

# RUBY/TK - TOPLEVEL WIDGET

[http://www.tutorialspoint.com/ruby/ruby\\_tk\\_toplevel.htm](http://www.tutorialspoint.com/ruby/ruby_tk_toplevel.htm)

Copyright © tutorialspoint.com

## Description:

A **Toplevel** is similar to a frame except that it is created as a top-level window. Its X parent is the root window of a screen rather than the logical parent from its path name.

The primary purpose of a toplevel is to serve as a container for dialog boxes and other collections of widgets. The only visible features of a toplevel are its background color and an optional 3-D border to make the toplevel appear raised or sunken.

## Syntax:

Here is a simple syntax to create this widget:

```
TkToplevel.new(root) {  
  .....Standard Options.....  
  .....Widget-specific Options.....  
}
```

## Standard Options:

- borderwidth
- cursor
- highlightbackground
- highlightcolor
- highlightthickness
- relief
- takefocus

These options have been described in previous chapter.

## Widget-specific Options:

### SN Options with Description

1 **background** => String

This option is the same as the standard **background** option except that its value may also be specified as an empty string. In this case, the widget will display no background or border, and no colors will be consumed from its colormap for its background and border.

2 **class** => String

Specifies a class for the window. This class will be used when querying the option database for the window's other options, and it will also be used later for other purposes such as bindings. The **class** option may not be changed with the **configure** method.

3 **colormap** => String

Specifies a colormap to use for the window. The value may be either **new**, in which case a new colormap is created for the window and its children, or the name of another window.

4 **height** => Integer

Specifies the desired height for the window.

5 **width** => Integer

Specifies the desired width for the window.

## Event Bindings:

When a new toplevel is created, it has no default event bindings: toplevels are not intended to be interactive.

## Examples:

```
require 'tk'

def make_win
  begin
    $win.destroy
  rescue
    end
    $win = TkToplevel.new
    TkButton.new($win) {
      text 'Window Dismiss'
      command "$win.destroy"
    }
    pack
  end

  TkButton.new {
    text 'make Window'
    command 'make_win'
  }
  pack('fill' => 'x')
}

Tk.mainloop
```

This will produce the following result:

