



POLYTECHNIC
FROM PORTO
SCHOOL
HIGHER
MEDIA ARTS
AND DESIGN

P.PORTO

ALGORITHMY AND DATA STRUCTURES

MODULE II

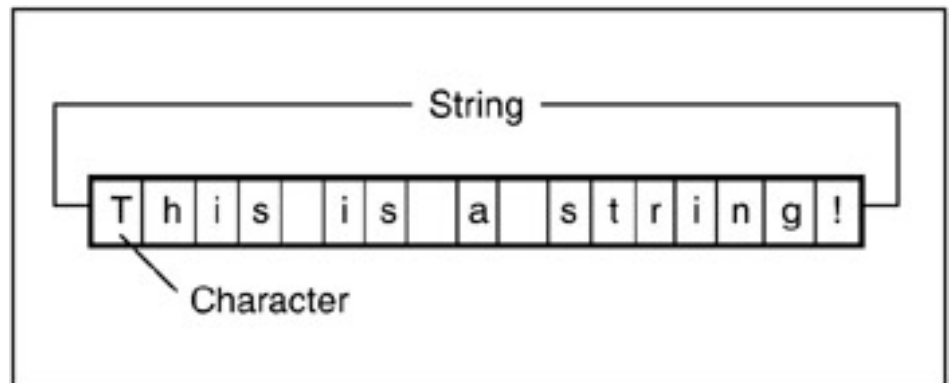
CHAINS OF CHARACTERS - STRINGS

TECHNOLOGIES AND INFORMATION SYSTEMS FOR THE WEB

1. Strings:

- ☐ Concept
- ☐ Concatenation of strings
- ☐ Index of a string
- ☐ Length of a string

2. Methods *built-in* Python



❖ String| Concept

- ❑ A string consists of a sequence of characters
- ❑ Content of a string is between “ ” or ' '
- ❑ One string without any characters consists on a null string

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

❖ String| Concatenation

- ❑ Concatenation of strings: It is an operation that can be seen as an operation analogous to addition in mathematics

```
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
```

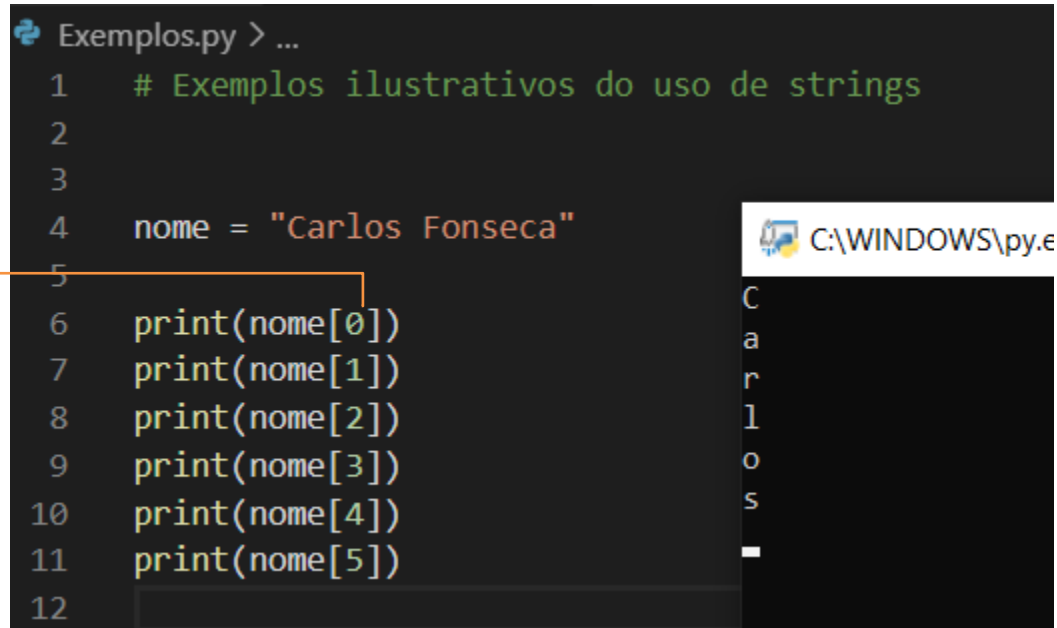
C:\WINDOWS\py.exe
Carlos Fonseca

❖ String| Index of a string

- ❑ A string consists on a set of characters, individually accessible by specifying their **index** (position) in the string
- ❑ The first character has the **position 0**.

Index (position)
In the 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| Index of a string

❑ **Substring**: subset of characters of string or string slicing

- ❑ We can extract substrings of a string, identifying the size of the substring across[]
- ❑ The ending position is excluded

```
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
-


Initial position

Final position (not included)

❖ String| Index of a string

- ❑ Use **negative indexes** to start the slice from the end of the string.
- ❑ The characters in a string can also be indexed counting back from the end of the string. This is known as negative indexing.
- ❑ The final character of the string has a negative index of -1. The second last character occupies position -2, and so on. If the index does not exist within the string, Python returns an error.
- ❑ The ending position is excluded

```
nome = "Carlos Fonseca"
```



..... -4 -3 -2 -1

❖ String| Index of a string

- ❑ Use **negative indexes** to start the slice from the end of the string.

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

```
nome = "Carlos Fonseca"
```

```
print(nome)  
print("\n\n")
```

```
print(nome[-7:-1])  
print("\n\n")
```

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

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

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

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

c:\Users\mario\OneDrive\AED\4 - Exercicios\Ficha 05 - VS Code Console

Carlos Fonseca

Fonsec

Fonseca

Carlos

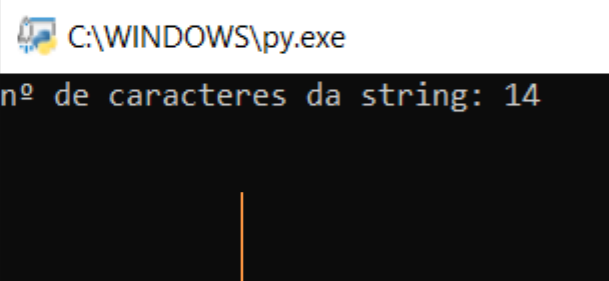
Carlos Fonseca

acesnoF solraC
Press any key to continue . . .

❖ String| Length of a string

- ❑ Length (size) of a string: **len**
- ❑ Function **len()** returns the length of a string (number of characters)

```
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
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\py.exe". It displays the output of the Python script: "nº de caracteres da string: 14". An orange arrow points from the number 14 in the output to the text below.

The string *nome* has **14** characters: from position **0** to position **13**

❖ String| Length of a string

❑ Length of a string: **len**

Returns the length of the string (number of characters)

number of characters inserted into the variable nome

cycle through all positions

print each position

```
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

❖ String| Length of a string

- ❑ Scroll through the character sequence of a string

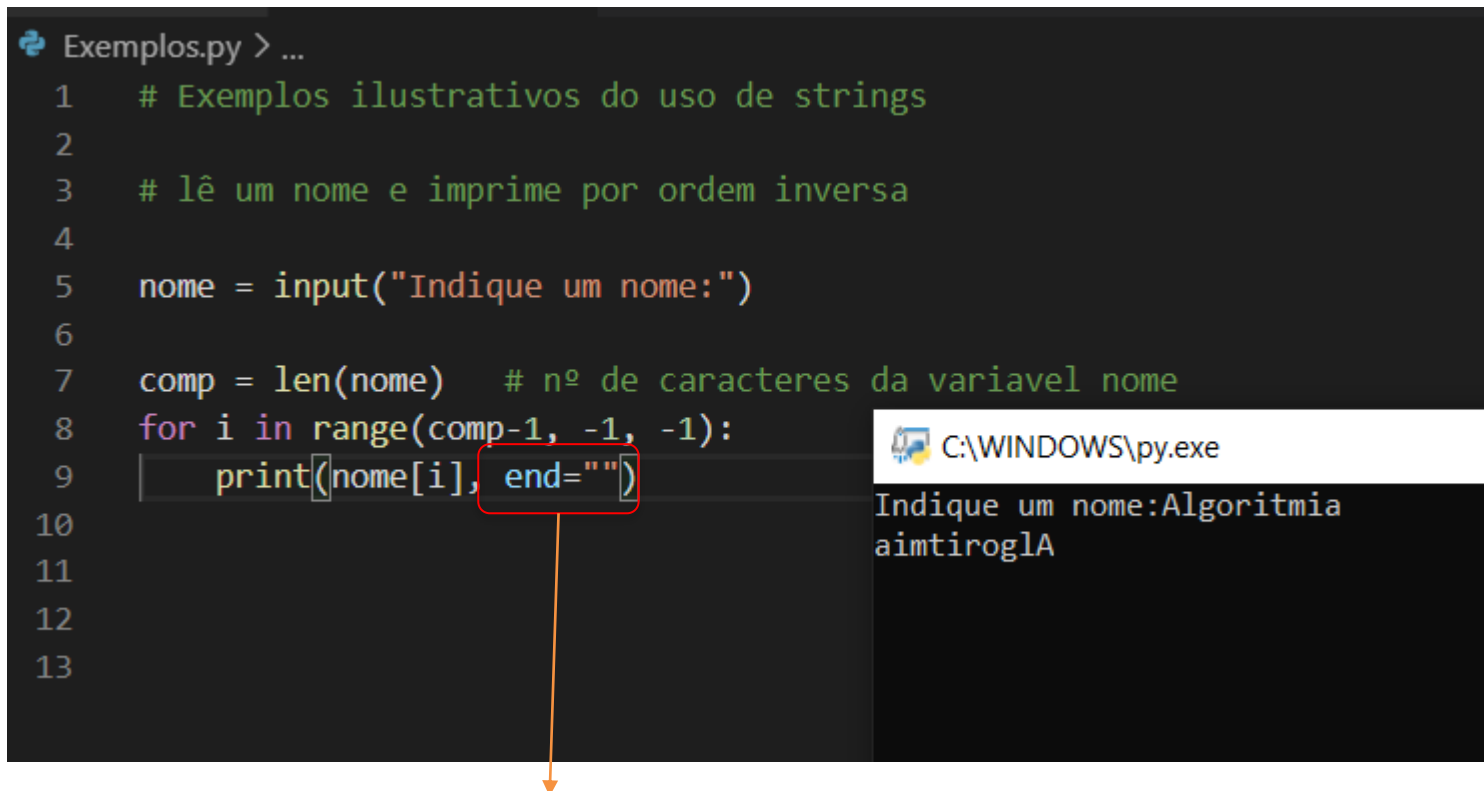
```
Exemplos.py > ...  
# Exemplos ilustrativos do uso de strings  
  
# lê um nome e imprime por ordem inversa  
  
nome = input("Indique um nome:")  
  
comp = len(nome) # nº de caracteres da variavel nome  
for i in range(comp-1, -1, -1):  
    print(nome[i])
```

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

❖ String| Length of a string

- ❑ Scroll through the character sequence of a 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

Without *new line* at the end of each print

❖ Strings | *built-in methods*

❑ Some methods that manipulate strings:

Method	Description
<code>upper()</code>	returns a string with all characters in capital letters
<code>lower()</code>	returns a string with all characters in lowercase
<code>find(search)</code>	returns the position, in the string, of the search characters (the first occurrence) If it does not exist, returns -1
<code>index(search)</code>	returns the position, in the string, of the search characters (the first occurrence) If it doesn't exist, it gives an error!
<code>rfind(standard)</code>	returns the position, in the string, of the search characters (<u>last occurrence</u>) If it does not exist, it returns -1
<code>rindex(standard)</code>	returns the position, in the string, of the search characters (<u>last occurrence</u>) If it doesn't exist, it gives an error!

❖ Strings | *built-in methods*

❑ Some methods that manipulate strings:

Method	Description
<code>split(character)</code>	splits the string into substrings, using the specified character
<code>replace(s1, s2)</code>	returns a string where all occurrences of s1 are replaced by s2
<code>count(search)</code>	returns the number of occurrences of a pattern (search characters) in the string

❖ Strings | *built-in methods*

❑ 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 | *built-in methods*

❑ `find(search)`

❑ `index(search)`

returns the position, in the string, of the search characters (first occurrence)

If it does not exist, returns -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 | *built-in methods*

❑ find(search)

❑ index(search)

Method ***find*** includes optional arguments to indicate start and end position of the search. Optional arguments start and end are interpreted as in slice notation.

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 input()
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 | *built-in methods*

❑ `find(search)`

❑ `index(search)`

Method ***find*** includes optional arguments to indicate start and end position of the search. Optional arguments start and end are interpreted as in slice notation.

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

```
nome = input("Indique um nome:")
```

```
print(nome.find(' '))
```

```
print(nome.find(' ', 20, 30))
```

C:\Users\mario\OneDrive\AED\4 - Exercicios\Ficha 05 - VS Code Console

Indique um nome:Algoritmia e estruturas de dados

10

23

Press any key to continue . . .

❖ Strings | *built-in methods*

❑ find(search)

❑ index(search)

returns the position, in the string, of the search characters (first occurrence)

If it doesn't exist, it gives an error!!

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 | *built-in methods*

❑ `rfind(standard)`

❑ `rindex(standard)`

returns the position, in the string, of the search characters (last occurrence)

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 | *built-in methods*

❑ `rfind(standard)`

❑ `rindex(standard)`

returns the position, in the string, of the search characters (last occurrence)

Method ***rfind*** includes optional arguments: start and end are interpreted as in slice notation.

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

```
nome = input("Indique um nome:")
```

```
print(nome.rfind('de'))
```

```
print(nome.rfind('de', 0, 10))
```

c:\Users\mario\OneDrive\AED\4 - Exercicios\Ficha 05 - VS Cod

Indique um nome:maria de de fatima

9

6

Press any key to continue . . .

❖ Strings | *built-in methods*

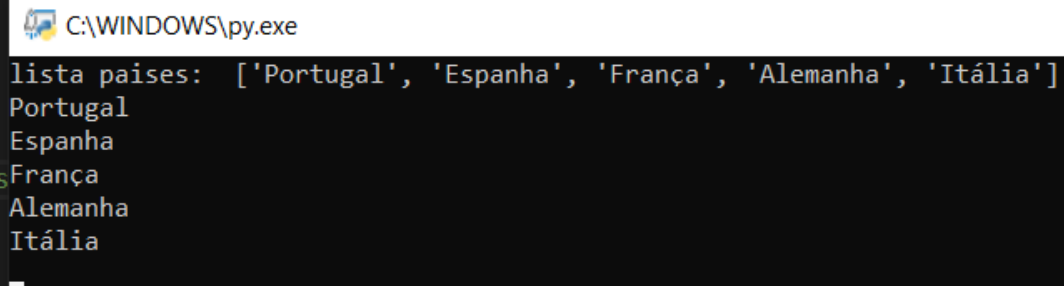
❑ `split(character)`

splits the string into substrings, using the specified character

```
# 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)
```



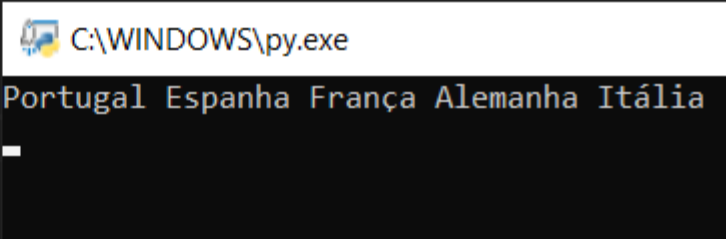
```
C:\WINDOWS\py.exe  
lista países: ['Portugal', 'Espanha', 'França', 'Alemanha', 'Itália']  
Portugal  
Espanha  
França  
Alemanha  
Itália  
-
```

❖ Strings | *built-in methods*

❑ `replace(s1,s2)`

returns a string where all occurrences of s1 are replaced by 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
```



`replace(s1, s2, number)`



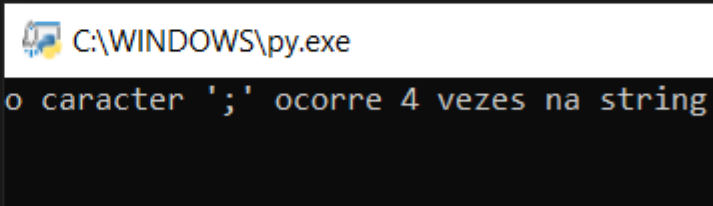
optional parameter: maximum number of occurrences to replace.
-1 (the default value) means replace all occurrences.

❖ Strings | *built-in methods*

❑ count(character)

returns the number of occurrences of a search characters in the 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  inpt  
12
```



❖ Strings | *built-in methods*

❑ `count(character)`

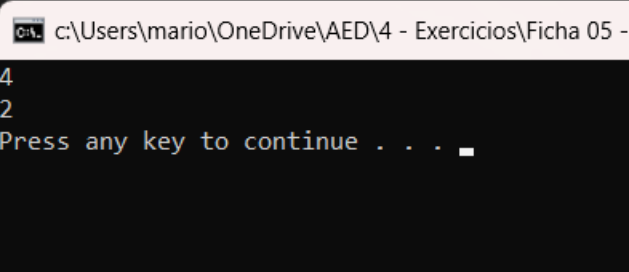
returns the number of occurrences of a search characters in the string

```
# Exemplos ilustrativos do uso de strings

países = "Portugal;Espanha;França;Alemanha;Itália"

print(países.count(";"))

print(países.count(";", 0, 20))
```



Optional arguments start and end are interpreted as in slice notation.