

# Práctica 1: Introducción a python

José Manuel Bustos Muñoz

## Ejercicios con cadenas

Implementa las siguientes funciones con cadenas y guárdalas en un fichero llamado cadenas.py (los ejercicios de la práctica han sido realizados en Python 3):

1. Implementa la función mezcla, que dada una cadena, devuelva otra cadena con los caracteres con índice par al principio y con índice impar después tras un guión. Ejemplo "ABCDEF" debe dar "ACE-BDF".

---

```
[MBP-de-Jose:p1Python josemanuel$ python cadenas.py ABCDEF
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 1
El resultado de la función utilizada es:
ACE-BDF
```

2. Implementa la función nombre\_propio, que transforme una cadena de forma que la primera letra esté en mayúscula y el resto en minúsculas.

```
[MBP-de-Jose:p1Python josemanuel$ python cadenas.py ABCDEF
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 2
El resultado de la función utilizada es:
Abcdef
```

3. Implementa la función normaliza\_nombre\_fichero, que cambie espacios por \_ y ponga en minúsculas la extensión tras el último punto.

```
[MBP-de-Jose:p1Python josemanuel$ python cadenas.py "nombre fichero.TXT"
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 3
El resultado de la función utilizada es:
nombre_fichero.txt
```

```
MBP-de-Jose:p1Python josemanuel$ python cadenas.py "dejar igual"
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 4
El resultado de la función utilizada es:
dejar igual
```

```
MBP-de-Jose:p1Python josemanuel$ python cadenas.py "texto"  
Opciones(Introducir el número correspondiente a la opción elegida)  
1.- Ejercicio 1  
2.- Ejercicio 2  
3.- Ejercicio 3  
4.- Ejercicio 4  
5.- Ejercicio 5  
6.- Ejercicio 6  
7.- Ejercicio 7  
Escoge una opción: 5  
El resultado de la función utilizada es:  
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>texto<<<<<<<<<<<<<<<<<<<<<<<<<  
MBP-de-Jose:p1Python josemanuel$ python cadenas.py "texto" 20  
Opciones(Introducir el número correspondiente a la opción elegida)  
1.- Ejercicio 1  
2.- Ejercicio 2  
3.- Ejercicio 3  
4.- Ejercicio 4  
5.- Ejercicio 5  
6.- Ejercicio 6  
7.- Ejercicio 7  
Escoge una opción: 5  
El resultado de la función utilizada es:  
>>>>>>>>>texto<<<<<<<
```

6. Implementa la función limpiar, que reciba una cadena c y una cadena de caracteres e, y devuelva la cadena c a la que se le han eliminado todos los caracteres contenidos en e.

```
[MBP-de-Jose:p1Python josemanuel$ python cadenas.py Salida Entrada
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 6
El resultado de la función utilizada es:
Sli
```

7. Implementa una función que reciba una frase y que devuelva la frase con la primera letra de cada palabra en mayúsculas excepto para las palabras de menos de x caracteres, donde x se pasa como parámetro. Adicionalmente, debe devolver el acrónimo de la frase.

```
[MBP-de-Jose:p1Python josemanuel$ python cadenas.py "La práctica me ha gustado pero es un poco difícil" 4
Opciones(Introducir el número correspondiente a la opción elegida)
1.- Ejercicio 1
2.- Ejercicio 2
3.- Ejercicio 3
4.- Ejercicio 4
5.- Ejercicio 5
6.- Ejercicio 6
7.- Ejercicio 7
Escoge una opción: 7
El resultado de la función utilizada es:
La frase capitalizada quedaría así: la Práctica me ha Gustado Pero es un Poco Difícil
El acrónimo sería: lPmhGPeuPD
```

## Procesado de un fichero de datos csv

Ejecución del ejercicio con fichero de ejemplo “small.csv”.

```
MBP-de-Jose:p1Python josemanuel$ python estads.py small.csv mMpñ
1,2.0,2.5,?,tc,38
2,4.0,3.0,?,none,39
1.67,2.83,2.75,?,none,38.5
3,3,2,0,3,2
MBP-de-Jose:p1Python josemanuel$ █
```

## Contar palabras

Ejecución del ejercicio con el fichero “elquijote.txt” y que saque las palabras que se repiten más de 200 veces, sin sacar las contenidas en el segundo fichero pasado “stopwords\_es.txt”.

```
MBP-de-Jose:p1Python josemanuel$ python contar_palabras.py elquijote.txt 200 stopwords_es.txt
allí: 421
amigo: 249
amo: 297
aquel: 486
aquella: 332
aun: 258
caballero: 661
caballeros: 293
cabeza: 214
camino: 245
casa: 334
cielo: 252
cosa: 447
cura: 313
desta: 232
digo: 309
dios: 524
don: 2652
dulcinea: 281
día: 274
días: 240
escudero: 242
fuese: 205
historia: 249
hombre: 259
mal: 453
mano: 304
manos: 239
mas: 239
merced: 899
mesmo: 219
mis: 333
```

Ejecución del ejercicio con los mismos ficheros, pero obteniendo las palabras que se repiten más de 1000 veces.

```
MBP-de-Jose:p1Python josemanuel$ python contar_palabras.py elquijote.txt 1000 stopwords_es.txt
don: 2652
quijote: 2180
respondió: 1062
sancho: 2148
señor: 1062
MBP-de-Jose:p1Python josemanuel$ █
```