RUBY/TK - ENTRY WIDGET

Description:

A **Entry** presents the user with a single-line text field that they can use to type in a value. These can be just about anything: their name, a city, a password, social security number, and so on.

Syntax:

Here is a simple syntax to create this widget:

```
TkEntry.new{
  .....Standard Options....
  .....Widget-specific Options....
}
```

Standard Options:

- background
- borderwidth
- cursor
- exportselection
- font
- foreground
- highlightbackground
- highlightcolor
- highlightthickness
- justify
- relief
- selectbackground
- selectborderwidth
- selectforeground
- takefocus
- textvariable
- xscrollcommand

These options have been described in previous chapter.

Widget-specific Options:

SN	Options with Description
1	**disabledbackground** => String Specifies the background color to use when the entry is disabled. If this option is the empty string, the normal background color is used.

2	**disabledforeground** => String Specifies the foreground color to use when the entry is disabled. If this option is the empty string, the normal foreground color is used.
3	**readonlybackground** => String Specifies the background color to use when the entry is read-only. If this option is the empty string, the normal background color is used.
4	**show** => String If this option is specified, then the true contents of the entry are not displayed in the window. Instead, each character in the entry's value will be displayed as the first character in the value of this option, such as ``*''. This is useful, for example, if the entry is to be used to enter a password. If characters in the entry are selected and copied elsewhere, the information copied will be what is displayed, not the true contents of the entry.
5	**state** => String Specifies one of three states for the entry: **normal**, **disabled**, or **readonly**. If the entry is **readonly**, then the value may not be changed using widget commands and no insertion cursor will be displayed, even if the input focus is in the widget; the contents of the widget may still be selected. If the entry is **disabled**, the value may not be changed, no insertion cursor will be displayed, the contents will not be selectable, and the entry may be displayed in a different color.
6	**validate** => String Specifies the mode in which validation should operate: **none**, **focus**, **focusin**, **focusout**, **key**, or **all**. It defaults to **none**. When you want validation, you must explicitly state which mode you wish to use.
7	**validatecommand** => String Specifies a script to eval when you want to validate the input into the entry widget.
8	**width** => Integer Specifies an integer value indicating the desired width of the entry window, in average-size characters of the widget's font. If the value is less than or equal to zero, the widget picks a size just large enough to hold its current text.

Validation of Entry:

We can validate entered value by setting the *validatecommand* option to a callback which will be evaluated according to the *validate* option as follows:

- **none:** Default. This means no validation will occur.
- **focus:** validatecommand will be called when the entry receives or loses focus.
- **focusin:** validatecommand will be called when the entry receives focus.
- **focusout:** validatecommand will be called when the entry loses focus.
- **key:** validatecommand will be called when the entry is edited.

- **all:** validatecommand will be called for all above conditions.

Manipulating Entries:

There are following useful methods to manipulate the content of an entry:

- **delete*first, ?last?*:** Delete one or more elements of the entry. First is the index of the first character to delete, and last is the index of the character just after the last one to delete. If last isn't specified it defaults to first+1, i.e. a single character is deleted. This command returns an empty string.
- **get:** Returns the entry's string.
- **icursor*index*:** Arrange for the insertion cursor to be displayed just before the character given by index. Returns an empty string.
- **index*index*:** Returns the numerical index corresponding to index.
- **insert*index, string*:** Insert the characters of string just before the character indicated by index. Returns an empty string.
- **xview*args*:** This command is used to query and change the horizontal position of the text in the widget's window.

Event Bindings:

Ruby/Tk automatically creates class bindings for entries that give them the following default behavior. :

- Clicking mouse button 1 positions the insertion cursor just before the character underneath the mouse cursor, sets the input focus to this widget, and clears any selection in the widget. Dragging with mouse button 1 strokes out a selection between the insertion cursor and the character under the mouse.
- Double-clicking with mouse button 1 selects the word under the mouse and positions the insertion cursor at the beginning of the word. Dragging after a double click will stroke out a selection consisting of whole words.
- Triple-clicking with mouse button 1 selects all of the text in the entry and positions the insertion cursor before the first character.
- The ends of the selection can be adjusted by dragging with mouse button 1 while the Shift key is down; this will adjust the end of the selection that was nearest to the mouse cursor when button 1 was pressed. If the button is double-clicked before dragging then the selection will be adjusted in units of whole words.
- Clicking mouse button 1 with the Control key down will position the insertion cursor in the entry without affecting the selection.
- If any normal printing characters are typed in an entry, they are inserted at the point of the insertion cursor.
- The view in the entry can be adjusted by dragging with mouse button 2. If mouse button 2 is clicked without moving the mouse, the selection is copied into the entry at the position of the insertion cursor.
- If the mouse is dragged out of the entry on the left or right sides while button 1 is pressed, the entry will automatically scroll to make more text visible *ifthereismoretextoff − screenonthesidewherethemouseleftthewindow*.
- The Left and Right keys move the insertion cursor one character to the left or right; they also clear any selection in the entry and set the selection anchor. If Left or Right is typed with the Shift key down, then the insertion cursor moves and the selection is extended to include the new character. Control-Left and Control-Right move the insertion cursor by words, and Control-Shift-Left and Control-Shift-Right move the insertion cursor by words and also extend the selection. Control-b and Control-f behave the same as Left and Right, respectively. Meta-b and Meta-f behave the same as Control-Left and Control-Right, respectively.

- The Home key, or Control-a, will move the insertion cursor to the beginning of the entry and clear any selection in the entry. Shift-Home moves the insertion cursor to the beginning of the entry and also extends the selection to that point.
- The End key, or Control-e, will move the insertion cursor to the end of the entry and clear any selection in the entry. Shift-End moves the cursor to the end and extends the selection to that point.
- The Select key and Control-Space set the selection anchor to the position of the insertion cursor. They don't affect the current selection. Shift-Select and Control-Shift-Space adjust the selection to the current position of the insertion cursor, selecting from the anchor to the insertion cursor if there was not any selection previously.
- Control-/ selects all the text in the entry.
- Control-\ clears any selection in the entry.
- The F16 key *labelledCopyonmanySunworkstations* or Meta-w copies the selection in the widget to the clipboard, if there is a selection.
- The F20 key *labelledCutonmanySunworkstations* or Control-w copies the selection in the widget to the clipboard and deletes the selection. If there is no selection in the widget then these keys have no effect.
- The F18 key *labelledPasteonmanySunworkstations* or Control-y inserts the contents of the clipboard at the position of the insertion cursor.
- The Delete key deletes the selection, if there is one in the entry. If there is no selection, it deletes the character to the right of the insertion cursor.
- The BackSpace key and Control-h delete the selection, if there is one in the entry. If there is no selection, it deletes the character to the left of the insertion cursor.
- Control-d deletes the character to the right of the insertion cursor.
- Meta-d deletes the word to the right of the insertion cursor.
- Control-k deletes all the characters to the right of the insertion cursor.
- Control-w deletes the word to the left of the insertion cursor.
- Control-t reverses the order of the two characters to the right of the insertion cursor.

If the entry is disabled using the **state** option, then the entry's view can still be adjusted and text in the entry can still be selected, but no insertion cursor will be displayed and no text modifications will take place.

Examples:

```
require 'tk'

root = TkRoot.new
root.title = "Window"

entry1 = TkEntry.new(root)
entry2 = TkEntry.new(root) do
     show '*'
end

variable1 = TkVariable.new
variable2 = TkVariable.new
entry1.textvariable = variable1
entry2.textvariable = variable2
variable1.value = "Enter any text value"
variable2.value = "Enter any confidential value"

entry1.place('height' => 25,
```

```
            'width'  => 150,
            'x'      => 10,
            'y'      => 10)

entry2.place('height' => 25,
            'width'  => 150,
            'x'      => 10,
            'y'      => 40)

Tk.mainloop
```

This will produce the following result: