



ALGORITMIA E ESTRUTURAS DE DADOS

STRINGS

Python String Methods

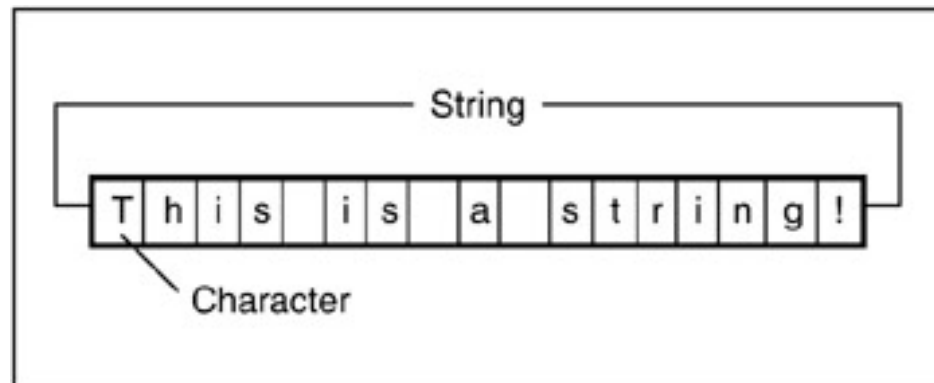
`str(), upper(),
lower(), count(),
find(), replace(),
len().`

LICENCIATURA EM
TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO PARA A WEB
#ESMAD #P.PORTO

1. Strings

- ☐ Conceito
- ☐ Concatenação de strings
- ☐ Índice de uma string
- ☐ Comprimento de uma string

2. Métodos *built-in* Python



❖ Strings | Conceito

- ❑ Cadeia de caracteres, isto é, consiste numa sequência de caracteres
- ❑ Conteúdo de uma string está entre “ ” ou ‘ ’
- ❑ Uma string sem quaisquer caracteres consiste numa string nula

```
Exemplos.py > ...  
1  # Exemplos ilustrativos do uso de strings  
2  
3  nome = "Algoritmia e Estruturas de Dados"  
4  |
```

❖ Strings | Concatenação

- ❑ Concatenação de strings: É uma operação que pode ser vista como uma operação análoga a adição na matemática

```
Exemplos.py > ...  
1  # Exemplos ilustrativos do uso de strings  
2  
3  
4  nome = "Carlos"  
5  sobrenome = "Fonseca"  
6  nome_completo = nome + " " + sobrenome  
7  print(nome_completo)  
8  
9
```

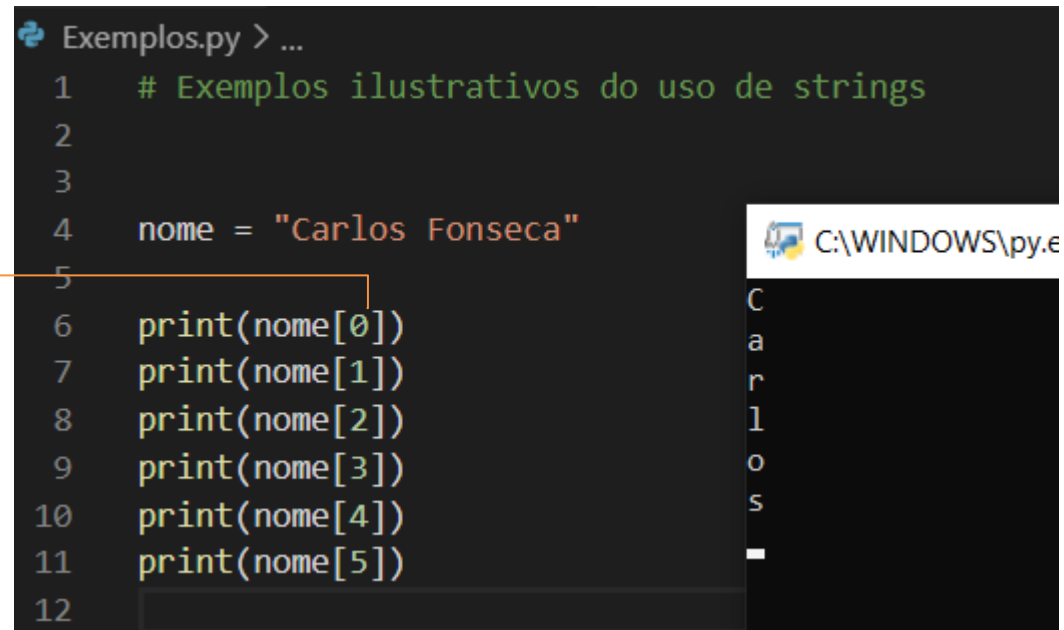
C:\WINDOWS\py.exe
Carlos Fonseca

❖ String | Índice de uma string

- ❑ Uma string consiste num conjunto de caracteres, acessíveis individualmente através da especificação do seu **índice** (posição) na string
- ❑ O primeiro caracter de uma string tem a **posição 0**.

Índice (posição)
na string

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  nome = "Carlos Fonseca"
5
6  print(nome[0])
7  print(nome[1])
8  print(nome[2])
9  print(nome[3])
10 print(nome[4])
11 print(nome[5])
12
```



❖ String | Índice de uma string

- ❑ **Substring:** subconjunto de caracteres de uma string
- ❑ Podemos extrair substrings de uma string, explicitando o tamanho da substring através de `[]`

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  nome = "Carlos Fonseca"
5
6  print(nome[0:3])    # substring da posição 0 à 3 (não incluída)
7  print(nome[0:5])    # substring da posição 0 à 5 (não incluída)
8  print(nome[:6])     # substring da posição 0 à 6 (não incluída)
9  print(nome[7:])     # substring da posição 7 à última
10
11
12
```

C:\WINDOWS\py.exe

Car
Carlo
Carlos
Fonseca
-

Posição inicial

Posição final (não incluída)

❖ String | Índice de uma string

❑ **Substring:** índices negativos

```
nome = "Carlos Manuel Fonseca"

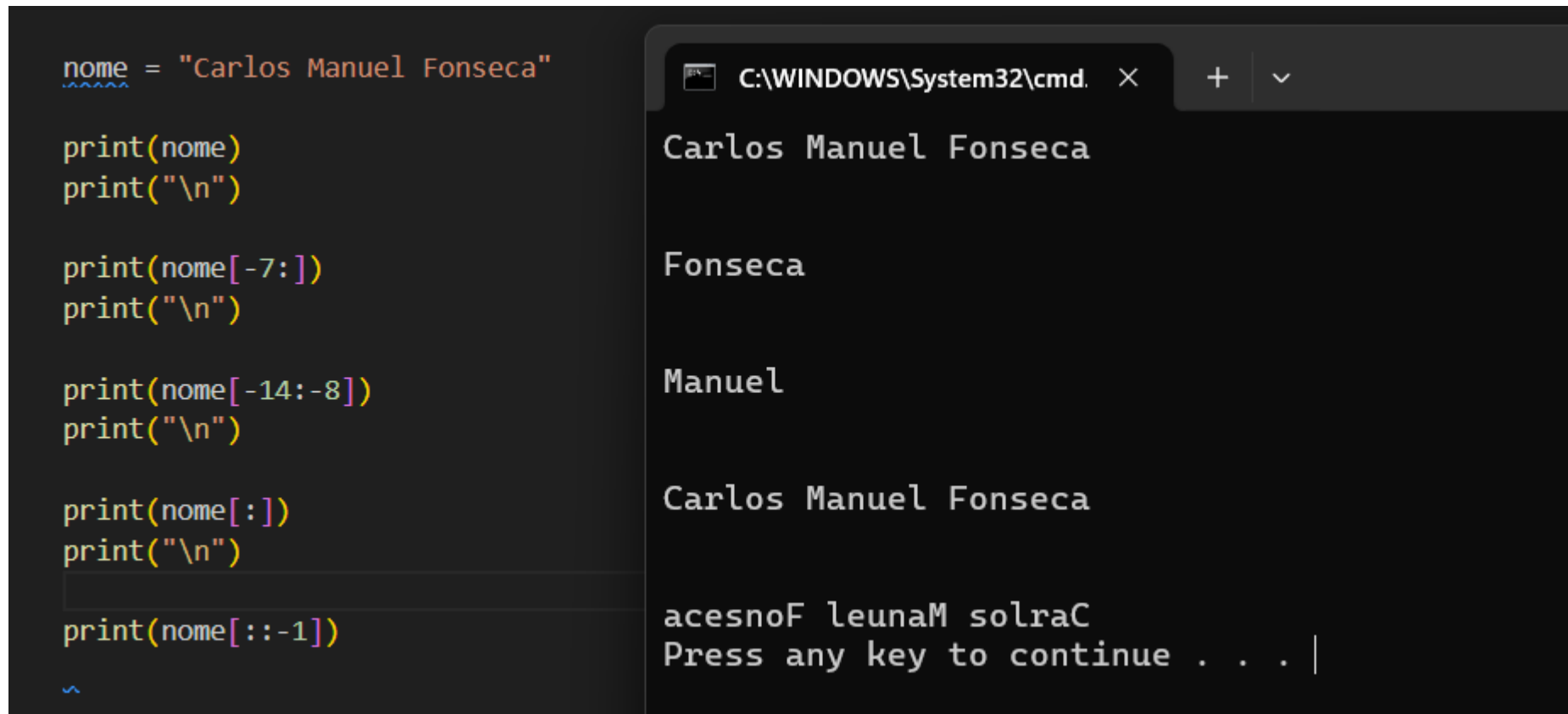
print(nome)
print("\n")

print(nome[-7:])
print("\n")

print(nome[-14:-8])
print("\n")

print(nome[:])
print("\n")

print(nome[::-1])
```



The image shows a Python script on the left and its execution output in a Windows command prompt on the right. The script defines a string 'nome' as 'Carlos Manuel Fonseca' and uses various slicing techniques to print parts of the string. The output in the command prompt shows the full string, followed by 'Fonseca' (last 7 characters), 'Manuel' (characters from index -14 to -8), the full string again, and finally the string reversed ('acesnoF leunaM solraC'). The command prompt window title is 'C:\WINDOWS\System32\cmd.'.

C:\WINDOWS\System32\cmd. × + ▾

Carlos Manuel Fonseca

Fonseca

Manuel

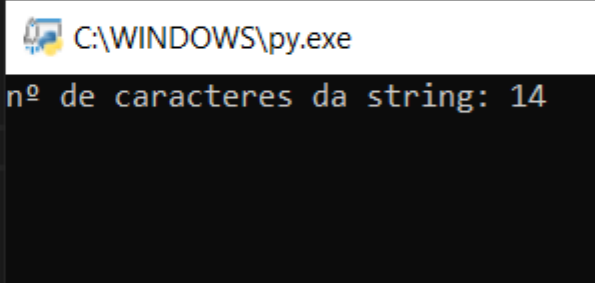
Carlos Manuel Fonseca

acesnoF leunaM solraC
Press any key to continue . . . |

❖ String | Comprimento de uma string

- ❑ Comprimento (tamanho) de uma string: **len**
- ❑ Função **len()** devolve o comprimento da string (número de caracteres)

```
Exemplos.py > ...  
1  # Exemplos ilustrativos do uso de strings  
2  
3  
4  nome = "Carlos Fonseca"  
5  comp = len(nome)  
6  print("nº de caracteres da string:", comp)  
7  
8  
9
```



A string nome tem **14** caracteres: da posição **0** à posição **13**

❖ String | Comprimento de uma string

- ❑ Comprimento de uma string: **len**
Devolve o comprimento da string (número de caracteres)

Nº caracteres
inseridos na
variável nome

Percorre todas
as posições

Imprime cada
posição

```
Exemplos.py > ...  
1  # Exemplos ilustrativos do uso de strings  
2  
3  
4  nome = input("Indique um nome:")  
5  comp = len(nome)  
6  
7  for i in range (comp):  
8      print (nome[i])  
9  
10  
11  
12
```

C:\WINDOWS\py.exe

Indique um nome:Algoritmia

A
l
g
o
r
i
t
m
i
a

```
4  
5  nome = input("Indique um nome: ")  
6  for character in nome:  
7      print(character)  
8  
9
```

❖ String | Comprimento de uma string

- ❑ Percorrer a sequência de caracteres de uma string

```
Exemplos.py > ...  
# Exemplos ilustrativos do uso de strings  
  
# lê um nome e imprime por ordem inversa  
  
nome = input("Indique um nome:")  
  
comp = len(nome) # nº de caracteres da variavel nome  
for i in range(comp-1, -1, -1):  
    print(nome[i])
```

C:\WINDOWS\py.exe
Indique um nome:Algoritmia
a
i
m
t
i
r
o
g
l
A
_

❖ String | Comprimento de uma string

- ❑ Percorrer a sequência de caracteres de uma string

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3  # lê um nome e imprime por ordem inversa
4
5  nome = input("Indique um nome:")
6
7  comp = len(nome)  # nº de caracteres da variavel nome
8  for i in range(comp-1, -1, -1):
9      print(nome[i], end="")
10
11
12
13
```

C:\WINDOWS\py.exe
Indique um nome:Algoritmia
aimtiroglA

Sem fazer *new line* em cada print

❖ Strings | Métodos *built-in* python

❑ Alguns métodos que manipulam strings:

| Método | Descrição |
|----------------|---|
| upper() | Devolve uma string com todos os caracteres em maiúsculas |
| lower() | Devolve uma string com todos os caracteres em minúsculas |
| find(padrao) | Devolve a posição em que se encontra, na string, o padrão de procura (primeira ocorrência) Se não existir devolve -1 |
| index(padrao) | Devolve a posição em que se encontra, na string, o padrão de procura (primeira ocorrência) Se não existir dá erro! |
| rfind(padrao) | Devolve a posição em que se encontra, na string, o padrão de procura (última ocorrência) |
| rindex(padrao) | Devolve a posição em que se encontra, na string, o padrão de procura (última ocorrência) |

❖ Strings | Métodos *built-in python*

❑ Alguns métodos que manipulam strings:

| Método | Descrição |
|--------------------|--|
| split(character) | Divide uma string em substrings, em função do caracter especificado |
| replace(s1, s2) | Devolve uma string em que todas as ocorrências de s1 são substituídas por s2 |
| count() | Devolve o número de ocorrências de um padrão na string |
| strip() | Remove espaços em branco no início e fim da string |
| startswith(padrao) | Devolve True se a string se inicia com o padrão de pesquisa. Caso contrário devolve False. |

❖ Strings | Métodos *built-in* python

- ❑ upper()
- ❑ lower()

```
Exemplos.py 7 ...  
1  # Exemplos ilustrativos do uso de strings  
2  
3  
4  nome = input("Indique um nome:")  
5  
6  nome1 = nome.upper()  
7  print (nome1)  
8  
9  nome1 = nome.lower()  
10 print(nome1)  
11  
12  
13
```

C:\WINDOWS\py.exe

Indique um nome:Algoritmia e Estruturas de Dados
ALGORITMIA E ESTRUTURAS DE DADOS
algoritmia e estruturas de dados

❖ Strings | Métodos *built-in python*

- ❑ `find(padão)`
- ❑ `index(padão)`

Devolve a posição em que se encontra, na string, o padrão de procura (primeira ocorrência). Se não existir devolve -1

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  nome = input("Indique um nome:")
5
6  pos = nome.find(" ")      # procura posição do 1º espaço
7  print ("posição do primeiro espaço=", pos)
8
9  pos = nome.find("r")      # procura posição do primeiro caracter "r"
10 print ("posição do primeiro r =", pos)
11
12 pos = nome.find("de")     # procura posição da primeira ocorrência de "de"
13 print ("posição do primeiro 'de' =", pos)
14
15 pos = nome.find("z")      # procura posição da primeira ocorrência de "z"
16 print ("posição do primeiro z =", pos)
17
18
```

C:\WINDOWS\py.exe

```
Indique um nome:Algoritmia e Estruturas de Dados
posição do primeiro espaço= 10
posição do primeiro r = 4
posição do primeiro 'de' = 24
posição do primeiro z = -1
```

❖ Strings | Métodos *built-in python*

- ❑ `find(padrao)`
- ❑ `index(padrao)`

Método ***find*** inclui parâmetros opcionais para indicar posição inicial e final da pesquisa

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  nome = input("Indique um nome:")
5
6  pos = nome.find(" ")      # procura posição do 1º espaço
7  print ("posição do primeiro espaço=", pos)
8
9  pos = nome.find(" ", pos+1)  # procura posição do 2º espaço
10 print ("posição do segundo espaço =", pos)
11
12 pos = nome.find(" ", pos+1)  # procura posição do 3º espaço
13 print ("posição do segundo espaço =", pos)
14
15
16
17
```

C:\WINDOWS\py.exe

```
Indique um nome:Algoritmia e Estruturas de Dados
posição do primeiro espaço= 10
posição do segundo espaço = 12
posição do segundo espaço = 23
```


❖ Strings | Métodos *built-in python*

❑ find(padão)

❑ index(padão)

Devolve a posição em que se encontra, na string, o padrão de procura (primeira ocorrência). Se não existir devolve um erro!!

Exemplos.py > ...

```
1 # Exemplos ilustrativos do uso de strings
2
3
4 nome = input("Indique um nome:")
5
6 pos = nome.index(" ") # procura posição do 1º espaço
7 print ("posição do primeiro espaço=", pos)
8
9 pos = nome.index("r") # procura posição do primeiro caracter "r"
10 print ("posição do primeiro r =", pos)
11
12 pos = nome.index("de") # procura posição da primeira ocorrência de "de"
13 print ("posição do primeiro 'de' =", pos)
14
15 #pos = nome.index("z") # procura posição da primeira ocorrência de "z"
16 #print ("posição do primeiro z =", pos)
17
```

C:\WINDOWS\py.exe

```
Indique um nome:Algoritmia e Estruturas de Dados
posição do primeiro espaço= 10
posição do primeiro r = 4
posição do primeiro 'de' = 24
```

```
File "c:\Users\mario\Dropbox\AED\2020-21\4 - Exercicios\Ficha 04\Exemplos.py", line 15, in <module>
    pos = nome.index("z") # procura posição da primeira ocorrência de "z"
ValueError: substring not found
PS C:\Users\mario\Dropbox\AED\2020-21\4 - Exercicios\Ficha 04>
```

❖ Strings | Métodos *built-in* python

❑ `rfind(padão)`

❑ `rindex(padão)`

Devolve a posição em que se encontra, na string, o padrão de procura (última ocorrência, isto é, a ocorrência mais à direita). O método `rindex` devolve um erro se o padrão de pesquisa não existir!

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  nome = input("Indique um nome:")
5
6  pos = nome.rfind(" ")      # procura posição do último espaço
7  print ("posição do último espaço=", pos)
8
9  pos = nome.rfind("a")      # procura posição do último de a
10 print ("posição do último 'a' =", pos)
11
12
13
14
```

C:\WINDOWS\py.exe

```
Indique um nome:Algoritmia e Estruturas de Dados
posição do último espaço= 26
posição do último 'a' = 28
```

❖ Strings | Métodos *built-in python*

❑ split(caracter)

Divide uma string em substrings, a partir do caracter especificado

```
# Exemplos ilustrativos do uso de strings
```

```
países = "Portugal;Espanha;França;Alemanha;Itália"  
print ("lista países: ", países.split(";"))
```

```
#split da string países para uma variável lista_países  
lista_países = países.split(";")  
# imprime o conteúdo da sequência lista_países  
for país in lista_países:  
    print(país)
```

C:\WINDOWS\py.exe

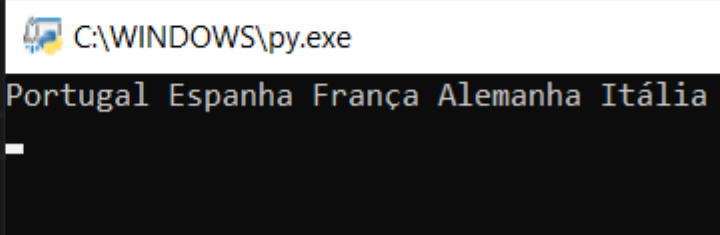
```
lista países: ['Portugal', 'Espanha', 'França', 'Alemanha', 'Itália']  
Portugal  
Espanha  
França  
Alemanha  
Itália  
-
```

❖ Strings | Métodos *built-in python*

❑ Replace(s1,s2)

Devolve uma string em que todas as ocorrências de s1 são substituídas por s2

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  paises = "Portugal;Espanha;França;Alemanha;Itália"
5  paises1 = paises.replace(";", " ")
6
7  print(paises1)
8
9
10
11
```



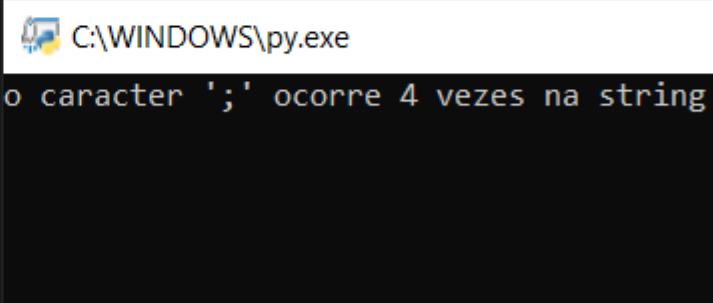
The screenshot shows a Windows command prompt window titled "C:\WINDOWS\py.exe". The output of the Python script is displayed as "Portugal Espanha França Alemanha Itália".

❖ Strings | Métodos *built-in python*

❑ count(padrão)

Devolve o número de ocorrências de um padrão na string

```
Exemplos.py > ...
1  # Exemplos ilustrativos do uso de strings
2
3
4  paises = "Portugal;Espanha;França;Alemanha;Itália"
5
6  num = paises.count(";")
7  print("o caracter ';' ocorre {0} vezes na string".format(num))
8
9
10
11
12
```

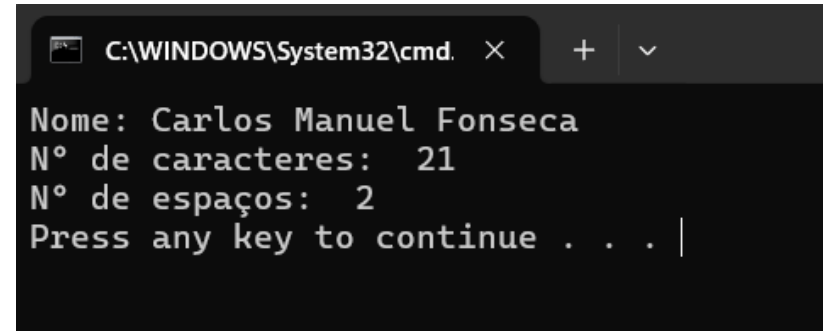


❖ Strings

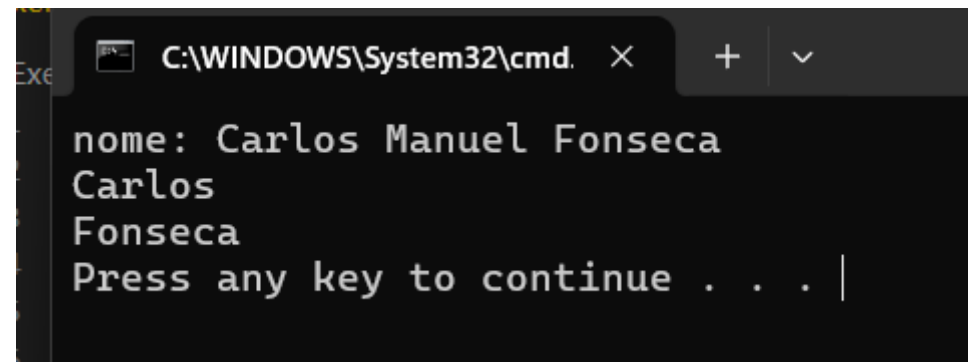
Avalia o teu conhecimento

- ☐ Ler um nome e no final imprimir:
 - ☐ O nº de caracteres
 - ☐ O nº de espaços

- ☐ Ler um nome completo e no final imprimir:
 - ☐ O primeiro nome próprio (até 1º espaço)
 - ☐ O último sobrenome (do último espaço até final)



```
C:\WINDOWS\System32\cmd. x + v
Nome: Carlos Manuel Fonseca
Nº de caracteres: 21
Nº de espaços: 2
Press any key to continue . . . |
```



```
C:\WINDOWS\System32\cmd. x + v
nome: Carlos Manuel Fonseca
Carlos
Fonseca
Press any key to continue . . . |
```