

## Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

### Tutorial Problema 2 de la Guía N° 4

#### Problema 2:

Escriba un script de PYTHON que su encabezamiento incluya:

```
texto = """
Los primeros días nos      pareció penoso      porque ambos habíamos dejado
en la parte      tomada muchas cosas      que queríamos. Mis libros de literatura
francesa , por      ejemplo      , estaban todos en la biblioteca .      Irene pensó
en una botella      de Hesperidina de muchos años .      con frecuencia (pero esto
solamente sucedió los      primeros días) cerrábamos      algún cajón de las
cómodas y nos mirábamos con tristeza.
"""
```

```
# de Casa tomada, de Julio Cortázar
```

Dado este `string` multilínea *texto*, el programa debe imprimir en pantalla el mismo texto pero corrigiendo los tres errores de formateo descriptos a continuación:

- Reemplazando espacios en blanco múltiples por espacios en blanco simples.
- Eliminando, si los hubiere, espacios en blanco que anteceden a los signos de puntuación “coma”, “punto y coma”, “punto” y “dos puntos”.
- Forzando que la primera palabra después de un punto comience con mayúscula.

---

#### Tutorial:

- Guarde en el archivo `p2.py` las siguientes instrucciones:

```
1
2 texto = """
3     Los primeros días nos      pareció penoso      porque ambos habíamos dejado
4 en la parte      tomada muchas cosas      que queríamos. Mis libros de literatura
5 francesa , por      ejemplo      , estaban todos en la biblioteca .      Irene pensó
6 en una botella      de Hesperidina de muchos años .      con frecuencia (pero esto
7 solamente sucedió los      primeros días) cerrábamos      algún cajón de las
8 cómodas y nos mirábamos con tristeza.
9 """
10
11 # de Casa tomada, de Julio Cortázar
12
13
14 print()
```

```

15 print("-> texto de entrada: -----")
16 print(texto)
17
18 print()
19 print("-> Primer método: -----")
20
21 def parse(s):
22     s = s.strip().lower().replace(" ", " ").replace(",",";").
23         .replace(";",";").replace(".", ".").replace(":", ":")
24     return str(s[0].upper()) + s[1:]
25
26 def parse2(s):
27     while s.find(" ")>-1 or s.find(",")>-1 or s.find(";")>-1 or s.find(
28         ".")>-1 or s.find(":")>-1:
29         s = parse(s)
30     return s
31
32 def parse3(s):
33     n1 = s.find(" ")
34     n2 = n1
35     while n2>-1:
36         s = s[0:n1] + s[n1:n1+3].upper() + s[n1+3:]
37         n2 = s[n1+1:].find(" ")
38         n1 = n2 + n1 + 1
39     return s
40
41
42 print('-> corrección tipo 1 -----')
43 texto1 = parse(texto)
44 print('    texto1 = parse(texto)')
45 print(texto1)
46 print()
47 print('-> corrección tipo 2 -----')
48 texto2 = parse2(texto)
49 print('    texto2 = parse2(texto)')
50 print(texto2)
51 print()
52 print('-> corrección tipo 3 -----')
53 texto3 = parse3(texto2)
54 print('    texto3 = parse3(texto)')
55 print(texto3)
56 print()
57
58 # -----
59 # Otro método con strip y join:
60 print()
61 print("-> Segundo método: -----")
62
63 # quitamos los extremos innecesarios
64 textoa = texto.strip()
65 # separamos por los puntos
66 textob = textoa.split('.')
67
68 # separamos cada oración en palabras
69 # y luego unimos las oraciones nuevamente
70 # lo que quita los espacios múltiples

```

```

71 textoc = []
72 for i in range(len(textob)-1):
73     textoc.append(' '.join(textob[i].split()))
74
75 print(textoc)
76 # Notar que join ha quitado las nuevas líneas \n
77
78 # pasamos las primeras letras a mayúsculas
79 textod = []
80 for i in range(len(textoc)):
81     textod.append(textoc[i][0].upper() + textoc[i][1:])
82
83 print(textod)
84 """
85 equivalentemente:
86 textodd = []
87 for i in range(len(textoc)):
88     textodd.append(textoc[i].capitalize() )
89
90 print(textodd)
91 """
92
93 # unimos todas las oraciones en un sólo texto
94 textoe = ''
95 for i in range(len(textod)):
96     textoe = textoe + textod[i] + '. '
97
98 print(textoe)
99
100 # quitamos los espacios extras delante de los otros signos de puntuación
101 textof = textoe.strip().replace(" ,",",").replace(" ;",";").replace(" :",":")
102 print(textof)
103
104 # Notar que el primer método mantiene las nuevas líneas \n
105
106 quit()
107
108 #####

```

- Desde la terminal ejecute:

```
python3 p2.py
```

e interprete el resultado.

Alternativamente ejecute:

```
python3
```

y vaya agregando uno a uno los bloques del programa.