

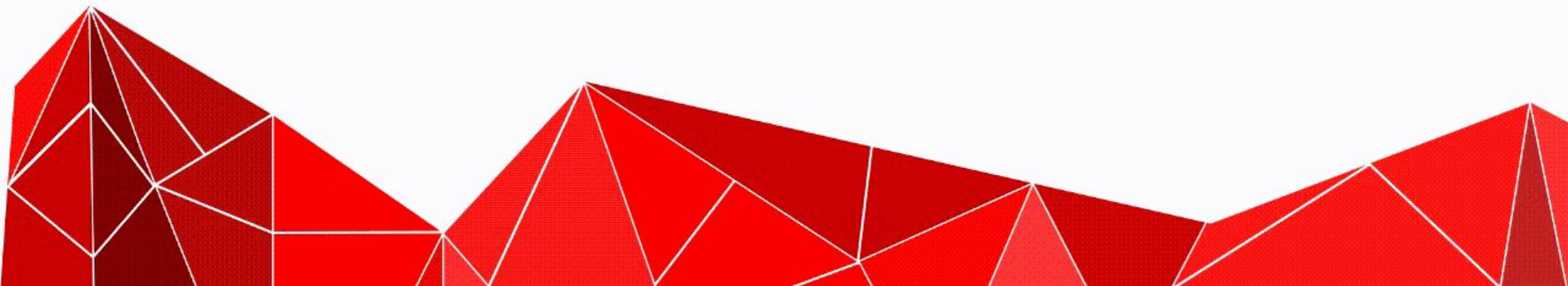


CODING

uma nova estrutura - FOR

Press -> to continue

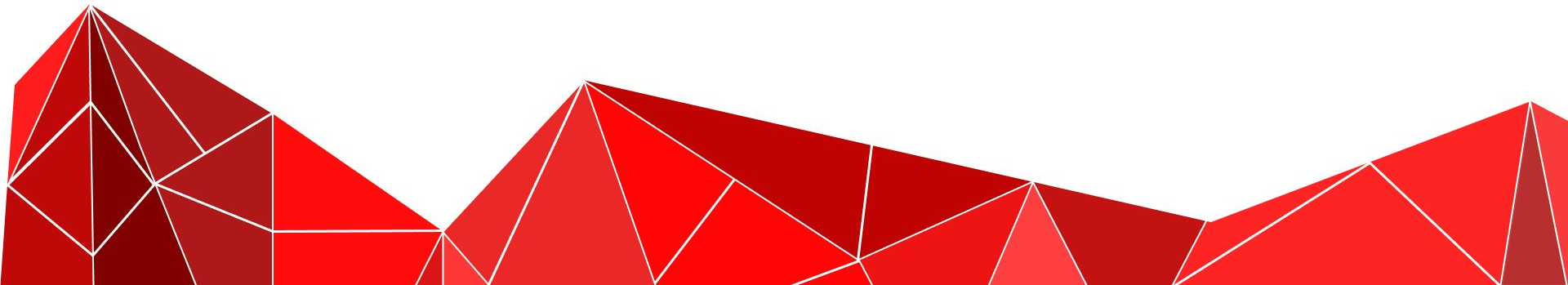
Powered by PET computação UFC



O que vamos ver hoje ?

Rápida definição

For





DEFINIÇÃO



Mas o que é isso ?

- As estruturas de repetição permitem executar mais de uma vez um mesmo trecho de código.
- As estruturas de repetição são úteis, por exemplo, para repetir uma série de operações semelhantes
- Através de determinadas condições
- Existem dois tipos no python: While e o For
- O while repete ENQUANTO uma condição estiver sendo satisfeita
- O for repete PARA cada elemento em uma determinada sequência

Mas o que é isso ?

- Imagine que você recebeu a missão de limpar o carro do senhor miyagi
- Toda vez que você for fazer o movimento de limpar o carro, você primeiro vai verificar se o carro ainda está sujo
- Vamos tomar o movimento como:
 - 1) Colocar a esponja no balde com sabão
 - 2) Tirar o excesso
 - 3) Fazer o movimento de limpeza
- Perceba que nesse exemplo você está fazendo uma determinada ação ENQUANTO uma certa condição está sendo satisfeita



Mas o que é isso ?

- No caso essa condição seria “o carro ainda está sujo”
- Se fossemos fazer um algoritmo seria mais ou menos assim:
- ENQUANTO o carro ainda estiver sujo FAÇA:
 - 1) Colocar a esponja no balde com sabão
 - 2) Tirar o excesso
 - 3) Fazer o movimento de limpeza
- Isso seria equivalente ao WHILE

Mas o que é isso ?

- Agora, Imagine que você trabalha em uma fábrica, mais especificamente na esteira de produção, apertando os parafusos das peças que passam pela esteira
- Você vai fazer o mesmo movimento de apertar os parafusos PARA CADA peça
- Nesse caso a “condição” seria PARA CADA peça NA esteira
- Se fossemos escrever um algoritmo, ele ficaria mais ou menos assim:



Mas o que é isso ?

- PARA CADA peça NA esteira FAÇA:
 - Apertar parafuso
- Agora, vamos ver a estrutura FOR no python



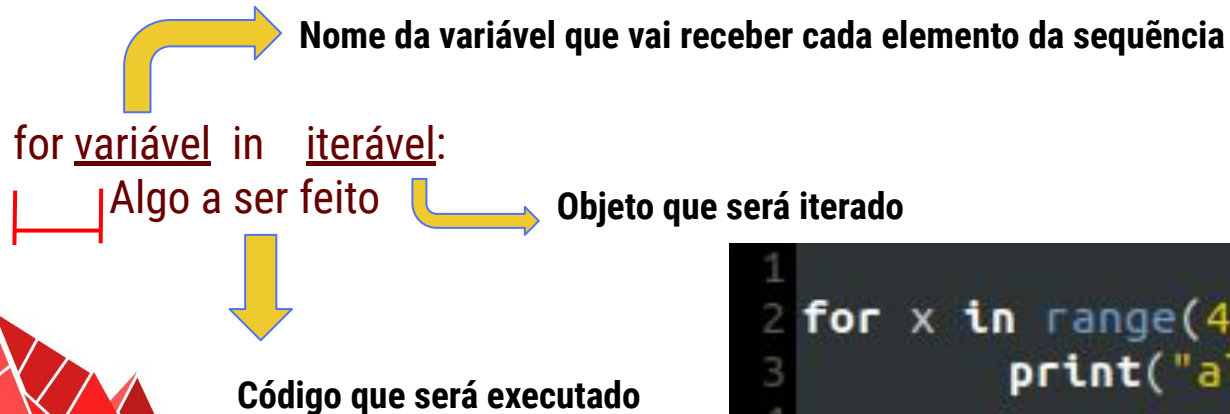


0 FOR

O FOR

- utilizado para percorrer ou iterar sobre uma sequência de dados
- Python utiliza indentação saber o'que está no for
- Podemos percorrer vários tipos de estruturas

- Estrutura básica:


for variável in iterável:
| Algo a ser feito | Objeto que será iterado
Código que será executado

```
1  
2 for x in range(4):  
3     print("algo a ser feito")  
4
```

O FOR

- Vamos ver um exemplo:



A screenshot of a Python IDE window titled `*firstProgram.py` with the file path `~/Documents`. The window contains a Python script that defines a list `lista` with values `[1, 2, 3, 4, 5]` and uses a `for` loop to iterate over each item in the list, printing each item. The code is as follows:

```
1 lista = [1, 2, 3, 4, 5]
2 for item in lista:
3     print(item)
4
5
6
```

The IDE interface includes a menu bar with `Open`, `Save`, and a hamburger menu icon. The status bar at the bottom indicates the current file is `Python`, the tab width is `8`, and the cursor is at `Ln 5, Col 1` in `INS` mode.

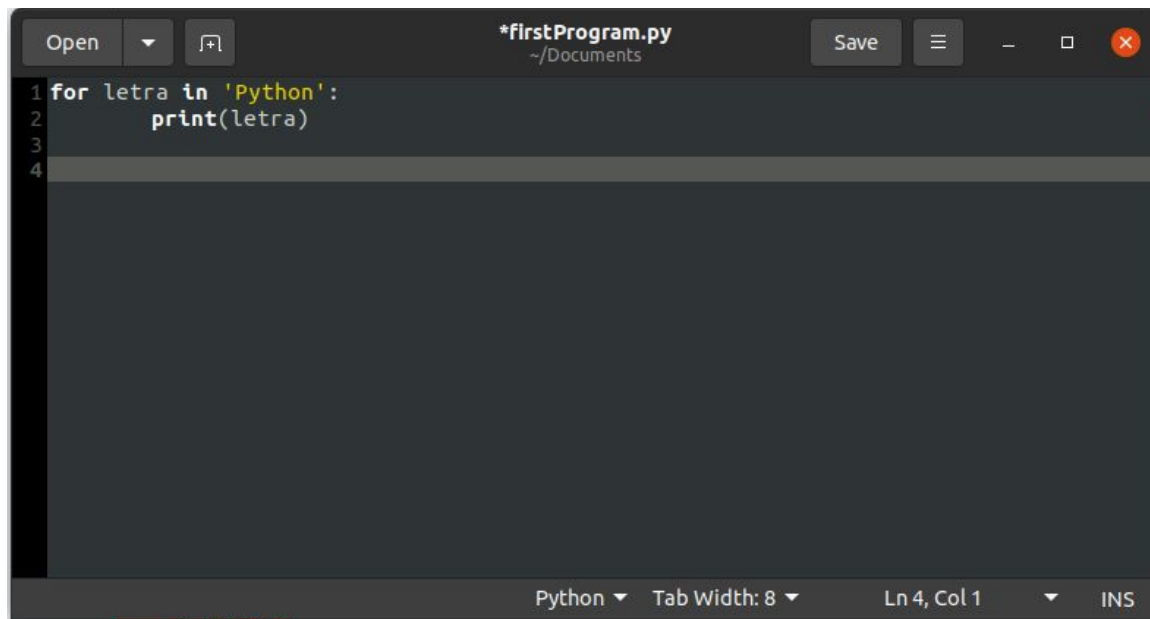
O FOR

- E o resultado é:

```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
rafael@rafael-Lenovo-ideapad-330-15IKB:~/Documents$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exec(open('firstProgram.py').read())
1
2
3
4
5
>>> 
```

O FOR

- Vamos ver um exemplo:



A screenshot of a Python IDE window titled `*firstProgram.py` with the path `~/Documents`. The window has a dark theme and includes buttons for `Open`, `Save`, and window management icons. The code editor contains the following Python code:

```
1 for letra in 'Python':  
2     print(letra)  
3  
4
```

The status bar at the bottom indicates the file is `Python`, the `Tab Width` is `8`, the cursor is at `Ln 4, Col 1`, and the input mode is `INS`.

O FOR

- E o resultado é:

```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents
>>> exec(open('firstProgram.py').read())
P
y
t
h
o
n
>>> 
```

O FOR

- A função range():
 - “Range” vem do inglês e significa alcance.
 - É uma função do python que utilizamos quando queremos que o for repita uma determinado quantidade de vezes.
 - Em outras palavras é como se fosse o “alcance” do nosso for
- Ela gera uma sequência de números pela qual podemos iterar!
- Com ela, conseguimos especificar:
 - O início de uma sequência;
 - O passo (ou pulo);
 - O valor final da sequência.
- Vamos ver um pouco mais detalhado.

O FOR

- A função `range()`:
 - A sintaxe básica da função `range()` é:
 - `range(start, stop, step)`
- Intervalo fechado e aberto
 - intervalo aberto `] x [`
 - Quando os colchetes se opõem ao número, temos um intervalo aberto. Assim, todos os números no intervalo estarão contidos na sequência numérica, menos os extremos.
 - intervalo fechado `[x]`
 - Quando o colchete estiver de "acordo" (no sentido normal) ao número, tem-se um intervalo fechado. Assim, todos os números contidos neste intervalo estarão contidos inclusive os extremos da sequência.

O FOR

- A função range():
 - A sintaxe básica da função range() é:
 - range(start, stop, step)
- De volta ao python
 - Na função range podemos colocar apenas o parâmetro “stop”
 - Para isso basta colocarmos -> stop(final)
 - Exemplo

```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents
>>> list(range(5))
[0, 1, 2, 3, 4]
>>>
```

- Veja que nesse caso o intervalo gerado é [0, 4[

O FOR

- Quando não colocamos o início, o python entende que o início é zero
- Mas podemos definir o inicio que quisermos.
- Vamos ver um exemplo:

```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
>>> list(range(2,6))
[2, 3, 4, 5]
>>> 
```

O FOR

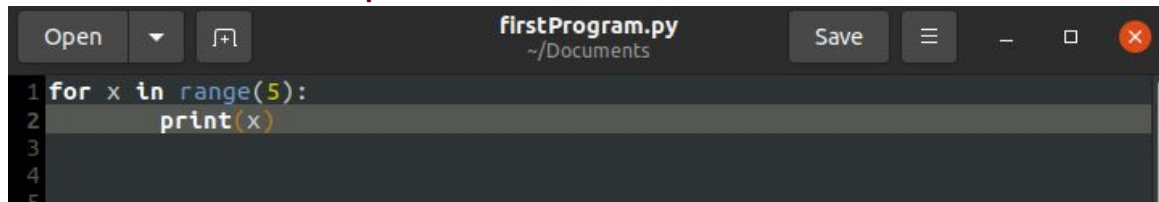
- Ainda podemos colocar o tamanho do salto.

```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents
>>> list(range(2,10,2))
[2, 4, 6, 8]
>>> 
```

- Lembre - se, como o intervalo da função range() é [a,b[, então como o passo é 2 e o próximo elemento seria o próprio 10, então a função range() não mostra o 10.

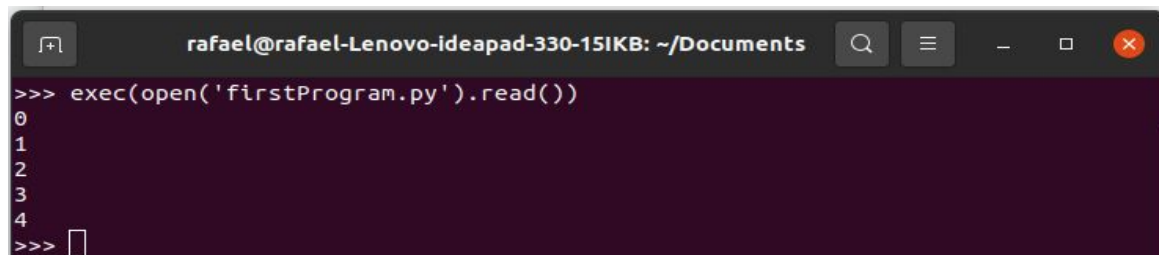
O FOR

- Usando a função range no for:
- Com ela geramos uma lista de número que podemos iterar sobre ela.
 - Vamos ver um exemplo:



```
firstProgram.py
~/Documents

1 for x in range(5):
2     print(x)
3
4
5
```



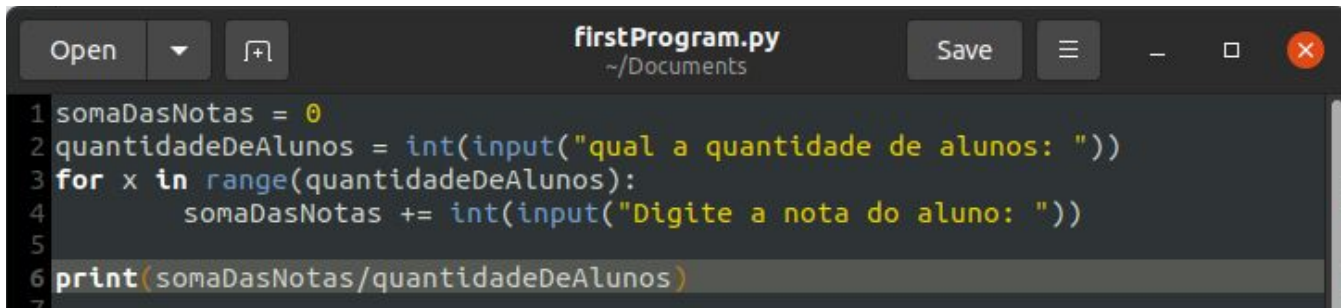
```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents

>>> exec(open('firstProgram.py').read())
0
1
2
3
4
>>>
```



O FOR

- Mas não necessariamente precisamos usar os números gerados pela função `range()`, podemos usar ela apenas para fazer o `for` “rodar” uma certa quantidade de vezes
- Imagine que você vai criar um programa que calcula a média geral de uma turma, e que esse programa vai pedir a nota de cada aluno, dependendo do número de alunos que o professor colocar.



```
1 somaDasNotas = 0
2 quantidadeDeAlunos = int(input("qual a quantidade de alunos: "))
3 for x in range(quantidadeDeAlunos):
4     somaDasNotas += int(input("Digite a nota do aluno: "))
5
6 print(somaDasNotas/quantidadeDeAlunos)
```

O FOR

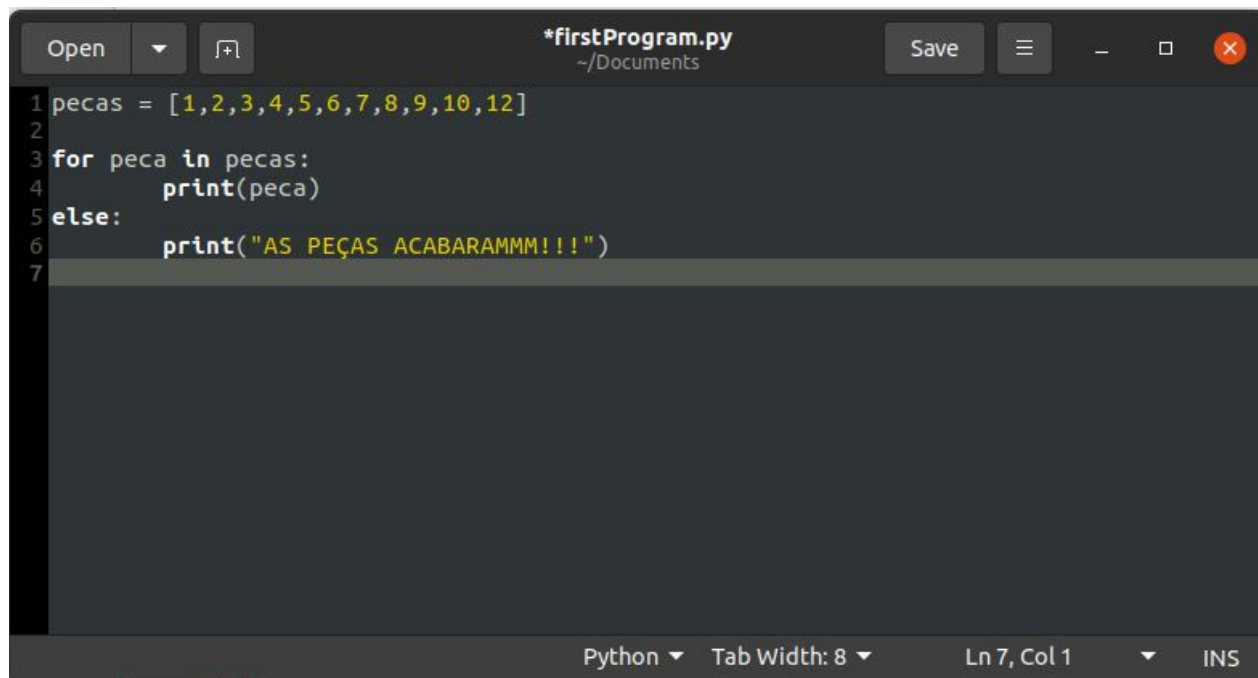
- O resultado é:

```
rafael@rafael-Lenovo-ideapad-330-151KB: ~/Documents
>>> exec(open('firstProgram.py').read())
qual a quantidade de alunos: 5
Digite a nota do aluno: 1
Digite a nota do aluno: 3
Digite a nota do aluno: 5
Digite a nota do aluno: 6
Digite a nota do aluno: 8
4.6
>>> █
```

O FOR

- O for com **else**
- Imagine que você voltou a trabalhar na fábrica
- O seu supervisor lhe falou para que quando as peças pararem de passar você teria que avisar que as peças acabaram
- No python isso seria equivalente ao FOR com o ELSE
- Podemos adicionar algo para ser executado depois que o loop acaba
- Vamos ver um exemplo:

0 FOR



A screenshot of a Python IDE window titled `*firstProgram.py` located in `~/Documents`. The window contains a Python script with a `for` loop. The code is as follows:

```
1 pecas = [1,2,3,4,5,6,7,8,9,10,12]
2
3 for peca in pecas:
4     print(peca)
5 else:
6     print("AS PEÇAS ACABARAMM!!!")
7
```

The IDE interface includes a top bar with buttons for `Open`, `Save`, and window management icons. The bottom status bar shows `Python`, `Tab Width: 8`, `Ln 7, Col 1`, and `INS`.

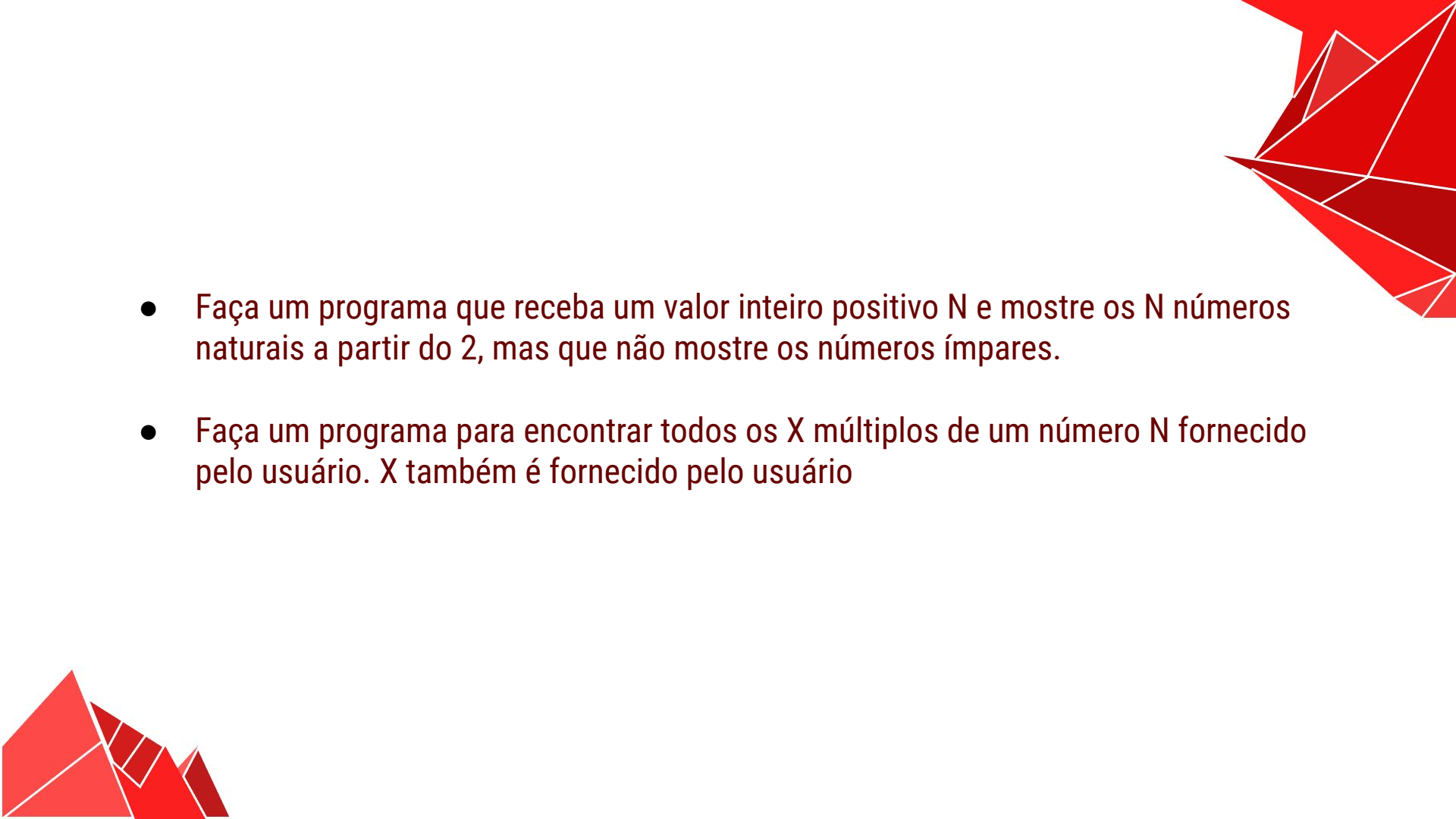
O FOR

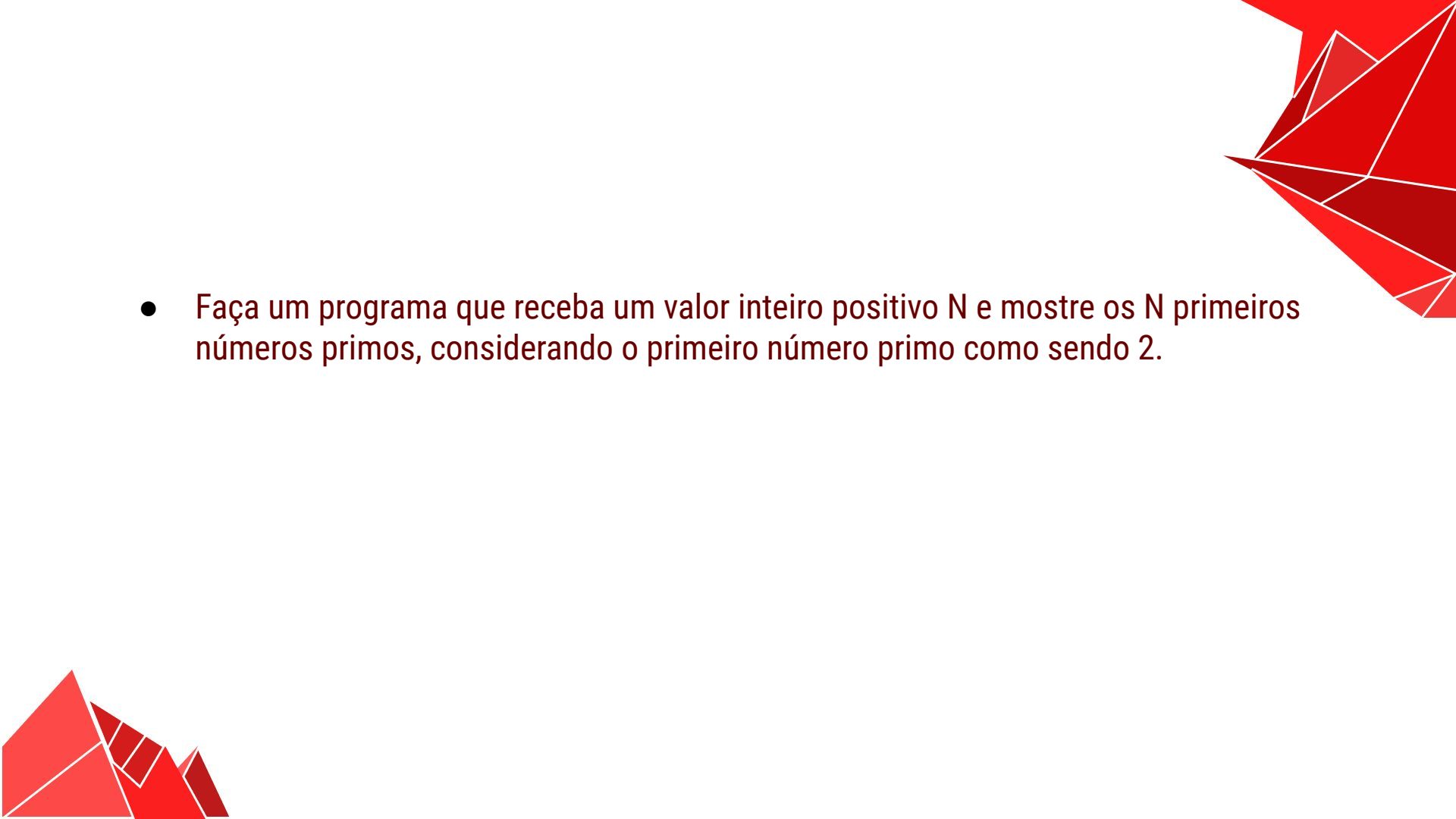
```
rafael@rafael-Lenovo-ideapad-330-15IKB: ~/Documents
>>> exec(open('firstProgram.py').read())
1
2
3
4
5
6
7
8
9
10
12
AS PEÇAS ACABARAMMM!!!
>>> 
```



VAMOS PRATICAR ! ! ! ! !

- 
- Faça um programa que receba um valor inteiro positivo N e mostre os N números naturais a partir do 1.

- 
- Faça um programa que receba um valor inteiro positivo N e mostre os N números naturais a partir do 2, mas que não mostre os números ímpares.
 - Faça um programa para encontrar todos os X múltiplos de um número N fornecido pelo usuário. X também é fornecido pelo usuário

- 
- Faça um programa que receba um valor inteiro positivo N e mostre os N primeiros números primos, considerando o primeiro número primo como sendo 2.

The background of the slide features a red grid pattern. The grid is composed of horizontal and vertical lines, with additional diagonal lines extending from the corners towards the center, creating a perspective effect. The text is centered within this grid.

Obrigado!!
Ainda com dúvida ?
Entre em contato