

----- TKINTER MENU BAR TUTORIAL -----

```
from tkinter import *
from tkinter import messagebox
from tkinter import ttk

# In this tutorial I'll cover
# Creating menu bars
# Triggering functions from the menu bar
# Using Checkboxes & Radio Buttons
# Adding shortcut keys to menu bar items

# When called this closes the app
def quit_app():
    root.quit()

# Shows an about message box
def show_about(event=None):
    messagebox.showwarning(
        "About",
        "Isn't this an awesome program?"
    )

# Output to console when the font is changed
def change_font(event=None):
    print("Font Changed to", text_font.get())

# Creates the main window
root = Tk()

# Add a title for your app
root.title("Menu Bar Example")

# Create the menu bar
the_menu = Menu(root)

# ----- FILE MENU ITEMS -----
# Create a pull down menu that can't be removed
# and tie it to your menu bar
file_menu = Menu(the_menu, tearoff=0)

# Add items that will show up when File is clicked on
file_menu.add_command(label="Open")
file_menu.add_command(label="Save")

# Add horizontal bar between items in menu
file_menu.add_separator()

# Add Quit to the menu and execute the function
# that quits the app
```

```

file_menu.add_command(label="Quit", command=quit_app)

# Add the label File and the pull down menu to
# the menu bar
the_menu.add_cascade(label="File", menu=file_menu)
# ----- END OF FILE MENU ITEMS -----

# ----- VIEW MENU ITEMS -----
# Create pull down for View
view_menu = Menu(the_menu, tearoff=0)

# Add a checkbox option to show line numbers and
# set the default to checked
line_numbers = IntVar()
line_numbers.set(1)

# Add checkbox to View with a label and
# bind line_numbers so we know if the box is
# checked or not
view_menu.add_checkbutton(label="Line Numbers",
                           variable=line_numbers)

# Add View to the menu
the_menu.add_cascade(label="View", menu=view_menu)
# ----- END OF VIEW MENU ITEMS -----

# ----- FONT MENU ITEMS -----
# Store the font chosen in a string variable
text_font = StringVar()

# Set the default font
text_font.set("Times")

# Create pull down for font
font_menu = Menu(the_menu, tearoff=0)

# Define radio buttons for the menu, store selection
# in text_font and call change_font when changed
font_menu.add_radiobutton(label="Times",
                           variable=text_font,
                           command=change_font)
font_menu.add_radiobutton(label="Courier",
                           variable=text_font,
                           command=change_font)
font_menu.add_radiobutton(label="Arial",
                           variable=text_font,
                           command=change_font)

# Add Font to the menu
the_menu.add_cascade(label="Font", menu=font_menu)
# ----- END OF FONT MENU ITEMS -----

# ----- HELP MENU ITEMS -----
# Create pull down for Help

```

```
help_menu = Menu(the_menu, tearoff=0)

# When About is clicked execute a function but
# also tie it to a shortcut
# Accelerator defines a shortcut that's available
help_menu.add_command(label="About",
                      accelerator="command-A",
                      command=show_about)

# Key substitutions on Windows and Mac
# Control (Windows) = Command (Mac)
# Alt (Windows) = Option (Mac)

# Add Help to the menu bar
the_menu.add_cascade(label="Help", menu=help_menu)

# Bind shortcut to the About and what we want
# to show
root.bind('<Command-A>', show_about)

# Also bind lowercase a
root.bind('<Command-a>', show_about)

# ----- END OF HELP MENU ITEMS -----

# Make your menu show on the screen
root.config(menu=the_menu)

# Keeps our program running until quit
root.mainloop()
```