

Programmieren – Praktikum 2

Ingenieurinformatik Teil 1, Wintersemester 2026/27

David Straub

Der Stand aus Praktikum 1

Heute lernt das Labor, mit Ihnen zu reden. Startpunkt ist dieser Stand (eine Beispielizele – Ihre eigenen Werte aus Praktikum 1 sind genauso gut):

```
kapazitaet_mAh = 3000
spannung_V = 3.7
masse_g = 45.0
strom_mA = 600

laufzeit_h = kapazitaet_mAh / strom_mA
energie_Wh = spannung_V * kapazitaet_mAh / 1000

print(laufzeit_h)
print(energie_Wh)
```

Übernehmen Sie ihn und führen Sie ihn einmal aus.

Aufgabe 1: Eingaben statt fester Werte

Feste Werte im Code bedeuten: Für jede neue Zelle muss jemand das Programm ändern. Besser: das Programm **fragt**.

1. Ersetzen Sie die vier festen Werte durch Eingaben (`input`)
2. Testen Sie mit den Werten Ihrer Zelle aus Praktikum 1 (oder mit den Beispielergebnissen von der vorigen Folie, falls Sie da nicht dabei waren)
3. Testen Sie mit den Werten der Nachbargruppe – ohne eine Zeile Code zu ändern

Aufgabe 2: Der Zell-Steckbrief

Rohe Zahlen wie `11.100000000000001` liest niemand gern. Bauen Sie die Ausgabe zu einem Steckbrief aus:

1. Geben Sie Laufzeit und Energieinhalt mit **einer Nachkommastelle** aus (`print` , f-String)
2. Ergänzen Sie Beschriftung und Einheit: `Laufzeit: 5.0 h`
3. Richten Sie die Zahlen **rechtsbündig** untereinander aus (Feldbreite!), sodass ein sauberer Block entsteht

Aufgabe 3: Plausibilität

Tippfehler passieren. Einzelwerte sind schwer zu beurteilen – aber die **spezifische Energie** (Wh/kg) liegt bei Li-Ionen-Zellen in einem engen typischen Bereich, egal wie groß die Zelle ist.

1. Berechnen Sie die spezifische Energie aus den Eingaben (wie in Praktikum 1)
2. Prüfen Sie mit `if`: Liegt sie grob zwischen 100 und 400 Wh/kg? Sonst geben Sie eine Warnung aus (`print`), dass die Eingaben nicht zusammenpassen
3. Testen Sie den Check: Was passiert, wenn Sie die Masse versehentlich in kg statt in g eingeben?

Aufgabe 4: Zellen einordnen

Zahlen bewerten ist Laborarbeit – Ihr Steckbrief soll die Zelle auch **einordnen**:

1. Ergänzen Sie mit `if / elif / else` eine Kategorie nach spezifischer Energie – z. B. über 250 Wh/kg: `Hochenergiezelle`, 150 bis 250: `Standard`, darunter: `Hochleistungs- oder ältere Zelle`
2. Die Grenzen und Namen der Kategorien sind Ihre Entscheidung – recherchieren Sie, was zu Ihrer Zelle passt
3. Nehmen Sie die Kategorie in den Steckbrief auf

Aufgabe 5 (falls Sie Schleifen schon hatten): Hartnäckig nachfragen

Eine Warnung ist gut – gar keine falschen Werte annehmen ist besser:

1. Wiederholen Sie die Kapazitäts-Eingabe mit einer `while`-Schleife, **bis** der Wert positiv und nicht absurd groß ist – die Obergrenze wählen Sie selbst
2. Geben Sie bei jeder ungültigen Eingabe aus (`print`), was falsch war
3. Wie muss die Schleifenbedingung lauten – `and` oder `or`? Diskutieren Sie in der Gruppe

Aufgabe 6 (falls Sie Schleifen schon hatten): Der Lade-Simulator

Jetzt wird geladen – in Prozentschritten:

1. Starten Sie bei `ladung_prozent = 0` und erhöhen Sie in einer `while`-Schleife um `schritt = 5`, bis 100 % erreicht sind
2. Geben Sie in jedem Durchlauf den Ladestand aus (`print`), rechtsbündig: `5 %`, `10 %`, ...
3. Zählen Sie die Durchläufe mit und geben Sie am Ende aus (`print`): `Voll nach 20 Schritten`
4. Machen Sie die Schrittweite zur Eingabe (`input`) – was passiert bei `schritt = 7`? Ist das Verhalten sinnvoll?

Zusatz für Schnelle

Ohne Schleifen:

- Holen Sie die **Pack-Rechnung** aus Praktikum 1 ins Programm: Zellenzahl als Eingabe (`input`), Pack-Spannung und Pack-Energie im Steckbrief
- Strompreis in €/kWh als Eingabe – was kostet eine volle Ladung Ihres Packs?
- Die C-Rate aus Praktikum 1 in den Steckbrief

Mit Schleifen:

- Erweitern Sie den Simulator: ab 80 % soll zusätzlich `Schnellladen beendet` erscheinen – aber nur einmal
- Der **Entlade**-Simulator: von 100 % herunter bis zur Abschaltgrenze Ihrer Wahl