

exp1_ws2022_ueb03_Dichten_STUD

November 1, 2022

Durch Ergänzung und Anwendung eines [Jupyter-Notebooks](#) auf dem [RWTH Jupyter Hub](#) sollen einige einfache Rechnungen mit den Dichten einiger bekannter Elemente gemacht werden, s.u.

Das Jupyter-Notebook besteht aus (meistens) abwechselnden Zellen von Text oder Python-Code. Dieser gesamte Textblock und die weiteren Beschreibungstexte sind in [Markdown](#) geschrieben. Auch [LaTeX-Mathesymbole](#) usw. lassen sich direkt im Text verwenden, z.B. wird aus α dann α .

Die Markdown-Zellen kann man nach einem Doppelklick darauf bearbeiten. Anschließend springt man durch gleichzeitiges Drücken von **Shift+Enter** aus der bearbeiteten Markdown-Zelle in die darunter befindliche Zelle.

Die Python-Codezellen kann man nach einem einfachen Mausklick bearbeiten. Durch gleichzeitiges Drücken von **Shift+Enter** wird der darin enthaltene Python-Code dann ausgeführt und z.B. Ergebnisse einer Berechnung unterhalb der Zelle ausgegeben.

Literatur

- LaTeX: <https://link.springer.com/book/10.1007/978-3-658-34430-6>
- Jupyter: <https://link.springer.com/book/10.1007/978-3-658-37723-6>
- Python: <https://link.springer.com/book/10.1007/978-3-658-28976-8>

0.0.1 Aufgabe 6: Dichten - 3 Punkte (1 + 1 + 1)

- a) In der Python-Codezelle unterhalb dieses Textes ist die Funktion `density()` abgebildet. Diese erwartet als Parameter (`elem`) einen aus zwei Buchstaben bestehenden String des Elementsymbols, also z.B. "Au" für Gold. Wenn das eingegebene Elementsymbol bekannt ist, liefert die Funktion die Dichte des jeweiligen Elements in g/cm^3 , andernfalls wird der Wert Null ausgegeben.

Definieren Sie die Funktion im Kontext dieses Jupyter-Notebooks durch einen Klick in die Codezelle und anschließendes Drücken von **Shift+Enter**.

```
[4]: def density(elem=''):      # Funktion zur Ausgabe der Dichten einiger chemischer
    ↪ Elemente in g/cm³
    if (elem=='Li'):          # Lithium
        return 0.534
    elif (elem=='Al'):        # Aluminium
        return 2.70
    elif (elem=='Fe'):        # Eisen
        return 7.87
```

```

elif (elem=='Cu'):    # Kupfer
    return 8.92
elif (elem=='Au'):    # Gold
    return 19.3
elif (elem=='Pb'):    # Blei
    return 11.3
else:
    return 0.

```

Nutzen Sie die `print()`-Funktion von Python (s.u.), um die Dichten von Gold, Kupfer und Aluminium mit drei Dezimalstellen auszugeben. Der Platzhalter `%.nf` steht für eine Dezimalzahl mit `n` Nachkommastellen.

```

[7]: print("Dies ist ein Test: %.3f"%density("Au")) # Test der print()-Funktion
     print("Dies ist ein Test: %.3f"%density("Cu"))
     print("Dies ist ein Test: %.3f"%density("Al"))

```

```

Dies ist ein Test: 19.300
Dies ist ein Test: 8.920
Dies ist ein Test: 2.700

```

- b) Die `NumPy`-Programmbibliothek stellt Datenstrukturen und Funktionen für numerische Berechnungen mit Python zur Verfügung. Wir benötigen Sie zur Berechnung der dritten Wurzel mit Hilfe der Funktion `cbirt()`.

Importieren Sie das Modul `numpy` unter dem Aliasnamen `np` durch einen Klick in die Codezelle und anschließendes Drücken von **Shift+Enter** und berechnen Sie mit der Funktion `cbirt()` die dritte Wurzel der Zahl 1,331, geben Sie ihr Ergebnis mit der `print()`-Funktion und zwei Dezimalstellen aus. Bedingt durch den Aliasnamen muss die Funktion hier als `np.cbirt()` verwendet werden.

```

[9]: import numpy as np    # das "numpy"-Modul importieren (unter dem Aliasnamen
     ↪ "np")

     # die numpy-Funktion "cbirt(Z)" berechnet die dritte Wurzel einer Zahl "Z"
     print(np.cbirt(1.331))

```

1.1

Definieren Sie nun zwei Umrechnungsfaktoren, von g/cm^3 in kg/cm^3 sowie von g/cm^3 in kg/m^3 . Ein Beispiel für die Umrechnung von Stunden in Sekunden finden Sie hier bereits:

```

[24]: U_h_in_s = 3600 # Umrechnungsfaktor von Stunden in Sekunden
     U_gcm3_in_kgcm3 = 1/1000 # Umrechnungsfaktor von Stunden in Sekunden
     U_gcm3_in_kgm3 = 1000 # Umrechnungsfaktor von Stunden in Sekunden

```

Berechnen Sie nun die Kantenlängen (in cm) von 1,31 kg schweren Würfeln aus Gold, Eisen und Aluminium. Geben Sie ihr Ergebnis mit der `print()`-Funktion und drei Dezimalstellen aus.

```
[25]: # V = 1.31kg/density(Element)
# r = V^(1/3)
print("Kantenlänge Gold: %.3f"%np.cbrt(1000*1.31/density("Au"))+"cm")
print("Kantenlänge Eisen: %.3f"%np.cbrt(1000*1.31/density("Fe"))+"cm")
print("Kantenlänge Aluminium: %.3f"%np.cbrt(1000*1.31/density("Al"))+"cm")
```

Kantenlänge Gold: 0.041cm

Kantenlänge Eisen: 0.055cm

Kantenlänge Aluminium: 0.079cm

- c) Berechnen Sie nun die Gewichte von Würfeln mit einem Volumen von $0,725 \text{ m}^3$ aus Blei, Kupfer bzw. Lithium. Geben Sie ihr Ergebnis mit der `print()`-Funktion und einer Dezimalstelle aus.

```
[29]: print("Gewicht Blei: %.1f"%(10**6*0.725*density("Pb"))+"g")
print("Gewicht Kupfer: %.1f"%(10**6*0.725*density("Cu"))+"g")
print("Gewicht Lithium: %.1f"%(10**6*0.725*density("Li"))+"g")
```

Gewicht Blei: 8192500.0g

Gewicht Kupfer: 6467000.0g

Gewicht Lithium: 387150.0g

Exportieren oder Drucken Sie ihr Jupyter-Notebook als PDF und hängen es an Ihre sonstigen Lösungen an. Laden Sie bitte in [moodle](#) zusätzlich auch die “ipynb”-Datei des Notebooks hoch.

[]: