
PyMsgBox Documentation

Release 0.9.0

Al Sweigart

Nov 20, 2017

Contents

1	Quickstart	3
2	PyMsgBox Basics	5
2.1	Installation	5
2.2	Usage	5
2.3	Timeout	7
2.4	Localization	8
3	Native Message Boxes	9
3.1	Support	9
3.2	Special Notes	9
4	Indices and tables	11

Contents:

CHAPTER 1

Quickstart

On Linux Python 2, you need to first install Tkinter by running: `sudo apt-get install python-tk`
All of the arguments to PyMsgBox functions are optional.

```
>>> import pymsgbox
>>> pymsgbox.alert('This is an alert.')
>>> pymsgbox.confirm('Click OK to continue, click Cancel to cancel.')
>>> pymsgbox.prompt('Enter your name.')
>>> pymsgbox.password('Enter your password. (It will appear hidden in this text field.
↪)')
```

Here are the default arguments for each function.

```
>>> pymsgbox.alert(text='', title='', button='OK')
>>> pymsgbox.confirm(text='', title='', buttons=['OK', 'Cancel'])
>>> pymsgbox.prompt(text='', title='', defaultValue='')
>>> pymsgbox.password(text='', title='', defaultValue='', mask='*')
```

To use native operating system message boxes, run the following:

```
>>> import pymsgbox.native as pymsgbox
```

(Only a few of the message boxes have native support. Unsupported message boxes will default to the normal message boxes.)

2.1 Installation

PyMsgBox can be installed from PyPI with pip:

```
pip install PyMsgBox
```

OS X and Linux may require sudo to install the module:

```
sudo pip install PyMsgBox
```

PyMsgBox uses Python's built-in Tkinter module for its GUI. It is cross-platform and runs on Python 2 and Python 3.

PyMsgBox's code is derived from Stephen Raymond Ferg's EasyGui. <http://easygui.sourceforge.net/>

2.2 Usage

There are four functions in PyMsgBox, which follow JavaScript's message box naming conventions:

- `alert(text='', title='', button='OK')`

Displays a simple message box with text and a single OK button. Returns the text of the button clicked on.

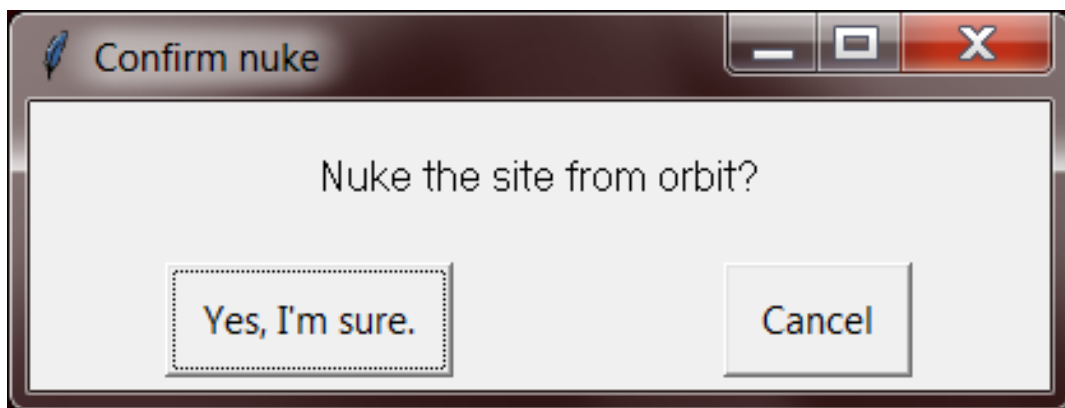
```
>>> pymsgbox.alert('This is an alert.', 'Alert!')
'OK'
```



- `confirm(text='', title='', buttons=['OK', 'Cancel'])`

Displays a message box with OK and Cancel buttons. Number and text of buttons can be customized. Returns the text of the button clicked on.

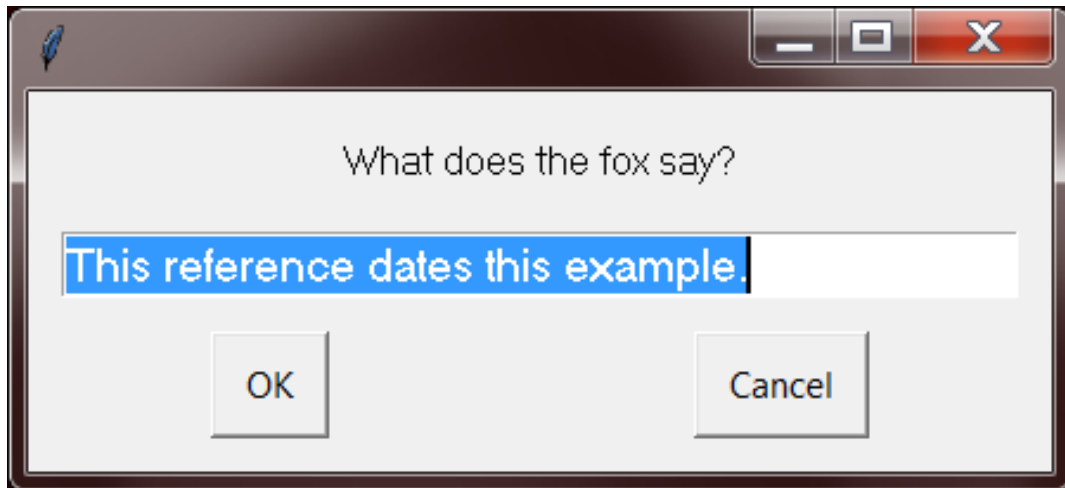
```
>>> pymsgbox.confirm('Nuke the site from orbit?', 'Confirm nuke', ["Yes, I  
↪'m sure.", 'Cancel'])  
"Yes, I'm sure."
```



- `prompt(text='', title='', defaultValue='')`

Displays a message box with text input, and OK & Cancel buttons. Returns the text entered, or None if Cancel was clicked.

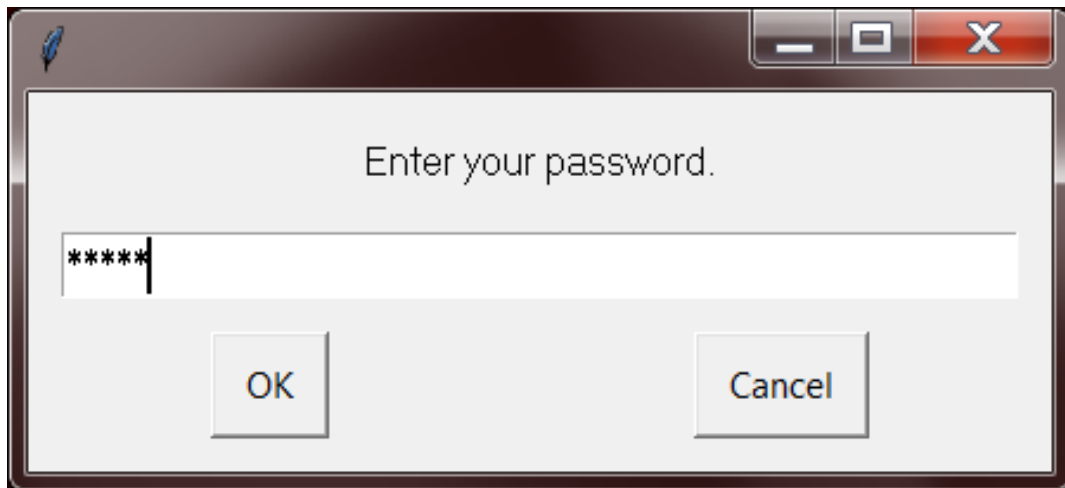
```
>>> pymsgbox.prompt('What does the fox say?', default='This reference_  
↪dates this example.')  
'This reference dates this example.'
```



- `password(text='', title='', defaultValue='', mask='*')`

Displays a message box with text input, and OK & Cancel buttons. Typed characters appear as *. Returns the text entered, or None if Cancel was clicked.

```
>>> pymsgbox.password('Enter your password.')
'12345'
```



2.3 Timeout

All four functions have a `timeout` parameter which takes a number of milliseconds. At the end of the timeout, the message box will close and have a return value of `'Timeout'`.

```
>>> pymsgbox.confirm('Nuke the site from orbit?', 'Confirm nuke', ["Yes, I'm sure.",
↳ 'Cancel'], timeout=2000) # closes after 2000 milliseconds (2 seconds)
'Timeout'
```

2.4 Localization

You can change the default 'OK', 'Cancel', and 'Timeout' strings by changing `pymsgbox.OK_TEXT`, `pymsgbox.CANCEL_TEXT`, and `pymsgbox.TIMEOUT_TEXT` variables respectively.

CHAPTER 3

Native Message Boxes

In addition to displaying message boxes using Python’s built-in TkInter GUI toolkit, PyMsgBox also has limited support for displaying message boxes by calling the operating system’s native functions. These exist in the `native` module. To use them, call:

```
>>> import pymsgbox
>>> pymsgbox.native.alert('This is an alert.', 'The title.')
```

Or, to avoid changing your function calls to use `pymsgbox.native`, run:

```
>>> import pymsgbox.native as pymsgbox
```

3.1 Support

These are the platforms and functions that have native message box support. Functions that do are not supported will default back to the normal TkInter-based message box.

Support as of v1.0.1:

	Windows	OS X	Linux	JVM / Jython
<code>alert()</code>	Yes	No	No	No
<code>confirm()</code>	Yes	No	No	No
<code>prompt()</code>	No	No	No	No
<code>password()</code>	No	No	No	No

3.2 Special Notes

3.2.1 Windows - `alert()`

The message box will only show a button with text “OK”, no matter what is passed for the `button` argument. The `button` argument will be the text that is returned by the function, just like the TkInter `alert()`.

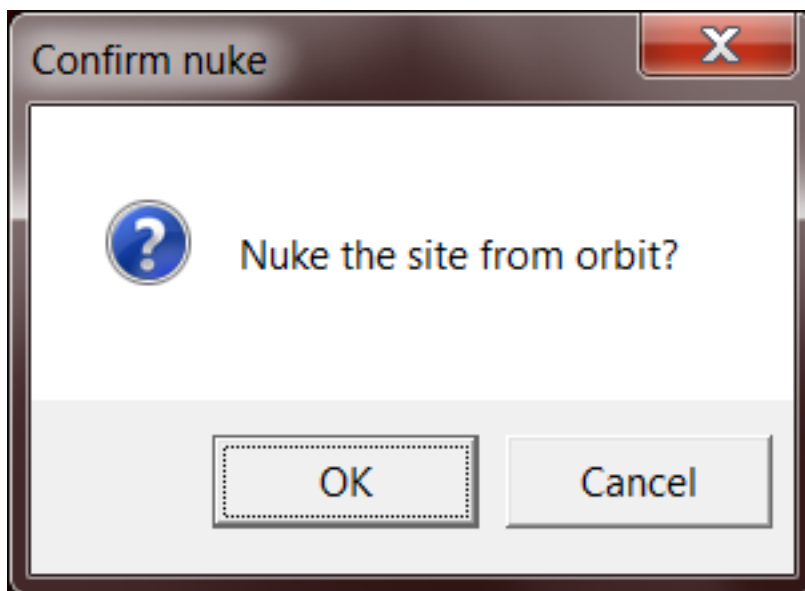
```
>>> pymsgbox.native.alert('This is an alert.', 'Alert!')
'OK'
```



3.2.2 Windows - confirm()

There will only be buttons “OK” and “Cancel”, no matter what is passed for the `buttons` argument. If “OK” is clicked, the first item in the `buttons` list is returned (by default, this is 'OK'). If “Cancel” is clicked, the second item in the `buttons` list is returned (by default, this is 'Cancel'), just like the TkInter `confirm()`.

```
>>> pymsgbox.native.confirm('Nuke the site from orbit?', 'Confirm nuke', ["Yes, I'm_sure.", 'Cancel'])
'Yes, I'm sure.'
```



CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`