

# Aufgabe 1: Hopsitexte

Team-ID: 00000

Team: Constantin

Bearbeiter/-innen dieser Aufgabe:  
Constantin

19. November 2024

## Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	2

## Lösungsidee

Die Aufgabe besagt, dass der Nutzer ein Tool haben möchte, dass ihn dabei unterstützt Hopsitexte zu schreiben. Am besten sollte das Tool also Hinweise darüber geben ob der geschriebene Text ein Hopsitext ist und falls nicht angibt ab wo. Außerdem ist die aktuelle Textlänge sinnvoll um herauszufinden wie lang unser Text bereits ist.

## Umsetzung

Die Umsetzung besteht aus drei Teilen.

### Tkinter

Wir definieren einen einfaches Tkinter Interface mit einem Eingabefeld und einer Scrollbar und zwei Label die unsere gewünschten Informationen aus der Lösungsidee (aktuelle Textlänge und Hinweis über den Status des Hopsitext's). Sobald sich der Text des Eingabefeld's ändert wird die Funktion *text\_ändern* ausgelöst.

### text\_ändern

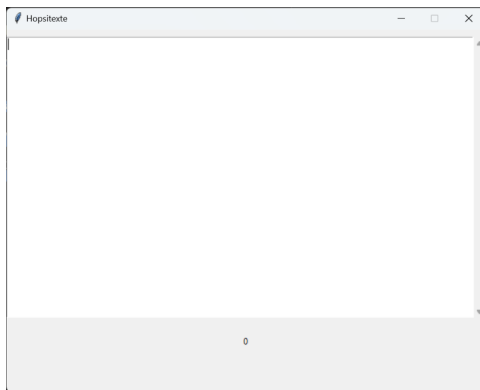
Diese Funktion evaluiert die aktuelle Textlänge und ändert darauf hin das Label für die aktuelle Textlänge im Tkinter Interface. Außerdem überprüft sie ob der Text gekürzt oder erweitert wurde. Wenn der Text gekürzt wurde wird die Variable *evaluierte\_textlänge* auf die aktuelle Textlänge gesetzt. Sollte der Text erweitert worden wird die Funktion *is\_hopsitext* mit dem Text aus dem Textfeld als Argument aufgerufen.

## is\_hopsitext

Diese Funktion implementiert das Hopsitext Spiel aus Junioraufgabe 2. Sollte der Text an irgendeinem Punkt kein Hopsitext mehr sein ändern wir das Label aus dem Tkinter Interface zu einem Hinweis, dass der Text an diesem Punkt kein Hopsitext mehr ist. Sollten wir den Text ändern und dann bemerken, dass der Text wieder ein Hopsitext ist wird der Hinweis entfernt.

## Beispiele

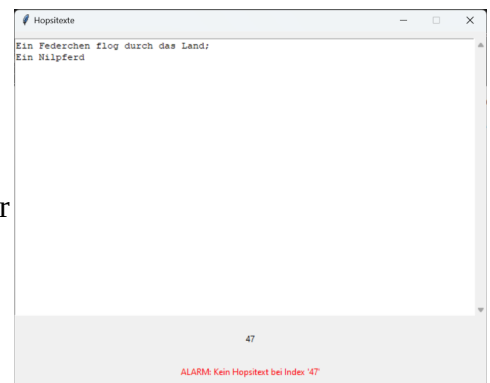
Da diese Aufgabe keine bereitgestellten Beispiele umfasst werde ich hier Ablauf des Hopsitexte schreiben durchgehen und erläutern, wie das Programm in bestimmten Fällen reagiert.



Wenn wir das Programm starten können wir unser leeres Textfeld und in der unteren Leiste, die Anzahl der getippten Charakter erkennen. In diesem Fall 0.

Sagen wir wir fangen an einen Text zu schreiben, zum Beispiel den Hopsen2 Text aus Junioraufgabe 2, sehen wir, dass nachdem wir das "d" eingeben eine

Warnung erscheint, dass der Text kein Hopsitext ist. Diese Warnung geht nicht sofort weg, wenn der Nutzer den Fehler löscht sondern erst wenn der Nutzer wieder anfängt zu tippen. So kann der Nutzer ganz einfach einen Hopsitext schreiben.



## Quellcode

```
import tkinter as tk
def is_hopsitext(text):
    player1 = 0
    player2 = 1
    for index, item in enumerate(list(text)):
        if player1 != player2:
            alert_label.config(text="")
            kein_hopsitext = False

        if player1 == player2:
            if kein_hopsitext == False:
                kein_hopsitext_index = index
                kein_hopsitext = True
                alert_label.config(text=f"ALARM: Kein Hopsitext bei Index '{kein_hopsitext_index}'")

    if item not in letter_dict:
        player1 += 1
```

```

        player2 += 1

    if index == player1:
        player1 += letter_dict.get(item)
    if index == player2:
        player2 += letter_dict.get(item)

    return not kein_hopsitext

def text_ändern(event):
    global evaluierte_textlänge
    text_eingabe = text_feld.get("1.0", tk.END)
    aktuelle_textlänge = len(list(text_eingabe))
    index_label.config(text=str(aktuelle_textlänge-1)) # von der aktuellen_textlänge wird 1
    subtriert, da Textlänge bei 1 und Index bei 0 startet.

    if aktuelle_textlänge > evaluierte_textlänge: # Wenn neuer Text hinzugefügt wurde
        is_hopsitext(text_eingabe)
        evaluierte_textlänge = aktuelle_textlänge

evaluierte_textlänge = 0
kein_hopsitext = False
letter_dict = { #Speichert alle Buchstaben und ihre zugehörigen Sprungweiten
    'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5, 'f': 6, 'g': 7, 'h': 8, 'i': 9, 'j': 10,
    'k': 11, 'l': 12, 'm': 13, 'n': 14, 'o': 15, 'p': 16, 'q': 17, 'r': 18, 's': 19,
    't': 20, 'u': 21, 'v': 22, 'w': 23, 'x': 24, 'y': 25, 'z': 26,
    'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'F': 6, 'G': 7, 'H': 8, 'I': 9, 'J': 10,
    'K': 11, 'L': 12, 'M': 13, 'N': 14, 'O': 15, 'P': 16, 'Q': 17, 'R': 18, 'S': 19,
    'T': 20, 'U': 21, 'V': 22, 'W': 23, 'X': 24, 'Y': 25, 'Z': 26,
    'Ä': 27, 'ä': 27, 'Ö': 28, 'ö': 28, 'Ü': 29, 'ü': 29, 'ß': 30,
}

# Definiert das Tkinter Fenster
fenster = tk.Tk()
fenster.title("Hopsitexte")
fenster.resizable(False, False)

# Definiert den Mainframe
frame = tk.Frame(fenster)
frame.pack(pady=10)

# Definiert das Eingabefeld
text_feld = tk.Text(frame)
text_feld.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

# Ermöglicht das scrollen bei langen Texten
scrollbar = tk.Scrollbar(frame, command=text_feld.yview)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

```

```
text_feld.config(yscrollcommand=scrollbar.set)

# Löst die Funktion text_ändern aus sobald der Text im Eingabefeld geändert wird
text_feld.bind("<KeyRelease>", text_ändern)

# Platzhalter Label für die Länge des Textes
index_label = tk.Label(fenster, text="", height=2)
index_label.pack(pady=5)
# Platzhalter Label für den "Kein Hopsitext" Alarm
alert_label = tk.Label(fenster, text="", height=2, fg="red")
alert_label.pack(pady=5)

fenster.mainloop()
```