

File - yarusoto.pyw

```
1 """
2 A simple tk to do list app
3 """
4
5 import tkinter as tk
6 from dataclasses import dataclass
7 from os import path
8 from tkinter import ttk
9 from typing import List, Optional
10
11
12 @dataclass
13 class TodoItem:
14     """
15     Represents a single task.
16
17     Fields:\n
18     name: The name of the task\n
19     checked: The status of the task (if is it checked off)
20     """
21     name: str
22     checked: Optional[tk.BooleanVar] = None
23
24
25 def add_todo(add_todo_text: ttk.Entry, todos_frame: ttk.Frame) -> None:
26     """
27     Add a new TodoItem to todo_list from the add_todo_text widget and re-renders the
28     list.
29     :param add_todo_text: ttk.Entry widget that hold the text that should be the new
30     TodoItem
31     :param todos_frame: The ttk.Frame that needs to be passed to update_todos to re-
32     render the list
33     """
34     todo_to_add: str = add_todo_text.get().strip()
35     if len(todo_to_add) > 0:
36         todo_list.append(TodoItem(todo_to_add))
37         add_todo_text.delete(0, "end")
38         update_todos(todos_frame, show_checked_var.get())
39
40
41 def update_todos(todos_frame: ttk.Frame, show_checked: bool) -> None:
42     """
43     Updates the todos_frame with the TodoItems in todo_list. This allows for hidden
44     checkboxes
45     to be removed immediately after they are checked .
46
47     :param todos_frame: The ttk.Frame that should display the TodoItems
48     :param show_checked: Whether just the checked or all the todos should be rendered
49     """
50     for old_todo in todos_frame.winfo_children():
51         old_todo.destroy()
```

File - yarusoto.pyw

```
49     for todo in todo_list:
50         # check if [TodoItem].checked is None, so it can be initialized
51         if todo.checked is None:
52             # need to instantiate tk var here because if it is done outside the
class or
53             # before tk is initialized the script crashes
54             todo.checked = tk.BooleanVar()
55             if not show_checked and todo.checked.get():
56                 continue
57             ttk.Checkbutton(
58                 todos_frame,
59                 text=todo.name,
60                 variable=todo.checked,
61                 onvalue=True,
62                 offvalue=False,
63                 command=lambda: update_todos(todos_frame, show_checked_var.get())
64             ).pack(anchor="w")
65
66
67 # global vars
68 todo_list: List[TodoItem] = []
69 show_checked_var: Optional[tk.BooleanVar] = None
70
71
72 def main() -> None:
73     """
74     Manages tk initialization and adds the non-dynamic widgets.
75     """
76     global show_checked_var
77
78     # ctypes.windll library is only found on Windows systems.
79     try:
80         from ctypes import windll
81     except AttributeError:
82         pass
83     else:
84         windll.shcore.SetProcessDpiAwareness(1)
85
86     root = tk.Tk()
87     root.title("Yarusoto")
88     root.geometry("400x600")
89     dir_path = path.abspath(path.dirname(__file__))
90     icon = tk.PhotoImage(file=path.join(dir_path, "icon.png"))
91     root.iconphoto(False, icon)
92
93     # vars
94     show_checked_var = tk.BooleanVar()
95
96     # frames
97     todos_frame: ttk.Frame = ttk.Frame(root)
98     add_todo_frame: ttk.Frame = ttk.Frame(root)
99
```

File - yarusoto.pyw

```
100     # widgets
101     title_label: ttk.Label = ttk.Label(root, text="Welcome to Yarusoto!")
102     show_checked: ttk.Checkbutton = ttk.Checkbutton(
103         root,
104         text="Show completed to-dos",
105         variable=show_checked_var,
106         onvalue=True,
107         offvalue=False,
108         command=lambda: update_todos(todos_frame, show_checked_var.get())
109     )
110     seperator: ttk.Separator = ttk.Separator(root, orient="horizontal")
111     todos_label = ttk.Label(text="To-dos:")
112
113     # add_todos_frame widgets
114     add_todo_text: ttk.Entry = ttk.Entry(add_todo_frame, width=30)
115     add_todo_button: ttk.Button = ttk.Button(
116         add_todo_frame,
117         text="Add",
118         command=lambda: add_todo(add_todo_text, todos_frame)
119     )
120     root.bind("<Return>", lambda _: add_todo(add_todo_text, todos_frame))
121
122     # pack
123     title_label.pack()
124     show_checked.pack()
125     seperator.pack(fill="x", padx=5)
126     todos_label.pack(anchor="w")
127
128     todos_frame.pack(anchor="w")
129
130     add_todo_text.pack(side="left")
131     add_todo_button.pack(side="left")
132     add_todo_frame.pack(side="bottom")
133
134     # loop
135     root.mainloop()
136
137
138 if __name__ == "__main__":
139     main()
140
```