

# Introdução ao Python

## Estruturas de repetição

Henrique Y. Shishido

Departamento de Computação  
Universidade Tecnológica Federal do Paraná

# Estrutura de repetição

---

- Uma estrutura de repetição permite executar mais uma vez um conjunto de instruções até que uma condição seja satisfeita
- O Python oferece dois comandos de repetição:
  - `while`
  - `for`

# Estrutura de repetição - while

---

- O comando `while` permite executar um conjunto de instruções enquanto uma condição é satisfeita.
  - Exemplo: imprimir o valor da variável `count` enquanto ela for menor que 10

```
1 count=0;
2 while(count < 10):
3     print(count)
4     count += 1
5
6 print("Print concluido")
```

# A instrução break

---

- A instrução break interrompe o laço de repetição mesmo se a condição do laço while seja verdadeira

```
1 count=0;
2 while(count < 10):
3     print(count)
4     if(count == 4)
5         break;
6     count += 1
7
8 print("Print concluido")
```

# A instrução continue

---

- A instrução continue interrompe a execução da iteração atual

```
1 count=0;
2 while(count < 10):
3     count += 1
4     if(count == 4)
5         continue;
6     print(count)
7
8 print("Print concluido")
```

# A instrução else

---

- A instrução else especifica um bloco de código a ser executado quando a condição de repetição seja falsa

```
1 count=0;
2 while(count < 10):
3     print(count)
4     count += 1
5 else:
6     print("count não é mais menor que 10")
```

# Estrutura de repetição - for

---

- O comando `for` é usado para iterar sobre uma lista, uma tupla, um dicionário, um conjunto ou uma string
- O `for` no Python é um pouco diferente de outras linguagens de programação. Ele se assemelha a um iterador encontrado em linguagens orientadas a objetos
- O comando `for` permite executar um conjunto de instruções para cada item em uma lista, tupla, conjunto, etc.

# Estrutura de repetição - for

---

## Iterando em elementos de uma lista

```
1 produtos = ["notebook", "mesa", "smartphone"]  
2 for p in produtos:  
3     print("Produto: " + p)
```

## Iterando em uma string

```
1 minha_string = "ciência de dados"  
2 for c in minha_string:  
3     print(c)
```



# Instrução break

---

- A instrução break permite interromper o laço mesmo sem passar por todos os elementos da lista, conjunto, string, etc.

```
1 paises = ["Brasil", "Argentina", "Colômbia"]
2 for p in paises:
3     print(p)
4     if(p == "Brasil"):
5         break
```

# Instrução continue

---

- A instrução continue interrompe somente a iteração atual e continua para a próxima iteração

```
1 paises = ["Brasil", "Argentina", "Colômbia"]
2 for p in paises:
3     if(p == "Brasil"):
4         continue
5     print(p)
```

# A função `range()`

---

- Para iterar sobre um conjunto de instruções por  $N$  vezes, pode-se utilizar a função `range()`
- Essa função retorna uma sequência de números partindo do 0 (zero) como padrão e os incrementa de 1 em 1

Chamada padrão da função. Retorna uma sequência de 0 a 9

```
1 for i in range(10):  
2     print(i)
```

# A função range()

---

Retorna uma sequência de 5 (inclusivo) a 10 (exclusivo)

```
1 for i in range(5,10):  
2     print(i)
```

Retorna uma sequência de 0 (inclusivo) a 10 (exclusivo) com incremento de 2 em 2

```
1 for i in range(0,10,2):  
2     print(i)
```

# Instrução else

---

- O comando else em um laço for especifica um bloco de instruções a ser executado quando o laço é concluído

Retorna uma sequência de 0 (inclusivo) a 10 (exclusivo) com incremento de 2 em 2

```
1 for i in range(10):  
2     print(i)  
3 else:  
4     print("Laço concluído")
```

# Instrução pass

---

- Estruturas de repetição não podem ser vazias. Caso precise criar uma estrutura sem nenhuma instrução, adicione o comando `pass` para evitar erros

Estrutura de repetição vazia

```
1 for i in range(10):  
2     pass
```