

```

import pygame
import random

pygame.init()

sw = 1300
sh = 500
screen = pygame.display.set_mode((sw, sh))
pygame.display.set_caption("Typing Speed Calculator")
text_font_20 = pygame.font.Font("fonts/SF-Pro-Text-Regular.otf", 20)
text_font_23 = pygame.font.Font("fonts/SF-Pro-Text-Regular.otf", 23)
text_font_30 = pygame.font.Font("fonts/SF-Pro-Text-Regular.otf", 30)
text_font_40 = pygame.font.Font("fonts/SF-Pro-Text-Regular.otf", 40)
heading_font_20 = pygame.font.Font("fonts/SF-Pro-Text-Bold.otf", 20)
heading_font_25 = pygame.font.Font("fonts/SF-Pro-Text-Bold.otf", 25)
heading_font_30 = pygame.font.Font("fonts/SF-Pro-Text-Bold.otf", 30)
heading_font_40 = pygame.font.Font("fonts/SF-Pro-Text-Bold.otf", 40)
heading_font_60 = pygame.font.Font("fonts/SF-Pro-Text-Bold.otf", 60)

def start_game():
    statements = [
        # Add more sentences later
        "Please take your dog, Cali, out for a walk – he really needs some exercise!",
        "What a beautiful day it is on the beach, here in beautiful and sunny Hawaii.",
        "Rex Quinfrey, a renowned scientist, created plans for an invisibility machine.",
        "Do you know why all those chemicals are so hazardous to the environment?",
        "You never did tell me how many copper pennies where in that jar; how come?",
        "Max Joykner sneakily drove his car around every corner looking for his dog.",
        "The two boys collected twigs outside, for over an hour, in the freezing cold!",
        "When do you think they will get back from their adventure in Cairo, Egypt?",
        "Trixie and Veronica, our two cats, just love to play with their pink ball of yarn.",
        "We climbed to the top of the mountain in just under two hours; isn't that great?",
        "Code is like humor. When you have to explain it, it's bad.",
        "In order to be irreplaceable, one must always be different.",
        "Optimism is an occupational hazard of programming: feedback is the treatment.",
        "Before software can be reusable it first has to be usable.",
        "If opportunity doesn't knock, build a door.",
        "When to use iterative development? You should use iterative development only on projects that you want to succeed.",
        "War does not bring anything good to the common people.",
        "After the death of the king, everyone wanted to be a king.",
        "The plots failed because of some trusted friends of the king.",
        "When I've built up my savings, I'll be able to travel to Mexico.",
        "Wouldn't it be lovely to enjoy a week soaking up the culture?",
        "What we know is a drop. What we don't know... is an ocean.",
        "Every decision for something is a decision against something else.",
    ]

```

```

    "The end is the beginning, and the beginning is the end.",
    "India or 'Republic of India' is a peninsular country in Asia i.e. it is surrounded by water from three sides.",
    "ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 123456789 !?@#%()[\|/*+<>.,;:'~"
]

def result(inp, statement, time_taken):
    c = 0
    C = 0
    inp_words = inp.split(" ")
    statement_words = statement.split(" ")

    wpm = (len(inp_words) / time_taken) * 60
    for i in range(len(inp_words)):
        for j in range(len(inp_words[i])):
            C += 1
            if inp_words[i][j] == statement_words[i][j]:
                c += 1
    return ((str(c / C * 100)[:5]), str(wpm)[:5])

statement = random.choice(statements)
inp = ''
enter = 0
start = 0

MainGame = True
while MainGame:
    screen.fill((10, 10, 10))

    sentence_msg = heading_font_30.render(("Sentence box"), True, (255, 255, 255))
    screen.blit(sentence_msg, (50, 40))

    type_text_msg = heading_font_30.render(("Type the text written above in the box below"), True, (255, 255, 255))
    screen.blit(type_text_msg, (50, 185))

    pygame.draw.rect(screen, (255, 255, 0), pygame.Rect(50, 80, 1200, 40), 3)
    sentence = heading_font_20.render(statement, True, (255, 255, 255))
    screen.blit(sentence, (60, 88))
    pygame.draw.rect(screen, (0, 255, 0), pygame.Rect(50, 230, 1200, 40), 3)

    Accuracy_msg = heading_font_30.render("Accuracy:", True, (255, 255, 255))
    screen.blit(Accuracy_msg, (505, 345))
    WPM_msg = heading_font_30.render("WPM:", True, (255, 255, 255))
    screen.blit(WPM_msg, (550, 385))

    if start == 0:
        inp_sentence = text_font_20.render(("Press TAB to start"), True, (128, 128, 128))
    else:

```

```

if inp in statement:
    color = (3, 165, 252)
    correct_msg = heading_font_25.render("Good going!", True, (3, 165, 252))
    screen.blit(correct_msg, (50, 275))
    wrong_msg = heading_font_25.render("Stop, error!", True, (64, 10, 10))
    screen.blit(wrong_msg, (50, 310))
else:
    color = (255, 38, 38)
    correct_msg = heading_font_25.render("Good going!", True, (0, 36, 56))
    screen.blit(correct_msg, (50, 275))
    wrong_msg = heading_font_25.render("Stop, error!", True, (255, 38, 38))
    screen.blit(wrong_msg, (50, 310))
inp_sentence = text_font_20.render(inp, True, (color))
game_time = pygame.time.get_ticks()
timer_start = 1
screen.blit(inp_sentence, (60, 238))

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        MainGame = False
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_TAB:
            start = 1
            start_time = pygame.time.get_ticks()
        elif event.key == pygame.K_RETURN:
            enter = 1
            end_time = pygame.time.get_ticks()
            accuracy, wps = result(inp, statement, (end_time - start_time) / 1000)
        elif event.key == pygame.K_ESCAPE:
            Display_Page()
        elif event.key == pygame.K_BACKSPACE:
            inp = inp[:-1]
        else:
            inp += event.unicode

if enter == 1:
    Accuracy_msg = heading_font_30.render("Accuracy: " + accuracy + "%", True, (255, 255, 255))
    screen.blit(Accuracy_msg, (505, 345))
    WPM_msg = heading_font_30.render("WPM: " + wps, True, (255, 255, 255))
    screen.blit(WPM_msg, (550, 385))

pygame.display.update()

```

```

def Display_Page():
    MainRun = True
    while MainRun:

```

```
screen.fill((10, 10, 10))
Game_msg = heading_font_60.render("Typing Speed Calculator", True, (255, 255, 0))
screen.blit(Game_msg, (300, 95))

Display_msg = text_font_40.render("PRESS SPACE TO START →", True, (255, 255, 200))
screen.blit(Display_msg, (420, 340))

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        MainRun = False

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_SPACE:
            MainRun = False
            start_game()

pygame.display.update()
```

Display_Page()