

LC-para-combinar.pdf



Ssarii



Lingüística Computacional



2º Grado en Lingüística y Lenguas Aplicadas



**Facultad de Filosofía y Letras
Universidad de Cádiz**



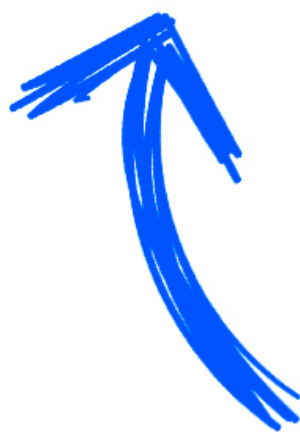
Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Estudiar sin publi es posible.



Compra Wuolah Coins y que nada
te distraiga durante el estudio



Les Roches

Despierta el líder que llevas dentro en Les Roches

Saber más

Lingüística computacional

Viernes 15

Que es un módulo – no todas las funciones del Python están accesibles al instalar Python. Hay cosas que este puede hacer que están encapsulados en Python que cuando se inicia el programa no están visibles. Hay que activarlos. Son ficheros en el que hay operaciones que se pueden hacer pero que no están directamente visibles. Para usarlos hay que importar el módulo.

Los datos mas importantes para nosotros serán las cadenas o “streams”. Para que Python lo reconozca tiene que ir entre comillas.

```
>>>A='23' cuando le damos a x 2 nos pondrá dos veces 23 porque es un stream.
```

El programa no sabe lo que hacer cuando se mezclan datos. El stream tiene que ir con el stream.

```
>>>Int(a) lo que hace es pasar el stream a número entero.
```

Open abrir primero ruta (directorio y luego archivo) luego decir para que quiero y final la codificación en la que está. Si esto nos da error es porque hemos puesto mal la codificación

Esto solo lo abre

Para leerlo hay que poner `texto1=f.read()` (f porque le asignamos f al fichero antes. El texto1 delante se pone para que lo asigne a una variable, si no lo ponemos nos abrirá en pantalla y nos petara el pc.

```
>>>f.close() después de leer un fichero siempre cerrarlo con esto para no dañar el fichero
```

Para los encoding: o es utf8 o es latin1

Consejo: Guardar todos los ficheros en una ruta por ejemplo “C:/user/Antonio/Documentos/”

```
>>>Ruta='C:/user/Antonio/Documentos/' (para guardarlos)
```

Luego pasar el texto1 a cadena

```
>>>Type(texto1)
```

Luego poner los caracteres

```
>>>Texto1[:200]
```

Esto nos dará los 200 primeros caracteres del documento

Si queremos los 200 ultimos ponemos `[-200:]`

```
>>>Len(texto1)
```

Esto nos dará la los caracteres que tenga el texto1

Para acceder a posiciones individuales de la cadena hay que poner texto1 [2] esto nos dará la posición 2 del texto1. Todos empiezan en 0

Para acceder al ultimo carácter ponemos [-1]

También se puede acceder a paquetes de elementos, indicando la primera posición y la segunda posición

>>>Texto1[45:89] Esto nos dará desde el elemento 45 hasta el 88 (no da el ultimo elemento)

Para decirle de cuantos en cuantos

>>>Texto1[45:89:3] de la posición 45 a 89 de 3 en 3

Las listas son variables puede haber strings números letras de todo

>>>Palabras=texto1.split() palabras dividirá el string texto 1 por los espacios en blanco
()

El punto . marca las operaciones

>>>'Hacia'.islower()

False

>>>'Hacia'[0].isupper()

True

Viernes 22

Las grandes ordenes son los bucles y el findall y los condicionales.

```
>>>N=10
```

```
>>>While= n>0:
```

```
    Print(n)
```

Eso lo que hará será escribir 10 infinitos.

Si se peta hacer control c y se interrumpe

While n>0:

```
    Print(n)
```

```
    n-=1
```

escribirá 10-9-8-7... hasta 1

Los bucles son útiles pero hay que ponerles una condición falsa para que no sigan hasta el infinito.

Lo único que se diferencian siempre en las tareas que haremos son los datos, pero muchas de esas tareas serán iguales.

El Python nos permite construir funciones que nos permiten hacer una determinada tarea.

¿Cuántas veces a parece una palabra en un texto?

Decidimos dos cosas cual es el texto (variable 1) y cual es la palabra (variable 2)

```
>>> def
```

(esto sirve para construir procedimientos)

Hay que ponerle un nombre a esa función (ponerle un nombre que recordemos)

```
>>>def buscar_palabra_texto(texto,palabra):    para una palabra
```

```
>>> def buscar_palabras_texto(texto,palabras):    para un grupo de palabras
```

```
    """busca palabras en un texto.    (aquí se pone lo que queremos que haga la función)
```

```
    (Esto se deja en blanco)
```

Texto es una variable tipo string (aquí ponemos el tipo de datos que tiene que analizar)

Palabras es una lista de string

Devuelve el numero de palabras en el texto"""

Import re,collections (hay que importar el módulo)

(a partir de aquí tenemos que construir nuestra función)

Les Roches

Despierta el líder que llevas dentro en Les Roches

Saber más

Texto=texto.lower() (esta orden convierte todo el texto a minúscula para que no afecte a la hora de bucar las palabras)

Texto1=re.findall(r"\w+",texto) (con esto devidimos el texto en palabras que la orden siguiente pueda contar)

Frecuencias=collections.Counter(texto1) (con esto nos contara las palabras del texto uno)

For x in palabras: (para buscar la palabra que queremos)
Print(x, frecuencias [x]) (para escribir la palabra que digamos)

Después de esto le damos intro y ya tenemos la función hecha

Recordar que para escribir las funciones empezamos con def

(escribir las funciones para poder hacer los ejercicios más 4ácilmente)

Para buscar las palabras con la función

Abrimos el texto como ya sabemos, es decir:

```
>>>with open (y la ruta donde este, y la codificación en la que esta normalmente 'latin1')
```

```
>>> text=f.read()
```

```
>>>text [:100] (para comprobar que es el texto corrector)
```

Y para contar las palabras que queremos se hace lo siguiente:

```
>>>words=['sombrero','corregidor','con'] (ponemos las palabras que queramos)
```

```
>>>buscar palabras texto(text,words) text porque es el nombre que le hemos dado a texto y words pq es el nombre que le hemos dado a esa variable.
```

(y saldrán el numero de palabras que le hayamos pedido que busque)

Recordar crear un modulo con nuestro nombre que contenga todas las funciones que necesitamos para los problemas. Y luego lo importamos.

```
>>>import.aleixperearnau (para importar el módulo)
```

Conclusión

Una parte importante de la programación consiste en crear funciones de uso general para cualquier texto y cualquier lista de palabras. Estas se escriben con la palabra def, y luego tenemos que usarla.

Para buscar expresiones no tendríamos que segmentar el texto entonces la orden sería:

```
>>>def buscar_palabras_texto(texto,palabras):
```

```
"""busca palabras en un texto.
```

texto es una variable tipo string

palabras es una lista de string

devuelve el numero de palabras en el texto"""

```
import re,collections
```

```
texto=texto.lower()
```

```
texto1=re.findall(r"\w+",texto)
```

```
———frecuencias=collections.Counter(texto1)
```

```
———for x in palabras:
```

```
———print(x,frecuencias[x])
```

Lo que esta tachado no se pondría

En su defecto pondríamos la orden

```
Texto.count()
```

Quedaría:

```
>>>def buscar_expresiones_texto(texto,palabras):
```

```
"""busca palabras en un texto.
```

texto es una variable tipo string

palabras es una lista de string

devuelve el número de palabras en el texto"""

```
import re,collections
```

```
texto=texto.lower()
```

```
texto.count()
```

Así escribiríamos la función, luego hay que usarla... Recordar

Para crear un modulo tenemos que guardarlo en un fichero pei es decir de texto plano, abrimos Python escribimos las funciones y las guardamos en fichero pei con nuestro nombre para recordarlo.

Otra función posible:

```
>>>def.ordenar_frecuencia_palabra(texto,palabras):
```

```
    Texto=texto.lower() para pasarla a minúscula
```

```
    A=[]                esto nos hará una lista con las palabras
```

For x in palabras:

a.append(x,texto,count(x)) para contra la palabra x en la lista y ordenarla

(Esta es un ejemplo no cuenta)