

File - yarusoto.pyw

```
1  """
2  A simple tkinter 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 Optional
10
11
12 @dataclass
13 class TodoItem:
14     """
15     Represents a single task.
16
17     :param name: The name of the task
18     :param checked: The status of the task (if is it checked off)
19     """
20     name: str
21     checked: Optional[tk.BooleanVar] = None
22
23
24 def add_todo(add_todo_text: ttk.Entry, todos_frame: ttk.Frame) -> None:
25     """
26     Add a new TodoItem to todo_list from the add_todo_text widget and re-renders the
27     list.
28
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     add_todo_text.delete(0, "end")
36     if len(todo_to_add) > 0:
37         todo_list.append(TodoItem(todo_to_add))
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()
52     # if only I could just do this
```

File - yarusoto.pyw

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

File - yarusoto.pyw

```
100     todos_frame: ttk.Frame = ttk.Frame(root)
101     add_todo_frame: ttk.Frame = ttk.Frame(root)
102
103     # widgets
104     title_label: ttk.Label = ttk.Label(root, text="Welcome to Yarusoto!")
105     show_checked: ttk.Checkbutton = ttk.Checkbutton(
106         root,
107         text="Show completed to-dos",
108         variable=show_checked_var,
109         onvalue=True,
110         offvalue=False,
111         command=lambda: update_todos(todos_frame, show_checked_var.get())
112     )
113     seperator: ttk.Separator = ttk.Separator(root, orient="horizontal")
114     todos_label = ttk.Label(text="To-dos:")
115
116     # add_todos_frame widgets
117     add_todo_text: ttk.Entry = ttk.Entry(add_todo_frame, width=30)
118     add_todo_button: ttk.Button = ttk.Button(
119         add_todo_frame,
120         text="Add",
121         command=lambda: add_todo(add_todo_text, todos_frame)
122     )
123     root.bind("<Return>", lambda *args: add_todo(add_todo_text, todos_frame))
124
125     # pack
126     title_label.pack()
127     show_checked.pack()
128     seperator.pack(fill="x", padx=5)
129     todos_label.pack(anchor="w")
130     todos_frame.pack(anchor="w")
131     add_todo_text.pack(side="left")
132     add_todo_button.pack(side="left")
133     add_todo_frame.pack(side="bottom")
134
135     # loop
136     root.mainloop()
137
138
139 if __name__ == "__main__":
140     main()
141
```