o seu tipo. Vamos ver isso novamente na prática:

A screenshot of a Google Colab notebook titled 'var&operadoresLógicos.ipynb'. The notebook shows two code cells. The first cell contains the code: `[2] var = 'olá mundo'` followed by `type(var)` on the next line. The output of this cell is `str`. The second cell contains the code: `var = 23` followed by `type(var)` on the next line. The output of this cell is `int`. The interface includes a top bar with 'Inteligência Artificial' and 'var&operadoresLógicos' tabs, a URL bar, and various icons. The bottom status bar shows '0s' and 'conclusão: 09:02'.

Aqui criamos uma variável `var` que inicialmente recebeu um valor String, mas após isso redefinimos esta mesma variável com um valor inteiro. E como eu já falei antes, em algumas linguagens isso não é possível, sendo esta mais uma das facilidades do python.

Vale destacar alguns pontos, em qualquer linguagem de programação existem regras que devem ser seguidas, por exemplo, em C sempre ao final de uma linha devemos colocar um ponto e vírgula para encerrar a instrução. Não existe essa regra no python, mas existem outras como por exemplo na hora de nomear uma variável.

1. Nomes de variáveis não podem ser iniciadas por números
2. Não pode haver espaços entre os nome de uma variável.
  - a. Ou seja, se a gente quiser nomear uma variável como nome completo, devemos fazer: `nome_completo`
  - b. Note que para separar as palavras foi usados `_` não é uma regra usar isso, mas é considerado uma boa prática de programação usar para dividir as palavras de uma variável.
3. Uma variável não pode conter alguns símbolos especiais tais como: `' " , < > / | \ ( ) @ # $ % ^ & * - + !`
4. Em qualquer linguagem de programação existem palavras reservadas da linguagem, que são usadas para funções específicas dentro do python, e essas palavras também

não podem ser usadas para nomear uma variável

## Palavras reservadas

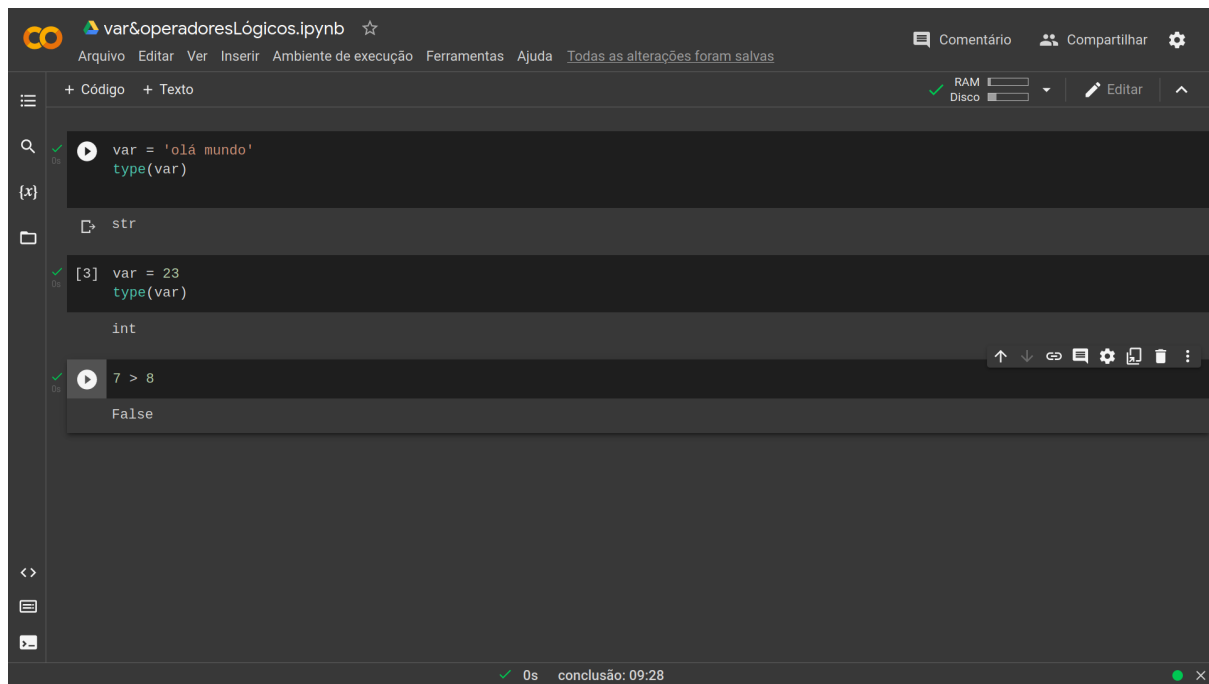
not	False	except	return	def
with	await	in	and	from
async	else	raise	continue	nonlocal
elif	import	True	for	while
if	pass	class	lambda	assert
or	None	finally	try	del
yield	break	is	as	global

Operadores:

Na aula anterior, vimos os operadores básicos que nos permitem efetuar cálculos matemáticos.

+	adição	>=	maior ou igual
-	subtração	<=	menor ou igual
/	divisão	>	maior
//	divisão inteira	<	menor
**	exponenciação	==	igual
*	multiplicação	!=	diferente

Aqui podemos ver, além dos operadores que matemáticos, alguns outros como < > ==  
Estes outros operadores são chamados de operadores relacionais, que nos permitem comparar valores, verificar se um valor é diferente de outro, maior ou menor vamos ver isso na prática:



A screenshot of a Jupyter Notebook interface. The title bar shows 'var&operadoresLógicos.ipynb'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', 'Ajuda', and 'Todas as alterações foram salvas'. The left sidebar has icons for file explorer, search, and code execution. The main area shows three code cells. The first cell contains `var = 'olá mundo'` and `type(var)`, with output `str`. The second cell contains `var = 23` and `type(var)`, with output `int`. The third cell contains `7 > 8`, with output `False`. The status bar at the bottom shows '0s' and 'conclusão: 09:28'.

```
var = 'olá mundo'
type(var)

str

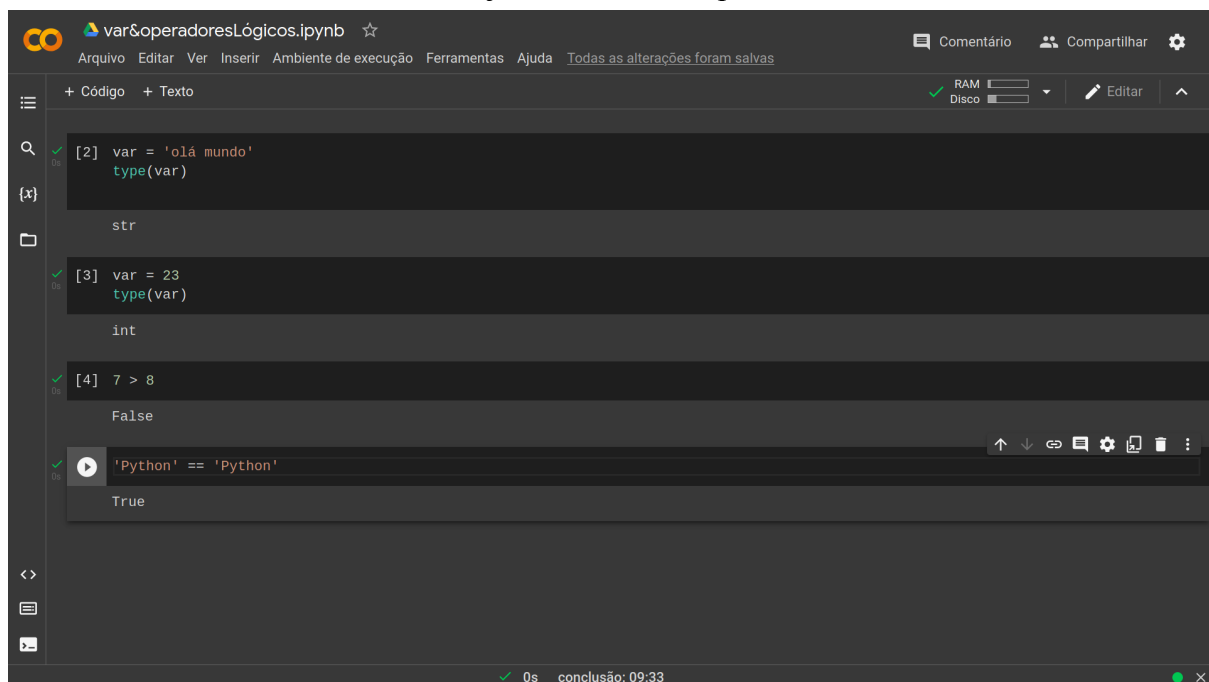
[3] var = 23
type(var)

int

7 > 8

False
```

Perceba que nesse código estamos basicamente perguntando se 7 é maior que 8, e o que aparece é uma mensagem: `False`, na verdade, esta mensagem também representa um tipo de variável, que é o tipo booleano. Uma variável do tipo booleano só pode assumir dois valores, verdadeiro ou falso: `True` ou `False`, vejamos outro exemplo:



A screenshot of a Jupyter Notebook interface. The title bar shows 'var&operadoresLógicos.ipynb'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', 'Ajuda', and 'Todas as alterações foram salvas'. The left sidebar has icons for file explorer, search, and code execution. The main area shows four code cells. The first cell contains `var = 'olá mundo'` and `type(var)`, with output `str`. The second cell contains `var = 23` and `type(var)`, with output `int`. The third cell contains `7 > 8`, with output `False`. The fourth cell contains `'Python' == 'Python'`, with output `True`. The status bar at the bottom shows '0s' and 'conclusão: 09:33'.

```
[2] var = 'olá mundo'
type(var)

str

[3] var = 23
type(var)

int

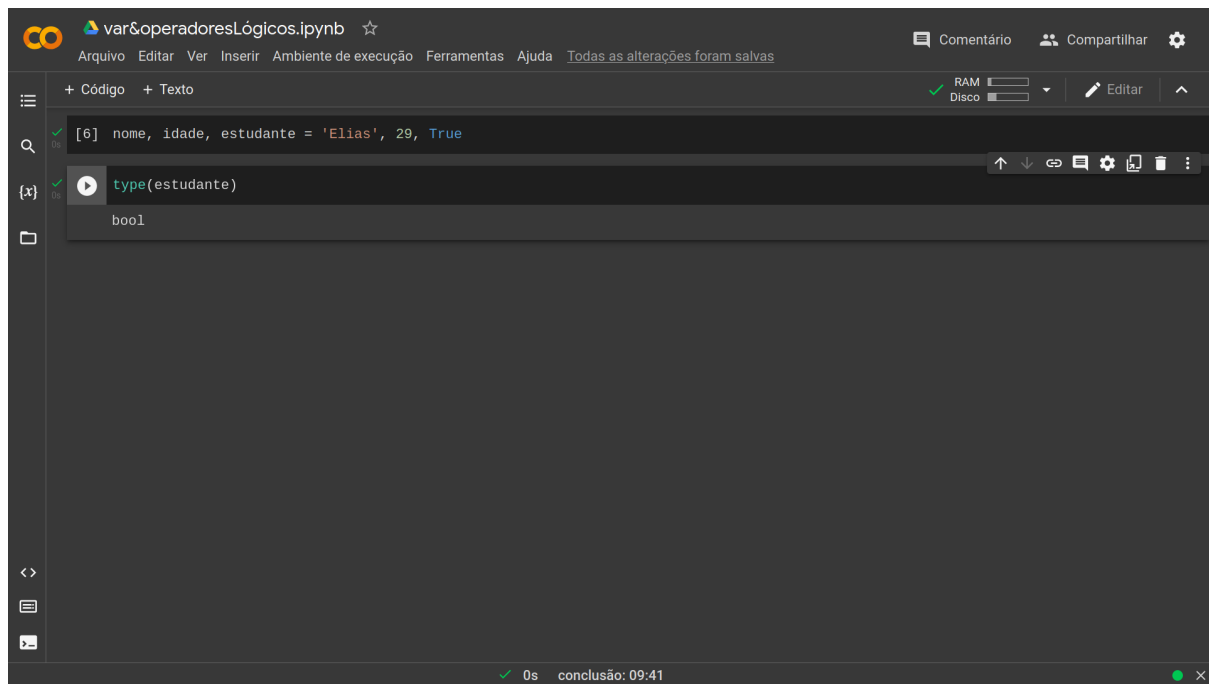
[4] 7 > 8

False

'Python' == 'Python'

True
```

Agora, usando os conhecimentos aprendidos, vamos tentar fazer um programa mais elaborado.



A screenshot of a Jupyter Notebook interface. The title bar shows 'var&operadoresLógicos.ipynb'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', 'Ajuda', and 'Todas as alterações foram salvas'. The toolbar shows 'RAM', 'Disco', and an 'Editar' button. The code cell contains the following Python code:

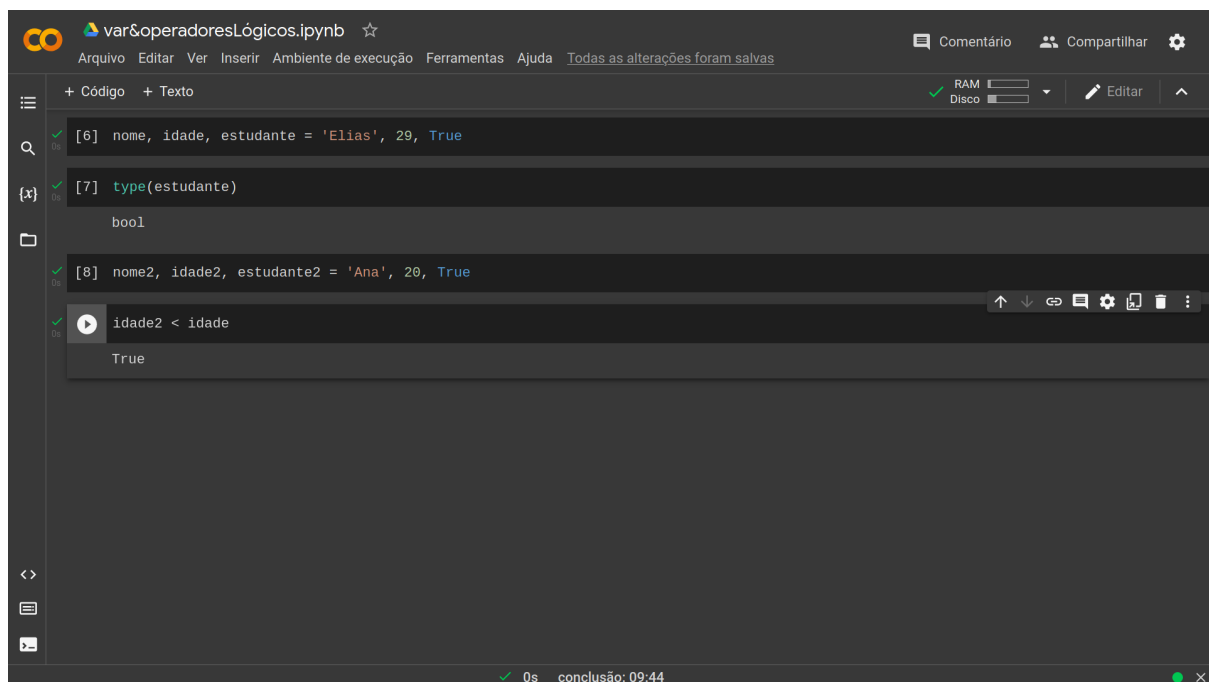
```
[6] nome, idade, estudante = 'Elias', 29, True
```

```
{x} type(estudante)
```

```
bool
```

The status bar at the bottom shows '0s' and 'conclusão: 09:41'.

Aqui definimos três variáveis e uma delas, recebe um valor booleano, que é se é estudante ou não;  
vamos criar outras variáveis:



A screenshot of a Jupyter Notebook interface. The title bar shows 'var&operadoresLógicos.ipynb'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', 'Ajuda', and 'Todas as alterações foram salvas'. The toolbar shows 'RAM', 'Disco', and an 'Editar' button. The code cell contains the following Python code:

```
[6] nome, idade, estudante = 'Elias', 29, True
```

```
[7] type(estudante)
```

```
bool
```

```
[8] nome2, idade2, estudante2 = 'Ana', 20, True
```

```
idade2 < idade
```

```
True
```

The status bar at the bottom shows '0s' and 'conclusão: 09:44'.

Aqui criamos outras variáveis para receber uma outra pessoa, e perguntamos se a idade desta pessoa é menor que a idade da pessoa anterior. Perceba que o Python nos retorna o valor True, ou seja verdadeiro.  
Mas por essa aula é isso, mas uma vez, obrigado por assistir até aqui e até a próxima aula.