



**TREINA RECIFE**  
DESENVOLVIMENTO GERENCIAL E PROFISSIONAL

*Prof. ME Rogério Aguiar C. Teixeira*

# Curso de Lógica de Programação

## Parte 04





# Quem sou eu?

**CEO and Founder da Empresa Treina Recife**  
**Mestre em Administração – UFPE**  
**Especialista em Gestão da TIC**  
**Especialista em Análise e Desenvolvimento de Sistemas**  
**Graduado em Matemática**  
**Analista de Sistemas da Emprer**  
**Consultor Sênior na Gestão Organizacional e de TI**  
**Instrutor de Lógica e SQL na Treina Recife**  
**Instrutor Cursos Executivos**  
**Professor de Pós-Graduação na FAFIRE, UNICAP e no SENAI-PE**



**Prof. ME Rogério Aguiar C. Teixeira**



@prof\_rogerio.aguiar



+55 81 9.9247-4667



[rogerio.aguiar@treinarecife.com.br](mailto:rogerio.aguiar@treinarecife.com.br)



<http://lattes.cnpq.br/1025838649183053>



<http://br.linkedin.com/pub/rogerio-aguiar-teixeira/23/634/28a>



**TREINA RECIFE**  
DESENVOLVIMENTO GERENCIAL E PROFISSIONAL

Fatiamento de Textos(str)



# Fatiamento de Textos(str)

Dentre as várias operações que podemos fazer com o conteúdo de uma variável **str** temos:

Variável s na memória RAM										
0	1	2	3	4	5	6	7	8	9	10
a	b	c	_	1	2	3	_	X	Y	Z

**s = 'abc\_123\_XYZ'**

Sintaxe	Efeito- Traz:	Exemplo	Resultado
<b>len(variável_str)</b>	O tamanho inteiro do conteúdo da variável	<b>print(len(s))</b>	<b>11</b>
<b>variável_str[pos]</b>	O conteúdo em determinada posição	<b>print(s[4])</b> <b>print(s[9])</b>	<b>1</b> <b>Y</b>
<b>variável_str[pos1:pos2]</b>	O conteúdo em um intervalo <b>pos1</b> até <b>pos2-1</b> [inicio:final[	<b>print(s[0:3])</b> <b>print(s[4:7])</b>	<b>abc</b> <b>123</b>
<b>variável_str[pos1:]</b>	O conteúdo a partir da <b>pos1</b> até o final do texto	<b>print(s[4:])</b>	<b>123_XYZ</b>
<b>variável_str[:pos2]</b>	O conteúdo a partir do início da string até <b>pos2-1</b> [inicio:final[	<b>print(s[:7])</b> <b>print(s[:3])</b>	<b>abc_123</b> <b>abc</b>
<b>variável_str[-n:]</b>	O conteúdo dos <b>n</b> últimos caracteres	<b>print(s[-3:])</b>	<b>XYZ</b>
<b>variável_str[:-n]</b>	O conteúdo sem os <b>n</b> últimos	<b>print(s[:-2])</b>	<b>abc_123_X</b>
<b>variável_str[::-1]</b>	O conteúdo invertido	<b>Print(s[::-1])</b>	<b>ZYX_321_cba</b>



## Fatiamento de Textos(str) - Relembrando

Suponha a variável `s = 'abc_123_XYZ'`



0	1	2	3	4	5	6	7	8	9	10
a	b	c	_	1	2	3	_	X	Y	Z

```
print(s[9])      # Y      - Item na posição 9
print(s[0:3])    # abc    - Da posição 0 até a 2 [0,3[
print(s[4:])     # 123_XYZ - Da posição 4 até o final
print(s[:7])     # abc_123 - Da posição 0 até a 6 [0,7[
print(s[:-2])    # abc_123_X - Retira os últimos 2
print(s[-3:])    # XYZ     - Os três últimos
print(s[::-1])   # zyx_321_cba - Inverte o nome
```



## Métodos da class str

0	1	2	3	4	5	6	7	8	9	10
a	b	c	_	1	2	3	_	X	Y	Z

Suponha `s = 'abc_123_XYZ'`

Método	Efeito	Exemplo	Resultado
<b>s.lower()</b>	Retorna um string com todas as letras <b>minúsculas</b>	<code>print(s.lower())</code>	<code>abc_123_xyz</code>
<b>s.upper()</b>	Retorna um string com todas as letras <b>maiúsculas</b>	<code>print(s.upper())</code>	<code>ABC_123_XYZ</code>
<b>s.title()</b>	Retorna uma string com a <b>primeira letra maiúscula</b> para cada sequência de letras	<code>print(s.title())</code>	<code>Abc_123_xyz</code>
<b>s.find('texto')</b>	Pesquisa um substring em s e retorna um número inteiro, indicando a posição se encontrar, ou -1 caso não encontre.	<code>print(s.find('123'))</code> <code>print(s.find('m'))</code>	4 -1
<b>s.replace('_', '*')</b>	Substitui no texto o primeiro caractere pelo segundo, caso exista	<code>print(s.replace('_', '*'))</code>	<code>abc*123*XYZ</code>



# Métodos da class str

0	1	2	3	4	5	6	7	8	9	10
a	b	c	_	1	2	3	_	X	Y	Z

Suponha s = 'abc\_123\_XYZ'

Método	Efeito	Exemplo	Resultado
<b>s.isalpha()</b>	Retorna <b>True</b> caso a string contenha <b>apenas letras</b> , se não <b>False</b>	print(s.isalpha())	False
<b>s.isnumeric()</b>	Retorna <b>True</b> caso a string contenha <b>apenas números</b> , se não False	print(s.isnumeric())	False
<b>s.split('_')</b>	Retorna uma lista de strings, utilizando o caractere passado como delimitador da separação	print(s.split('-'))	['abc', '123', 'XYZ']
<b>s.count('texto')</b>	Retorna quantas vezes a string 'texto' aparece no texto pesquisado, podemos delimitá-lo <b>s.count('X', inicio, fim)</b>	print(s.count('X'))	1



# Exemplo

Faça um programa que receba uma data em uma variável tipo **str** no formato "DD/MM/AAAA" e calcule a soma do DD+MM+AAAA e exiba na tela.

0	1	2	3	4	5	6	7	8	9
D	D	/	M	M	/	A	A	A	A

Teste	Exemplos de Entrada	Exemplos de Saída
1°	data = "22/05/2020"	soma = 2047
2°	data = "30/06/2020"	soma = 2056
3°	data = "02/12/1964"	soma = 1978

Entrada	<i>data</i>
Saída	<i>soma = ?</i>
Auxiliares	<i>dia , mes , ano</i>
Processamento	
<i>Usar os métodos da classe <b>str</b> para separar <b>dia</b>, <b>mes</b> e <b>ano</b>, logo após somar os três e atribuir na variável <b>soma</b></i>	

```
data = input('Informe uma data no formato DD/MM/AAAA: ')
dia = int(data[0:2]) → Inicia na posição 0 e vai até a posição 1
mes = int(data[3:5]) → Inicia na posição 3 e vai até a posição 4
ano = int(data[6:]) → Inicia na posição 6 e vai a até a posição final
soma = dia + mes + ano
print( 'A soma calculada foi:', soma)
```





## Exercício

**Ex31 – Operação\_Frase:** Faça um programa que receba uma frase qualquer informada via teclado e escreva no vídeo o que se segue:

- a) Os Primeiros 5(cinco) caracteres da frase;
- b) Os últimos 5(cinco) caracteres da frase;
- c) Os primeiros 5(cinco) caracteres da frase invertidos;
- d) A quantidade de caracteres contidos na frase;

Teste	Exemplos de Entrada	Exemplos de Saída
1°	frase = 'Curso da Treina Recife'	rA = 'Curso' rB = 'ecife' rC = 'osruC' rD = 22
2°	frase = 'Aprendendo Python'	rA = 'Apren' rB = 'ython' rC = 'nerpA' rD = 17



## Exercício

**Ex32** - Receba um nome com, no mínimo, 10 caracteres via teclado. Com ele você deve gerar uma senha no seguinte formato: dividir o nome recebido em duas partes e concatenar os dois primeiros caracteres da segunda parte, com os caracteres “%%” e os três últimos caracteres da primeira parte. Exiba ao final a parte1, a parte 2 e a senha.

Teste	Exemplos de Entrada	Exemplos de Saída
1°	nome = 'TREINA_RECIFE' 0123456789112 012	parte1 = 'TREINA' parte2 = '_RECIFE' ' Senha = '_R%%INA'
2°	frase = 'JOSEFINA_SILVA' 01234567891111 0123	parte1 = 'JOSEFIN' parte1 = 'A_SILVA' senha = 'A_%%FIN'



## Exercício

**Ex33** - Elabore um programa que permita receber uma frase via teclado e exibir no vídeo a quantidade de letras "a" existentes na frase e a frase recebida invertida

Teste	Exemplos de Entrada
1°	frase = 'Aulas de lógica com Python para iniciantes'
	Exemplos de Saída
	qtd_A = 6 frase_Inv = 'setnaicini arap nohtyP moc acigól ed saluA'

Teste	Exemplos de Entrada
2°	frase = 'Exercitando fatiamento de texto'
	Exemplos de Saída
	qtd_A = 3 frase_Inv = 'otxet ed otnemaitaf odnaticrexE'



## Exercício

**Ex34** - Elabore um programa que receba 50 frases via teclado e ao final exiba a maior frase digitada.

Teste	Exemplos de Entrada
1°	frase = 'Aulas de lógica com Python para iniciantes' frase = 'Exercitando fatiamento de texto' frase = 'Eu vou ficar expert após este curso' Frase = 'Não se esqueça o segredo está nos exercícios' frase = 'Curso Python da Treina Recife'
	Exemplos de Saída
	maior_Frase = 'Não se esqueça o segredo está nos exercícios'

---

*Obrigado!*

**Prof. ME Rogério Aguiar C. Teixeira**

+55 81 9.9247-4667

@prof\_rogerio\_aguiar

@treina\_recife

[rogerio.aguiar@treinarecife.com.br](mailto:rogerio.aguiar@treinarecife.com.br)

<http://br.linkedin.com/pub/rogerio-aguiar-teixeira/23/634/28a>