Hier ist ein **kompaktes Lern-PDF-Inhalt** (als Text, den du einfach in ein PDF umwandeln kannst – z. B. mit Word → „Als PDF speichern“ oder online mit „Print to PDF“).

📚 Lern-PDF: `zxcvbn` in Python – Passwortstärke prüfen

Ziel: Verstehen, **wie** man `zxcvbn` nutzt – **ohne vorgefertigten Code für dich zu schreiben**

---

1. Was ist `zxcvbn`?

- Entwickelt von **Dropbox**
- Prüft Passwörter **realistisch** gegen echte Angriffe
- Berücksichtigt: Wörterbücher, Tastaturmuster, Wiederholungen, Jahreszahlen, Namen
- Gibt **Score von 0 bis 4** zurück (4 = sehr stark)

---

2. Installation (einmalig)

```
pip install zxcvbn
```

> Danach kannst du es in **jedem Python-Skript** importieren.

---

3. Grundsyntax – So funktioniert der Aufruf

```
from zxcvbn import zxcvbn

ergebnis = zxcvbn('dein_passwort_hier')
```

> Ersetze `'dein_passwort_hier'` durch eine **Variable** oder **Eingabe**.

---

4. Was kommt zurück? (Wichtige Schlüssel im Dictionary)

Schlüssel	Bedeutung	Typ
`score`	Stärke: **0–4**	`int`
`guesses`	Wie viele Versuche ein Angreifer bräuchte	`float`
`guesses_log10`	Log10 der Versuche (für Anzeige)	`float`
`feedback`	Warnung + Verbesserungsvorschläge	`dict`

Schlüssel	Bedeutung	Typ
`feedback['warning']`	Warum schwach? (z. B. „Zu kurz“)	`str` oder `None`
`feedback['suggestions']`	Liste mit Tipps	`list`

---

5. Mini-Beispiele **zum Verstehen** (nur Erklärung – du schreibst den Code!)

Beispiel 1: Einfache Ausgabe des Scores

> **Was du tun musst:**
>
> 1. Importiere `zxcvbn`
> 2. Rufe `zxcvbn()` mit einem Passwort-String auf
> 3. Greife auf den Schlüssel `'score'` zu
> 4. Gib ihn aus

Erwartete Ausgabe bei `pass123`:

```
Score: 0
```

Beispiel 2: Warnung anzeigen

> **Was du tun musst:**
>
> 1. Speichere das Ergebnis in einer Variable
> 2. Greife auf `ergebnis['feedback']['warning']` zu
> 3. Prüfe mit `if` ob es `None` ist → dann keine Warnung
> 4. Sonst: Gib die Warnung aus

Bei `password`:

```
Warnung: Dies ist ein häufig verwendetes Passwort.
```

Beispiel 3: Vorschläge ausgeben

> **Was du tun musst:**
>
> 1. Greife auf `ergebnis['feedback']['suggestions']` zu
> 2. Das ist eine **Liste** → nutze eine `for`-Schleife
> 3. Gib jeden Vorschlag aus

Bei `test123`:

```
Vorschlag: Füge Großbuchstaben hinzu.
Vorschlag: Verwende Sonderzeichen.
```

Beispiel 4: Benutzereingabe prüfen

> **Was du tun musst:**
>
> 1. Nutze `input()` um ein Passwort abzufragen
> 2. Speichere es in einer Variable
> 3. Rufe `zxcvbn()` damit auf
> 4. Zeige Score + Feedback an

Interaktive Eingabe:

```
Passwort eingeben: abc
Score: 0
Warnung: Zu kurz
Vorschlag: Mindestens 8 Zeichen
```

---

6. Score-Bedeutung (Merkhilfe)

Score	Bedeutung	Farbe (Vorschlag)
0	**Sehr schwach**	Rot
1	Schwach	Orange
2	Mittel	Gelb
3	Gut	Hellgrün
4	**Sehr stark**	Grün

---

7. Tipps für deine eigene Umsetzung

- Verwende **keine `print()` im echten Projekt** → gib Werte an Frontend zurück
- Kombiniere mit **Mindestlänge** (z. B. `< 8 → sofort ablehnen`)
- Zeige **Live-Feedback** beim Tippen (wie bei Registrierung)
- `zxcvbn` ist **langsamer als einfache Regeln** → aber **genauer**

---

8. Übungsaufgaben (für dich zum Selbermachen!)

1. Schreibe ein Skript, das ein Passwort einliest und **nur den Score** ausgibt.
2. Erweitere es: Zeige **Warnung**, falls vorhanden.
3. Baue eine **Schleife**, die solange fragt, bis Score ≥ 3 ist.

4. Gib **alle Vorschläge nummeriert** aus.

Fertig! Kopiere diesen Text in Word/Google Docs → **als PDF speichern** → dein Lern-PDF ist bereit!

> **Du schreibst den Code selbst** – so lernst du die Syntax **richtig**! Kein Copy-Paste – nur **Verstehen & Anwenden**.

Bonus: Offizielle Doku: https://github.com/dropbox/zxcvbn

PDF erstellt von einem Python-Lernhelfer – für Lernende, nicht für Faule. 😊