

Ja, sehr gern – hier ist dein **ausführlicher Lernzettel** zur **Formulareingabe mit mehreren Feldern gleichzeitig** in `prompt_toolkit`.



Lernzettel: Mehrfeld-Formulare mit `prompt_toolkit`

🎯 Ziel

Du lernst, **mehrere Eingabefelder** gleichzeitig in einer Eingabemaske (z. B. Login, Setup, Settings) mit `prompt_toolkit` zu erstellen – ähnlich wie ein grafisches Formular, aber **im Terminal**.

Das Ganze läuft **strukturiert mit Tab-Wechsel** zwischen den Feldern, Enter zur Bestätigung, und kann später erweitert werden (z. B. Validierung, Pflichtfelder, Buttons etc.).



1. Voraussetzungen

Du brauchst:

```
pip install prompt_toolkit
```



2. Wichtige Klassen & Konzepte

Baustein	Funktion
Window, Label	Zeilen & Texte anzeigen
TextArea	Eingabefelder
HSplit, VSplit	Layout-Manager für vertikale/horizontale Gruppen
Frame	Rahmen um Formulare (optional, fürs Aussehen)
Application	Hauptklasse zur Ausführung
Focus + TabHandler	Navigieren zwischen Feldern mit Tab / Shift+Tab

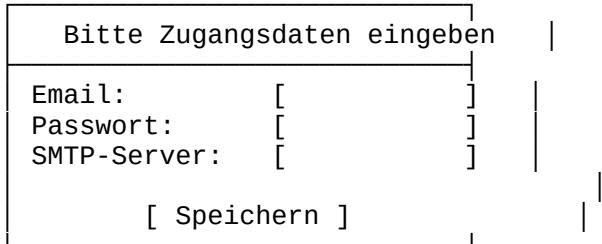


3. Formularstruktur (Beispiel mit 3 Feldern)

Wir definieren drei Felder:

- E-Mail-Adresse
 - Passwort
 - SMTP-Server
-

Aufbauidee (ohne Code):



Du kannst mit **Tab/Shift+Tab** zwischen Feldern wechseln und später mit **Enter speichern**.

4. Erklärungen zu den Bausteinen

Element	Beschreibung
TextArea()	Eingabefeld mit optionalem password=True
Label()	Beschriftung eines Feldes
VSplit()	Zeile mit Label + Textfeld nebeneinander
HSplit()	Mehrere Zeilen untereinander
Frame()	Visueller Rahmen um die Eingabe
focus()	Setzt den Anfangsfokus
kb.add("tab")	Tab-Steuerung zwischen Feldern

5. Beispielhafte Struktur (als Lernhilfe, nicht fix für dein Projekt!)

```
from prompt_toolkit.application import Application
from prompt_toolkit.layout import Layout, HSplit, VSplit
from prompt_toolkit.widgets import Label, TextArea, Button, Frame
from prompt_toolkit.key_binding import KeyBindings

# 1. Felder
email_input = TextArea(prompt='Email: ')
password_input = TextArea(prompt='Passwort: ', password=True)
smtp_input = TextArea(prompt='SMTP: ')

# 2. Speicher-Button
def speichern():
    print("\nErfasst:")
    print("Email:", email_input.text)
    print("Passwort:", password_input.text)
    print("SMTP:", smtp_input.text)
    app.exit()

save_button = Button("Speichern", handler=speichern)

# 3. Layout (eine vertikale Liste von horizontalen Einträgen)
form = HSplit([
    Label("Bitte Zugangsdaten eingeben:"),  

    VSplit([Label("Email:      "), email_input]),  

    VSplit([Label("Passwort:   "), password_input]),  

    VSplit([Label("SMTP-Server: "), smtp_input]),
```

```

        save_button
    ])

# 4. Tabbing
kb = KeyBindings()
@kb.add("tab")
def _(event):
    event.app.layout.focus_next()
@kb.add("s-tab")
def _(event):
    event.app.layout.focus_previous()

# 5. App bauen
app = Application(
    layout=Layout(form, focused_element=email_input),
    key_bindings=kb,
    full_screen=False
)
app.run()

```



6. Was du daraus lernen kannst

Lektion	Bedeutung
Jedes TextArea ist ein Eingabefeld	Du kannst beliebig viele Felder bauen
Fokus springt mit Tab	Benutzerfreundliche Navigation
Button kann Funktionen triggern	Z. B. Speichern oder Weiterleitung
Du kannst mit Frame oder Farben arbeiten	Für übersichtliches Aussehen



7. Erweiterungsmöglichkeiten

Feature	Umsetzung
Validierung	<code>if not email_input.text.strip(): ...</code>
Dynamischer Button-Text	<code>Button(text_func=...)</code>
Felder optional/erforderlich	Manuell prüfen beim Speichern
Tooltips/Helptext	<code>bottom_toolbar=</code> oder <code>Label()</code> darunter
Farbliche Hervorhebung	Mit <code>Style()</code> oder <code>FormattedTextControl</code>
Autovervollständigung	Mit <code>Completer</code> beim <code>TextArea</code>



Fazit

Mit `prompt_toolkit` kannst du **komfortable, interaktive Formulare im Terminal** bauen – inklusive Tab-Navigation, Passwortfeldern, Buttons und mehr.

So etwas ist **perfekt für den ersten Programmstart**, Setup-Wizard oder Konfigurationen.

Wenn du willst, kann ich dir gern in einem weiteren Lernzettel zeigen:

-  Wie du Formularfelder **validierst**
-  Wie du **Completer** nutzt (z. B. für Domains)
-  Wie du Farben und Themes einfügst
-  Wie du die Ergebnisse speicherst (JSON, DB, etc.)

Sag einfach: „**Ja, ich will mehr!**“ oder gib ein Thema vor!