

Elementos comunes en PySimpleGUI

Actualizado : 16 – Nov - 21

Botón (BUTTON)



El elemento más común es el Botón. Cuando es presionado se ejecuta la acción definida en el ciclo while del programa.

Creación de un botón

Button text!

```
button1 = sg.Button('Button text!', key = 'BTN1')
```

```
button1 = sg.Button('Button text!', property1 = value, property2 = value, ... key = 'BTN1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#button-element>

Property	Value
visible	True/False
disabled	True/False
size	(width, height)
image_filename	'image.png'

Define a set of actions for a button

Dentro del ciclo **while**, usa un **if** para verificar si la variable **event** es igual == a la **key** del Botón

```
#Initiate an infinite loop
while True:
    #Use the window.read() method to catch any button pressed.
    event, values = window.read()
    # End program if user closes window
    if event == sg.WIN_CLOSED:
        break
    #Do something if the button| is pressed
    if event == 'BTN1': 
        #Actions
        print('Hello')
        window.close()
```

TEXTO (Etiqueta)

This element is a Text box

Este elemento es utilizado para mostrar información o para dar alguna instrucción al usuario.

Como crear una etiqueta

```
text1 = sg.Text('Text')
```

This element is a Text box

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#text-element>

```
text1 = sg.Text('Text', property1 = value, property2 = value, ... key = 'TXT1')
```

Property	Value
visible	True/False
text_color	<u>check the name colors</u>
size	(characters, lines)
font	('font', size)

INPUT (Entrada de datos)



Permite al usuario proporcionar datos mediante el teclado.

Como crear un elemento de Entrada



```
input1 = sg.Input('Text', key = 'INP1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#input-element>

```
input1 = sg.Input('Text', property1 = value, property2 = value, ... key = 'INP1')
```

Property	Value
visible	True/False
disabled	True/False
size	(characters, None)
password_char	'*' (or any symbol)

Image

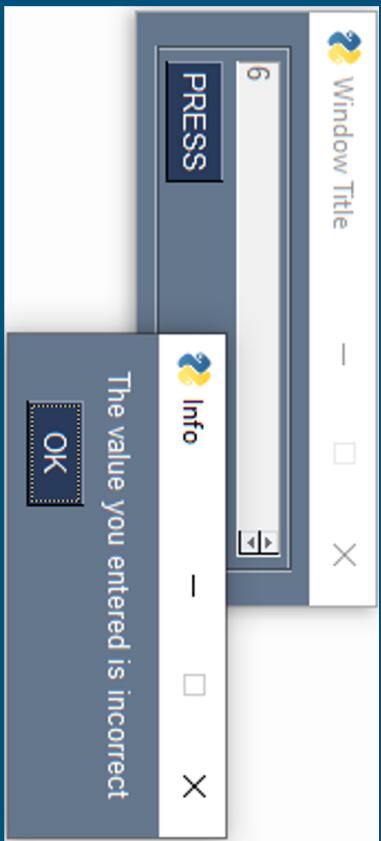


Usa la siguiente sintaxis para insertar una imagen (imágenes PNG o GIF)

```
img = sg.Image(filename = 'image.png', key = 'IMG1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#image-element>

Popup message (ventana emergente)



Utiliza la siguiente sintaxis para invocar una ventana emergente.

```
sg.popup('Message', title = 'Title')
```

Cambiar las propiedades de los elementos

Si quieras modificar alguna propiedad de los elementos (excepto el *tamaño*) usa la sig. sintaxis.

```
window['KEY'].update('Text', property1 = value, property2 = value,...)
```

Para cambiar el tamaño, se utiliza:

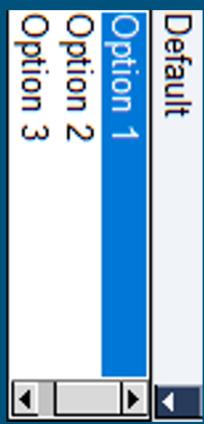
```
window['KEY'].set_size(width, height))
```

Combo (lista desplegable)



Se utiliza un Combo cuando se desea seleccionar una opción de una lista de opciones.

Creación de un Combo



```
choices1 = ('Option 1', 'Option 2', 'Option 3', 'Option 4')
combo1 = sg.Combo(choices1, key = 'COMBO1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#combo-element>

```
combo1 = sg.Combo(choices1, property1 = value, property2 = value,... key = 'COMBO1')
```

Property	Value
visible	True/False
disabled	True/False
size	(characters, num_of_options)
default_value	'string'

Cómo obtener el valor de un Input

Para recuperar la información proporcionada por el usuario en un Input, se utiliza la siguiente sintaxis.

```
var = values['INP1']
```

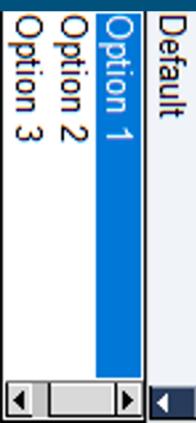
Nota: el valor tecleado en un Input es de tipo string (str)

Obtener el valor de un Combo

Para guardar en una variable el valor elegido por el usuario, se utiliza la sig. sintaxis:

```
var = values['COMBO1']
```

Nota: el valor elegido de un Combo conserva su tipo de dato.



Checkbox (botón de selección)



Se utiliza un Checkbox cuando se desea determinar un valor de verdadero o falso de una selección.

Como crear un Checkbox



```
checkbox1 = sg.Checkbox('Option', key = 'CHK1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#checkbox-element>

```
checkbox1 = sg.Checkbox('Option', property1 = value, property2 = value, ... key = 'CHK1')
```

Property	Value
visible	True/False
disabled	True/False
size	(characters, None)
default	True/False

Cómo determinar si un Checkbox fue seleccionado o no

Para determinar si la opción de un Checkbox fue seleccionada o no, se utiliza la sig. sintaxis

```
var = values['CHK1']
```

Nota: los únicos valores posibles son True o False

Radio buttons (grupo de botones)



Se utilizan los Radio Buttons para seleccionar una sola opción de un conjunto posible de opciones (similar al Combo)

Como crear un grupo de Radio Buttons

Radio 1 Radio 2

Observa que el grupo_id debe ser el mismo

```
radio1 = sg.Radio('Radio 1', group_id = 'RadGroup', key = 'RB1')
radio2 = sg.Radio('Radio 2', group_id = 'RadGroup', key = 'RB2')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#radio-element>

```
radio1 = sg.Radio('Radio 1', group_id = 'RadGroup', property1 = value,... key = 'RB1')
```

Property	Value
visible	True/False
disabled	True/False
size	(characters, None)
default	True/False

Como determinar si un Radio Button fue seleccionado

Para saber si un Radio Button fue seleccionado o no, se utiliza la sig. sintaxis

```
var = values['RB1']
```

Nota: los únicos valores posibles son True o False.

Slider (lista deslizable)



Un slider permite elegir un número dentro de un rango de valores

Cómo Crear un Slider

Esto crea un slider vertical del 1 al 10.



```
slider1 = sg.Slider(range = (1, 10), key = 'SL1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#slider-element>

```
slider1 = sg.Slider(range = (1, 10), property1 = value, property2 = value, ... key = 'SL1')
```

Property	Value
default_value	Any number in the given range
resolution	How much the numbers change when you move the slider.
orientation	'horizontal' or 'vertical'
size	(width, height)

Cómo obtener el valor de un Slider

Para guardar el número elegido por el usuario en una variable, se utiliza la sig. sintaxis.

```
var = values['SL1']
```

Nota: el valor regresado siempre es de tipo numérico.

Spinbox (ruleta)



Similar a un slider, el usuario debe presionar las flechas para elegir el número deseado.

Creación de un Spinbox

Esto crea un Spinbox del 1 al 9

```
spin1 = sg.Spin(tuple(range(1, 10)), key = 'SPN1')
```

<https://pysimplegui.readthedocs.io/en/latest/call%20reference/#spin-element>

```
spin1 = sg.Spin(tuple(range(1, 10)), property1 = value, ..., key = 'SPN1')
```

Property	Value
initial_value	Any number in the given range
disabled	True/False
visible	True/False
size	(characters, None)

Cómo obtener el valor de un Spinbox

—
Usa la siguiente sintaxis:

```
var = values['SPN1']
```

Nota: var es de tipo numérico.