

# Practica1

March 17, 2020

## 1 Practica 1 - Seminario de Lenguaje Python 2020



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.

### 1.1 Parte I: Conceptos Básicos

Algunos comandos para manejo de **String**

`s.split('delim')` > devuelve una lista de substrings separados por el delimitador dado.

```
[1]: texto = ' Esto es un texto con agrego numeros y espacio al principio/ 4567813_
      ↪312/      / espacio y 1 números'
      texto.split('/')
```

```
[1]: [' Esto es un texto con agrego numeros y espacio al principio',
      ' 4567813 312',
      ' ',
      ' espacio y 1 números']
```

El texto no cambio, para guardar los cambios tenemos que igualarlo a una variable

```
[3]: texto_mod = texto.split('t')
      print(texto_mod)
```

```
[' Es', 'o es un ', 'ex', 'o con agregó números y espacio al principio/ 4567813 312/ / espacio y 1 números']
```

Ahora la variable 'texto\_mod' es una lista con varios elementos, para acceder a cada uno podemos hacerlo:

```
[6]: texto_mod[0]
```

```
[6]: ' Es'
```

- **s.lower(), s.upper()** > convertir texto a minúsculas o mayúsculas.

```
[30]: print(texto_mod[0].lower())
```

```
esto es un texto con espacio al principio
```

- **s.strip()** > devuelve un string sin espacios del principio o del final.

```
[32]: texto = ''' En la cursada vamos a trabajar con Python 3.6 '''
      texto.strip()
```

```
[32]: 'En la cursada vamos a trabajar con Python 3.6'
```

**s.isalpha(), s.isdigit(), s.isspace()**: verifica si el string es sólo caracteres, sólo dígitos, sólo espacios y podemos concatenar las funciones

```
[34]: texto.split('/')[0].strip().lower()
```

```
[34]: 'en la cursada vamos a trabajar con python 3.6'
```

- **s.startswith('otro'), s.endswith('otro')** > verifica si el string comienza o termina con el sub-string pasado como parámetro
- **s.join(lista)** > Realiza lo opuesto a **split()**, une elementos de una lista dada generando un string que interpone entre cada palabra de la lista el string dado.

```
[54]: print('@'.join(texto_mod))
```

```
Esto es un texto con espacio al principio@ 4567813 312@ @ espacio y números
```

Para mostrar en pantalla 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: Python `print(", ".join(e for e in conj))`

```
[1]: soyconj= {'4', '5', '6', 'u', 'canción'}
```

```
[61]: print(", ".join(e for e in soyconj))
```

5, u, canción, 6, 4

```
[2]: for each in soyconj:
      print(each)
```

canción

5

4

u

6

## 2 Ejercicios

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.

Modificar el programa para imprima las palabras sin importar mayúsculas ni minúsculas de la palabra que se busca.

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 2, 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."
```

```
[ ]:
```