

# Práctica 1

Seminario de Python - Facultad de Informática - Universidad Nacional de La Plata

2019

*En la cursada vamos a trabajar con Python 3.6, se podrán descargar desde <http://www.python.org/download/>. Las prácticas presentadas en la cursada podrán ser realizadas utilizando cualquier editor de texto (EditPlus, NotePad, etc) o IDE (IDLE, Eclipse, NetBeans, etc) que usted crea conveniente sobre la plataforma que sea de su agrado (Windows, Linux, etc). La cátedra provee una máquina virtual con las herramientas que utilizaremos durante la cursada.*

## 1. Parte I: Conceptos Básicos

- `s.lower()`, `s.upper()`: convertir texto a minúsculas o mayúsculas.
- `s.strip()`: devuelve un string sin espacios del principio o del final.
- `s.isalpha()`/`s.isdigit()`/`s.isspace()`: verifica si el string es sólo caracteres, sólo dígitos, sólo espacios.
- `s.startswith('otro')`, `s.endswith('otro')`: verifica si el string comienza o termina con el substring pasado como parámetro,
- `s.split('delim')`: devuelve una lista de substrings separados por el delimitador dado. `'aaa,bbb,ccc'.split(',')`  
-> `['aaa', 'bbb', 'ccc']`.
- `s.join(lista)`: opuesto a `split()`, une elementos de una lista dada generando un string que interpone entre cada palabra de la lista el string dado. Ej. `'—'.join(['aaa', 'bbb', 'ccc'])`  
-> `aaa—bbb—ccc`
- La impresión de los elementos de un conjunto que contiene palabras acentuadas no puede realizarse usando `print(conjunto)`, dado que su implementación tira errores al ejecutarse, conviene usar:

```
print(", ".join(e for e in conj))
```

1. Dado el siguiente string, generar una lista donde cada elemento sea una palabra, utilizando el espacio (' ') como separador:  
**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."
2. Dada una frase y un string ingresados por teclado (en ese orden), genere una lista de palabras (separadas por ' '), y sobre ella, informe la cantidad de palabras en las que se encuentra el string. No importan las mayúsculas y minúsculas.

3. Genere una nueva lista con todas las palabras de la **frase** dada en el ejercicio 1 en mayúsculas.
4. Dada la lista de palabras generada en el ejercicio 3, arme un string con la frase armada con todas ellas separadas por un único espacio en blanco.
5. 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."

6. Dada la lista generada en el ejercicio 3 genere una estructura a la que se pueda acceder directamente a una lista de strings según la cantidad de caracteres de cada uno.
7. Para registrar las actividades de un usuario en un juego dado se requiere guardar los siguientes datos: nombre del jugador, nivel alcanzado en el juego, puntaje máximo y tiempo de juego. Utilizando la estructura que considere más adecuada, almacenar la información de varios usuarios ingresados desde teclado. Tener en cuenta que se necesita acceder a un usuario determinado en forma directa sin recorrerla.
8. Con la estructura generada en el ejercicio 7, imprimir los datos de un usuario dado sin recorrer la estructura. ¿La elección sobre la estructura fue adecuada? ¿Cuál usó?
9. Con la estructura generada en el ejercicio 7, imprima solos los nombres de los usuarios que jugaron sin recorrer la estructura.
10. Dada la estructura generada en el ejercicio 7 imprimir el usuario que obtuvo mayor puntaje.
11. Dada la estructura del ejercicio 7, ingresar los datos correspondientes a la jugada de un usuario. Si el usuario existe previamente, guardar los datos de mayor puntaje. Luego imprimir el ranking de los 10 mejores puntajes.
12. Dado el siguiente diccionario donde las claves son nombres de animales y los valores representan posiciones:

```
anim={'Gato Montes':2,'Yacare overo':4,'Boa acuática':5}
```

Imprimir un string por cada animal de la estructura, reemplazando sus caracteres por el string '\_' (inclusive los espacios en blanco) salvo el caracter correspondiente al valor del mismo.

Ejemplo: Gato Montes tiene asociado el *valor* 2:

```

_ _ t _ _ _ _ _ _ _
0 1 2 3 4 5 6 7 8 9 10
```

13. Dada una lista con nombres de imágenes:

```
imagenes=['im1','im2','im3']
```

Generar una estructura que asocie 3 coordenadas ingresadas por teclado  $(x_1, y_1)$ ,  $(x_2, y_2)$  y  $(x_3, y_3)$ , con cada elemento de la lista (en el mismo orden en que son ingresadas). Además verifique, mientras se van ingresando las coordenadas, que no hayan repetidas para una misma imagen; en dicho caso deberá volver a ingresarla.

14. Dada una estructura:

```
tam={'im1':[(x_1,y_1),(x_2,y_2)],'im2':[(x_1,y_1),(x_2,y_2)],'im3':[(x_1,y_1),(x_2,y_2)]}
```

Donde  $(x_1, y_1), (x_2, y_2)$  representan el extremo izquierdo inferior y el extremo derecho superior del tamaño de cada imagen, verifique con la estructura generada en el ejercicio 13 si cada coordenada no queda fuera del tamaño de la imagen.

## 2. Parte II: Encontramos errores

1. Corrija los errores de los siguientes códigos:

```
# -*- encoding: utf-8 -*-
puntaje = {}
puntaje[30] = 'Ada'
puntaje[40] = 'Hedy Lammar'
puntaje[45] = 'Colossus'
puntaje[47] = 'Angela Ruiz Robles'
for puntaje, nombre in puntaje.keys():
    print('{} puntaje {}'.format(nombre, puntaje))
```

2. Usando el mismo diccionario del ejercicio anterior, ejecutar:

```
for puntaje in puntaje.keys():
    print(puntaje)

for cada in sorted(puntaje, reverse=True):
    print(puntaje[cada], cada)
```

3. ¿Qué sucede con el siguiente código?

```
lista1 = ['perro', 4, True, (6,7)]
lista2 = [False, 'casa', 9, [3, 4, 'gato']]
lista3 = []

lista_aux = lista1.extend(lista2)

for elem in lista_aux:
    if type(elem) is str:
        lista3.append(elem)

print(lista3)
```

4. Resuelva el problema

```
bb = ['Si', 'mucho', 'con', 'computadoras,', 'eventualmente', 'que', 'te']

for each in bb
    print(len(each))
```