Estruturas de Controle

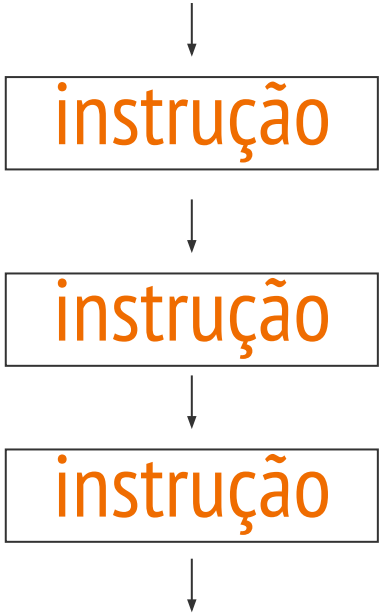
Estrutura Sequencial, Condicional e Repetição

Sumário

- ESTRUTURA SEQUENCIAL
- ESTRUTURA CONDICIONAL
- ESTRUTURA DE REPETIÇÃO

Estrutura Sequencial

Fluxograma



Linguagem Python

```
x,y=2,3

y,x=x,y

print(x,y)
```

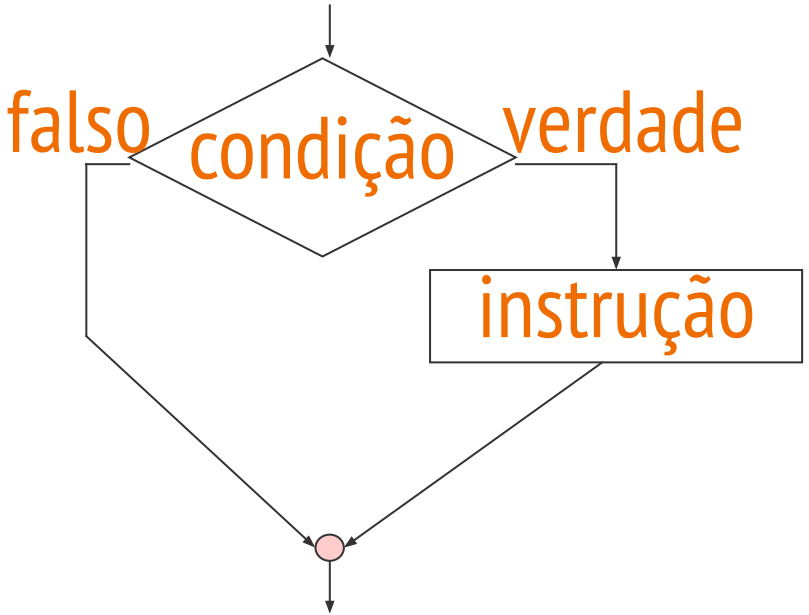
Estruturas Condicionais

- Estrutura Condicional Simples

- Estrutura Condicional Composta

Estruturas Condicionais Simples

Fluxograma



Linguagem Python

```
if (condição):
    <instrução>
```

Estruturas Condicionais Simples

Linguagem Python

a condição deve ser uma expressão lógica

```
if (condição):
    <instrução>
```

A instrução só será executado se a condição for verdadeira

Estruturas Condicionais Simples

Linguagem Python

```
if (condição):
    <instrução>
```

se mais de uma instrução deve ser executada quando a condição for verdadeira, essas instruções devem ser transformadas em instruções compostas.

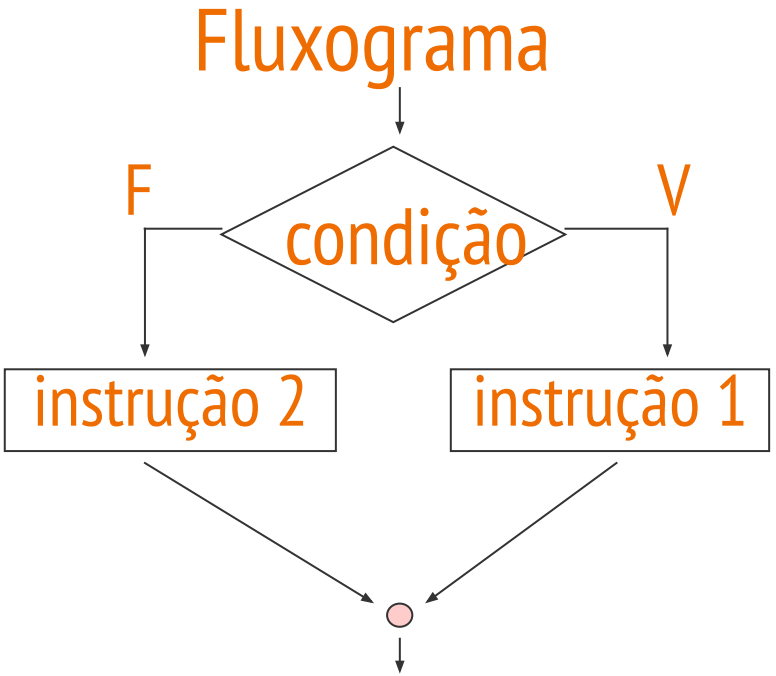
Estruturas Condicionais Simples

Linguagem Python

```
if (condição):
 <instrução>
 <instrução>
 <instrução>
```

As instruções compostas a serem executadas dentro de uma condição devem estar indentadas

Estruturas Condicionais Compostas



Python

```
if (condição):
    <instrução 1>
else:
    <instrução 2>
```

Estruturas Condicionais Compostas

```
if (condição):
        <instrução>
        ...
      <instrução>
else:
        <instrução>
        ...
      <instrução>
```

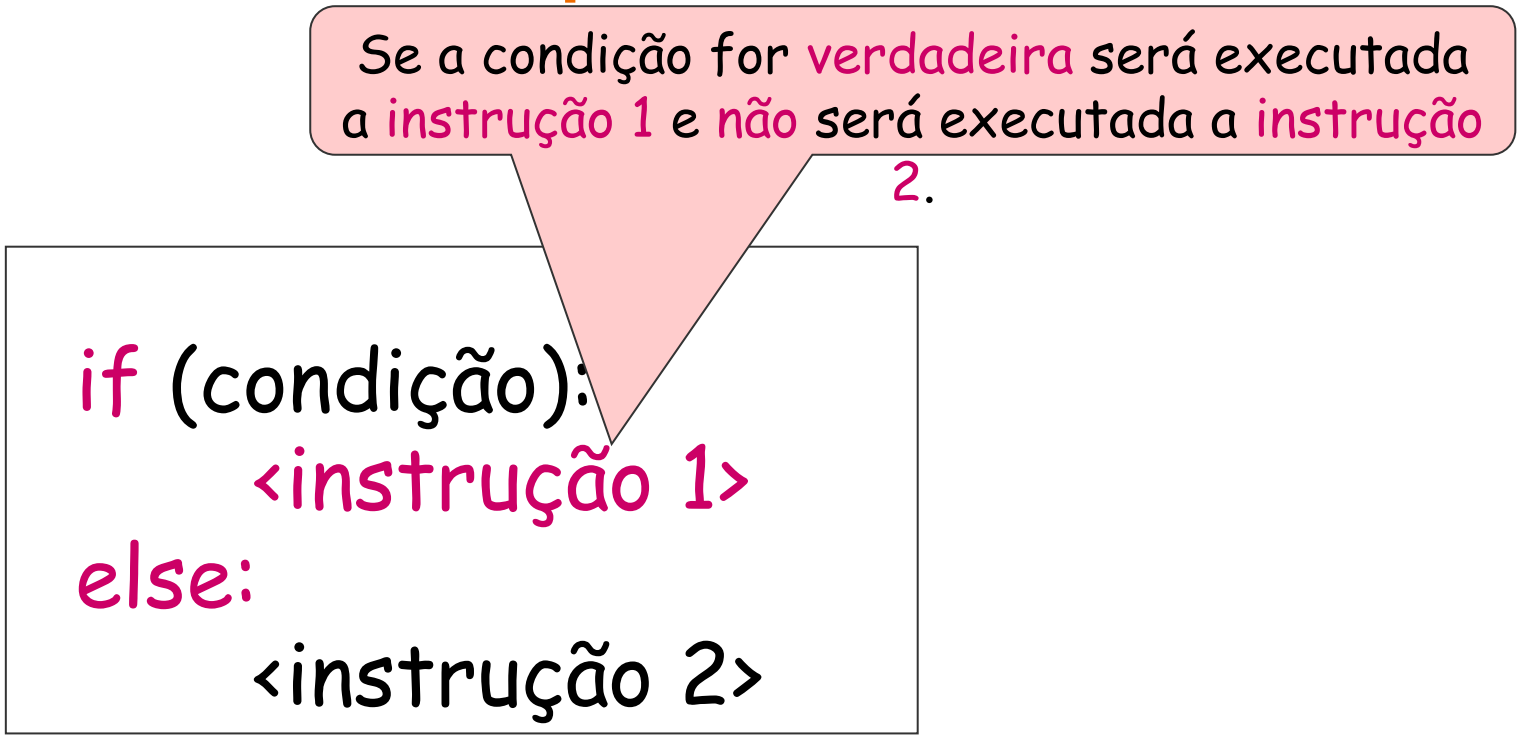
```
if (condição):
        <instrução>
        ...
      <instrução>
elif(condição):
      <instrução>
      if(cond):
          <instrução>
      else:
          <instrução>
else:
        <instrução>
        ...
      <instrução>
```

Estruturas Condicionais Compostas

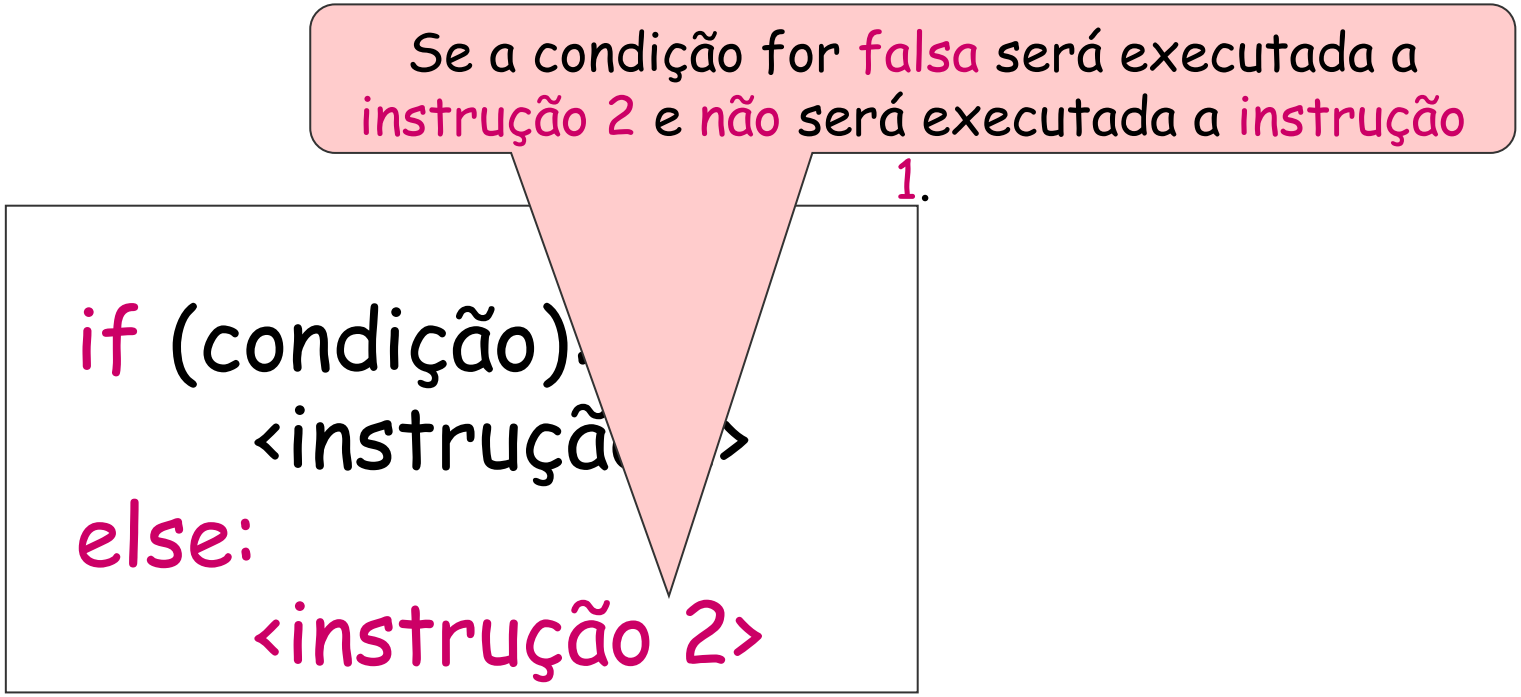
a condição deve ser uma expressão lógica

```
if (condição):
        <instrução 1>
else:
        <instrução 2>
```

Estruturas Condicionais Compostas



Estruturas Condicionais Compostas



Estruturas Condicionais Compostas

Operador Ternário

<expr1> if <conditional_expr> else
<expr2>

```
if (condição):
    <instrução 1>
else:
    <instrução 2>
```

```
<instrução 1> if (condição) else <instrução 2>
```

Estruturas Condicionais Compostas

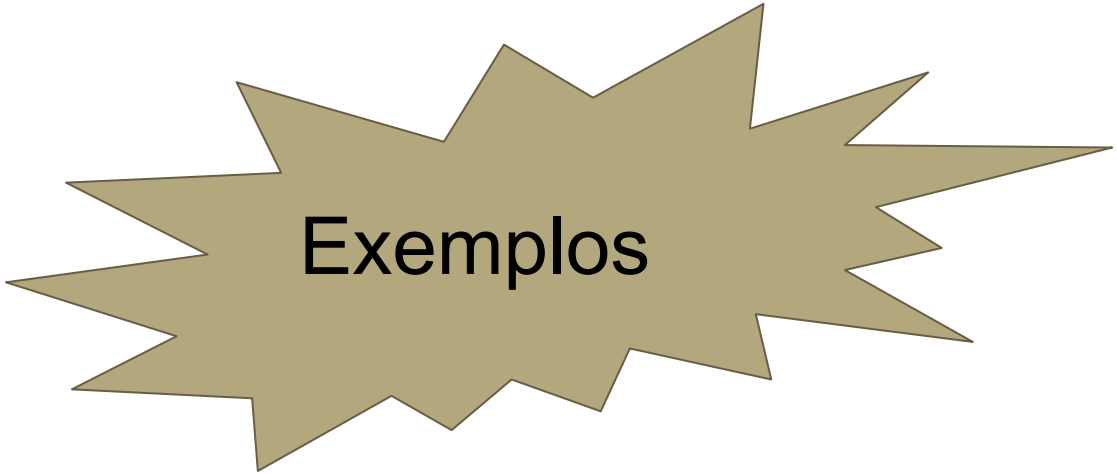
pass

```
if (condição):
    pass

<instrução>
```

- If deve ser seguido por alguma instrução.
- A ausência de instrução leva a um erro.
- Utiliza-se pass, se há alguma razão para não incluir uma instrução.

Estruturas Condicionais Compostas



Estruturas de Repetição

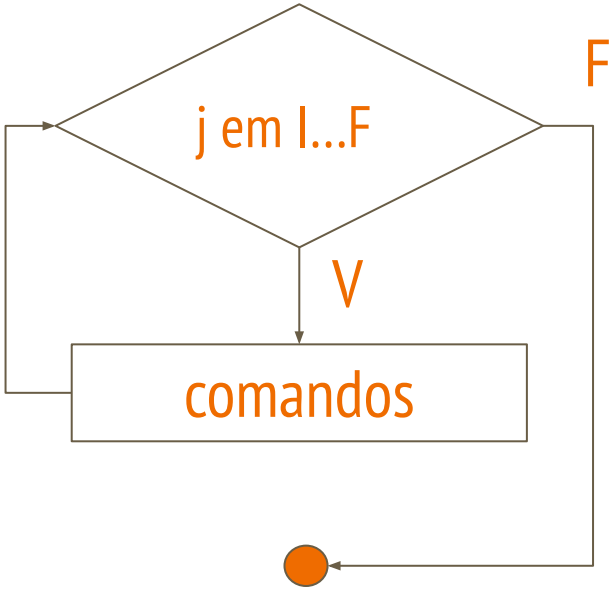
- Uma estrutura de repetição é utilizada quando um comando ou um bloco de comandos deve ser repetido.

- A quantidade de repetições pode ser fixa ou pode depender de uma determinada condição.

Estruturas de Repetição

Repetição Contada

Fluxograma



Linguagem Python

```
for j in range(I,F):
    <comando>
    ...
    <comando>
```

Estruturas de Repetição

```
for <variável-contador> in
range(<valores_possíveis>):
        instrução
```

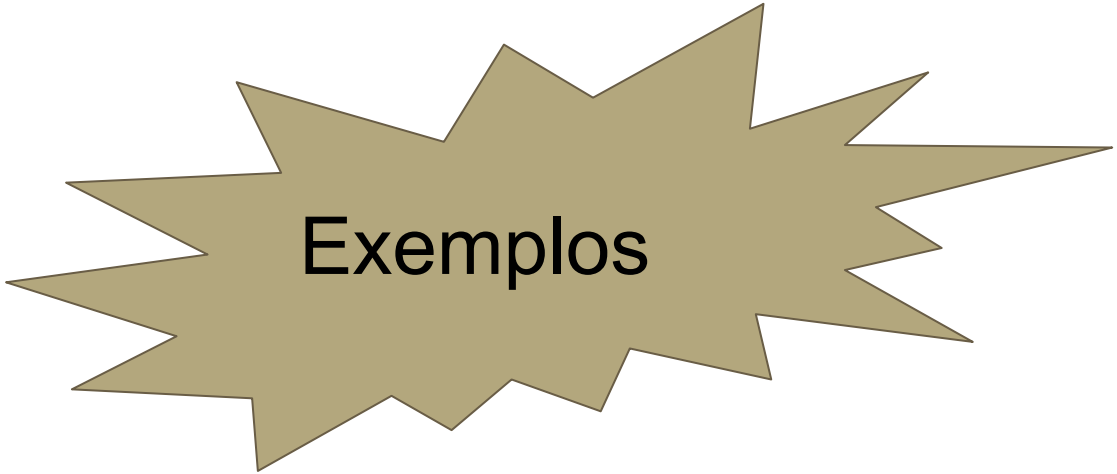
```
for <variável-contador> in
range(<valores_possíveis>):
        instrução_1
        instrução_2
        ...
        instrução_n
```

Estruturas de Repetição

range(início,fim,passo)

- início: primeiro elemento do conjunto de valores
  - default: início = 0
- fim: último elemento do conjunto de valores
- passo: tamanho do incremento ou decremento
  - default: passo = 1
- A repetição persiste até contador = fim -1

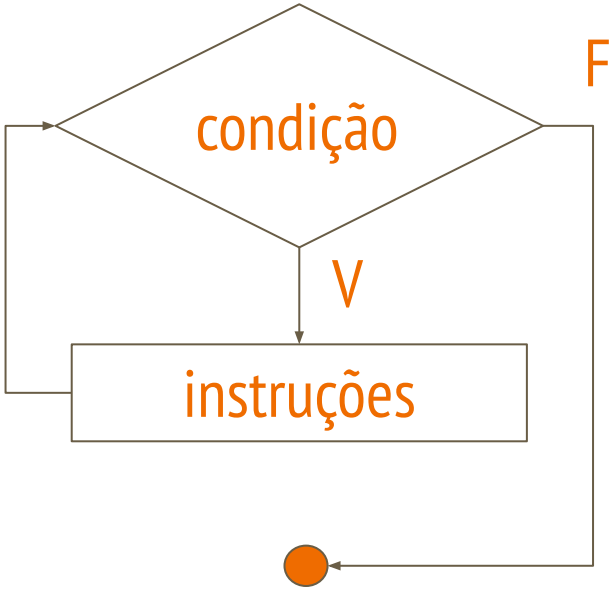
Estruturas de Repetição



Estruturas de Repetição

Repetição com teste no início

Fluxograma



Linguagem Python

```
while(<condição>) :
    <instrução>
    ...
    <instrução>
```

Estruturas de Repetição

"while" pode substituir o comando "for"

- ➔ Inicialização da variável de teste - início
- ➔ while(teste-fim):
  - ◆ incremento da variável de teste - passo

Estruturas de Repetição

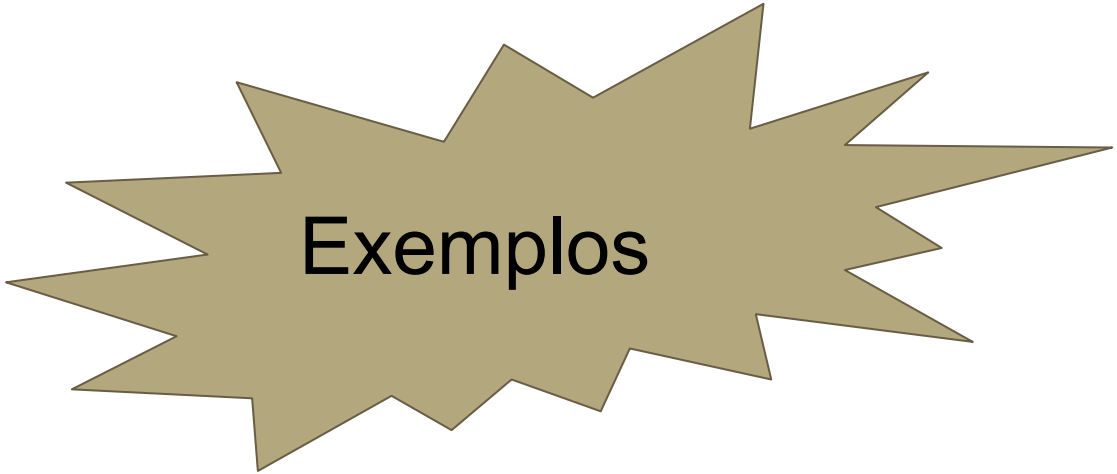
"while" pode substituir o comando "for"

```
for i in range(a,b,passo):
        <instrução>
```

```
i = a
while(i!=b):
    <instrução>
    i+=passo

#a e b números
```

Estruturas de Repetição



Estruturas de Repetição

Comando break

- Utilizado no corpo de qualquer estrutura de laço.
- Causa a saída imediata do laço, desviando o programa para a próxima instrução após o laço atual.
- Se estiver em laços aninhados, o break afetará somente o laço que o contém e seus laços internos.
- No caso de laços como for e while, o laço é interrompido e os comandos do programa são retomados a partir da primeira linha fora do laço.

Estruturas de Repetição

Comando break

```
1   import math
2   sum=0
3   while(1):
4       x=int(input("x="))
5       if(x<0):
6           break
7       sum += math.sqrt(x)
8   print("sum=",sum)
```

Estruturas de Repetição

Comando continue

- Força a execução da próxima iteração do laço, não executando o código que vem a seguir.
- Esse comando ocorre apenas nos comandos “for” e “while”.

```
1   for valor in range(1,10):
2       if(valor%2):
3           continue
4       print(valor,end=' ')
```

```
2 4 6 8
```

Estruturas de Repetição

Comando continue

- No caso do “while”, a execução é desviada para o teste condicional e depois segue para o corpo do laço.
  - Interrompe a sequência de execuções dentro do laço e verifica a condição.
- No caso do “for”, o desvio é feito para o incremento, seguido pelo teste condicional e corpo do laço.
  - Interrompe a sequência de execuções dentro do laço.
  - Incrementa ou decrementa o contador, verificando a condição.

Estruturas de Repetição

```
1   for valor in range(1,10):
2       if(valor%2):
3           continue
4       print(valor,end=' ')
```

```
2 4 6 8
```

```
1   valor=1
2   while(valor<10):
3       if(valor%2):
4           continue
5       valor+=1
6       print(valor,end=' ')
```

```
1   valor=1
2   while(valor<10):
3       if(valor%2):
4           valor+=1
5           continue
6       valor+=1
7       print(valor,end=' ')
```

Estruturas de Repetição

Não confunda pass, continue e break

- pass permite incluir uma condição sem instruções relacionadas, mas que podem ser incluídas no futuro (placeholder)
- break altera o fluxo de execução do laço que termina quando certa condição é satisfeita.
- continue desvia o fluxo de execução para a próxima iteração.

Estruturas de Repetição