

EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!
EU ADORO PROGRAMAR! EU ADORO PROGRAMAR!



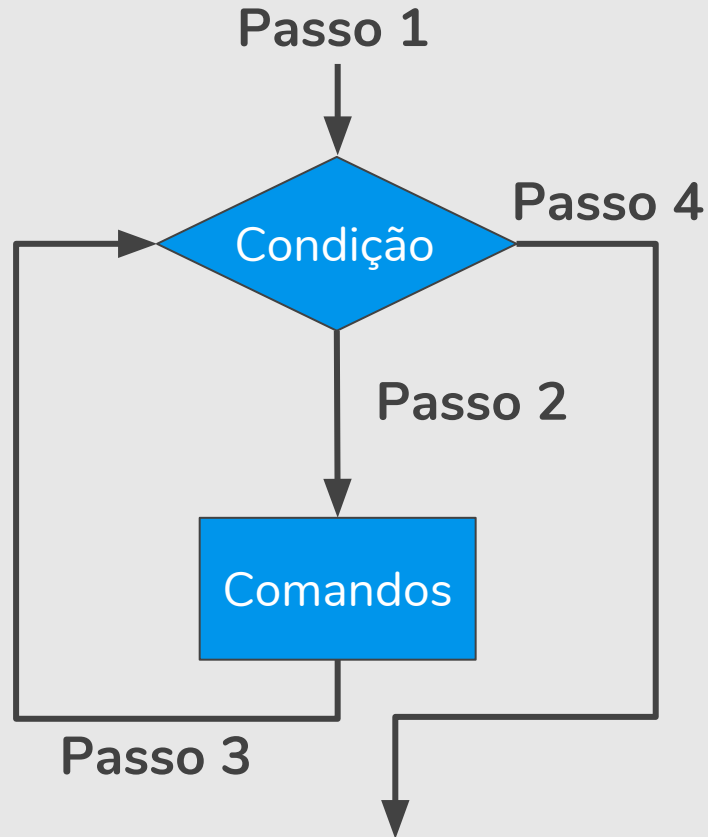
Comando while

Comando `while`

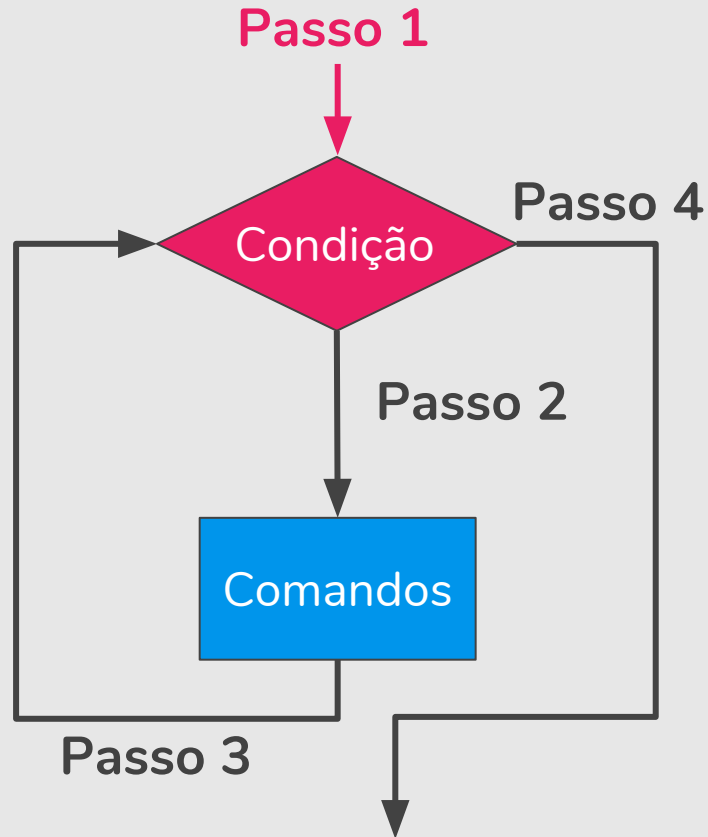
- Executa um bloco de comando(s) enquanto a condição é verdadeira (`True`).

```
while condicao:  
    comando(s)
```

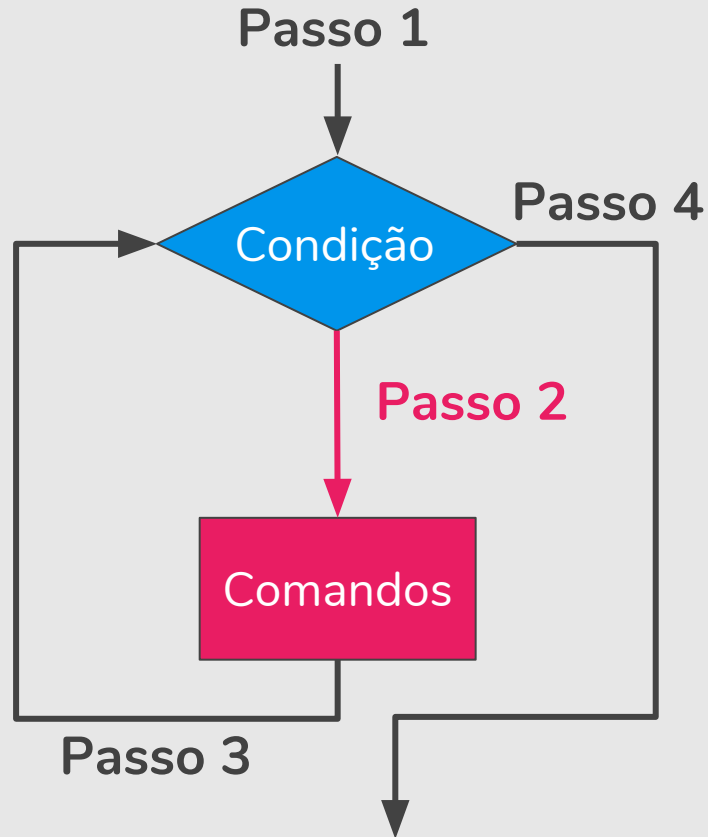
Comando `while`



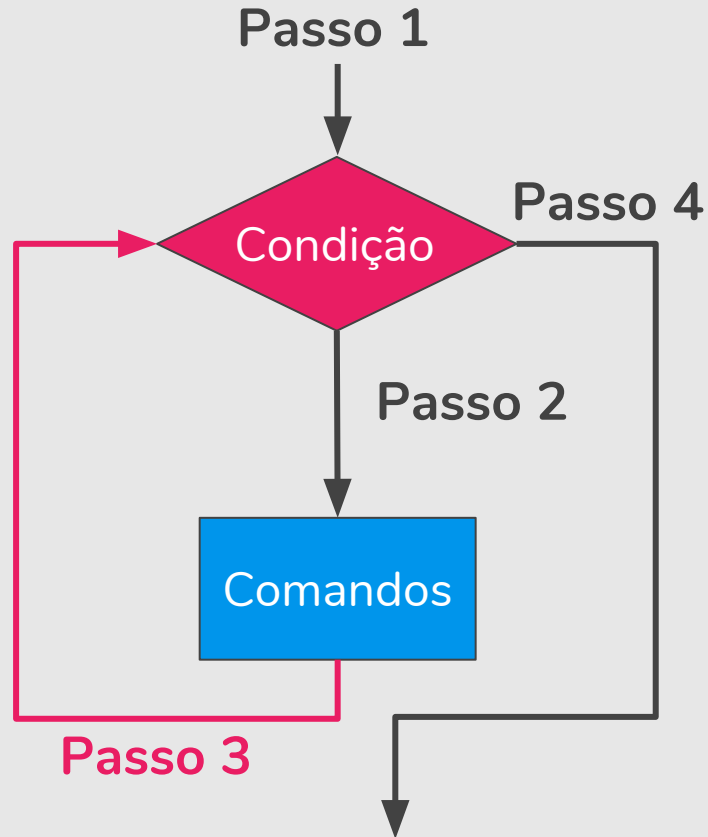
Comando `while`



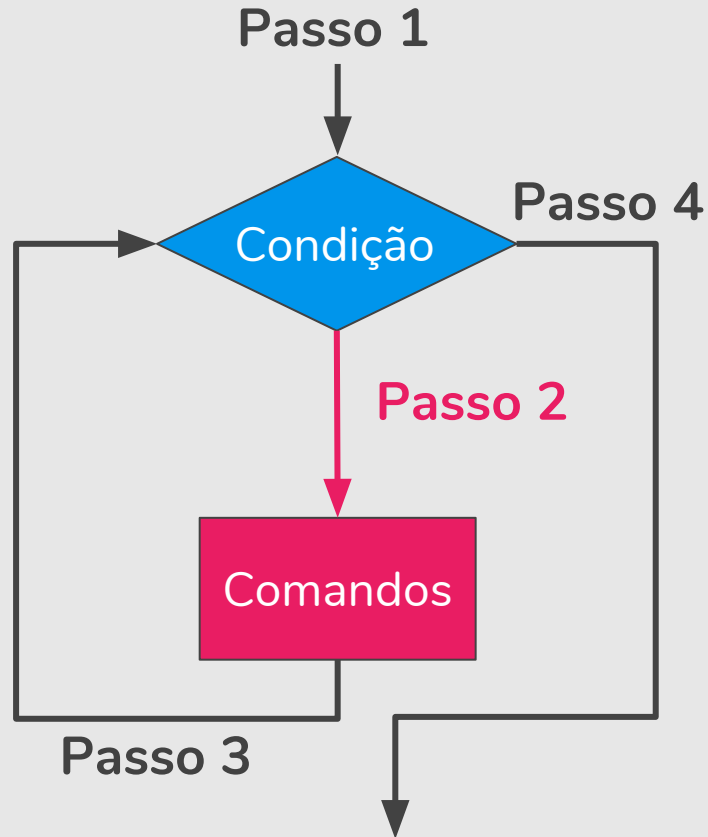
Comando `while`



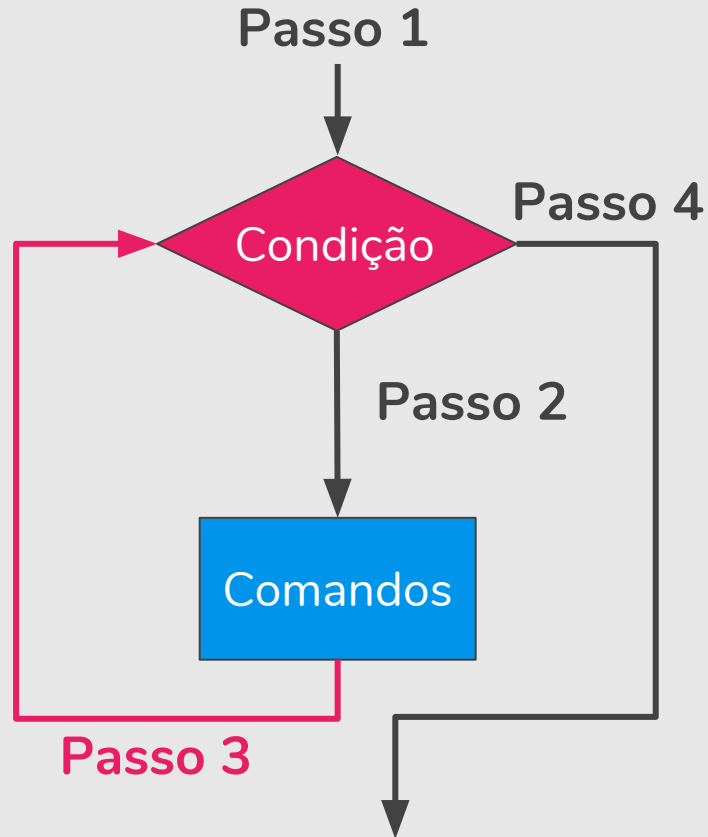
Comando `while`



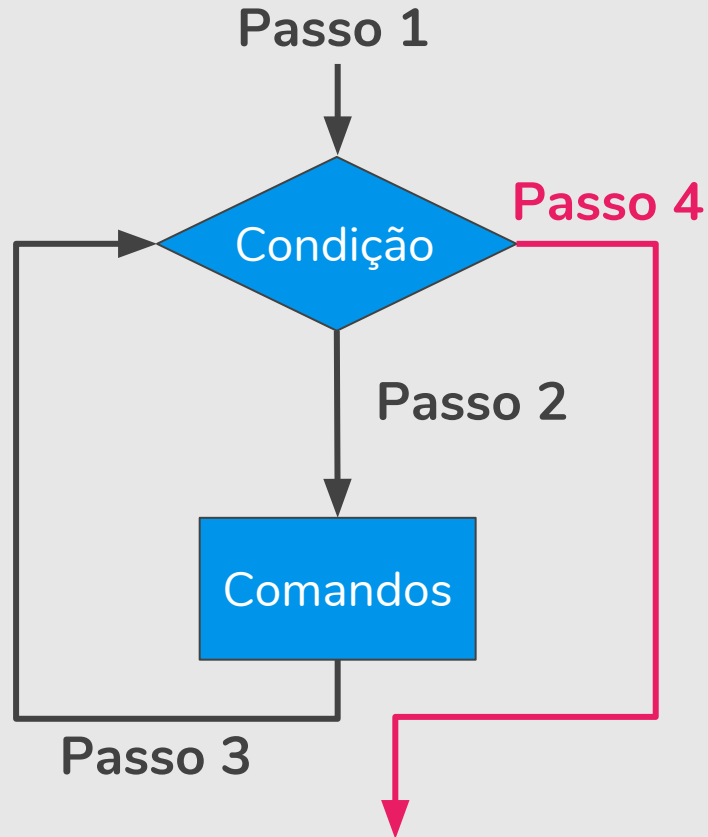
Comando `while`



Comando `while`



Comando `while`



Comando `while`

- Programa que imprime todos os números de 1 a 100.

```
# Imprime todos os números de 1 a 100  
numero = 1  
while numero <= 100:  
    print(numero)  
    numero = numero + 1
```

Comando `while`

- Programa que imprime os n primeiros números.

```
# Imprime os n primeiros números
n = int(input("Digite um número: "))
numero = 1
while numero <= n:
    print(numero)
    numero = numero + 1
```

Listas

Listas (Breve Introdução)

- Uma lista em Python é uma estrutura que armazena vários dados, que **podem ser de um mesmo tipo ou não**.
- Uma lista é criada como a construção: $[\text{dado}_1, \text{dado}_2, \dots, \text{dado}_n]$

```
lista1 = [10, 20, 30, 40]
lista2 = ["programação", "mc102", "python"]
lista3 = ["oi", 2.0, 5, [10, 20]]
```

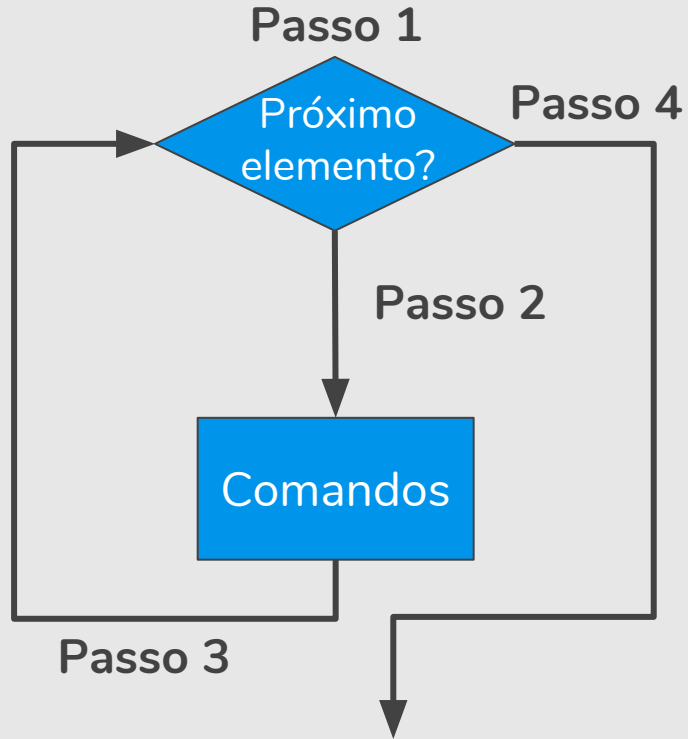
Comando for

Comando `for`

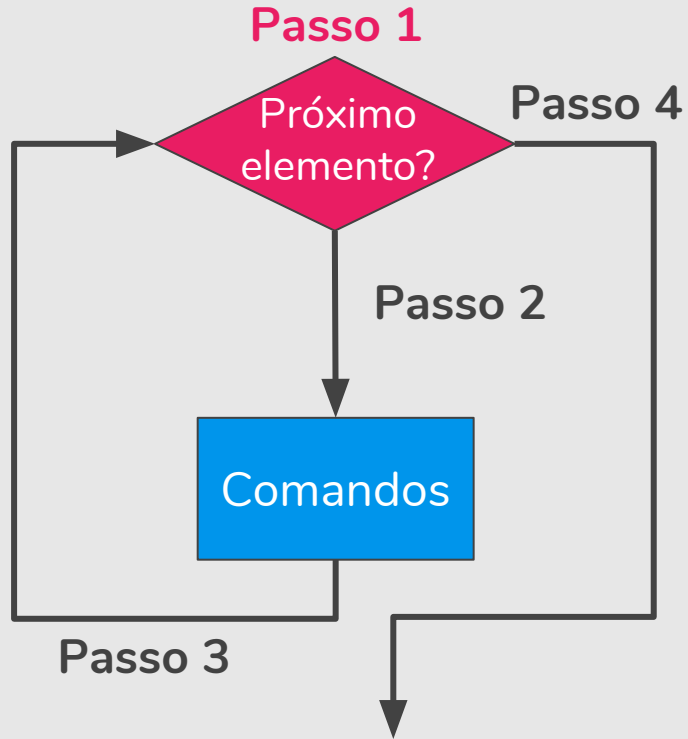
- É a estrutura de repetição mais usada no Python.
- Para cada elemento da lista, em ordem de ocorrência, é atribuído este elemento à variável e então é executado o(s) comando(s).

```
for variável in lista:  
    comando(s)
```

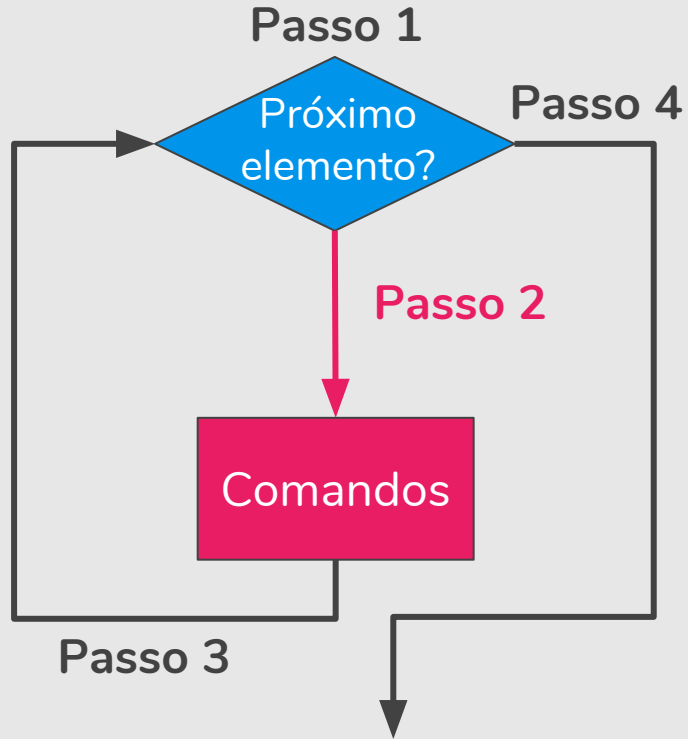

Comando for



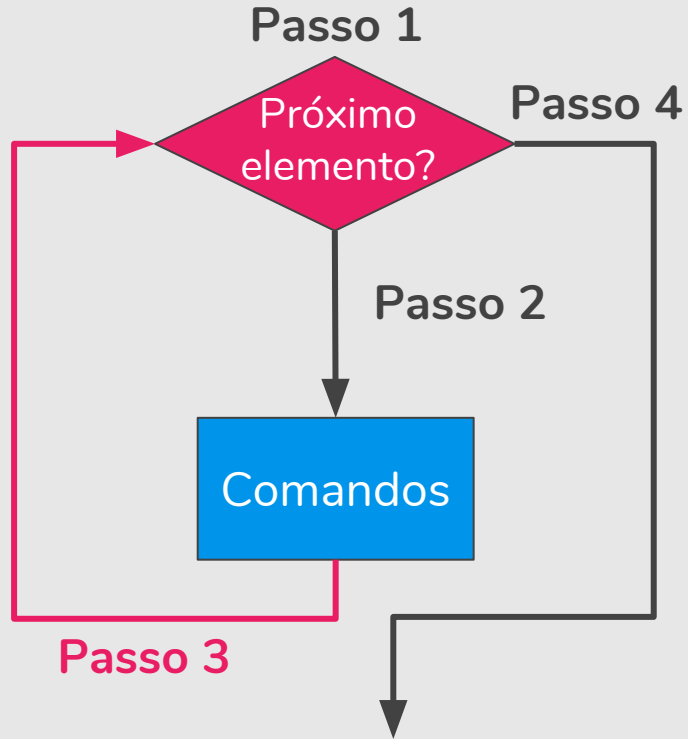
Comando for



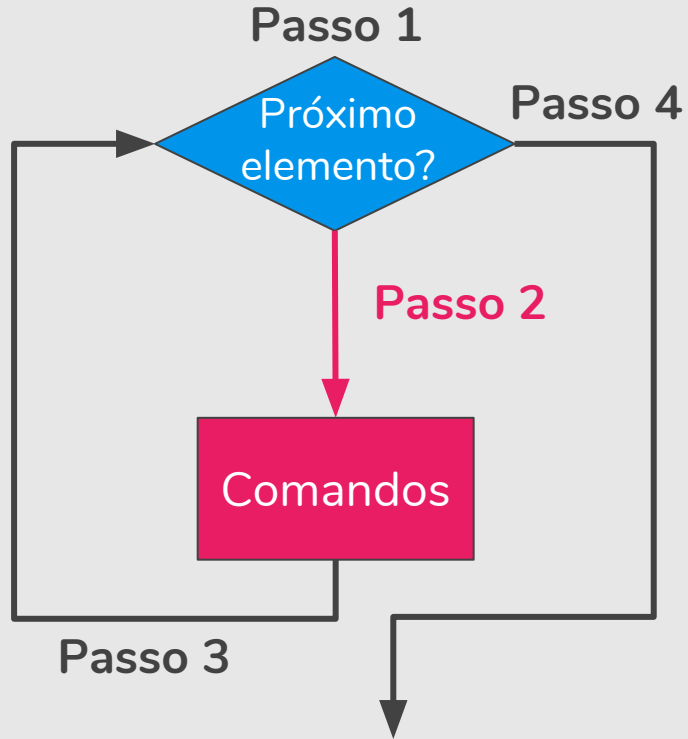
Comando for



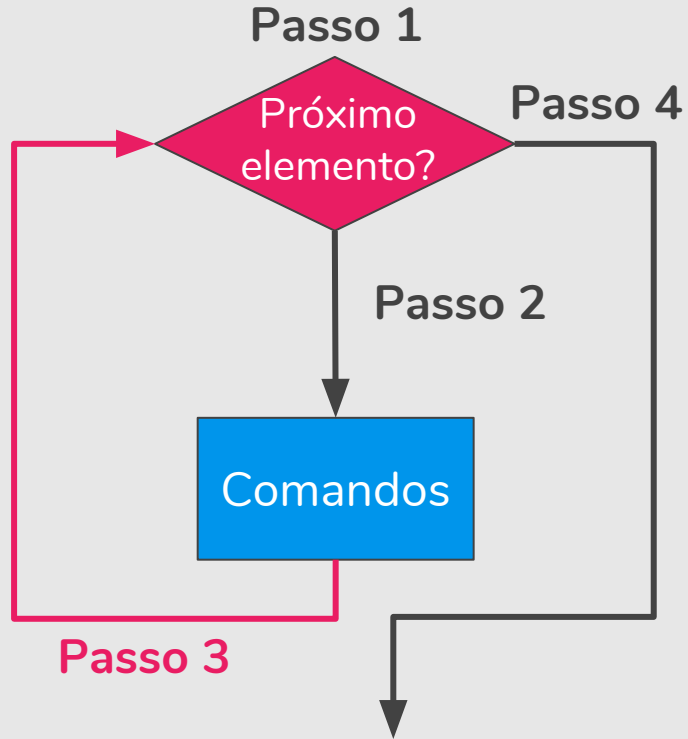
Comando for



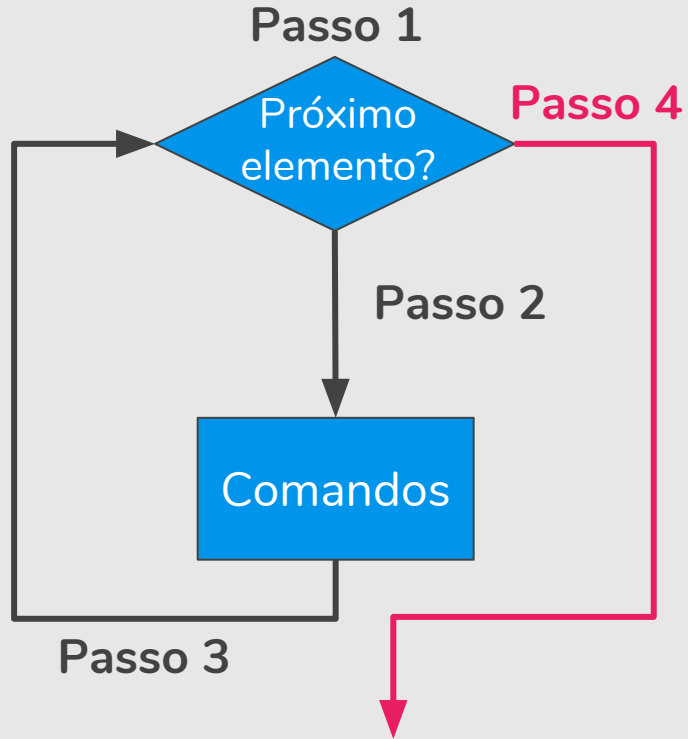
Comando for



Comando for



Comando for



Comando `for`

- Programa que imprime todos os números de uma lista.

```
# Imprime todos os números de uma lista  
lista_numeros = [1, 2, 3, 4, 5]  
for numero in lista_numeros:  
    print(numero)
```

```
1  
2  
3  
4  
5
```


A Função `range`

- É comum fazermos um laço `for` iterar sobre valores numéricos.
- Em Python, a função `range(n)` gera uma lista com valores de 0 até $n-1$.
- Programa que imprime todos os números de 0 a 9.

```
# Imprime todos os números de 0 a 9  
for numero in range(10):  
    print(numero)
```

A Função `range`

- Podemos especificar um intervalo de valores na função `range(n)`
 - `range(inicio, fim)`: gera-se números de `inicio` até `fim-1`.
- Programa que imprime todos os números de 5 a 9.

```
# Imprime todos os números de 5 a 9  
for numero in range(5,10):  
    print(numero)
```

A Função `range`

- Podemos especificar um passo a ser considerado no intervalo de valores na função `range(n)`
 - `range(inicio, fim, passo)`: gera-se números de `inicio` com incremento de `passo` até `fim-1`.

A Função range

- Programa que imprime todos os números pares entre 0 e 13.

```
# Imprime todos os números pares entre 0 e 13  
for numero in range(0,13,2):  
    print(numero)
```

```
0  
2  
4  
6  
8  
10  
12
```

while e for

- Programa que imprime os n primeiros números.

```
# Imprime os n primeiros números
n = int(input("Digite um número: "))
numero = 1
while numero <= n:
    print(numero)
    numero = numero + 1
```

```
# Imprime os n primeiros números
n = int(input("Digite um número: "))
for numero in range(1, n+1):
    print(numero)
```

Jogo de Adivinhação

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    palpites = palpites + 1
    if meu_palpite > numero:
        print(meu_palpite, "está acima.")
    elif meu_palpite < numero:
        print(meu_palpite, "está abaixo.")
    meu_palpite = int(input("tente novamente: "))
print("\nÓtimo, você acertou em", palpites, "tentativas!")
```

Variável Acumuladora

Jogo de Adivinhação

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    palpites = palpites + 1
    if meu_palpite > numero:
        print(meu_palpite, "está acima.")
    elif meu_palpite < numero:
        print(meu_palpite, "está abaixo.")
    meu_palpite = int(input("tente novamente: "))
print("\nÓtimo, você acertou em", palpites, "tentativas!")
```


Laços e os comandos
`break` e `continue`

Laços e o Comando `break`

- O comando `break` faz com que a execução de um laço seja terminada, passando a execução para o próximo comando depois do final do laço.

```
while condicao:  
    comando(s)  
    break  
comando(s)
```

```
for variável in lista:  
    comando(s)  
    break  
comando(s)
```

Laços e o Comando `break`

- O que será impresso?

```
for numero in range(1,11):  
    if (numero >= 5):  
        break  
    print(numero)  
print("Terminou o laço.")
```

```
1  
2  
3  
4  
"Terminou o laço."
```

Laços e o Comando `continue`

- O comando `continue` faz com que a execução de um laço seja alterada para o final do laço.

```
numero = 1
while numero <= 10:
    if (numero == 5):
        numero = numero + 1
        continue
    print(numero)
    numero = numero + 1
print("Terminou o laço.")
```

- O que será impresso?

Laços e o Comando `continue`

- O comando `continue` faz com que a execução de um laço seja alterada para o final do laço.

```
numero = 1
while numero <= 10:
    if (numero == 5):
        numero = numero + 1
        continue
    print(numero)
    numero = numero + 1
print("Terminou o laço.")
```

```
1
2
3
4
6
7
8
9
10
```

```
"Terminou o laço."
```

- O que será impresso?

Exercícios

1. Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .
2. Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .
3. Repita o Jogo de Adivinhação dando a opção do jogador de desistir, por exemplo, escolhendo o número 0.

Exercícios

1. Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .
2. Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .
3. Repita o Jogo de Adivinhação dando a opção do jogador de desistir, por exemplo, escolhendo o número 0.

Exercício 1: Usando `while`

Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

Exercício 1: Usando `while`

Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

```
base = int(input("Digite a base: ")) # base a
expoente = int(input("Digite o expoente: ")) # expoente b

contador = 0
resultado = 1

while (contador < expoente):
    # base ** expoente = base * base (expoente vezes)
    resultado = resultado * base
    contador = contador + 1
print(base, "elevado a", expoente, "=", resultado)
```

Exercício 1: Usando `while`

Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

```
base = int(input("Digite a base: ")) # base a
expoente = int(input("Digite o expoente: ")) # expoente b

contador = 0
resultado = 1

while (contador != expoente):
    # base ** expoente = base * base (expoente vezes)
    resultado = resultado * base
    contador = contador + 1
print(base, "elevado a", expoente, "=", resultado)
```

Exercício 1: Usando `for`

Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

Exercício 1: Usando `for`

Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

```
base = int(input("Digite a base: ")) # base a
expoente = int(input("Digite o expoente: ")) # expoente b

resultado = 1

for numero in range(1, expoente+1):
    # base ** expoente = base * base (expoente vezes)
    resultado = resultado * base
print(base, "elevado a", expoente, "=", resultado)
```

Exercício 1

Faça um programa que lê dois números inteiros ~~positivos~~ a e b. Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .

```
base = int(input("Digite a base: ")) # base a
expoente = int(input("Digite o expoente: ")) # expoente b

contador = 0
resultado = 1

if (expoente < 0):
    while contador > expoente:
        # base ** expoente = 1 / (base * base) (expoente vezes)
        resultado = resultado / base
        contador = contador - 1
else:
    while (contador < expoente):
        # base ** expoente = base * base (expoente vezes)
        resultado = resultado * base
        contador = contador + 1
print(base, "elevado a", expoente, "=", resultado)
```

```
base = int(input("Digite a base: ")) # base a
expoente = int(input("Digite o expoente: ")) # expoente b

contador = 0
resultado = 1

if (expoente < 0):
    while contador > expoente:
        # base ** expoente = 1 / (base * base) (expoente vezes)
        resultado = resultado / base
        contador = contador - 1
    print(base, "elevado a", expoente, "=", format(resultado, ".2f"))
else:
    while (contador < expoente):
        # base ** expoente = base * base (expoente vezes)
        resultado = resultado * base
        contador = contador + 1
    print(base, "elevado a", expoente, "=", resultado)
```

Exercícios

1. Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .
2. Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .
3. Repita o Jogo de Adivinhação dando a opção do jogador de desistir, por exemplo, escolhendo o número 0.

Exercício 2: Usando `for`

Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .

```
n = int(input("Digite um número inteiro positivo: "))

for numero in range(2, n+1):
    if (n % numero == 0): # se n é divisível por numero
        print(numero, end=" ")
```

Exercício 2: Usando `while`

Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .

```
n = int(input("Digite um número inteiro positivo: "))

numero = 2
while numero <= n:
    if (n % numero == 0): # se n é divisível por numero
        print(numero, end=" ")
    numero = numero + 1
```

Exercícios

1. Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .
2. Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .
3. Repita o Jogo de Adivinhação dando a opção do jogador de desistir, por exemplo, escolhendo o número 0.

Exercício 3: Jogo de Adivinhação

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    palpites = palpites + 1
    if meu_palpite > numero:
        print(meu_palpite, "está acima.")
    elif meu_palpite < numero:
        print(meu_palpite, "está abaixo.")
    meu_palpite = int(input("tente novamente: "))
print("\nÓtimo, você acertou em", palpites, "tentativas!")
```

```
import random # módulo random
numero = random.randrange(1, 101) # número entre 1 e 100

palpites = 0
meu_palpite = int(input("Adivinhe meu número entre 1 e 100: "))

while meu_palpite != numero:
    if meu_palpite == 0:
        print("Ah. Você desistiu do jogo.")
        break
    else:
        palpites = palpites + 1
        if meu_palpite > numero:
            print(meu_palpite, "está acima.")
        elif meu_palpite < numero:
            print(meu_palpite, "está abaixo.")
        meu_palpite = int(input("tente novamente: "))

if meu_palpite != 0:
    print("\nÓtimo, você acertou em", palpites, "tentativas!")
```

Mais Exercícios =)

- <https://wiki.python.org.br/EstruturaDeRepeticao>: 51 exercícios \o/
- Curso de Python:
 - <https://www.codecademy.com/learn/learn-python>