

## 2.2. EL MÓDULO messagebox



Todos los sistemas operativos visuales definen ventanas con mensajes sencillos con los que cualquier aplicación informa al usuario de algún hecho o le hace una pregunta. Estas ventanas bloquean la aplicación hasta que el usuario responde pulsando uno de sus botones.

El módulo `tkinter.messagebox` define la clase `Message` a partir de la clase `tkinter.commondialog.Dialog`. Ambas clases se usan a nivel interno, por lo que el desarrollador no suele usarlas directamente en sus aplicaciones, limitándose a usar una de las ocho funciones definidas en el módulo.

Tres de ellas muestran un mensaje con el botón OK y un icono que da información visual sobre el tipo de mensaje.

`showinfo()`      `showwarning()`      `showerror()`

Las tres devuelven un string que representa al botón pulsado por el usuario, 'ok' si no se ha modificado el tipo de mensaje.

Las otras cinco funciones formulan una pregunta al usuario y presentan dos o tres botones por que responde

`askyesno()` presenta los botones YES y NO, devolviendo el valor bool True o False.

`askquestion()` muestra el mismo diálogo que `askyesno()` pero devuelve los string 'yes' o 'no'.

`askokcancel()` presenta los botones OK y CANCEL y devuelve True o False.

`askretrycancel()` presenta los botones RETRY y CANCEL y devuelve True o False.

`askyesnocancel()` presenta los botones YES, NO y CANCEL y devuelve el bool True o False en los dos primeros casos y None en el tercero.

Todas las funciones tienen el mismo prototipo, sus argumentos son (`title=None`, `message=None`, `**options`). Los dos primeros parámetros son de tipo string. Las opciones son del tipo `clave=valor`, donde la clave puede ser, además de `title` o `message`:

```
from tkinter import TK
import tkinter.messagebox as mb
v = TK()
mb.showinfo('Ventana de bienvenida', '¡Hola!')
v.mainloop()
```

**default** especifica el botón por defecto, pulsado con la tecla RETURN. Puede tomar el valor 'abort', 'retry', 'ignore', 'ok', 'cancel', 'yes' o 'no'. La tecla ESPACIO pulsa el botón que tiene el foco del teclado, no necesariamente el botón por defecto.

**detail** añade una explicación (más detallada) al mensaje, con fuente más pequeña y sin usar negrita.

**icon** puede tomar el valor 'error', 'info', 'question' o 'warning'.

**parent** no debemos modificarlo.

**type** permite cambiar el tipo de mensaje especificando uno de los valores 'abort', 'retry', 'ignore', 'ok', 'okcancel', 'retrycancel', 'yes' o 'yesnocancel'.

**command** se usa para llamar a una función con el string que represente al botón pulsado ('abort'...). En la función llamada podemos usar su único parámetro para gestionar la respuesta del usuario. Si se especifica un comando en cualquiera de las funciones `show*()`, la función `show*()` devuelve siempre una cadena vacía, los `ask*()` devuelven siempre False.

```

from tkinter import TK
import tkinter.messagebox as mb
def f(r):
    m = 'Has pulsado ' + r
    mb.showinfo('Tu respuesta', m)
v = TK()
mb.showwarning('Decide', 'Pulsa un botón', detail='No lo pienses...',
              type='yesnocancel',
              v.mainloop()
              default='yes', command=f)
  
```