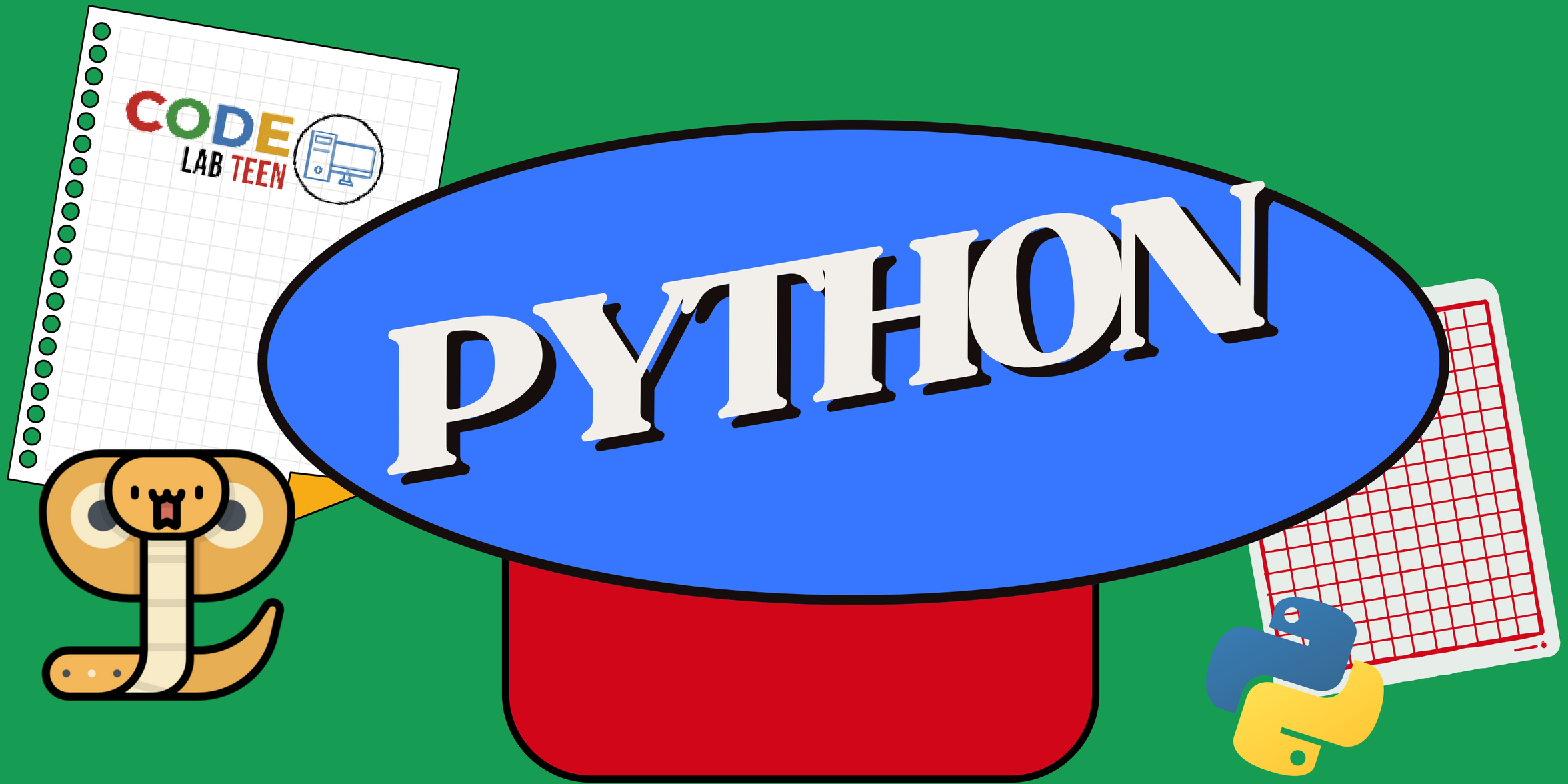

CODE
LAB TEEN
PYTHON

BEM - VINDOS!

AGENDA

Hoje, iremos relembrar alguns conceitos de aulas passadas:

- Revisão de Programação

E depois iremos apresentar uma nova linguagem de programação o python

REVISÃO DE PROGRAMAÇÃO

Variáveis

Váriáveis são armazenamentos de informações. Elas guardam algo para ser usado em algum código que vem em sequência

```
VARIAVEL1 = 14
VARIAVEL2 = "OLÁ"
VARIAVEL3 = TRUE
```

1 – Não podem iniciar com números
2 – Não podem ter caractéres especiais (ex: ´, @,#...)
3 – Não podem ter espaços
4 – Possuem tipos

CODE LAB TEEN

REVISÃO DE PROGRAMAÇÃO

Entrada e Saída

Também chamado de **input** e **output**, é a forma que dados são **inseridos** em um programa e como suas **saídas** são visisualizadas

ENTRADA (INPUT)

```
VARIAVEL =
input('insira aqui')
```

SAÍDA (OUTPUT)

```
print("O usuário
inseriu", VARIAVEL)
>>O usuário inseriu 2
```

1 - O programa fica bloqueado enquanto a entrada não for inserida
2 - print() é a função de impressão, para imprimir variáveis colocamos ela após aspas e a vírgula "",

CODE LAB TEEN

REVISÃO DE PROGRAMAÇÃO

Condicionais

Fazem com que o código seja executado mediante uma condição, um exemplo prático seria “se for chover hoje, então leve guarda-chuva, se não, deixe em casa”

SE

```
if (chover == True):
    guarda_chuva = True
```

SE NÃO

```
else:
    guarda_chuva = False
```

1 – Podem ser aninhados (repetidos) um dentro do outro para criar condições bem especificas
2 – Podem existir mais de uma condição dentro de um if
3 – Nem todo if precisa de um else

CODE LAB TEEN

REVISÃO DE PROGRAMAÇÃO

Laços

Garantem **repetição** de um código até uma **condição** ser satisfeita, um exemplo prático seria, “**enquanto** não acaba a aula, preste atenção”

FOR

```
for i in range(0,2):
    print(i * 2)
```

WHILE

```
while(i != 1)
    i = 0
    print(i)
    i += 1
```

O que será imprimido nos dois casos e quatas vezes o laço é repetido?

CODE LAB TEEN

INTRODUÇÃO

O QUE É PYTHON?

Python é uma linguagem de programação versátil e fácil de aprender, usada para desenvolver uma ampla variedade de aplicativos, incluindo sites, jogos, automação de tarefas, análise de dados, inteligência artificial e muito mais.

CODE LAB TEEN

CODE LAB TEEN

Vamos começar acessando o compilador onde vamos aprender python, clique no link

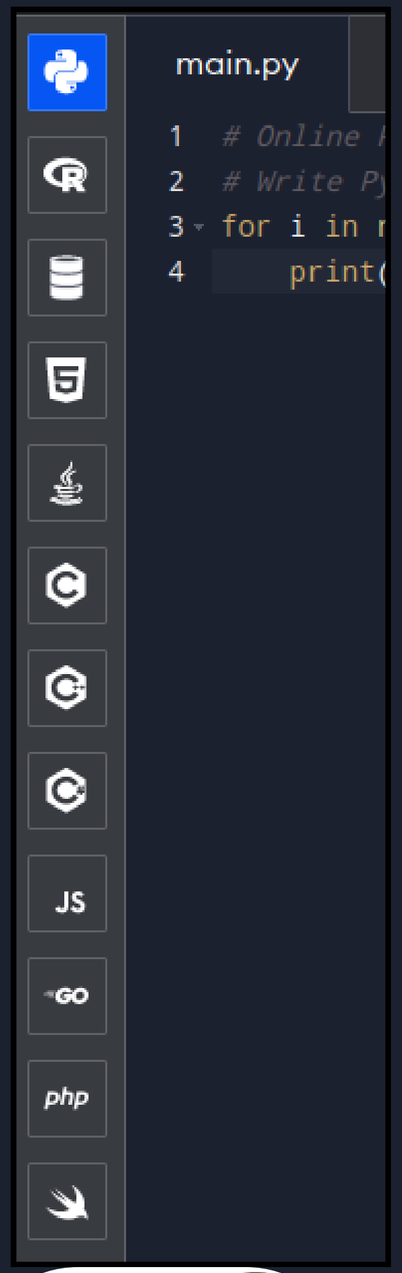
START

main.py

Save Run Output Clear

```
1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 print("Try programiz.pro")
```

CODE LAB TEEN



Essa parte no canto é onde escolhemos a linguagem que vamos programar, então tenha certeza que python esteja selecionado, assim como na imagem

CODE LAB TEEN

main.py

Save Run

```
1  # Online Python compiler (interpreter) to run Python online.
2  # Write Python 3 code in this online editor and run it.
3  print("Try programiz.pro")
```

Nesta parte aqui, vamos usar para escrever o nosso código

Já temos um código ali, que imprime a frase que está entre aspas

Vamos mudar e escrever, “Oii Codelab!”

CODE LAB TEEN

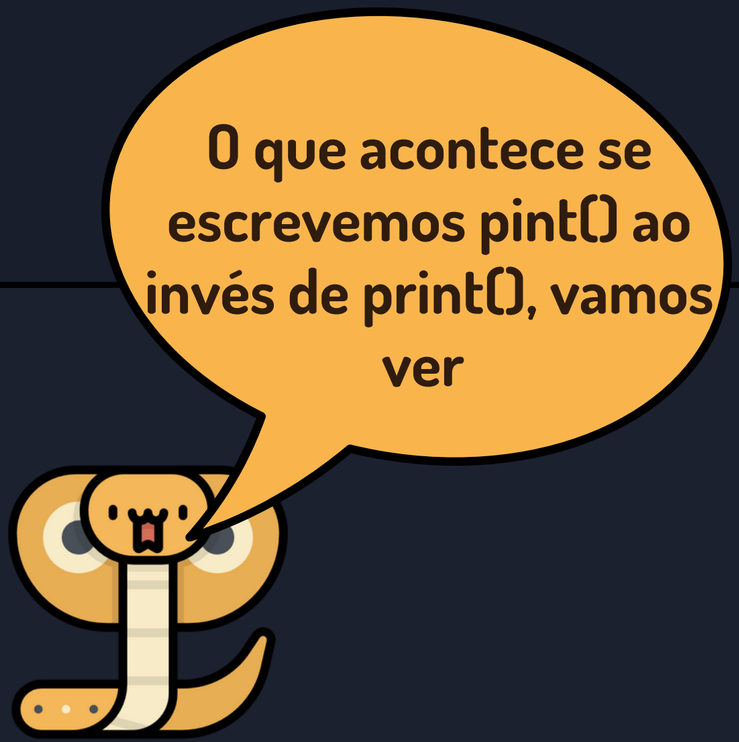
```
Output                                                    Clear
Oii Codelab!

=== Code Execution Successful ===
```

Essa parte é o output, aqui vai sair o resultado do nosso código

podemos ver que foi impressa nossa mensagem

Aparece também:
“=== Code Execution Successful ===”
Isso significa que ocorreu tudo certo

O botão “Clear” limpa o texto do output, aperte ele para ver



CODE LAB TEEN

main.py

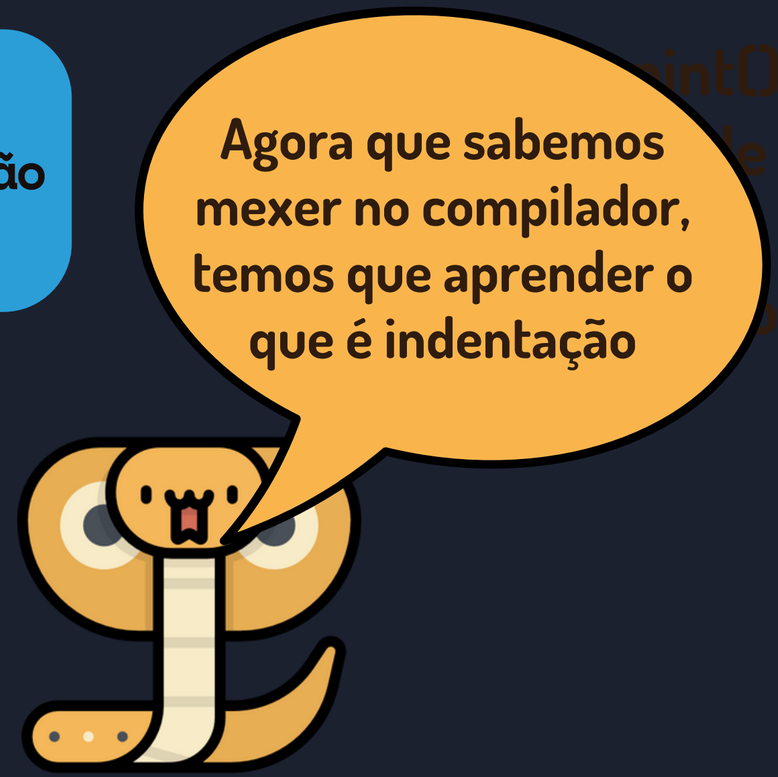
Save Run

```
1 pint("Oii Codelab!")
2
3
```

Output Clear

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
NameError: name 'pint' is not defined

=== Code Exited With Errors ===
```

Se escrevemos algum comando errado, no nosso terminal será imprimido um erro sintático, pois aquele comando não está definido no Python



CODE LAB TEEN

Indentação

No Python, para separar um bloco de código que deve ser executado em conjunto de outro usamos espaços em branco no início de cada linha, esse espaço é chamado de indentação

Exemplo, se temos uma condição if/else, e queremos que algo seja executado dentro de cada uma das condições, temos que indentar o bloco de código, o que significa que você o move um pouco para a direita.

```
1  x = 2
2  if x == 2:
3  print("x eh 2")
4  else:
5  print("x não eh 2")
6
```

Sem Indentação

```
1  x = 2
2  if x == 2:
3      print("x eh 2")
4  else:
5      print("x não eh 2")
6
```

Com Indentação

Indentação

Em python, diferente de outras linguagens, o uso da indentação faz parte da sintaxe da linguagem e é usada para delimitar blocos de código

```
1  doce = 5
2  for i in range(4):
3      print("comi um doce")
4  doce = doce - 1
5  print(f'eu tenho {doce} doce')
6
```

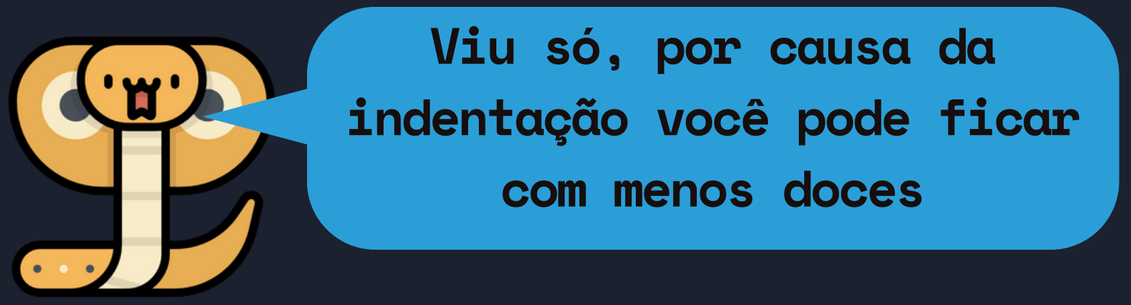
```
1  doce = 5
2  for i in range(4):
3      print("comi um doce")
4      doce = doce - 1
5  print(f'eu tenho {doce} doce')
6
```

Apesar dos códigos parecerem iguais, eles tem resultados bem diferentes

Indentação

```
Output
comi um doce
comi um doce
comi um doce
comi um doce
eu tenho 4 doce

=== Code Execution Successful ===
```

```
Output
comi um doce
comi um doce
comi um doce
comi um doce
eu tenho 1 doce

=== Code Execution Successful ===
```



Operações

Soma:

```
1  num1 = 10
2  num2 = 5
3  soma = num1 + num2
4  print("Adição:", soma)
5
```

Subtração:

```
1  num1 = 10
2  num2 = 5
3  subtracao = num1 - num2
4  print("Subtração:", subtracao)
5
```

Vamos aprender a fazer algumas operações básicas

Operações

Multiplicação:

```
1  num1 = 10
2  num2 = 5
3  multiplicacao = num1 * num2
4  print("Multiplicação:", multiplicacao)
5
```

Divisão:

```
1  num1 = 10
2  num2 = 5
3  divisao = num1 / num2
4  print("Divisão:", divisao)
5
```

Nos também podemos juntar as operações umas com as outras

Operações

```
1  num1 = 10
2  num2 = 6
3  num3 = 2
4  num4 = 2
5  resultado = num4*num1-num2/num3
6  print(resultado)
7
```

```
1  num1 = 10
2  num2 = 6
3  num3 = 2
4  num4 = 2
5  resultado = num4*(num1-num2)/num3
6  print(resultado)
7
```

Podemos usar parênteses para dar prioridade para as operações

Prioridade

A prioridade das operações em Python segue a convenção matemática padrão, conhecida como "PEMDAS", que significa

1. Parênteses
2. Expoentes
3. Multiplicação e Divisão (da esquerda para a direita)
4. Adição e Subtração (da esquerda para a direita)

Que tal um exemplo pra ver se vocês aprenderam ?

Prioridade

```
1  resultado = 10 + 2 * 3
2  print(resultado)
3
```

```
1  resultado = (10 + 2) * 3
2  print(resultado)
3
```

Qual o resultado que essas operações vão dar ?

Prioridade

```
1  resultado = 10 + 2 * 3
2  print(resultado)
3
```

```
16

=== Code Execution Successful ===
```

```
1  resultado = (10 + 2) * 3
2  print(resultado)
3
```

```
36

=== Code Execution Successful ===
```

Eai acertaram?
se não acertou não precisa ficar chateado, é so treinar que você aprende rapidinho

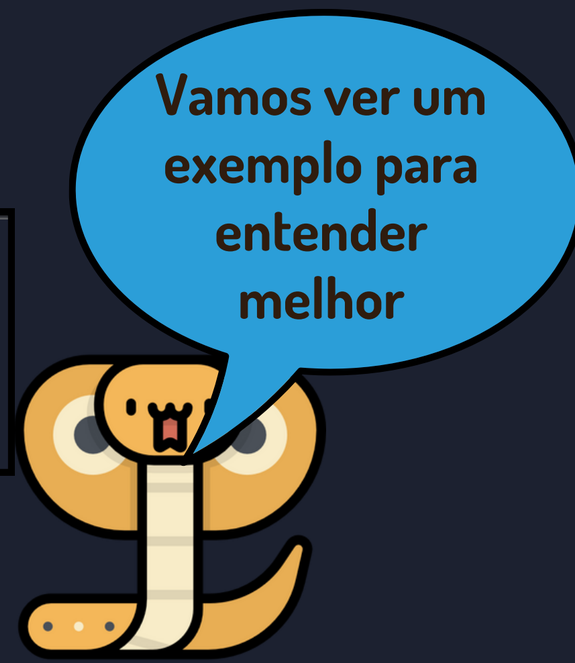
Iniciando bloco de código

O dois-pontos (“:”) é uma maneira de informar ao Python que um bloco de código está começando e que tudo indentado após ele faz parte desse bloco.

Este bloco de código pode ser loops (como o for e o while) e condicionais (como o if, else, elif).

```
1
2 for i in range(0,10):
3       print(i+1)
4
```

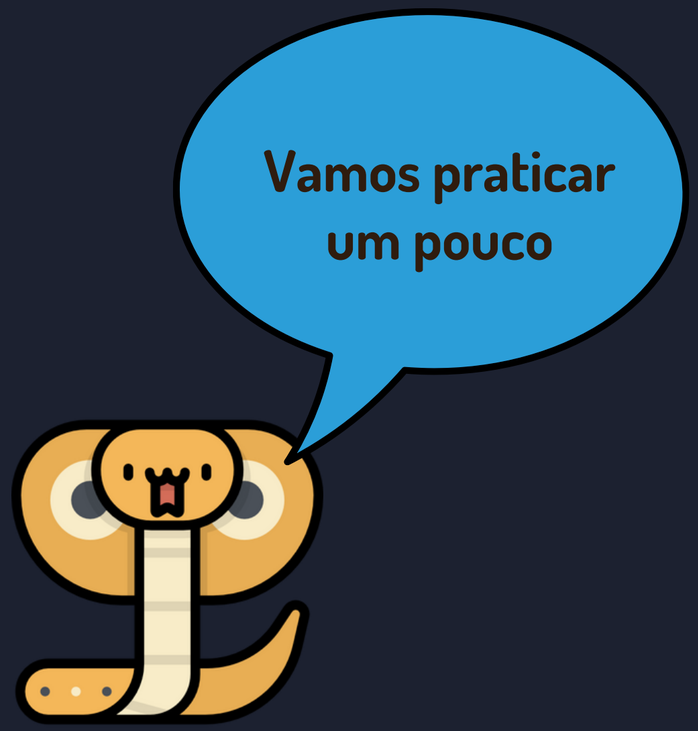
```
1 while i!=0:
2       print(i)
3       i = i+1
4
```



CODE LAB TEEN

Exercício

Faça uma variável para receber como input o seu nome, verifique com um if se é mesmo seu nome e se for imprima-o na tela, caso não seja seu nome imprima “Nome errado”



Dica, você usará os seguintes comandos

```
input(‘’)
if:
else:
print(“”)
```

OBRIGADO!

https://forms.gle/jy35CB6ZS2torZQz6

CODELAB TEEN