Python - Aula 5

Estruturas de controle de fluxo II

Python - Estruturas de Repetição

1) WHILE

O comando **WHILE** permite a repetição de um conjunto de instruções enquanto uma determinada condição for **VERDADEIRA**.

Python - Estruturas de Repetição

1) WHILE

O comando **WHILE** permite a repetição de um conjunto de instruções enquanto uma determinada condição for **VERDADEIRA**.

Quando a condição for **FALSA**, o execução é interrompido, saindo assim do loop e passando para o próximo **bloco** de código.

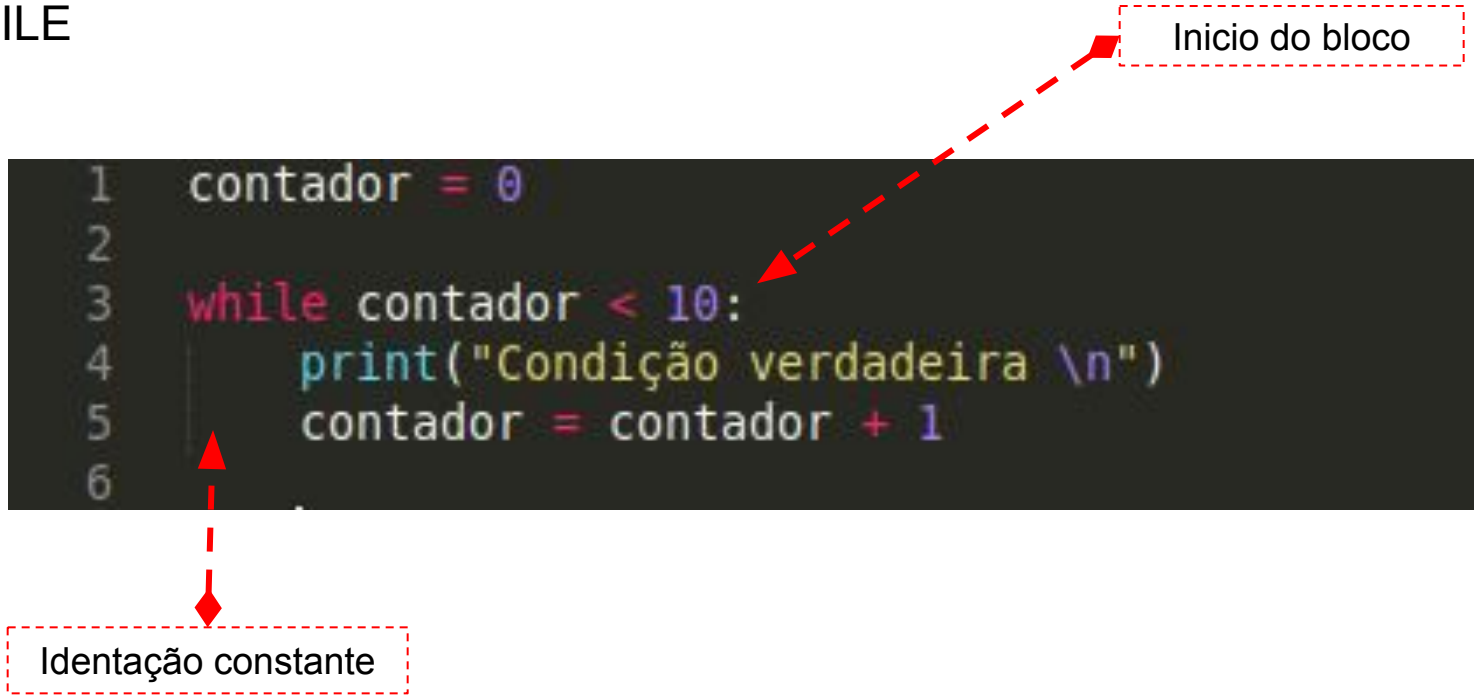
Python - Estruturas de Repetição

1) WHILE

```
1  contador = 0
2
3  while contador < 10:
4      print("Condição verdadeira \n")
5      contador = contador + 1
6
```

Python - Estruturas de Repetição

1) WHILE



Python - Estruturas de Repetição

1) WHILE

EXEMPLO:

```
1  contador = 0
2
3  while contador < 10:
4      print("Condição verdadeira")
5      contador = contador + 1
6      print("Contador: %d \n" % contador)
7
```

Python - Estruturas de Repetição

#EXERCÍCIOS

1) Faça um programa que calcule o número médio de alunos por turma. Para isto, peça a quantidade de turmas e a quantidade de alunos para cada turma. As turmas não podem ter mais de 40 alunos.

Python - Estruturas de Repetição

#EXERCÍCIOS

1) Faça um programa que calcule o número médio de alunos por turma. Para isto, peça a quantidade de turmas e a quantidade de alunos para cada turma. As turmas não podem ter mais de 40 alunos.

2) Faça um programa que leia um nome de usuário e a sua senha e não aceite a senha igual ao nome do usuário. Mostre uma mensagem de erro e volte a pedir as informações.

Python - Estruturas de Repetição

2) FOR

O **FOR** permite percorrer todos os itens de uma coleção, executando o bloco de instruções uma vez para cada um deles.

Python - Estruturas de Repetição

2) FOR

O **FOR** permite percorrer todos os itens de uma coleção, executando o **bloco** de instruções uma vez para cada um deles.

Não é necessário um ponto de parada. O **FOR** encerra após a execução do **bloco** para o último item da coleção.

Python - Estruturas de Repetição

2) FOR

```
1  nomes = ["Fulano", "Beltrano", "Sicrano"]
2
3  for nome in nomes:
4      print(nome)
5
```

Python - Estruturas de Repetição

2) FOR

```
1  nomes = ["Fulano", "Beltrano", "Sicrano"]
2
3  for nome in nomes:
4      print(nome)
5
```

Python - Estruturas de Repetição

3) CONTINUE e BREAK

O **CONTINUE** é utilizado para **iniciar** imediatamente a próxima volta do loop.

Python - Estruturas de Repetição

3) CONTINUE e BREAK

O **CONTINUE** é utilizado para **iniciar** imediatamente a próxima volta do loop.

O **BREAK** é utilizado para **encerrar** imediatamente o loop.

Python - Estruturas de Repetição

3) **CONTINUE** e BREAK

```
1  nomes = ["Fulano", "Beltrano", "Sicrano"]
2
3  for nome in nomes:
4      if nome == "Beltrano":
5          continue
6      else:
7          print(nome)
8
```

Python - Estruturas de Repetição

3) CONTINUE e **BREAK**

```
1  nomes = ["Fulano", "Beltrano", "Sicrano"]
2
3  for nome in nomes:
4      if nome == "Beltrano":
5          break
6      else:
7          print(nome)
8
```

Python - Estruturas de Repetição

4) ENUMERATE

O **ENUMERATE** cria um índice numérico para cada um dos elementos de uma coleção.

Python - Estruturas de Repetição

4) ENUMERATE

```
1  nomes = ["Fulano", "Beltrano", "Sicrano"]
2
3  for chave, nome in enumerate(nomes):
4      print(chave, nome)
5
```

Python - Estruturas de Repetição

#EXERCÍCIOS

3) Faça um programa que receba o nome de um aluno e 3 notas referentes as avaliações feitas por este. O programa deve continuar recebendo informações até que o nome do aluno esteja em branco. Ao final, o programa deve mostrar um relatório de cada aluno contendo:

a) O nome do aluno;

b) As notas do aluno;

c) A média do aluno.