

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

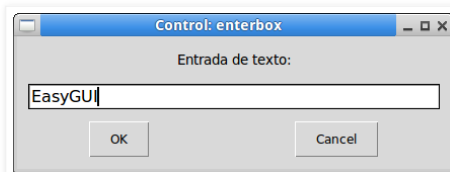
Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

martes, 20 de mayo de 2014

Entrada de datos con EasyGUI

Después de ver en el [capítulo anterior](#) las funciones que se utilizan para mostrar ventanas de mensajes, continuamos con algunas de las funciones que permiten a los usuarios crear ventanas de diálogo para introducir datos en un programa: **enterbox()**, **integerbox()**, **codebox()**, **textbox()** y **multenterbox()**.

1) enterbox: entrada de texto

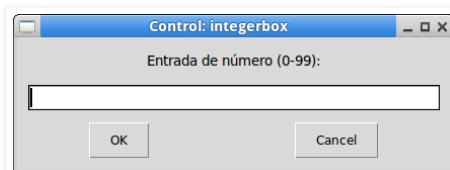


La función **enterbox()** se utiliza para mostrar una ventana donde se permite introducir una cadena de texto.

El argumento **strip** si tiene el valor **True** suprime los espacios al principio y final de la cadena introducida, **default** contiene la cadena inicial y editable que será mostrada en la ventana, **title** es el título de la ventana, **msg** es el mensaje de la ventana e **image** sirve para indicar la ruta y el nombre de una imagen a mostrar.

```
texto = eg.enterbox(msg='Entrada de texto:',
                    title='Control: enterbox',
                    default='EasyGUI', strip=True,
                    image=None)
```

2) integerbox: entrada de número entero



La función **integerbox()** se usa para mostrar una ventana con una caja donde es posible introducir un número entero. El valor introducido no será aceptado hasta que sea un número entero. Si se han establecido los límites mínimo y máximo (con los argumentos **lowerbound** y **upperbound**) el número deberá ajustarse a éstos.

```
num = eg.integerbox(msg='Entrada de número (0-99):',
                    title='Control: integerbox',
                    default='',
                    lowerbound=0,
                    upperbound=99,
                    image=None)
```

3) codebox y textbox: cajas de texto editable

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6 . El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [..], ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valores...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

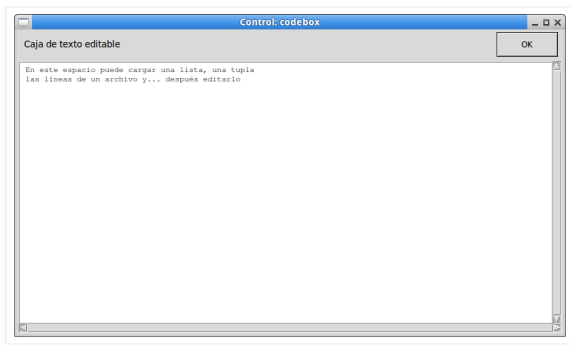
Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

[Threading: programación con hilos \(I\)](#)

En programación, la técnica que permite que una aplicación ejecute



Las funciones `codebox()` y `textbox()` se utilizan para mostrar una ventana con una caja de texto que le permite al usuario introducir o editar texto en varias líneas. También, estas cajas pueden emplearse para mostrar los elementos de una lista, una tupla o las líneas de un archivo con la posibilidad de editar la información. La diferencia entre `codebox()` y `textbox()` está en la fuente que se utiliza por defecto para mostrar el texto. Con `codebox()` la fuente es proporcional, que es más adecuada para editar el código fuente de los programas.

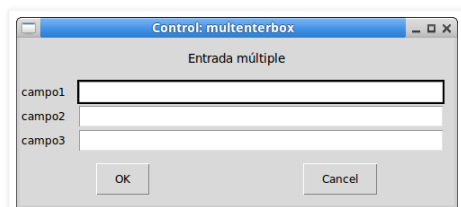
En la función `codebox()` el argumento `msg` es para el mensaje, `title` es para el título de la ventana y `text` es para establecer el texto inicial que será mostrado en la ventana. La función `textbox()` cuenta además con el argumento `codebox` que con el valor "1" fija la fuente en proporcional.

```
texto = ['En este espacio puede cargar una lista, una tupla\n',
        'las líneas de un archivo y... después editarlo']

texto1 = eg.codebox(msg='Entrada de fuente',
                   title='Control: codebox',
                   text=texto)

texto2 = eg.textbox(msg='Entrada de texto:',
                   title='Control: textbox',
                   text='', codebox=0)
```

4) multenterbox: entrada múltiple



La función `multenterbox()` muestra una ventana que permite introducir al usuario los datos de varios campos.

El argumento `fields` es una lista con los literales que identifican a cada campo y el argumento `values` es una tupla que contiene los valores iniciales de cada campo y es opcional.

```
campos = ['Nombre', 'Correo', 'Tel']
datos = []
datos = eg.multenterbox(msg='Entrada múltiple',
                      title='Control: multenterbox',
                      fields=campos, values=())

cadena = ''
if datos != None:
    for cam, dat in zip(campos, datos):
        cadena = cadena + cam + ': ' + dat + '\n'

eg.msgbox(cadena, 'multenterbox', ok_button='Seguir')
```

[Ir al índice del tutorial de EasyGUI](#)

Publicado por Pherkad en [14:04](#)



Etiquetas: [EasyGUI](#), [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

simultáneamente varias operaciones en el mismo espacio de proceso se...

Archivo

mayo 2014 (6) ▼

python.org



pypi.org



Sitios

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España
. Tema Sencillo. Con la tecnología de [Blogger](#).