

PROYECTO PYTHON

Ordenamiento burbuja

15/02/2019

Fernando García Ballesteros

Saúl Lozano Selva

Índice

Introducción	2
Código	2
Explicación del código.	3
Conclusión	4

Introducción

En esta ocasión hemos realizado un programa que se encarga de recoger los números que le escribamos y ordenarlos de menor a mayor. El proyecto está elaborado en Python, y para llevarlo a cabo hemos contado con una semana y media de trabajo, la cual en su mayoría hemos dedicado a la búsqueda de información debido a la complejidad y novedad de este código para nuestro actual nivel.

Código

```
from tkinter import *
ventana = Tk()
ventana.title("BOTONES")
ventana.geometry("2000x1500")
LISTA=[]

etiqueta2 = Label(ventana, text=";Dime un numero!", width=20)
etiqueta2.place(x=0, y=100)
texto1 = Entry(ventana,width=40)
texto1.place(x=150, y=100)

def clicleado():
    respuesta = texto1.get()
    LISTA.append (respuesta)
    etiqueta3=Label(ventana,text= LISTA, width=40)
    etiqueta3.place(x=150, y=200)
    return (LISTA)

def ordenamientoBurbuja():
    for numPasada in range(len(LISTA)-1,0,-1):
        for i in range(numPasada):
            if LISTA[i]>LISTA[i+1]:
                temp = LISTA[i]
                LISTA[i] = LISTA[i+1]
                LISTA[i+1] = temp
    etiqueta4=Label(ventana,text= LISTA, width=40)
    etiqueta4.place(x=150, y=220)

boton = Button(ventana, text="ACEPTAR",width=20,bg="black",fg="red",command=clicleado)
boton.place(x=150,y=120)
boton = Button(ventana, text="ORDENAR",width=20,bg="black",fg="red",command=ordenamientoBurbuja)
boton.place(x=150,y=150)
ventana.mainloop()
```

ORDENAMIENTO BURBUJA

-5 3 7 1 9 2

Pon un número

4

ACEPTAR

ORDENAR

-5 1 2 3 7 9

Explicación del código.

```
from tkinter import *
ventana = Tk()
ventana.title("BOTONES")
ventana.geometry("2000x1500")
LISTA=[]

etiqueta2 = Label(ventana, text="¡Dime un numero!", width=20)
etiqueta2.place(x=0, y=100)
texto1 = Entry(ventana,width=40)
texto1.place(x=150, y=100)
```

En la primera parte del código, importamos la librería tinker en la primera línea. En las siguientes líneas, abrimos una ventana y la configuramos, para que sus medidas sean las que aparecen y el título igual.

Posteriormente, establecemos la configuración de una etiqueta en la que pondrá "¡Dime un número!", diciendo su posición y la del texto dentro de esta.

```
def clicleado():
    respuesta = texto1.get()
    LISTA.append (respuesta)
    etiqueta3=Label(ventana,text= LISTA, width=40)
    etiqueta3.place(x=150, y=200)
    return (LISTA)
```

Definimos la función que se lleva a cabo a la hora de clicar a la etiqueta y que se imprimen los valores que ponemos en el cuadro de texto y que los guarde.

```
def ordenamientoBurbuja():
    for numPasada in range(len(LISTA)-1,0,-1):
        for i in range(numPasada):
            if LISTA[i]>LISTA[i+1]:
                temp = LISTA[i]
                LISTA[i] = LISTA[i+1]
                LISTA[i+1] = temp
    etiqueta4=Label(ventana,text= LISTA, width=40)
    etiqueta4.place(x=150, y=220)
```

Definimos la función que se encarga de ordenar los números que hemos introducido anteriormente en la ventana y los imprime en la misma.

```
boton = Button(ventana, text="ACEPTAR",width=20,bg="black",fg="red",command=clicleado)
boton.place(x=150,y=120)
boton = Button(ventana, text="ORDENAR",width=20,bg="black",fg="red",command=ordenamientoBurbuja)
boton.place(x=150,y=150)
ventana.mainloop()
```

Aquí definimos y posicionamos los botones que nos ayudaran a poder ejecutar los comandos anteriores, el primero depende de la primera función, la cual guarda los números que queramos, y el segundo de la otra, la cual ordena estos números.

Conclusión

Como conclusión es destacable la dificultad de este proyecto debido a nuestro poco conocimiento de este lenguaje de programación. Además en esta ocasión, a parte del poco conocimiento, se ha unido la dificultad del proyecto en sí, pues no sabíamos cómo relacionar los números guardados con el comando que los ordenaba. El único fallo de nuestro código es que solo ordena números de un solo dígito, por lo que se mueve de un rango de -9 a 9.

