Tkinter Entry

If this Python Tutorial saves you hours of work, please whitelist it in your ad blocker 😭 and

Donate Now

(https://www.pythontutorial.net/donation/)

to help us ❤️ pay for the web hosting fee and CDN to keep the website running.

Summary: in this tutorial, you'll learn how to use the Tkinter Entry widget to create a textbox.

Introduction to Tkinter Entry widget

The `Entry` widget allows you to enter a sing-line text. In Tkinter, to create a textbox, you use the `Entry` widget:

```
textbox = ttk.Entry(container, **options)
```

In this syntax:

- The `container` is the parent frame (https://www.pythontutorial.net/tkinter/tkinter-frame/) or window (https://www.pythontutorial.net/tkinter/tkinter-window/) . on which you want to place the widget.
- The `options` is one or more keyword arguments used to configure the `Entry` widget.

Note that if you want to enter multi-line text, you should use the `Text` `(https://www.pythontutorial.net/tkinter/tkinter-text/)` widget.

To get the current text of a `Entry` widget as a string, you use the `get()` method:

```
textbox.get()
```

Typically, you associate the current value of the textbox with a StringVar object like this:

```
text = tk.StringVar()
textbox = ttk.Entry(root, textvariable=text)
```

In this syntax:

- First, create a new instance of the `StringVar` class. The text will be the value holder for a string variable.
- Second, assign the text variable to the `textvariable` of the `Entry` widget.

In this case, you can use call the `get()` method of the `StringVar()` object to get the current value of the entry widget:

```
text.get()
```

Setting the focus to the Tkinter Entry widget

To provide a better user experience, you can place move the focus to the first `Entry` widget after the window appears. Once the `Entry` widget has focus, it's ready to accept the user input.

To do it, you use the `focus()` method of the `Entry` widget like this:

```
textbox.focus()
```

Showing a Tkinter password entry

To hide sensitive information on the `Entry` widget e.g., a password, you can use the `show` option.

The following creates a password entry. When you enter a password, it doesn't show the actual characters but the asterisks (*) specified in the `show` option:

```
password = tk.StringVar()
password_entry = ttk.Entry(
    root,
    textvariable=password,
    show='*'
)
password_entry.pack()
```

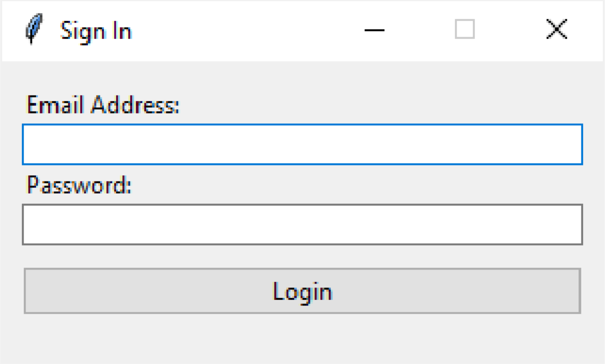
Tkinter Entry widget example

The following program shows how to use the `Entry` widgets to create a sign-in form:

```
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showinfo

# root window
root = tk.Tk()
root.geometry("300x150")
root.resizable(False, False)
root.title('Sign In')

# store email address and password
email = tk.StringVar()
password = tk.StringVar()


def login_clicked():
    """ callback when the login button clicked
    """
    msg = f'You entered email: {email.get()} and password: {password.get()}'
    showinfo(
        title='Information',
        message=msg
    )
```

```
# Sign in frame
signin = ttk.Frame(root)
signin.pack(padx=10, pady=10, fill='x', expand=True)


# email
email_label = ttk.Label(signin, text="Email Address:")
email_label.pack(fill='x', expand=True)

email_entry = ttk.Entry(signin, textvariable=email)
email_entry.pack(fill='x', expand=True)
email_entry.focus()

# password
password_label = ttk.Label(signin, text="Password:")
password_label.pack(fill='x', expand=True)

password_entry = ttk.Entry(signin, textvariable=password, show="*")
password_entry.pack(fill='x', expand=True)

# login button
login_button = ttk.Button(signin, text="Login", command=login_clicked)
login_button.pack(fill='x', expand=True, pady=10)


root.mainloop()
```

Output:



How it works.

First, create two string variables to hold the current text of the email and password `Entry` widgets:

```
# store email address and password
email = tk.StringVar()
password = tk.StringVar()
```

Second, create the email `Entry` widget and associate it with the email variable:

```
email_entry = ttk.Entry(signin, textvariable=email)
```

The following sets focus on the `email` entry:

```
email_entry.focus()
```

Third, create the password entry widget and assign the password variable to its `textvariable` :

```
password_entry = ttk.Entry(signin, textvariable=password, show="*")
```

Finally, display a message box that shows the entered email and password if the login button is clicked.

Summary

- Use the `ttk.Entry` widget to create a textbox.
- Use an instance of the `StringVar()` class to associate the current text of the `Entry` widget with a string variable.

- Use the `show` option to create a password entry.