



Aula 10 – Estruturas de repetição

Professor Rodrigo Maciel

Controle de Fluxo

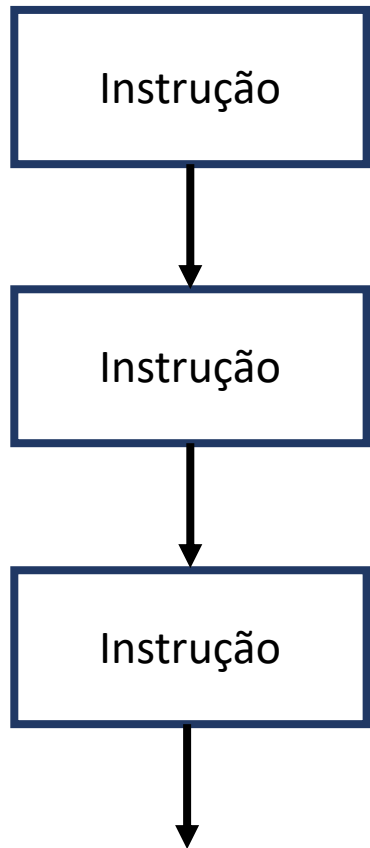
- Com o controle de fluxo, é possível fazer com que o código **tome decisões** ou execute alguns procedimentos **repetidas vezes**.
- Tais ações equivalem a dizer para o compilador instruções como:

"se isso, faça aquilo"

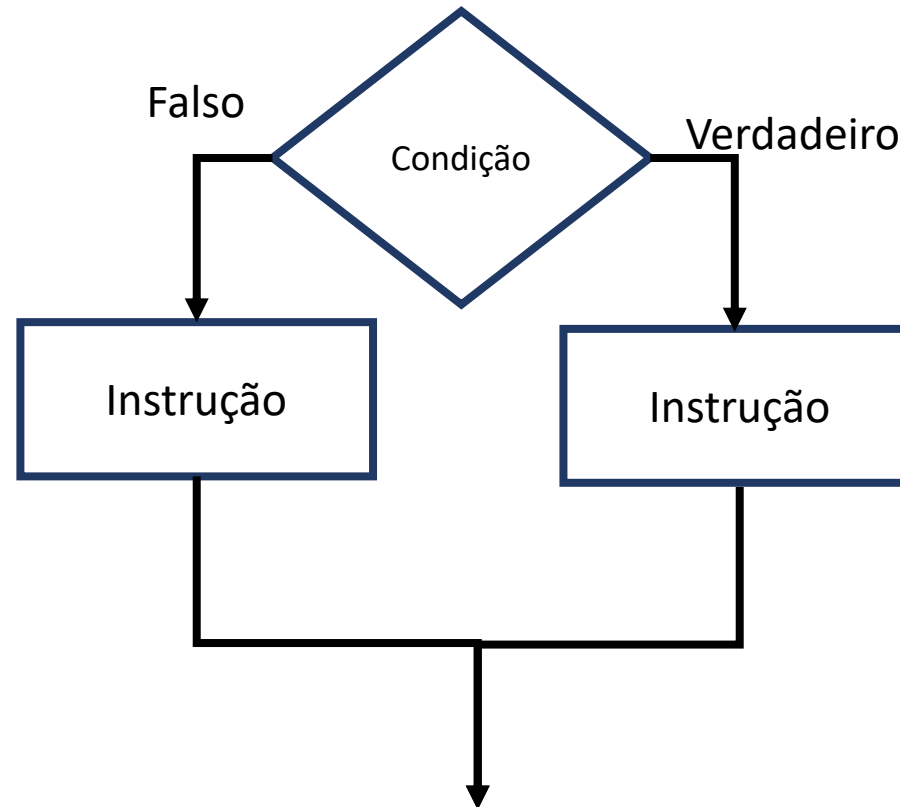
"enquanto isso, execute aquilo"

Controle de Fluxo

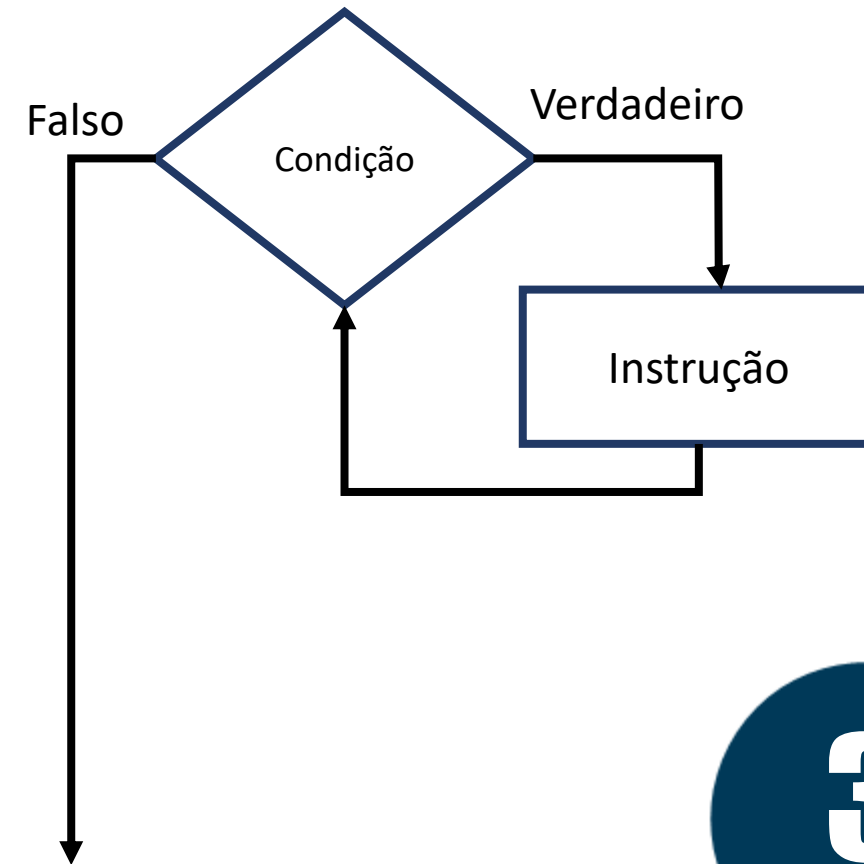
Sequência



Condição



Repetição





Estruturas de repetição em Python

A estrutura de repetição é utilizada para executar uma mesma sequência de comandos várias vezes.

Ela é utilizada para executar uma mesma sequência de comandos várias vezes.

A repetição está associada ou a uma condição , que indica se deve continuar ou não.



Estruturas de repetição em Python

As estruturas de repetição são conhecidas também como laços (*loops*).

Cada repetição é dada o nome de iteração.

Laço **while**

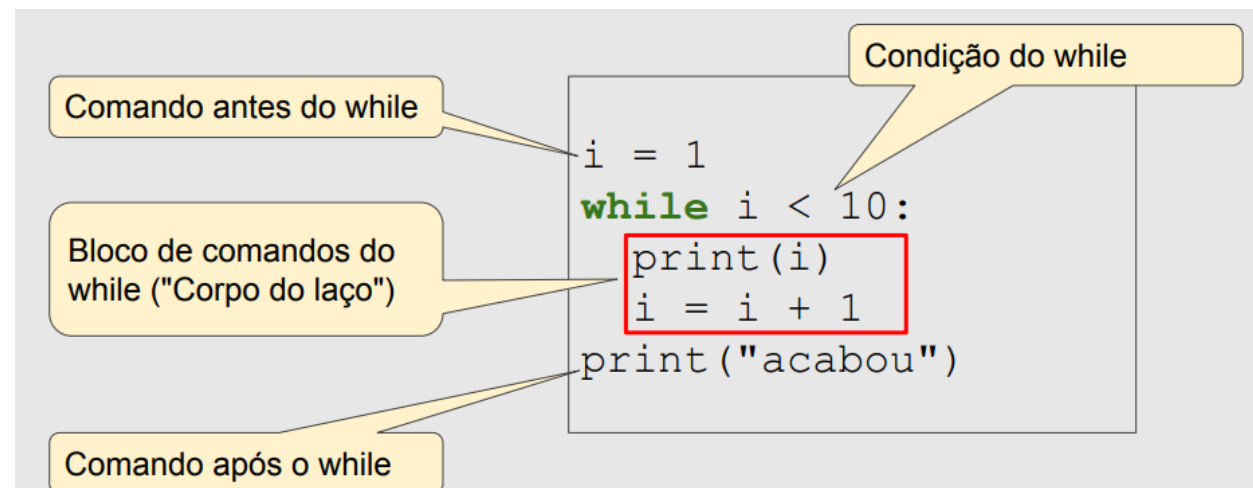
Laço **for**

Estruturas de repetição – Laço While

No laço while o trecho de código da repetição está associado a uma condição.

Enquanto a condição tiver valor verdadeiro, o trecho é executado. Quando a condição passa a ter valor falso , a repetição termina.

`while <condição>:`
`<bloco de comandos>`



Estruturas de repetição – Laço While

```
i = 0  
  
while i<=10:  
    print(i)  
    i += 1
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Estruturas de repetição – Laço While

```
senha = '010203'
usuario = 'yuri'

leituraSenha = ''
leituraUsuario = ''

while((leituraUsuario!=usuario) and (leituraSenha!=senha)):
    leituraUsuario = input('Digite o nome do usuário: ')
    leituraSenha = input('Digite o senha do usuário: ')

    if (leituraUsuario==usuario) and (leituraSenha==senha):
        print('Usuário e Senha correto ! Acesso Liberado...')
    else:
        print('Usuário ou Senha incorretos ! Tente Novamente...')
```

```
Digite o nome do usuário: A
Digite o senha do usuário: 51231
Usuário ou Senha incorretos ! Tente Novamente...

Digite o nome do usuário: yuri
Digite o senha do usuário: 010203
Usuário e Senha correto ! Acesso liberado...
```


Estruturas de repetição – Laço for

Ao invés de avaliar uma condição, como o while, o for, atribui valores a uma variável até que a lista deles acabe.

Pode ser utilizado com uma sequência numérica (gerada com o comando range) ou associado a uma lista.

```
for <variável> in range(início, limite, passo):  
    <bloco de comandos>
```

```
for <variável> in <lista>:  
    <bloco de comandos>
```

```
for x in (0,1,2,3,4):  
    print(x)
```

Estruturas de repetição – Laço for

```
listaUsuarios = ['Pedro', 'Maria', 'Joao', 'Carlos', 'Ronaldo']  
  
for usuario in listaUsuarios:  
    print(usuario)
```

```
Pedro  
Maria  
Joao  
Carlos  
Ronaldo
```

Estruturas de repetição – Laço for

```
listaUsuarios = ['Pedro', 'Maria', 'Joao', 'Carlos', 'Ronaldo']  
listaSenha = ['123456', '5524123', '77423467', '0000000', '52dd23545']  
  
for i in range(len(listaUsuarios)):  
    print('O usuário ({}) e senha ({}).format(listaUsuarios[i], listaSenha[i]))
```

```
O usuário (Pedro) e senha (123456)  
O usuário (Maria) e senha (5524123)  
O usuário (Joao) e senha (77423467)  
O usuário (Carlos) e senha (0000000)  
O usuário (Ronaldo) e senha (52dd23545)
```

Desvios Incondicionais

Os desvios incondicionais são mudanças da execução do programa para outra linha, isto é, desviar o código para outra parte.

Há três comandos de desvio:

break
continue
pass

Desvio Incondicional - Break

O **break**, serve para forçar uma saída do laço de repetição

```
while True:
    print('----Menu de opções----')
    print(' 1 - Cadastrar Usuário')
    print(' 2 - Alterar Usuário')
    print(' 3 - Sair do programa')
    opMenu = int(input('Digite a opção:'))

    if(opMenu == 3):
        break
```

```
In [9]: runfile('C:/Users/yuric/untitled4.py',
----Menu de opções----
      1 - Cadastrar Usuário
      2 - Alterar Usuário
      3 - Sair do programa

Digite a opção:2
----Menu de opções----
      1 - Cadastrar Usuário
      2 - Alterar Usuário
      3 - Sair do programa

Digite a opção:3

In [10]:
```

Desvio Incondicional - Continue

O **continue**, serve para, a partir do momento que é atingindo, pular para a próxima iteração do laço, ignorando os procedimentos abaixo dele

```
listaUsuarios = ['Pedro', 'Maria', 'Joao', 'Carlos', 'Ronaldo']
listaSenha = ['123456', '5524123', '77423467', '0000000', '52dd23545']

for i in range(len(listaUsuarios)):
    if(listaUsuarios[i]=='Maria'):
        continue
    print('O usuário ({}) e senha ({}).format(listaUsuarios[i],listaSenha[i]))
```

```
O usuário (Pedro) e senha (123456)
O usuário (Joao) e senha (77423467)
O usuário (Carlos) e senha (0000000)
O usuário (Ronaldo) e senha (52dd23545)
```

Desvio Incondicional - Pass

O **pass** nada mais é que uma forma de fazer um código que não realiza operação nenhuma.

Usamos o **pass** para dizer ao Python que o bloco de código está vazio.

```
for numero in range(10):  
    pass
```



Praticando Python

Faca um programa que apresente um menu de opções para o cálculo das seguintes operações entre dois números:

- adição (opção 1)
- subtração (opção 2)
- multiplicação (opção 3)
- divisão (opção 4)
- saída (opção 5)

O programa deve possibilitar ao usuário a escolha da operação desejada, a exibição do resultado e a volta ao menu de opções. O programa são termina quando for escolhida a opção de saída (opção 5).