

Práctica 2

March 26, 2021

1 Práctica 2 - Seminario de Lenguaje Python 2021

2 Ejercicios

1. Tomando el texto del README.md de numpy, imprima todas las líneas que contienen 'http' o 'https'.
2. Indique la palabra que aparece mayor cantidad de veces en el texto del README.md de numpy.
3. Dado un texto solicite por teclado una letra e imprima las palabras que comienzan con dicha letra. En caso que no se haya ingresado una letra, indique el error. *Ver: módulo string*
4. Retomamos el código visto en la teoría, que informaba si los caracteres "@" o "!" formaban parte de una palabra ingresada

```
[ ]: cadena = input("Ingresa la clave (debe tener menos de 10 caracteres y no
↳contener los símbolos:@ y !):")
if len(cadena) > 10:
    print("Ingresaste más de 10 caracteres")
cant = 0
for car in cadena:
    if car == "@" or car == "!":
        cant = cant + 1
if cant >= 1:
    print("Ingresaste alguno de estos símbolos: @ o !" )
else:
    print("Ingreso OK")
```

Ingresa la clave (debe tener menos de 10 caracteres y no contener los símbolos:@ y !):@@ggg@@!!!

Ingresaste alguno de estos símbolos: @ o !

¿Cómo podemos simplificarlo?

5. Dada una frase y un string ingresados por teclado (en ese orden), e informe la cantidad de veces que se encuentra el string en la frase. No distinguir entre mayúsculas y minúsculas.

Ejemplo 1

- **Para la frase:** "Tres tristes tigres, tragaban trigo en un trigal, en tres tristes trastos, tragaban trigo tres tristes tigres."
- **Palabra:** "tres"

- **Resultado:** 3

Ejemplo 2

- **Para la frase:** “Tres tristes tigres, tragaban trigo en un trigal, en tres tristes trastos, tragaban trigo tres tristes tigres.”
- **Palabra:** “tigres”
- **Resultado:** 2

Ejemplo 3

- **Para la frase:** “Tres tristes tigres, tragaban trigo en un trigal, en tres tristes trastos, tragaban trigo tres tristes tigres.”
- **Palabra:** “TRISTES”
- **Resultado:** 3

6. Dada una frase donde las palabras pueden estar repetidas e indistintamente en mayúsculas y minúsculas, imprimir una lista con todas las palabras sin repetir y en letra minúscula.

frase = """

Si trabajás mucho CON computadoras, eventualmente encontrarás que te gustaría automatizar alguna tarea. Por ejemplo, podrías desear realizar una búsqueda y reemplazo en un gran número DE archivos de texto, o renombrar y reorganizar un montón de archivos con fotos de una manera compleja. Tal vez quieras escribir alguna pequeña base de datos personalizada, o una aplicación especializada con interfaz gráfica, o UN juego simple.

"""

7. Trabajando con los contenidos de los archivos que pueden acceder en el curso:

- `nombres`
- `eval1`
- `eval2`

Copiar el contenido de los archivos en variables de tipo string y realizar.

- generar una estructura con los nombres de los estudiantes y la suma de ambas.
- Calcular el promedio de las notas totales e informar quiénes obtuvieron menos que el promedio.

notas.

8. Con la información de los archivos de texto que se encuentran disponibles en el curso:

- `nombres_1`
- `nombres_2`

Nota: Trabaje con los datos en variables de tipo string.

- Indique los nombres que se encuentran en ambos. **Nota:** pruebe utilizando list comprehension.
 - Genere dos variables con la lista de notas que se incluyen en los archivos: `nombres_1`, `eval1.txt` y `eval2.txt` e imprima con formato los nombres de los estudiantes con las correspondientes nota y la suma de ambas como se ve en la [imagen](#)
9. Escriba un programa que solicite que se ingrese una palabra o frase y permita identificar si la misma es un [Heterograma](#) (tenga en cuenta que el contenido del enlace es una traducción del

inglés por lo cual las palabras que nombra no son heterogramas en español). Un Heterograma es una palabra o frase que no tiene ninguna letra repetida entre sus caracteres.

Tener en cuenta - Lo que no se puede repetir en la frase son sólo aquellos caracteres que sean letras. - No se distingue entre mayúsculas y minúsculas, es decir si en la frase o palabra tenemos la letra “T” y la letra “t” la misma NO será un Heterograma. - Para simplificar el ejercicio vamos a tomar como que las letras con tilde y sin tilde son distintas. Ya que Python las diferencia:

```
>>> 'u' == 'ú'
False
```

Ejemplos

Entreda	¿Heterograma?
cruzamiento	Sí
centrifugados	Sí
portón	Sí
casa	No
día de sol	No
con diez uñas	Sí
no-se-duplica	Sí

10. Escriba un programa que solicite por teclado una palabra y calcule el valor de la misma dada la siguiente tabla de valores del juego Scrabble:

Letra	valor
A, E, I, O, U, L, N, R, S, T	1
D, G	2
B, C, M, P	3
F, H, V, W, Y	4
K	5
J, X	8
Q, Z	10

*Tenga en cuenta qué estructura elige para guardar estos valores en Python

Ejemplo 1

- **Palabra:** “solo”
- **valor:** 4

Ejemplo 2

- **Palabra:** “tomate”
- **valor:** 8

11. ¿Conoces [Pypi](#)? Es un sitio con gran variedad de librerías que podés instalar libremente a través de la herramienta [pip](#). Te permite buscar proyectos según el área de interés. Queremos procesar esta lista de categorías como un string para poder saber las subcategorías tiene cada una. Copia el contenido del archivo [categorias](#) como contenido de una variable string para

poder obtener información de la **cantidad de subcategorías** y la **lista que incluye cada categoría**.

Nota: no es necesario diferenciar las subcategorías que se encuentren anidadas.

Ejemplo: De la categoría ‘Communications’ hay 23 subcategorías, y se debe contar con la lista que se encuentran dentro de la misma.

```
_____  
'Communications':  
_____  
'cant': 23,  
'nombres':  
'BBS',  
'Chat',  
'ICQ',  
'Internet Relay Chat',  
.....  
_____
```

12. La idea es tratar de programar una de las partes principales del juego “Buscaminas”. La idea es que dado una estructura que dice que celdas tienen minas y que celdas no las tienen, como la siguiente:

```
[  
    '-*-*- ',  
    '---*-- ',  
    '-----* ',  
    '*----- ',  
]
```

Generar otra que indique en las celdas vacías la cantidad de bombas que la rodean, para el ejemplo anterior, sería:

```
[  
    '1*3*1 ',  
    '12*32 ',  
    '1212* ',  
    '*1011 ',  
]
```

Nota: Defina al menos una función en el código (si hay mas mejor) y documente las mismas con **docstring** que es lo que hacen.

3 Ejercitación por Python Plus (10 pts)

Elija un ejercicio entre el 7 y el 12 de esta práctica y suba la resolución a su repositorio personal de Github donde subieron archivos en para la Práctica 1.