

**Instituto de
Computação**

UNIVERSIDADE ESTADUAL DE CAMPINAS



MC102 - Aula 09

Strings

Algoritmos e Programação de Computadores

Turmas

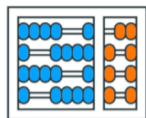
OVXZ

Prof. Lise R. R. Navarrete

lrommel@ic.unicamp.br

Terça-feira, 19 de abril de 2022

21:00h - 23:00h (CB06)



**Instituto de
Computação**

UNIVERSIDADE ESTADUAL DE CAMPINAS



UNICAMP

MC102 – Algoritmos e Programação de Computadores

Turmas

OVXZ

<https://ic.unicamp.br/~mc102/>

Site da Coordenação de MC102

Aulas teóricas:

Terça-feira, 21:00h - 23:00h (CB06)

Quinta-feira, 19:00h - 21:00h (CB06)

Conteúdo

- Strings
 - Formatação de Strings
 - Operações com Strings
 - Exemplos usando Strings

Strings

- Uma String é uma sequência de caracteres.
- Em Python, Strings são representadas como listas imutáveis de caracteres.
- Podemos representar uma String como uma sequência de caracteres entre aspas simples (') ou aspas duplas (").
- Exemplo:

```
1 msg = "hello world"  
2 print(msg)  
3 # hello world
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O seguinte trecho de código apresenta erros.

```
1 print("Você respondeu "SIM".")
2 # SyntaxError: invalid syntax
3 print("\")
4 # SyntaxError: EOL while scanning string literal
```

- Isso acontece porque " e \ são caracteres reservados da linguagem.
- Para representar os caracteres " e \ precisamos utilizar o seguinte código.

```
1 print("Você respondeu \"SIM\".")
2 # Você respondeu "SIM".
3 print("\\")
4 # \
```

- Outros exemplos:

```
1 print("Joe\'s Car")
2 # Joe's Car
3 print("Quebra de\nLinha")
4 # Quebra de
5 # Linha
6 print("\tTabulação")
7 #         Tabulação
```

- Por padrão, um <tab> (\t) tem tamanho equivalente a oito espaços em branco.

- Podemos acessar elementos de uma String como acessamos elementos de uma lista.

```
1 msg = "hello world"
2 print(msg[0])
3 # h
4 print(msg[1])
5 # e
6 print(msg[-1])
7 # d
8 print(msg[-5])
9 # w
10 print(msg[12])
11 # IndexError: string index out of range
```


<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Como Strings são listas imutáveis (assim como as tuplas), não é possível alterar uma posição da String.

```
1 msg = "hello world"
2 msg[0] = "y"
3 # TypeError: 'str' object does not support item assignment
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Também podemos selecionar um trecho de uma String utilizando `string[start:stop:step]`.
- O trecho inicia na posição `start` (inclusive) e vai até a posição `stop` (exclusive), selecionando de `step` em `step` caracteres.
- Caso o parâmetro `step` não seja especificado, Python utilizará o valor 1 como padrão (assim como em listas ou tuplas).

```
1 msg = "hello world"
2 print(msg[3:8])
3 # lo wo
4 print(msg[:5])
5 # hello
6 print(msg[6:])
7 # world
8 print(msg[::-2])
9 # hlowrd
10 print(msg[::-1])
11 # dlrow olleh
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi



Run >

Result Size: 421 x 402

Get your website

```
a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.
```

```
"""
```

```
print(a)  
print("End")
```

```
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.
```

End

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi



Run >

Result Size: 421 x 402

Get your website

```
a = '''Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.
```

```
...
```

```
print(a)  
print("End")
```

```
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.
```

End



Run >

Result Size: 421 x 402

Get your website

```
a = '''Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

```
...
```

```
print(['',a, '' ])
```

```
...
```

```
Comentarios 1
```

```
'''
```

```
"""
```

```
Comentarios 2
```

```
"""
```

```
print("End")
```

```
[ Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

```
.
]
End
```



Run >

Result Size: 421 x 402

Get your website

```
a = '''Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

```
...
```

```
print(['',a, ''])
```

```
...
```

```
Comentarios 1
```

```
'''
```

```
"""
```

```
Comentarios 2
```

```
"""
```

```
print(['', "End", ""])
```

```
[Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

```
.
]
[End]
```



Run >

Result Size: 411 x 402

Get your website

```
a = "Hello, World!"  
print(a[1])  
  
print(10*"-")  
  
for x in "banana":  
    print(x)
```

```
e  
-----  
b  
a  
n  
a  
n  
a
```



Run >

Result Size: 411 x 402

Get your website

```
a = "Hello, World!"
print(a[1])

print(10*" - ")

for x in a:
    print(x)

print(10*" - ")

print(len(a))
```

```
e
-----
H
e
l
l
o
,
W
o
r
l
d
!
-----
13
```




Run >

Result Size: 411 x 402

Get your website

```
# Check String
txt = "The best things in life are free!"
print("free" in txt)

print("-----")

txt = "The best things in life are free!"
if "free" in txt:
    print("Yes, 'free' is present.")

print("-----")

# Check if NOT
txt = "The best things in life are free!"
print("expensive" not in txt)

print("-----")

txt = "The best things in life are free!"
if "expensive" not in txt:
    print("No, 'expensive' is NOT present.")
```

```
True
-----
Yes, 'free' is present.
-----
True
-----
No, 'expensive' is NOT present.
```



Run >

Result Size: 411 x 402

Get your website

Slicing

`b = "Hello, World!"``print(b[2:5])``print(b[:5])``print(b[2:])``print(b[-5:-2])`

```
llo
Hello
llo, World!
orl
```

Formatação de Strings

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Uma das formas de formatar uma String é utilizando a função `format`.
- A função `format` recebe como parâmetros um valor e uma string com a formatação desejada.
- Como resposta a função retorna uma string formatada.
- A string de formatação possui uma especificação para cada tipo de dados.
- Especificação completa em:
<https://docs.python.org/3/library/string.html#formatspec>
- Iremos focar na formatação de valores do tipo inteiro (`int`) e real (`float`) em Strings.

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatando um número inteiro:

```
1 print(format(10, "d"))  
2 # 10
```

- Formatando um número inteiro com sinais:

```
1 print(format(13, "+d"))  
2 # +13  
3 print(format(-7, "+d"))  
4 # -7
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatando um número real:

```
1 print(format(3.14159265359, "f"))  
2 # 3.141593
```

- Formatando um número real com sinais:

```
1 print(format(3.14159265359, "+f"))  
2 # +3.141593
```

- Formatando um número real com sinais e precisão:

```
1 print(format(3.14159265359, "+.10f"))  
2 # +3.1415926536
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Outra forma de formatar uma String é utilizando o método `format`.
- O método `format` gera uma nova String como resposta.
- O método recebe como parâmetros uma sequência de valores que são utilizados para criar a String no formato desejado.
- O método `format` também segue uma especificação para formação de Strings.
- Especificação completa em:
<https://docs.python.org/3/library/string.html#formatstrings>

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatação com Strings:

```
1 frutas = "Frutas: {0}, {1} e {2}"
2 print(frutas.format("abacaxi", "banana", "caqui"))
3 # Frutas: abacaxi, banana e caqui
4 pets = "Quem é mais inteligente: {1} ou {0}?"
5 print(pets.format("gato", "cachorro"))
6 # Quem é mais inteligente: cachorro ou gato?
```


<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatação com números inteiros:

```
1 soma = "{0} + {1} = {2}"
2 print(soma.format(3, 4, 3 + 4))
3 # 3 + 4 = 7
4 valores = "Valor mínimo/médio/máximo: {0}/{1}/{2}"
5 print(valores.format(10, 35, 100))
6 # Valor mínimo/médio/máximo: 10/35/100
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatação com números reais:

```
1 pi = "O valor de pi é: {0:.4f}"
2 print(pi.format(3.14159265359))
3 # O valor de pi é: 3.1416
4 notas = "A média das notas da turma foi {0:.2f}."
5 print(notas.format(8.7525))
6 # A média das notas da turma foi 8.75.
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Formatação com vários tipos de dados:

```
1 cabeçalho = "{0}, {1} de {2} de {3}"
2 print(cabeçalho.format("Campinas", 7, "maio", 2020))
3 # Campinas, 7 de maio de 2020
4 temperatura = "{0:02d}/{1:02d}/{2}: {3:.1f}C"
5 print(temperatura.format(7, 5, 2020, 28.765))
6 # 07/05/2020: 28.8C
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi

Run >

Result Size: 832 x 116

Get your website

```
age = 36

txt = "My name is John, and I am {}"
print(txt.format(age))

quantity = 3
itemno = 567
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."

print(myorder.format(quantity, itemno, price))

myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))
```

```
My name is John, and I am 36
I want 3 pieces of item 567 for 49.95 dollars.
I want to pay 49.95 dollars for 3 pieces of item 567.
```

Operações com Strings

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O operador + concatena duas Strings.

```
1 msg = "hello"
2 msg2 = "y" + msg[1:] + "w"
3 print(msg2)
4 # yellow
```

- O operador * replica uma String.

```
1 s = "abc"
2 print(s * 3)
3 # abcabcabc
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- A função `len` retorna o tamanho (quantidade de caracteres) de uma String.

```
1 msg = "hello"
2 print(len(msg))
3 # 5
4 msg2 = "Hello World"
5 print(len(msg2))
6 # 11
7 msg3 = "Hello\nWorld"
8 print(len(msg3))
9 # 11
```

- Observe que qualquer tipo de caractere é contado pela função `len`, inclusive espaços, quebra de linhas ou tabulações.

- O operador `==` verifica se duas Strings são iguais. <https://ic.unicamp.br/~mc102/aulas/aula06.pdf>
- O operador `!=` verifica se duas Strings são diferentes.

```
1 a = "Python"
2 b = "Py" + "thon"
3 c = "p" + "ython"
4 print(a == b)
5 # True
6 print(a == c)
7 # False
8 print(b != c)
9 # True
```

- O operador `in` verifica se uma String é parte de outra String.

```
1 print("thon" in "Python")
2 # True
3 print("thor" in "Python")
4 # False
```


<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `startswith` verifica se a String recebida como parâmetro é um prefixo da String base.

```
1 msg = "Hello World"
2 print(msg.startswith("Hello"))
3 # True
4 print(msg.startswith("World"))
5 # False
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `index` retorna a primeira posição em que uma String fornecida como parâmetro ocorre na String base.
- Se a String fornecida como parâmetro não está contida na String base, então é gerado um erro (similar ao que ocorre com listas).

```
1 bond = "My name is Bond, James Bond"
2 print(bond.index("Bond"))
3 # 11
4 msg = "Hello World"
5 print(msg.index("World"))
6 # 6
7 print(msg.index("Bond"))
8 # ValueError: substring not found
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `find` também retorna a primeira posição em que uma String fornecida como parâmetro ocorre na String base.
- Se a String fornecida como parâmetro não está contida na String base, então é retornado o valor `-1`.

```
1 bond = "My name is Bond, James Bond"
2 print(bond.find("Bond"))
3 # 11
4 msg = "Hello World"
5 print(msg.find("World"))
6 # 6
7 print(msg.find("Bond"))
8 # -1
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `strip` remove todos os espaços em branco (incluindo quebras de linhas e tabulações) no início e no fim da String.

```
1 msg = " \n Hello World \t"  
2 print(msg.strip())  
3 # Hello World
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `split` divide uma String em uma lista de acordo com um padrão de caracteres (separador).
- Por padrão, o separador é igual a qualquer sequência de espaços em branco (incluindo quebras de linhas e tabulações).

```
1 str1 = "  MC102  Algoritmos\t\tProgramação\nComputadores  "  
2 dados = str1.split()  
3 print(dados)  
4 # ['MC102', 'Algoritmos', 'Programação', 'Computadores']  
5 str2 = "abacaxi, banana, caqui, damasco"  
6 frutas = str2.split(", ")  
7 print(frutas)  
8 # ['abacaxi', 'banana', 'caqui', 'damasco']
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Por exemplo, o método `split` pode ser usado para separar múltiplos valores lidos numa única linha.

```
1 # Lendo duas strings separadas por um espaço
2 s1, s2 = input().split()
3 # Lendo três números inteiros separados por espaços
4 a, b, c = [int(i) for i in input().split()]
5 # Lendo múltiplos números separados por espaços
6 numeros = [float(i) for i in input().split()]
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `join` junta uma lista de Strings usando a String base como concatenador.

```
1 frutas = ['abacaxi', 'banana', 'caqui', 'damasco']
2 txt = ", ".join(frutas)
3 print(txt)
4 # abacaxi, banana, caqui, damasco
```

- A função `list()` pode ser utilizada para transformar uma String em uma lista de caracteres.

```
1 str = "aeiou"
2 lista = list(str)
3 print(lista)
4 # ['a', 'e', 'i', 'o', 'u']
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- O método `replace` cria uma nova String onde todas as ocorrências de um padrão de caracteres numa String dada são trocadas por outro.

```
1 x = "Algoritmos e Programação de Computadores"
2 y = x.replace("a", "_")
3 print(y)
4 # Algoritmos e Progr_m_ção de Comput_dores
5 y = x.replace("Algoritmos", "$" * len("Algoritmos"))
6 print(y)
7 # $$$$$$$$$$ e Programação de Computadores
8 x = "a,b,c,d,e"
9 y = x.replace(",", "")
10 print(y)
11 # abcde
```


<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- `capitalize()`: converte o primeiro caractere para maiúsculo.

```
1 print("meu teste".capitalize())  
2 # Meu teste
```

- `lower()`: converte a String para letras minúsculas.

```
1 print("Meu TESTE".lower())  
2 # meu teste
```

- `upper()`: converte a String para letras maiúsculas.

```
1 print("mEU tESte".upper())  
2 # MEU TESTE
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- `isnumeric()`: testa se todos os caracteres são dígitos.

```
1 print("1234".isnumeric())
2 # True
3 print("teste123".isnumeric())
4 # False
```

- `isalpha()`: testa se todos os caracteres são letras.

```
1 print("MeuTeste".isalpha())
2 # True
3 print("teste123".isalpha())
4 # False
```

- `isalnum()`: testa se todos os caracteres são letras ou dígitos.

```
1 print("teste123".isalnum())
2 # True
3 print("Meu teste".isalnum())
4 # False
```

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Exemplo:

```
1 sc = input("Entre com uma sequência de caracteres: ")
2
3 if sc.isalpha():
4     print(sc, "possui apenas letras")
5 elif sc.isnumeric():
6     print(sc, "possui apenas dígitos")
7 elif sc.isalnum():
8     print(sc, "possui letras e dígitos")
9 else:
10    print(sc, "não possui apenas letras e dígitos")
```

- Podemos utilizar o comando `for` para percorrer uma String. <https://ic.unicamp.br/~mc102/aulas/aula06.pdf>
- Exemplo:

```
1 s = "abc"
2 for c in s:
3     print(c)
4 # a
5 # b
6 # c
```

- Outro exemplo:

```
1 for c in "Algoritmos":
2     if c in "AEIOUaeiou":
3         print("A String possui a vogal:", c)
4 # A String possui a vogal: A
5 # A String possui a vogal: o
6 # A String possui a vogal: i
7 # A String possui a vogal: o
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi



Run >

Result Size: 411 x 402

Get your website

```
# Modify Strings
```

```
a = " Hello, World!"
```

```
print(a.upper())
```

```
print(a.lower())
```

```
print(a.strip())
```

```
print(a.replace("H", "J"))
```

```
HELLO, WORLD!
```

```
hello, world!
```

```
Hello, World!
```

```
Jello, World!
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi

Run >

Result Size: 832 x 164

Get your website

```
txt = "welcome to the jungle"  
x = txt.split()  
print(x)
```

```
txt = "hello, my name is Peter, I am 26 years old"  
x = txt.split(", ")  
print(x)
```

```
txt = "apple#banana#cherry#orange"  
x = txt.split("#")  
print(x)
```

```
['welcome', 'to', 'the', 'jungle']  
['hello', 'my name is Peter', 'I am 26 years old']  
['apple', 'banana', 'cherry', 'orange']
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string_multi



Run >

Result Size: 832 x 149

Get your website

```
txt = "apple#banana#cherry#orange"
# setting the maxsplit parameter to 1, will return a list with 2 elements!
x = txt.split("#", 1)
print(x)

x = txt.split("#", 2)
print(x)

x = txt.split("#", 4)
print(x)
```

```
['apple', 'banana#cherry#orange']
['apple', 'banana', 'cherry#orange']
['apple', 'banana', 'cherry', 'orange']
```

Exemplos usando Strings

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

1. Escreva um programa que, dada uma sequência de números inteiros (todos fornecidos na mesma linha, separados por espaços), imprima a média desses números.

```
1 texto = input("Entre com uma sequência de números: ")
2 numeros = texto.split()
3 soma = 0
4
5 for n in numeros:
6     soma = soma + int(n)
7 media = soma / len(numeros)
8
9 print("A média é: ", format(media, ".2f"))
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 149

Get your website

```
a = "1 3 45 1 0"  
l = [int(x) for x in a.split()]  
print(sum(l)/len(l))
```

10.0

<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

- Escreva um programa que, dada uma String representando um texto, imprima o número de palavras existentes. Observação: você deve remover os sinais de pontuação (".", ",", ":", ";", "!" e "?") antes de realizar a contagem das palavras.

```
1 texto = input("Entre com um texto: ")
2 pontuacao = [".", ",", ":", ";", "!", "?"]
3
4 # remove os sinais de pontuação
5 for p in pontuacao:
6     texto = texto.replace(p, " ")
7
8 numPalavras = len(texto.split())
9 print("Número de palavras:", numPalavras)
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 149

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""

print(texto)
```

```
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 135

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""

pontuacao = [".", ",", ":", ";", "!", "?"]

for letra in texto:
    if letra in pontuacao:
        texto = texto.replace(letra, " ")

print(texto)
```

```
Aonde fica a saída Perguntou Alice ao gato que ria
Depende respondeu o gato
De quê replicou Alice
Depende de para onde você quer ir
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 135

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""

pontuacao = [".", ",", ":", ";", "!", "?"]

for letra in texto:
    if letra in pontuacao:
        texto = texto.replace(letra, " ")

print(texto.split())
```

```
['Aonde', 'fica', 'a', 'saída', 'Perguntou', 'Alice', 'ao', 'gato', 'que', 'ria', 'Depende',
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 135

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""

pontuacao = [".", ",", ":", ";", "!", "?"]

for letra in texto:
    if letra in pontuacao:
        texto = texto.replace(letra, " ")

print(len(texto.split()))
```

25

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 108

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""

pontuacao = [".", ",", ":", ";", "!", "?"]

for letra in texto:
    if letra in pontuacao:
        texto = texto.replace(letra, " ")

texto_split = texto.split()
print(texto_split, len(texto_split), sep="\n")
```

```
['Aonde', 'fica', 'a', 'saída', 'Perguntou', 'Alice', 'ao', 'gato', 'que', 'ria', 'Depende',
25
```


<https://ic.unicamp.br/~mc102/aulas/aula06.pdf>

3. Escreva um programa que, dada uma String texto e uma String palavra, ache todas as posições de ocorrência da palavra no texto. O seu programa deve desconsiderar se as letras são maiúsculas ou minúsculas.

```
1 texto = input("Entre com um texto: ")
2 palavra = input("Entre com uma palavra: ")
3 texto = texto.lower()
4 palavra = palavra.lower()
5
6 removido = 0
7 while palavra in texto:
8     posicao = texto.find(palavra)
9     print(removido + posicao)
10    texto = texto[posicao + 1:]
11    removido = removido + (posicao + 1)
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1



Run >

Result Size: 832 x 108

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""
palavra = "Depende"
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 108

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""
palavra = "Depende"

lista = [i for i in range(len(texto)) if texto.startswith(palavra, i) ]

print(lista)
```

```
[55, 108]
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 108

Get your website

```
texto = """
Aonde fica a saída?, Perguntou Alice ao gato que ria.
Depende!, respondeu o gato.
De quê?, replicou Alice;
Depende de para onde você quer ir...
"""
palavra = "Depende"

lista = [i for i in range(len(texto)) if texto.startswith(palavra, i) ]

print(lista)

for x in lista:
    print(texto[x:x+len(palavra)])
```

```
[55, 108]
Depende
Depende
```

- Um palíndromo é uma palavra ou frase que pode ser lida da mesma forma tanto da esquerda para a direita como da direita para a esquerda (desconsiderando os espaços em branco). Considere que a entrada não possui sinais de pontuação ou acentos. Escreva um programa que, dada uma String, verifique se ela é um palíndromo.

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 70

Get your website

```
txt = "A torre da derrota"
txt = txt.lower()
txt = txt.replace(" ", "")
i = 0
j = len(txt) - 1
palindromo = True
while i < j:
    if txt[i] != txt[j]:
        palindromo = False
        break
    else:
        i = i + 1
        j = j - 1
if palindromo:
    print("Palíndromo")
else:
    print("Não palíndromo")
```

Palíndromo

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 70

Get your website

```
txt = "A torre da derrota"  
txt = txt.lower().replace(" ", "")
```

```
i = 0  
j = len(txt) - 1  
palindromo = True  
while i < j:  
    if txt[i] != txt[j]:  
        palindromo = False  
        break  
    else:  
        i = i + 1  
        j = j - 1  
  
print("Palindromo:", palindromo)
```

```
Palindromo: True
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 70

Get your website

```
txt = "A diva em Argel alegre-me a vida"  
txt = txt.lower().replace(" ", "")
```

```
i = 0  
j = len(txt) - 1  
palindromo = True  
while i < j:  
    if txt[i] != txt[j]:  
        palindromo = False  
        break  
    else:  
        i = i + 1  
        j = j - 1
```

```
print("Palindromo:", palindromo)
```

```
Palindromo: False
```


https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 70

Get your website

```
txt = "A diva em Argel alegre-me a vida"
txt = txt.lower().replace(" ", "").replace("-", "")

i = 0
j = len(txt) - 1
palindromo = True
while i < j:
    if txt[i] != txt[j]:
        palindromo = False
        break
    else:
        i = i + 1
        j = j - 1

print("Palindromo:", palindromo)
```

```
Palindromo: True
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 147

Get your website

```
txt = "A diva em Argel alegre-me a vida"
txt = txt.lower().replace(" ", "").replace("-", "")

palindromo = False
if txt == txt[::-1]:
    palindromo = True

print("Palindromo:", palindromo)
```

```
Palindromo: True
```

https://www.w3schools.com/python/trypython.asp?filename=demo_string1

Run >

Result Size: 832 x 147

Get your website

```
txt = "A diva em Argel alegre-me a vida"
tmp = txt.lower().replace(" ", "").replace("-", "")

palindromo = False
if tmp == tmp[::-1]:
    palindromo = True

print("Palindromo:", palindromo)
```

```
Palindromo: True
```

Perguntas

Referências

- Zanoni Dias, MC102, Algoritmos e Programação de Computadores, IC/UNICAMP, 2021. <https://ic.unicamp.br/~mc102/>
 - Aula Introdutória [[slides](#)] [[vídeo](#)]
 - Primeira Aula de Laboratório [[slides](#)] [[vídeo](#)]
 - Python Básico: Tipos, Variáveis, Operadores, Entrada e Saída [[slides](#)] [[vídeo](#)]
 - Comandos Condicionais [[slides](#)] [[vídeo](#)]
 - Comandos de Repetição [[slides](#)] [[vídeo](#)]
 - Listas e Tuplas [[slides](#)] [[vídeo](#)]
 - Strings [[slides](#)] [[vídeo](#)]
 - Dicionários [[slides](#)] [[vídeo](#)]
 - Funções [[slides](#)] [[vídeo](#)]
 - Objetos Multidimensionais [[slides](#)] [[vídeo](#)]
 - Algoritmos de Ordenação [[slides](#)] [[vídeo](#)]
 - Algoritmos de Busca [[slides](#)] [[vídeo](#)]
 - Recursão [[slides](#)] [[vídeo](#)]
 - Algoritmos de Ordenação Recursivos [[slides](#)] [[vídeo](#)]
 - Arquivos [[slides](#)] [[vídeo](#)]
 - Expressões Regulares [[slides](#)] [[vídeo](#)]
 - Execução de Testes no Google Cloud Shell [[slides](#)] [[vídeo](#)]
 - Numpy [[slides](#)] [[vídeo](#)]
 - Pandas [[slides](#)] [[vídeo](#)]
- Panda - Cursos de Computação em Python (IME -USP) <https://panda.ime.usp.br/>
 - Como Pensar Como um Cientista da Computação <https://panda.ime.usp.br/pensepy/static/pensepy/>
 - Aulas de Introdução à Computação em Python <https://panda.ime.usp.br/aulasPython/static/aulasPython/>
- Fabio Kon, Introdução à Ciência da Computação com Python <http://bit.ly/FabioKon/>
- Socratica, Python Programming Tutorials <http://bit.ly/SocraticaPython/>
- Google - online editor for cloud-native applications (Python programming) <https://shell.cloud.google.com/>
- w3schools - Python Tutorial <https://www.w3schools.com/python/>
- Outros, citados nos Slides.