

Estruturas de Repetição



Estrutura de Repetição

- Utilizadas quando é necessário repetir (iterar) um ou vários comandos

Exemplos:

- Exibir 10 vezes o nome “IFPB”;
- Pedir números, enquanto o usuário não informa o valor 0 (zero);
- Exibir os números de 1 até 1000.

≡ Estrutura de Repetição

- De forma geral, a repetição de comandos ocorre considerando
 - uma contagem pré-estabelecida de iterações (**número de repetições é conhecido**), ou
 - a ocorrência (satisfação) de uma determinada condição (**número de repetições é desconhecido**).
- Voltando ao exemplo...
 - Exibir 10 vezes o nome “IFPB”;
 - Pedir números, enquanto o usuário não informa o valor 0 (zero);

≡ Estrutura de Repetição

- As **linguagens de programação** normalmente **suportam** os chamados **laços de repetição (*loops*)**, ou seja, estruturas que permitem executar repetidas vezes uma sequência de comandos
- Em Python...
 - *for*
 - *while*



for

- **Tradicionalmente**, na grande maioria das linguagens de programação, o comando **for** permite executar instruções um **número fixo de vezes**
- Em **Python**, o comando **for** permite **percorrer itens de uma coleção**, e assim, implementar a **repetição** para um **número fixo de vezes**

- **Sintaxe:**

```
for <variável> in <coleção>:  
    comandos  
else:  
    comandos
```

- O **else** no laço de repetição é **opcional**, mas caso seja utilizado, apenas será executado no final da iteração.

≡ A função *range*

- A função ***range*** é frequentemente utilizada com o comando `for` para controle das iterações
- Sintaxe:
 - **`range (start, stop, step)`**
 - **start** - valor inicial da sequência (opcional, default=0)
 - **stop** - valor final (menos 1) da sequência (obrigatório)
 - **step** - intervalo entre os elementos (opcional, default=1)
- **Retorna** uma **série numérica** no intervalo enviado como argumento da função
- A **série** (sequência numérica) **retornada** pela função ***range*** é um **objeto iterável**.

≡ *for*: Exemplos

- Faça um programa que gere uma sequencia numérica de **0** a **3**

Exemplo 1

```
for i in range(4):  
    print(i)
```



≡ *for*: Exemplos

Exemplo 2

```
for i in range(0, 4, 1):  
    print(i)
```



≡ *for*: Exemplos

Exemplo 3

```
for i in range (4, 0, -1):  
    print(i)
```



≡ *for*: Exemplos

Exemplo 4

```
for i in range(2, 11, 2):  
    print(i)
```



≡ *for*: Exemplos

- Vamos construir juntos um programa para:
 - Exibir 10 vezes o nome “IFPB”
 - Exibir os números de 1 a 50
 - Solicitar N números do usuário
 - Solicitar N números do usuário e a cada leitura mostrar o dobro desse número
 - Solicitar N números do usuário e ao final da leitura mostrar a soma de todos os números lidos



while



while

- Executa uma sequência de comandos de acordo com uma condição pré-estabelecida.
- Assim como no for, o **else** é opcional, mas caso seja utilizado, apenas será executado no final da iteração.

- Sintaxe:

while condição :

comandos

else:

comandos

≡ *while*: Exemplos

- Faça um programa que **solicite** do usuário **vários números até que seja informado o número 0**. Sempre que o usuário informar um número o programa deverá mostrar o dobro desse número

Exemplo 1

```
n=int(input("Informe um numero: "))
while (n!=0):
    print ("O dobro de", n, "=", 2*n )
    n=int(input("Informe um numero: "))
else:
    print("O usuario digitou zero")
print("O programa terminou!!!!")
```





while: Exercícios

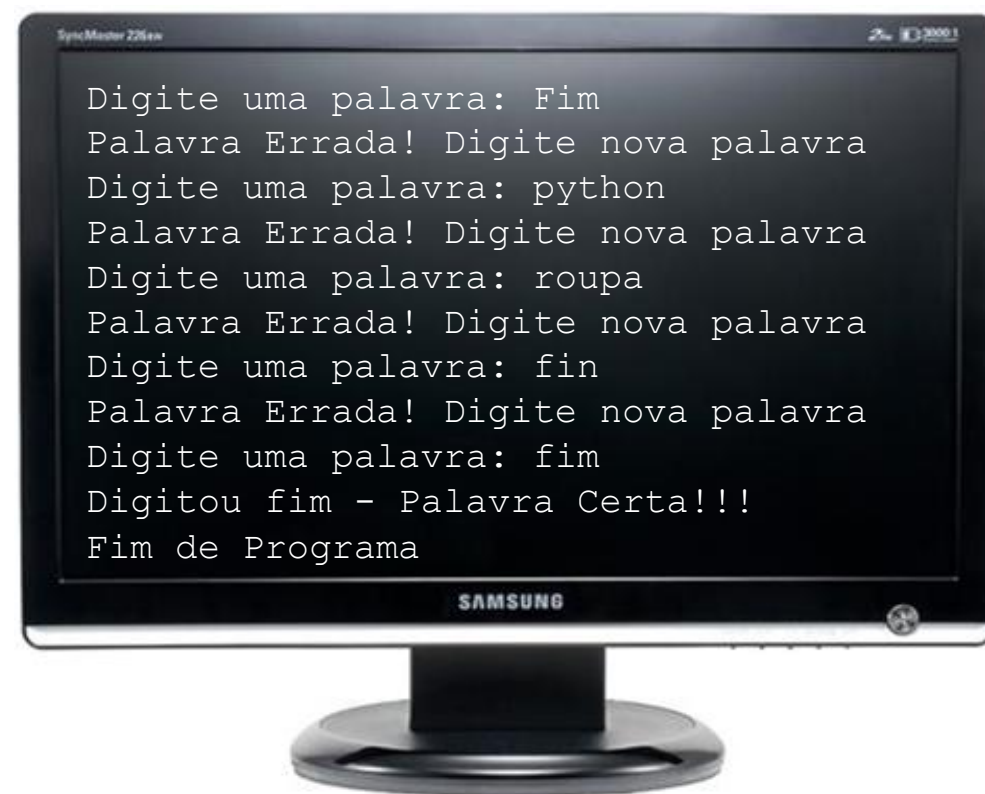
- Vamos construir juntos, **agora usando o comando while**, um programa para:
 - Exibir 10 vezes o nome “IFPB”
 - Exibir os números de 1 a 50
 - Solicitar N números do usuário, verificar e exibir se o número lido é par ou ímpar
 - Solicitar N números do usuário (até que o usuário digite o número zero) e a cada leitura mostrar o dobro desse número
 - Solicitar N números do usuário (até que o usuário digite o número zero ou a soma dos números lidos seja maior do que 100) e ao final da leitura mostrar a soma de todos os números lidos

≡ while: Exemplos

- Faça um programa que **solicite** do usuário **uma palavra**.
 - Se a **palavra** informada for igual a **“fim”** o programa **deverá** **exibir “Digitou fim – Palavra Certa!!!”** e **parar** a sua execução.
 - Caso contrário, o programa **deverá** **exibir “Palavra Errada! Digite nova palavra”** e uma **nova palavra** **deverá** ser solicitada ao usuário.

Exemplo 2

```
while(input("Digite uma palavra: ")!="fim"):
    print("Palavra Errada! Digite nova palavra")
else:
    print("Digitou fim - Palavra Certa!!!")
print("Fim de Programa")
```



≡ *while*: Exemplo usando os comandos aninhados

Faça um programa que **solicite** do usuário **um número** e, **para cada número** informado, **gere sua sequência numérica**. O programa deverá **parar** quando for informado o **número 0**

Exemplo 3

```
n = int(input("Informe um número: "))
while (n != 0):
    for i in range(1, n + 1, 1):
        print(i, end=" ")
    n = int(input("\nInforme um número: "))
else:
    print("saiu")
print("saiu mesmo")
```





break e continue

≡ *break*

- Finaliza um laço de repetição (***for*** ou ***while***) de forma forçada.
- Quando o comando ***break*** é utilizado em um loop (laço), ele terminará o loop e o controle será transferido para fora do corpo do loop.
- Na utilização de ***break*** em loops aninhados, apenas o loop interno será finalizado.
- Se um loop tiver a cláusula ***else*** e for utilizado um ***break*** para sair do loop, então o ***else*** será ignorado.
- Sintaxe:

break



continue

- Interrompe a iteração atual de um loop e o controle é transferido para o início do loop.
- O laço não será terminado, mas continuará com a próxima iteração.
- Tanto o ***break*** como o ***continue*** são comumente usados dentro de um ***if*** com uma condição para que seja executado.
- Sintaxe:
`continue`

≡ Exemplo de *break* e *continue*

- Faça um programa para ler várias idades e calcular a média delas. **Obs:** a leitura da **idade 0** encerra a entrada de dados (**flag**).

Exemplo

```
cont = 0
soma = 0
while True:
    idade = int(input('\nDigite a idade (ou 0 p/encerrar): '))
    if idade == 0:
        break
    if idade < 0:
        print('Idade inválida. Digite novamente...')
        continue
    cont += 1
    soma += idade
media = soma / cont
print(f'\nMédia das idades = {media:.0f}')
```

