Escalona_Joaquin_4

September 3, 2018

Hola, pueden encontrar este documento y los códigos aquí copiados para que puedan ejecutarlos y evaluarlos con mayor facilidad pinchando aquí :) GitHub

Ejercicio 1

Escriba una función que, dada una cadena de characteres y un desplazamiento, devolverá la cadena cifrada para ese cambio. Tenga en cuenta que la misma función se puede utilizar para descifrar un mensaje, pasando un cambio negativo. Solo debe aceptar y devolver letras minúsculas, y los espacios no deben cambiarse. Descifre el siguiente mensaje, que fue encriptado con un desplazamiento de 13:

pbatenghyngvbaf lbh unir fhpprrqrq va qrpelcgvat gur fgevat

Solución

```
#-*- coding: utf-8 -*-
def des_encriptacion(cadena,desp):
        abc = 'abcdefghijklmnopqrstuvwxyz'
        #respuesta final
        res = ""
        for letra in cadena:
        #encontrar la posicion de cada letra
                pos = abc.find(letra)
        #el método .find retorna -1 si no encuentra el caracter en el abc
        #ejemplo, el espacio(" ").
        #https://www.tutorialspoint.com/python/string_find.htm
                if pos == -1:
        #si no encuentra caracter en el abc se debe añadir este caracter a res. (Idea de Diego Salvador)
                        res = res + letra
                        continue
        #posicion de la letra desplazada desp letras
                des = pos + desp

        #al hacer len(abc) retorna 26, si la variable desplazamiento es mayor a 26
        #no tiene sentido buscar el elemento, por ejemplo 27 en el abc, ya que retornará un "fuera de rango"
        #para solucionarlo:
                if des >=26:
                        des = des - 26
        #ejemplo, si desp = 3, al llegar a una "z" debiera retornar "c", si no estuviera el if, en la linea
        #de abajo saldrá un error ya que abc [25+3] no existe. Sin embargo 28-26 = 2 y abc[2] = 'c'.
        #Esto sirve para cualquier desplazamiento.
                res = res + abc[des]
        return res

print('--------- METODO DE CESAR ---------\n')

ask = raw_input('Quieres encriptar o descrifar un mensaje? (ingresa 1 para encriptar, 2 para descrifrar) = ')

#while con múltiples condiciones
#https://stackoverflow.com/questions/2146419/how-to-do-while-loops-with-multiple-conditions
while not ask == "1" and not ask == "2":
```

```
        ask = None
        ask = raw_input('Ingresa 1 [encriptar] o 2 [descifrar] (CTRL + C para cerrar) = ')

if ask == '1':
        print '******* ENCRIPTACION *******'
        enp = raw_input('Ingresa un mensaje para ser encriptado \n')
        dsp = int(raw_input('Ingresa el desplazamiento de cifrado (debe ser negativo)\n'))
        while dsp > 0:
                dsp = None
                dsp = int(raw_input('Debe ser un valor negativo!\n'))
        print "********************"
        print "El siguiente mensaje está encriptado con un desplazamiento de",abs(dsp),"letras\n"
        print des_encriptacion(cadena=enp,desp=dsp)

elif ask == '2':
        print '******* DESENCRIPTACION *******'
        enp = raw_input('Ingresa un mensaje para ser descifrado \n')
        dsp = int(raw_input('Ingresa el desplazamiento con el cual fue encriptado (debe ser positivo)\n'))
        while dsp < 0:
                dsp = None
                dsp = int(raw_input('Debe ser un valor positivo!\n'))
        print "********************"
        print " Mensaje revelado = "
        print des_encriptacion(cadena=enp,desp=dsp)
```

Al solicitar al programa que descifre el código propuesto por la profesora, con un desplazamiento igual a 2, el programa imprime:

```
********************
Mensaje revelado =
congratulations you have succeeded in decrypting the string
```

Ejercicio 2

Intente descifrar la frase siguiente y encontrar el desplazamiento: **gwc uivioml bw nqvl bpm zqopb apqnb**

Para ambos ejercicios: Puede usar las funciones incorporadas "chr" y "ord" (y recuerde que puede obtener más información sobre una función al usar? En IPython). Otra es configurar el alfabeto en una cadena y usar el acceso a elementos ([4]) para convertir de números a letras, y el método de índice para convertir de letras a números.

Solución

Para la resolución de este ejercicio utilicé el mismo programa del Ejercicio 1, sólo con unas leves modificaciones (que son donde van los nuevos comentarios)

```
#-*- coding: utf-8 -*-
def des_encriptacion(desp):
        #imprime el valor de desp para saber cuanto es el desplazamiento
        print desp
        abc = 'abcdefghijklmnopqrstuvwxyz'
        #cadena a descifrar
        cad = "gwc uivioml bw nqvl bpm zqopb apqnb"
        res = ""
        for letra in cad:
                pos = abc.find(letra)
                if pos == -1:
                        res = res + letra
                        continue
                des = pos + desp
                if des >=26:
                        des = des - 26
                res = res + abc[des]
        return res

#probar en un rango de 1 hasta 26, que es la cantidad de letras del alfabeto
#luego buscar hasta encontrar una frase con sentido (desp = 18)
for i in range(1,27):
        print des_encriptacion(i)
```

Entre las soluciones que imprimió el programa están:

```
15
vlr jxkxdba ql cfka qeb ofdeq pefcq
16
wms kylyecb rm dglb rfc pgefr qfgdr
17
xnt lzmzfdc sn ehmc sgd qhfgs rghes
18
you managed to find the right shift
19
zpv nbobhfe up gjoe uif sjhiu tijgu
20
aqw ocpcigf vq hkpf vjg tkijv ujkhv
```

Vemos que con un desplazamiento de 18, el mensaje encriptado "gwc uivioml bw nqvl bpm zqopb apqnb" se revela como **"you managed to find the right shift"**