

Python - Strings



Disciplina: Programação para Internet

Prof. Bruno Gomes



@profbrunogomes



Aula de Hoje

2

- » **Strings**
- » **Funções**
- » **Substrings**
- » **Formatação**



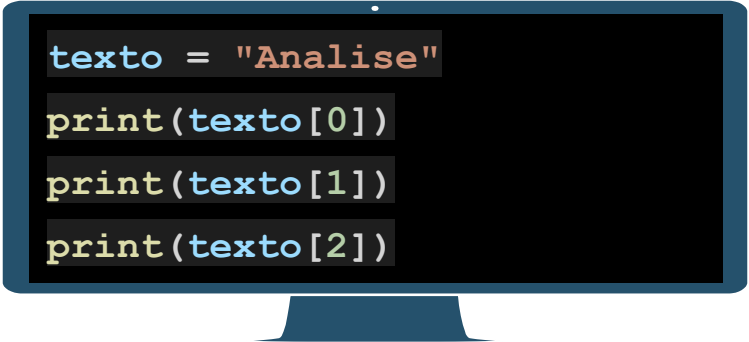
1.

Introdução

String

4

- » É uma sequência de caracteres (lista);
- » Em Python, podemos acessar diretamente cada caractere desta sequência através do índice e colchetes.




```
texto = "Analise"  
print(texto[0])  
print(texto[1])  
print(texto[2])
```

Strings - Concatenação

5

- » + - concatena textos (não adiciona espaço entre elas)




```
texto = "Analise "  
print(texto + "de dados")
```

Strings - Concatenação

6

- » , - usada na função para concatenar também (como se fosse parâmetros), Adiciona espaço automaticamente entre elas.



```
texto = "Analise"  
print(texto, "de dados")
```

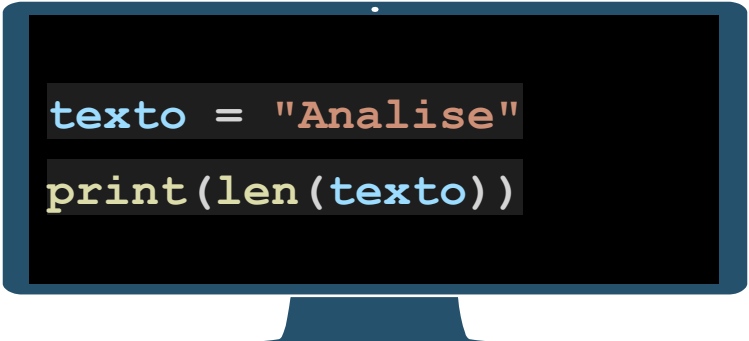
2.

Funções

Strings - Funções

8

- » **len()** - retorna o tamanho da String



```
texto = "Analise"  
print(len(texto))
```


Strings - Funções

9

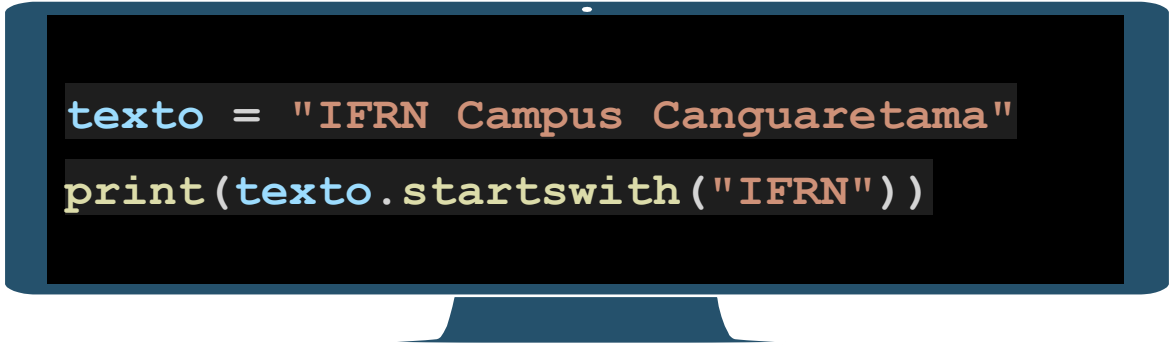
- » **replace()** - substitui uma substring por outro texto

```
texto = "IFRN"  
novo_texto = texto.replace("IFRN", "Instituto Federal do RN")  
print(novo_texto)
```

Strings - Funções

10

- » **startswith()** - verifica se inicia com algum texto

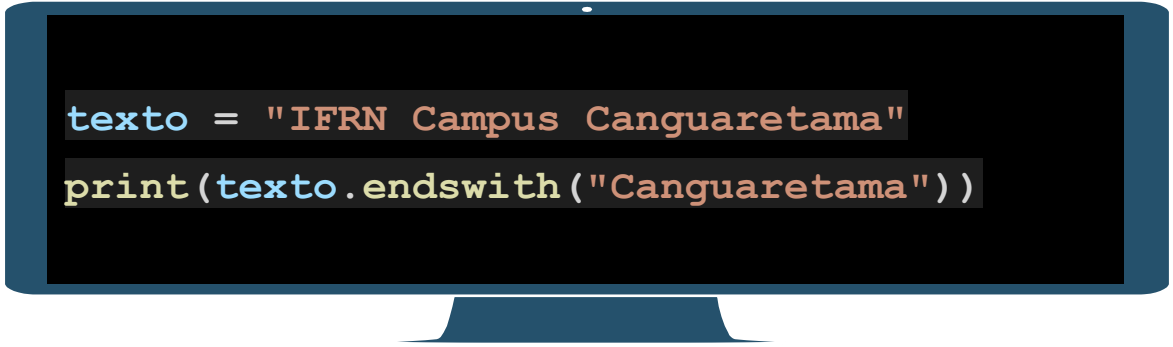


```
texto = "IFRN Campus Canguaretama"  
print(texto.startswith("IFRN"))
```

Strings - Funções

11

- » **endswith()** - verifica se termina com algum texto




```
texto = "IFRN Campus Canguaretama"  
print(texto.endswith("Canguaretama"))
```

Strings - Funções

12

- » **capitalize()** - Transforma primeira letra em maiúscula



```
texto = "campus canguaretama"  
print(texto.capitalize())
```

Strings - Funções

13

- » **isdigit()** - Verifica se uma string só possui números



```
texto = "1234"  
print(texto.isdigit())
```

Strings - Funções

14

- » **isalnum()** - Verifica se uma string é alfanumérica



```
texto = "1234abc"  
print(texto.isalnum())
```

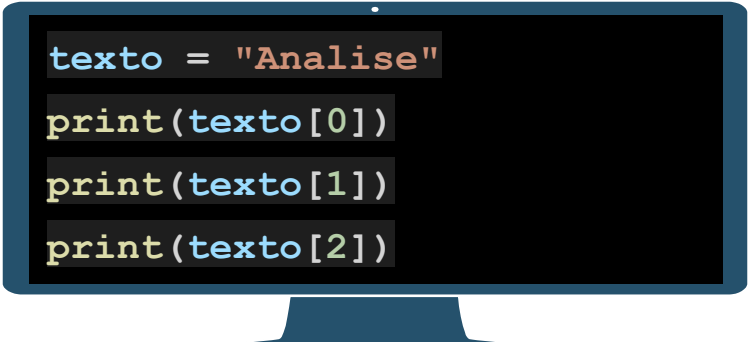
3.

Substring

Substrings

16

- » [] - usa colchetes para acessar caracteres específicos:
 - » Sempre inicia em 0;

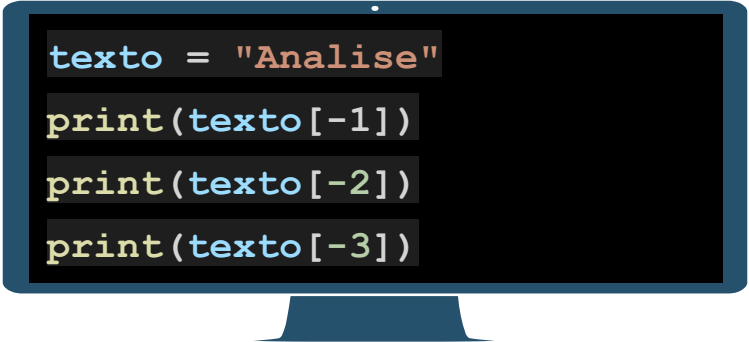


```
texto = "Analise"  
print(texto[0])  
print(texto[1])  
print(texto[2])
```

Substrings

17

- » [] - usa colchetes para acessar caracteres específicos:
 - » Para acessar em ordem inversa, inicia em -1, depois em -2, e assim por diante.



```
texto = "Analise"  
print(texto[-1])  
print(texto[-2])  
print(texto[-3])
```

Slices - fatias

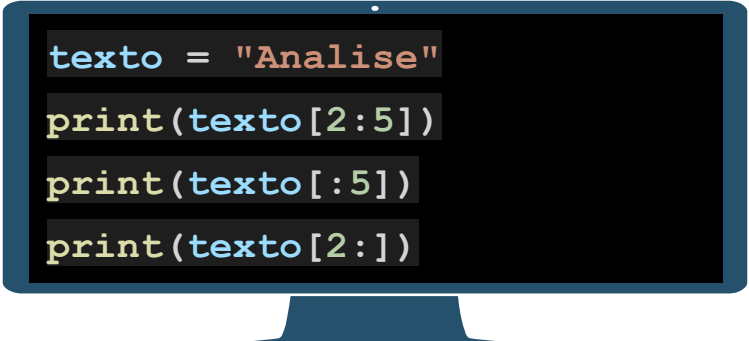
18

- » [início:fim:salto] - Sintaxe para fatiar uma String
 - » início: primeiro índice
 - » fim: último índice
 - » salto: quantos caracteres devem ser saltados (1 é o padrão, se colocar número negativo será lido ao contrário).

Slices - fatias

19

- » **[início : fim]** - acessa uma fatia da String informando o índice de início e o de fim:

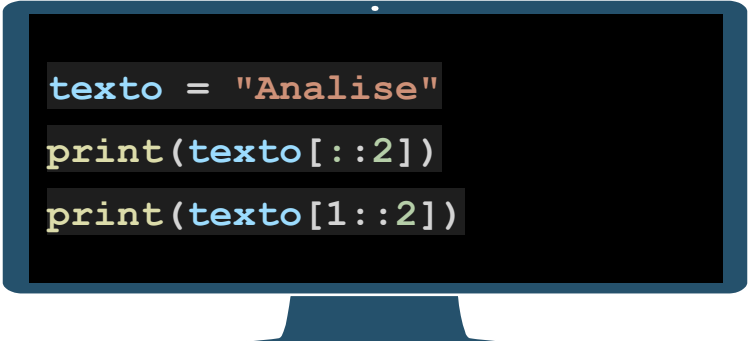


```
texto = "Analise"  
print(texto[2:5])  
print(texto[:5])  
print(texto[2:])
```

Slices - fatias

20

- » `::2` - retorna os caracteres nos índices pares
- » `1::2` - retorna os caracteres nos índices ímpares

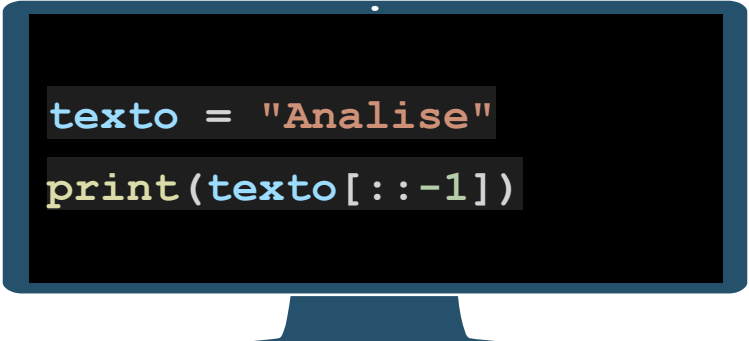


```
texto = "Analise"  
print(texto[:2])  
print(texto[1::2])
```

Slices - fatias

21

- » `[::-1]` - Inverte a String

A stylized illustration of a computer monitor with a dark blue frame and a black screen. The screen displays two lines of Python code in a monospaced font with syntax highlighting: the first line is `texto = "Analise"` and the second line is `print(texto[::-1])`.

```
texto = "Analise"  
print(texto[::-1])
```

4.

Formatação

Slices - fatias

23

- » **format()** e **{ }** - serve para formatar e substituir valores na String

```
texto = 'Aluno possui {0} anos e nasceu em {1}'  
print(texto.format(22, 2020))
```

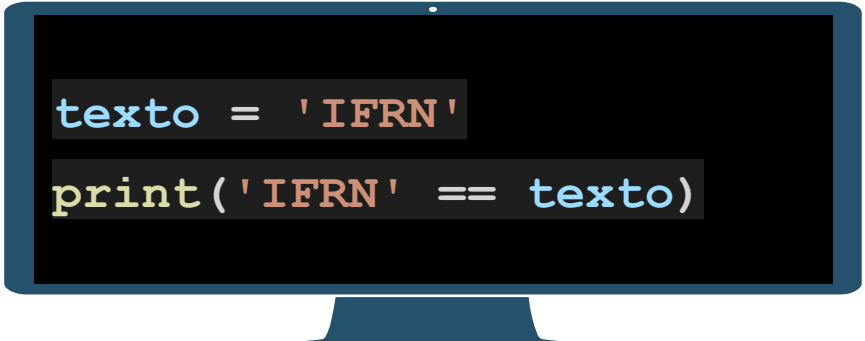
5.

Comparação

Slices - fatias

25

- » == - utiliza o operador de comparação



```
texto = 'IFRN'  
print('IFRN' == texto)
```

Dúvidas?



Vamos Praticar!!

Lista de Exercícios

