**TYPING SPEED TEST**

**Project Description:**

This is a Python script that creates a simple GUI application using the tkinter module for calculating typing speed and accuracy. The application displays a random quote as a typing test, and the user types the quote in a text box. The user can start the timer, stop it, and calculate the speed and accuracy of their typing.

The above code is a Python script that creates a graphical user interface (GUI) using the tkinter module for calculating typing speed and accuracy. The application displays a random quote as a typing test, and the user types the quote in a text box. The user can start the timer, stop it, and calculate the speed and accuracy of their typing.

Let's take a closer look at how the code works:

The MainWindow class is defined, which takes a root argument that is used to create the main window of the GUI. The init method sets up the GUI by creating the label, text box, and buttons for the application. It also sets up some variables to track the speed, accuracy, and start/end time of the typing test.

The text variable is a list of quotes that will be displayed as the typing test. The speed and accuracy variables are initialized to 0, and the time\_start and time\_end variables are initialized to None.

The GUI consists of a label that displays the quote, a text box where the user types, and three buttons: "Start/Restart", "Stop", and "New Text". The label\_text variable is a Label widget that displays the quote. The user\_text variable is a Text widget that the user types into. The btn\_start, btn\_stop, and btn\_newtext variables are Button widgets that start the timer, stop the timer and calculate the speed and accuracy, and generate a new random quote for the user to type, respectively.

The start method is called when the user clicks the "Start/Restart" button. It sets the time\_start variable to the current time, which marks the beginning of the typing test.

The stop method is called when the user clicks the "Stop" button. It sets the time\_end variable to the current time, which marks the end of the typing test. It then calculates the speed and accuracy of the typing test by dividing the number of words in the quote by the time it took the user to type them and multiplying by 60 to get the speed in words per minute (WPM). The accuracy is calculated using the SequenceMatcher class from the difflib module, which compares the text in the label to the text the user typed and returns a ratio that represents the similarity between the two texts. This ratio is then multiplied by 100 to get a percentage of accuracy. The label\_speed and label\_accuracy variables are updated to display the speed and accuracy, respectively.

The new\_text method is called when the user clicks the "New Text" button. It generates a new random quote from the text variable and sets the label\_text variable to display the new quote. It also clears the user\_text variable so that the user can start typing the new quote from scratch.

The main function creates the main window of the GUI using the Tk() function, and then creates an instance of the MainWindow class using the root variable. It then starts the main event loop using the mainloop() method, which waits for user input and responds to events such as button clicks.

Overall, this code provides a simple and user-friendly typing test application that can help users improve their typing speed and accuracy. The GUI is easy to navigate and the application provides instant feedback on the user's performance.

**Python Concepts Used:**

The above code uses the Tkinter library, which provides a graphical user interface (GUI) toolkit for Python. It includes various functions and widgets that allow developers to create GUI applications. Here are the key concepts and functions used in the above code:

**Classes**: The code defines a class named "MainWindow," which represents the main window of the application. It contains various methods and properties to manage the UI and handle user actions.

**Widgets**: Tkinter provides a wide range of widgets that can be used to build the UI of the application. The code uses the following widgets:

**Label**: Used to display text on the UI.

Text: Used to provide a text editor where the user can type.

Button: Used to create buttons that the user can click to perform specific actions.

Grid layout: Tkinter allows developers to create a grid layout for the UI, where widgets can be arranged in rows and columns. The code uses the grid layout to arrange the widgets on the main window.

**Text manipulation**: The code uses various functions to manipulate the text entered by the user, such as getting the text from the Text widget, splitting the text into words, and calculating the typing accuracy using the SequenceMatcher function from the difflib module.

**Event handling**: The code uses event handling to respond to user actions, such as clicking on a button. It defines methods that are called when the user clicks on a button to perform specific actions.

**Time module**: The code uses the time module to calculate the time taken by the user to type the text.

**Random module**: The code uses the random module to select a random text from a list of predefined texts to be typed by the user.

**Round function**: The code uses the round function to round off the calculated typing speed and accuracy to the nearest integer.

**References:**

* *Stack Overflow*. Available at: <https://stackoverflow.com/>
* Python Docs. Available at: <https://docs.python.org/3/reference/>
* W3chools. Available at: <https://www.w3schools.com/python/>

**CODE:**

from tkinter import \*

from tkinter import ttk

import time

import random

import difflib

class MainWindow:

def \_\_init\_\_(self, root):

self.text = [

"The greatest glory in living lies not in never falling, but in rising every time we fall.",

"The way to get started is to quit talking and begin doing.",

"Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma – which is living with the results of other people's thinking.",

"If life were predictable it would cease to be life, and be without flavor.",

"If you look at what you have in life, you'll always have more. If you look at what you don't have in life, you'll never have enough.",

"If you set your goals ridiculously high and it's a failure, you will fail above everyone else's success.",

"Life is what happens when you're busy making other plans.",

"One day the people that don’t even believe in you will tell everyone how they met you.",

"The true meaning of life is to plant trees, under whose shade you do not expect to sit.",

"The quick brown fox jumps over the lazy dog."

]

self.speed = 0

self.accuracy = 0

self.time\_start = 0

self.time\_end = 0

root.title("Typing Speed Test")

root.minsize(800, 500)

for row in range(5):

root.grid\_rowconfigure(row, weight=1)

for col in range(3):

root.grid\_columnconfigure(col, weight=1)

self.label\_text = Label(root,

text="Welecom to typeing speed calculator",

wraplength=500)

self.label\_text.grid(row=0, column=0, columnspan=3, sticky="nsew")

self.user\_text = Text(root)

self.user\_text.grid(column=0, row=1, columnspan=3, sticky="nsew")

self.btn\_start = Button(root, text="Start/Restart", command=self.start)

self.btn\_start.grid(column=0, row=2, columnspan=1, sticky="nsew")

self.btn\_stop = Button(root, text="Stop", command=self.stop)

self.btn\_stop.grid(column=1, row=2, columnspan=1, sticky="nsew")

self.btn\_newtext = Button(root, text="New Text", command=self.new\_text)

self.btn\_newtext.grid(column=2, row=2, columnspan=1, sticky="nsew")

self.label\_speed = Label(root,

text=f"Your typing speed is {self.speed} WPM")

self.label\_speed.grid(row=3, column=0, columnspan=3, sticky="nsew")

self.label\_accuracy = Label(root,

text=f"Your typing accuracy is {self.speed} %")

self.label\_accuracy.grid(row=4, column=0, columnspan=3, sticky="nsew")

def start(self):

self.time\_start = time.time()

def stop(self):

self.time\_end = time.time()

words = self.label\_text.cget("text").split(' ')

self.speed = round(len(words) / ((self.time\_end - self.time\_start) / 60))

self.label\_speed.config(text=f"Your typing Speed is {self.speed} WPM")

self.accuracy = round(

difflib.SequenceMatcher(None, self.label\_text.cget("text"),

self.user\_text.get("1.0", 'end-1c')).ratio() \*

100)

self.label\_accuracy.config(

text=f"Your typing accuracy is {self.accuracy} %")

def new\_text(self):

self.label\_text.config(

text=self.text[random.randint(0,

len(self.text) - 1)])

self.user\_text.delete('1.0', END)

def main():

root = Tk()

myapp = MainWindow(root)

root.mainloop()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT:**

