



Unterrichtsentwurf

für die Beurteilung der Unterrichtspraxis

Vor- und Nachname:	Lorenz Bung
Schule (Anschrift, Telefon):	Walther-Rathenau-Gewerbeschule (Friedrichstr. 51, 79098 Freiburg, 0761 201-7942)
Schulleiterin, Schulleiter:	Renate Storm
Prüfer/in:	Jochen Pogrzeba
Prüfungsvorsitzende/r:	Jürgen Schnaiter

Datum: 21.02.2026	Uhrzeit: 09:45 – 10:30
Klasse: FTE1-1	Schulart: Fachschule für Technik
Raum: 033	Fach: Informatik

Thema des Unterrichts:	Python: Dictionaries
-------------------------------	----------------------

Inhaltsverzeichnis

1.	Überblick und zentrales Anliegen.....	4
1.1	Thema.....	4
1.2	Lehrplanbezug.....	4
1.3	Zentrales Anliegen.....	4
1.4	Lehr-Lernarrangement.....	4
2.	Begründungszusammenhänge und Vertiefung.....	5
2.1	Rahmenbedingungen und Einbettung des Unterrichts.....	5
2.2	Lernziele und Kompetenzentwicklung.....	5
2.3	Inhalte.....	5
2.4	Gestaltung des Lehr-Lernarrangement.....	5
3.	Anhang.....	6
3.1	Unterrichtsverlaufsplan.....	6
3.2	Quellenverzeichnis.....	7
3.3	Weitere Materialien.....	7

Versicherung

Ich versichere, dass ich den Entwurf zur unterrichtspraktischen Prüfung selbstständig, also ohne Hilfen Dritter, und nur mit den angegebenen Hilfsmitteln angefertigt habe. Für alle Stellen und Materialien, die dem Wortlaut oder Sinn nach anderen Werken, auch elektronischen Medien, entnommen wurden, habe ich die Quellen angegeben. Materialien aus dem Internet sind durch Ausdruck der ersten Seite belegt. Auf Nachfrage werde ich diese Materialien als kompletten Ausdruck oder auf einem elektronischen Speichermedium (möglichst im PDF-Format) zur Verfügung stellen.

Datum

Unterschrift (handschriftlich)



1. Überblick und zentrales Anliegen

1.1 Thema

Python: Dictionaries

1.2 Lehrplanbezug

Als Teil von BPE 1: Strukturierte Programme umsetzen findet sich unter BPE 1.3: Algorithmen umsetzen unter Anderem *"Erweiterte Datentypen: Arrays, Zeichenketten, Strukturen"*.

1.3 Zentrales Anliegen

Ziel der Stunde ist es, dass die Schüler das Konzept des Dictionaries als Schlüssel-Wert-Zuweisung kennenlernen und es konkret in Python um- und einsetzen können.

Die Schüler sollen dabei sowohl eigenständig Dictionaries implementieren als auch damit arbeiten können, beispielsweise durch Verwendung in Kombination mit einer Schleife.

1.4 Lehr-Lernarrangement

Eine Audio-Sequenz aus der Sendung mit der Maus auf Rumänisch führt in die Stunde ein. Nach Diskussion der Möglichkeiten, einen Sinn aus der fremden Sprache zu ziehen, wird das Gespräch auf die Datenstruktur der Dictionaries in Python gelenkt.

Durch Programmierung eines Beispiels am Lehrer-PC wird die Syntax des Dictionaries erklärt. Hierbei wird auch auf Unterschiede zu den bereits bekannten Datenstrukturen *Liste* und *Set* eingegangen.

In der anschließenden ersten Übungsphase sollen die Schüler am Beispiel ihres Personalausweises ein steckbriefartig aufgebautes Dictionary selbst umsetzen. In weiteren Arbeitsschritten soll das erstellte Dictionary auch in Kombination mit einer Schleife ausgegeben werden sowie eine Erweiterung der Datenstruktur durch Schachtelung mit einem weiteren Dictionary vorgenommen werden.

Die Aufgabe wird anschließend im Plenum besprochen und mögliche Fragen geklärt.

Die daran anschließende zweite Übungsphase hat zum Ziel, das Anfangsproblem wieder aufzugreifen und mithilfe eines rumänisch-deutschen Wörterbuchs eine rudimentäre Übersetzung des Maus-Intros zu erstellen. Die Schüler sollen auf Basis ihrer Übersetzung Vermutungen anstellen, worin es in der Folge wohl geht.

Nach der anschließenden Besprechung der Aufgabe im Plenum wird durch die deutsche Version des Intros aufgelöst, um was es sich in der Folge tatsächlich gedreht hat und ob die Voraussage der Schüler korrekt war.



2. Begründungszusammenhänge und Vertiefung

2.1 Rahmenbedingungen und Einbettung des Unterrichts

Die Klasse FTE1-1 besteht aus 12 ausschließlich männlichen Schülern im Alter von 21 – 40 Jahren, wobei die meisten Schüler um die 25 Jahre alt sind. Einige der Schüler haben vor Beginn der Technikerschule ebenfalls an der Walther-Rathenau-Gewerbeschule ihre Ausbildung absolviert und sind daher bereits sowohl mit der Schulgemeinschaft als auch den Räumlichkeiten vertraut. Die Klasse arbeitet generell gut im Unterricht mit, Unterrichtsstörungen treten so gut wie gar nicht auf.

Der Raum 033 verfügt über Schüler-PCs, welche ringförmig an den Außenwänden platziert sind. In der Mitte des Raumes stehen weiterhin Tische ohne PCs in Reihen. Viele der Schüler nutzen im Unterricht selbst ihr eigenes digitales Endgerät, weshalb die meisten auch während den Arbeitsphasen in der Mitte des Raumes sitzen bleiben. Diejenigen Schüler, die die Schulrechner benutzen, sitzen während dem Unterricht neben den Tischen in der Mitte an den Außenwänden. Zusätzlich verfügt der Raum neben dem Lehrer-PC über eine Dokumentenkamera sowie eine digitale Tafel, welche auch zur Wiedergabe von Multimediadateien genutzt werden kann. Eine analoge Tafel befindet sich nicht im Raum.

Eine Besonderheit des Raumes ist das angrenzende Büro einiger Kollegen, dessen Türe sehr schwergängig ist und deswegen häufiger laut zugezogen werden muss. Es kann daher zu – leider nahezu unvermeidbaren – akustischen Störungen kommen. Weiterhin verfügt der Raum über keine eigenen Fenster, sondern kann nur über die Fenster des angrenzenden Büros bei gleichzeitiger Öffnung der angrenzenden Türe mitbelüftet werden. Infolgedessen ist die Luft innerhalb des Raums leider oft abgestanden, was bei den Schülern in der Vergangenheit schon zu Konzentrationsschwierigkeiten und Kopfschmerzen geführt hat.

Stofflich stand bisher ausschließlich die strukturierte Programmierung im Vordergrund. Begonnen bei Variablen und Datentypen sowie dem EVA-Prinzip wurden bisher Verzweigungen, match-Blöcke sowie for- und while-Schleife behandelt. Das Konzept der Funktion wurde in der letzten Woche eingeführt, allerdings aus zeitlichen Gründen noch nicht vertieft eingeübt. Weiterhin kennen die Schüler bereits die Datenstrukturen der Liste und des Sets.



2.2 Lernziele und Kompetenzentwicklung

Das Grobziel der Stunde ist es, dass die Schüler das Konzept des Dictionaries als Schlüssel-Wert-Zuweisung kennenlernen und es konkret in Python um- und einsetzen können.

Dieses Grobziel lässt sich in folgende Feinziele aufteilen, die innerhalb der Stunde als Teilziele erreicht werden sollen:

1. Die Schüler *nennen* Unterschiede zwischen den Datenstrukturen *Liste*, *Set* und *Dictionary*. (AFB I)
2. Die Schüler *implementieren* die Datenstruktur *Dictionary* in Python. (AFB II)
3. Die Schüler *entwerfen* komplexere Datenstrukturen durch Schachtelung von Dictionaries. (AFB III)
4. Die Schüler *wenden* Dictionaries in Kombination mit anderen bekannten Konzepten *an*, um Probleme zu lösen. (AFB II)

2.3 Inhalte

Die Stunde beschäftigt sich mit der Datenstruktur der *Dictionaries* als Paaren von Schlüsseln und zugehörigen Werten. Dabei liegt der Vergleich mit den ebenfalls bereits eingeführten Datenstrukturen *Liste* und *Set* nahe – insbesondere in Hinblick auf die vergleichbare Syntax der unterschiedlichen Datenstrukturen.

Weiterhin sollen die Eigenschaften der verschiedenen Datenstrukturen miteinander verglichen werden, beispielsweise die Relevanz der Reihenfolge bei den Listen (und, seit Python 3.7 ebenfalls bei den Dictionaries), welche im Gegensatz zu den unsortierten Sets steht. Auch die Eindeutigkeit der Schlüssel, welche keine Duplikate erlaubt, soll thematisiert werden und dabei Parallelen zu den Elementen eines Sets gezogen werden.

Ebenso geht es um die konkrete Umsetzung der Dictionaries in der Programmiersprache Python, welche im Unterricht eingesetzt wird. Hierbei soll zunächst die eigenständige Implementierung eines Dictionaries im Vordergrund stehen. Anschließend soll auch der Zugriff auf Elemente geübt werden, beispielsweise in Kombination mit einer Schleife.

Weiterführend soll ebenfalls die Schachtelung mehrerer Dictionaries ineinander angesprochen werden. Aufgrund der begrenzten Zeit ist dies jedoch ein Thema, was in einer nachfolgenden Stunde noch vertieft thematisiert werden muss und hier nur kurz angeschnitten wird. Die entsprechende Aufgabe ist daher so formuliert, dass die Schüler Ideen für dieses Problem sammeln sollen und es wird davon ausgegangen, dass nur sehr leistungsstarke Schüler direkt auf die richtige Lösung kommen.

Zuletzt soll das Konzept des Dictionaries mit anderen bereits bekannten Elementen der strukturierten Programmierung kombiniert werden: Einer Schleife sowie einer Verzweigung.



2.4 Gestaltung des Lehr-Lernarrangement

Der Unterrichtseinstieg wurde gewählt, da es sich bei der Sendung mit der Maus um etwas allgemein Bekanntes handelt und da das Intro jeder Folge zweisprachig ist. Die Bedeutung des Dictionarys als Wörterbuch steckt schon im Namen.

Bei der Erarbeitungsphase wurde bewusst keine eigenständige Erarbeitung des Konzepts gewählt, sondern das "vorprogrammieren". Einer der Gründe dafür war der zeitliche Aspekt: da nur 45 Minuten zur Verfügung standen, wurde so eine zusätzliche Sicherungsphase und damit wertvolle Zeit gespart. Weiterhin ist eine eigenständige Erarbeitung einer bestimmten Syntax nicht besonders spannend und keine kognitive Herausforderung, allerdings kann es zusätzlich zu unterschiedlichen Ergebnissen und damit Verunsicherung bei den Schülern führen.

Während der Erarbeitung sollen jedoch im Lehrer-Schüler-Gespräch auch gleichzeitig Parallelen zu bereits bekannten Datenstrukturen und deren Syntax gezogen werden. Dies soll einerseits eine Erleichterung beim Merken der Syntax liefern, als gleichzeitig auch auf eine Diskussion der möglichen Unterschiede und unterschiedlichen Anwendungsfälle der verschiedenen Datenstrukturen hinführen.

In der ersten Übungsphase sollen einerseits das eigenständige Umsetzen eines Dictionarys in Python als auch der Umgang damit (Zugriff auf Elemente, Iteration über die Schlüssel mithilfe einer Schleife) geübt werden. Darüber hinausgehend vertieft Teilaufgabe c) das Konzept jedoch um die Schachtelung mehrerer Dictionarys ineinander. Es ist zu erwarten, dass einige der leistungsschwächeren Schüler Probleme mit der Komplexität dieser Teilaufgabe haben werden oder aus zeitlichen Gründen gar nicht erst zu ihrer Bearbeitung kommen. Die Aufgabe wurde daher bewusst ans Ende gestellt, um stärkeren Schülern zusätzliche, vertiefende Inhalte zu bieten, gleichzeitig aber sicherzustellen, dass auch leistungsschwächere Schüler das Grundkonzept eingeübt haben.

Aufgabe 2 hat zum Ziel, das Anfangsproblem zu lösen und die an der Realität angelehnte Verwendung des Dictionarys als tatsächliches *Wörterbuch* zu thematisieren. Das zur Aufgabe 2 gehörige Wörterbuch wurde hierbei zunächst manuell mit den Wörtern gefüllt, die für das Übersetzen des rumänischen Satzes benötigt werden. Damit die Lösung jedoch nicht direkt aus dem Wörterbuch an sich abgelesen werden kann, wurden anschließend noch weitere Wörter mit passender Übersetzung durch ChatGPT generiert (OpenAI, 2026) und das Wörterbuch so erweitert.

Zusätzlich zur Verwendung des Dictionarys müssen hier auch andere Konzepte verwendet werden: Neben einer Schleife zur Iteration über die einzelnen Wörter ist auch eine Verzweigung nötig, um Wörter, die nicht im Wörterbuch auftauchen, trotzdem (z. B. unverändert) in die fertige Übersetzung mit einzubeziehen.

Aufbauend auf dieses eigenständig entwickelte Übersetzungs-Programm sollten die Schüler damit in der Lage sein, die Themen des Maus-Intros vorherzusehen.



3. Anhang

3.1 Unterrichtsverlaufsplan

Phase	Unterrichtsstruktur (mit Zeitplanung)	Lehrerhandeln	Schülerhandeln	Lernziele (fachliche und überfachliche)	Medien
Unterrichtseinstieg (09:45 – 09:50)	Maus-Intro auf Rumänisch (5 min.)	Begrüßung der Schüler Vorstellung der Prüfungskommission Abspielen des Maus-Intros auf Rumänisch Erklären der Problemsituation	Begrüßung Anhören des Maus-Intros Ideensammlung: Möglichkeiten zur Lösung des Problems mit Python evtl. erste Fragen an die LK	Motivationsaufbau Fokussierung der Schüler auf den Lerngegenstand Herstellung eines geeigneten Arbeitsklimas	Digitale Tafel Audiodatei "maus-rumaenisch.mp3"
Erarbeitung (09:50 – 09:55)	Lehrervortrag, darbietend (5 min.)	Erklärung der Datenstruktur <i>Dictionary</i> als Schlüssel-Wert-Paar Programmierung eines Beispiel-Dictionaries Beantwortung von Fragen der Schüler	Vergleich des neuen Konzepts mit den bereits bekannten Datenstrukturen <i>Liste</i> und <i>Set</i> Fragen an LK stellen	1. TZ: Die Schüler <i>nennen</i> Unterschiede zwischen den Datenstrukturen <i>Liste</i> , <i>Set</i> und <i>Dictionary</i> . (AFB I)	Lehrer-PC Digitale Tafel
Übung und Anwendung 1 (09:55 – 10:10)	Einzelarbeit Aufgabe "Personalausweis" (15 min.)	Freischalten des ABs in Moodle Erklärung des Arbeitsauftrags Beantwortung von Fragen der Schüler technische Hilfestellung	Bearbeitung der Aufgabe Verständnisfragen an LK stellen evtl. Diskussion mit Sitznachbarn (Aufgabe 1c)	2. TZ: Die Schüler <i>implementieren</i> die Datenstruktur <i>Dictionary</i> in Python. (AFB II) 3. TZ: Die Schüler <i>entwerfen</i> komplexere Datenstrukturen durch Schachtelung von Dictionaries. (AFB III)	Arbeitsblatt, Aufgabe 1 Moodle-Kurs



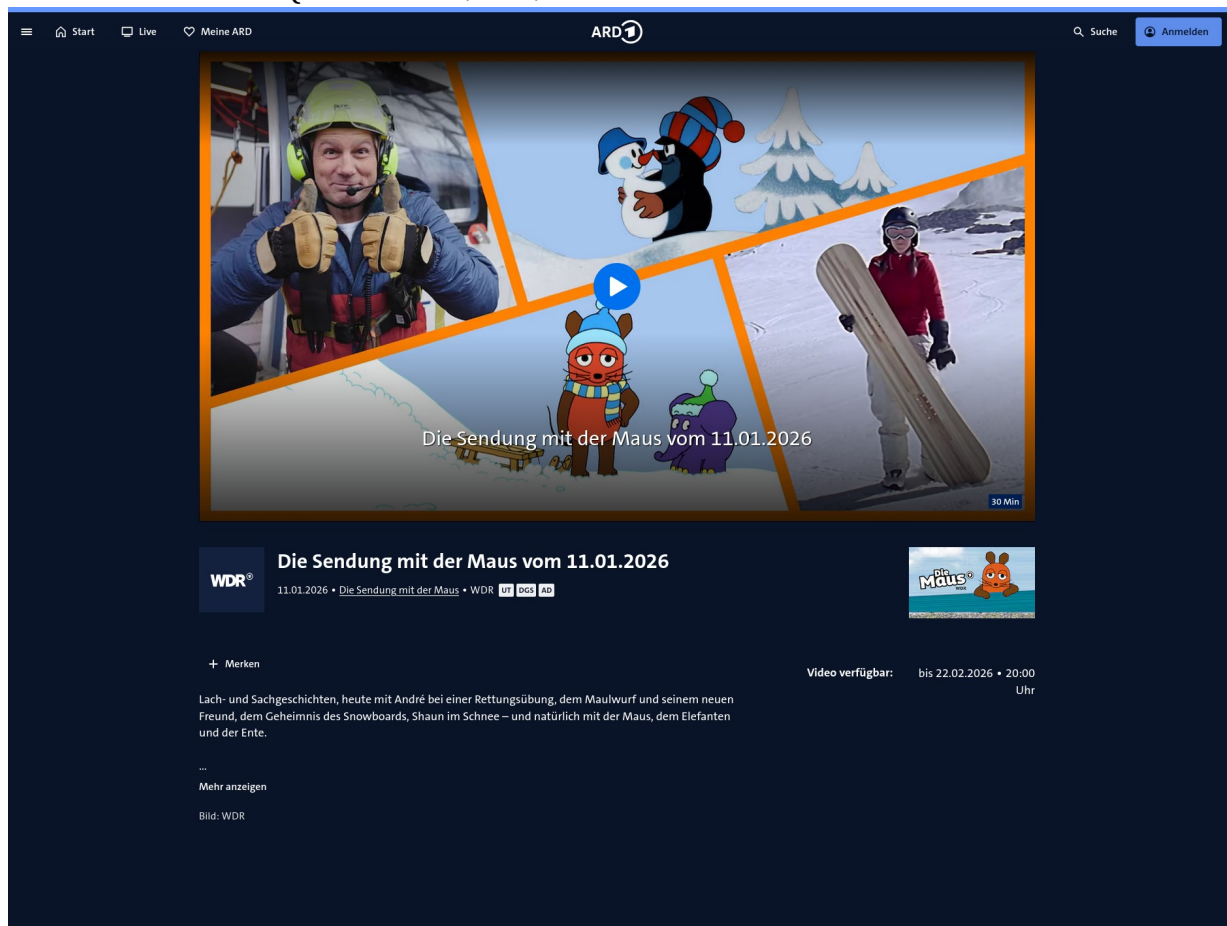
Sicherung 1 (10:10 – 10:17)	Plenumsdiskussion Besprechung der Aufgabe "Personal- ausweis" (7 min.)	Moderation der Schülermeldungen Ergänzung der vorgestellten Lösung Fragen an Schüler stellen Beantwortung von Fragen	Präsentation der Ergebnisse aus der Arbeitsphase Ergänzung / Diskussion der vorgestell- ten Lösung Stellen von Verständnisfragen	Fachliche Konsolidierung der Schüler	Moodle-Kurs Lehrer-PC Digitale Tafel
Übung und Anwendung 2 (10:17 – 10:24)	Partnerarbeit Aufgabe "Übersetzung des Maus- Intros" (7 min.)	Erklärung des Arbeitsauftrags Beantwortung von Fragen der Schüler technische Hilfestellung	Bearbeitung der Aufgabe Verständnisfragen an LK stellen evtl. Diskussion mit Sitznachbarn	4. TZ: Die Schüler <i>wenden</i> Dictionaries in Kombination mit anderen bekannten Konzepten <i>an</i> , um Probleme zu lösen. (AFB II)	Arbeitsblatt, Aufgabe 2 Moodle-Kurs
Sicherung 2 (10:24 – 10:30)	Besprechung der Aufgabe "Übersetz- ung des Maus-Intros" Interpretation und Auflösung (6 min.)	Moderation der Schülermeldungen Ergänzung der vorgestellten Lösung Fragen an Schüler stellen Beantwortung von Fragen	Präsentation der Ergebnisse aus der Arbeitsphase Ergänzung / Diskussion der vorgestell- ten Lösung Stellen von Verständnisfragen	Fachliche Konsolidierung der Schüler Lösen des Einstiegsproblems	Moodle-Kurs Lehrer-PC Digitale Tafel Videodatei "maus- deutsch.mp4"
Maximalplanung	Einzel- oder Partnerarbeit Bonusaufgabe "Personalregister" (bis zu 10 min.)	Erklärung des Arbeitsauftrags Beantwortung von Fragen der Schüler technische Hilfestellung	Bearbeitung der Aufgabe Verständnisfragen an LK stellen evtl. Diskussion mit Sitznachbarn	3. TZ: Die Schüler <i>entwerfen</i> komplexere Datenstrukturen durch Schachtelung von Dictionaries. (AFB III) sowie 4. TZ: Die Schüler <i>wenden</i> Dictionaries in Kombination mit anderen bekannten Konzepten <i>an</i> , um Probleme zu lösen. (AFB II)	Arbeitsblatt, Bonusaufgabe Moodle-Kurs

3.2 Quellenverzeichnis

- Ministerium für Kultus, Jugend und Sport Baden-Württemberg (2023). *Bildungsplan für die Fachschule für Technik, Fachrichtung Elektrotechnik*. URL: https://bildungsplaene-bw.de/site/bildungsplan-rebrush2024/get/documents_E-784390660/lsbw/Bildungsplaene-BERS/fs/FS-FT_FR-Elektrotechnik_13_3838_2023.pdf. Abgerufen am 19.01.2026.
- WDR (2026): *Die Sendung mit der Maus vom 11.01.2026*. URL: <https://www.ardmediathek.de/video/die-sendung-mit-der-maus/die-sendung-mit-der-maus-vom-11-01-2026/wdr/Y3JpZDovL3dkci5kZS9CZWl0cmFnLXNvcGhvcmEtNGU5YmVky2EtMDk4MC00NWE2LWEyMGQtYTc4MjZlMzM0Y2Ex>. Abgerufen am 16.01.2026.
- OpenAI (2026): ChatGPT (Version Januar 2026) [Large language model]. <https://chatgpt.com/share/696ced0a-0758-8011-a1ca-6ed8770ac40c>. Abgerufen am 18.01.2026.

3.3 Weitere Materialien

- Screenshot zur Quelle WDR (2026):





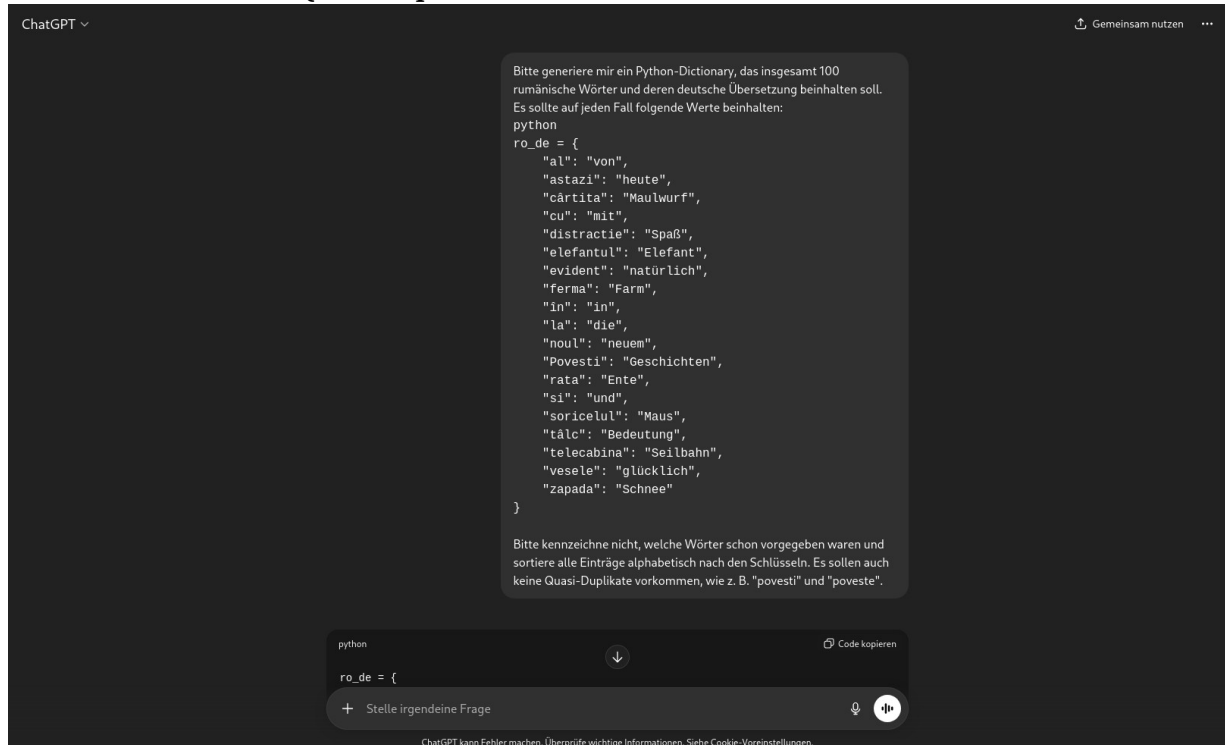
- **Prompt zur Quelle OpenAI (2026):**

Bitte generiere mir ein Python-Dictionary, das insgesamt 100 rumänische Wörter und deren deutsche Übersetzung beinhalten soll. Es sollte auf jeden Fall folgende Werte beinhalten:

```
python
ro_de = {
    "al": "von",
    "astazi": "heute",
    "cãrtita": "Maulwurf",
    "cu": "mit",
    "distractie": "Spaß",
    "elefantul": "Elefant",
    "evident": "natürlich",
    "ferma": "Farm",
    "în": "in",
    "la": "die",
    "noul": "neuem",
    "Povesti": "Geschichten",
    "rata": "Ente",
    "si": "und",
    "soricelul": "Maus",
    "tâlc": "Bedeutung",
    "telecabina": "Seilbahn",
    "vesele": "glücklich",
    "zapada": "Schnee"
}
```

Bitte kennzeichne nicht, welche Wörter schon vorgegeben waren und sortiere alle Einträge alphabetisch nach den Schlüsseln. Es sollen auch keine Quasi-Duplikate vorkommen, wie z. B. "povesti" und "poveste".

- **Screenshot zur Quelle OpenAI (2026):**



- **Arbeitsblatt "Python: Dictionaries"**
- **Lösungsvorschlag zum Arbeitsblatt**

Dictionaries

Eine häufig verwendete Datenstruktur ist das sogenannte **Dictionary**. Dabei werden Paare von Werten – bestehend aus einem **Schlüssel** und einem zugehörigen **Wert** – einander zugewiesen. Die Schlüssel sind dabei eindeutig, Werte können aber doppelt vorkommen.

Ein Dictionary wird in Python mit geschweiften Klammern angelegt, zwischen denen die Schlüssel-/Wert-Paare kommagetrennt stehen. Zwischen Schlüssel und Wert steht ein Doppelpunkt. Das könnte zum Beispiel so aussehen:

```
woerter = {"eins": 1, "zwei": 2, "drei": 3}
```

Auf die einzelnen Elemente kann man nun ähnlich wie bei einer Liste zugreifen:

```
eins = woerter["eins"] #Der Wert der Variablen ist 1
```

Auch nach dem Anlegen kann man neue Werte hinzufügen und entfernen:

```
woerter["vier"] = 4 #Jetzt steht "vier" : 4 auch im Dictionary
```

```
woerter.pop("eins")#Der Eintrag "eins" : 1 wird gelöscht
```

Die Schlüssel kann man auch als Liste verwenden (z. B. für die Verwendung in einer Schleife):

```
schluessel = woerter.keys()#schluessel = ["zwei", "drei", "vier"]
```

Aufgabe 1: Personalausweis

(🕒 15 min.)

a) Legen Sie ein Dictionary an, das Sie in Form eines Steckbriefs beschreibt. Es soll die folgenden Schlüssel haben:

- Vorname
- Nachname
- Geschlecht
- Geburtsdatum
- Körpergröße

b) Erweitern Sie den angelegten Personalausweis nachträglich um Ihre Augenfarbe und Nationalität. Geben Sie zusätzlich alle Einträge des Ausweises aus.

c) Legen Sie ein weiteres Dictionary an, das ein Datum speichern soll: Es soll die Schlüssel `day`, `month` und `year` haben. Haben Sie eine Idee, wie man dieses neue Dictionary im Personalausweis für das Geburtsdatum nutzen könnte?

Aufgabe 2: Übersetzung des Maus-Intros

(🕒 7 min.)

Laden Sie sich das vorgefertigte Python-Programm `maus.py` aus Moodle herunter. Darin sind bereits Variablen mit dem rumänischen Text sowie ein rumänisch-deutsches Wörterbuch angelegt.

Schreiben Sie nun ein Programm, das den Text Wort für Wort durchgeht und übersetzt und die Übersetzung anschließend ausgibt ¹. Können Sie mithilfe der Übersetzung sagen, worum es in der Folge wohl geht?

Bonusaufgabe 3: Personalregister für die Klasse

Legen Sie mit Ihrer Lösung aus Aufgabe 1 Personalausweise für Ihre Mitschüler an. Speichern Sie diese in einem weiteren Dictionary, welches als Personalregister dient.

Passen Sie das Programm so an, dass der Nutzer einen Namen angeben kann, zu dem er anschließend den entsprechenden Personalausweis ausgegeben bekommt.

¹Die Python-Funktion `split()` zum Aufteilen eines Strings könnte hilfreich sein: `"Hallo 123".split(" ")` wird zu einer Liste `["Hallo", "123"]`.

Aufgabe 1: Personalausweis

Der folgende Lösungsvorschlag beinhaltet Aufgabe 1a), b) und c):

```
1 # Teilaufgabe a)
2 personalausweis = {
3     "vorname" : "Max",
4     "nachname" : "Mustermann",
5     "geschlecht" : 'm',
6     # Ohne Teilaufgabe c)
7     # "geburtsdatum" : "15.11.1989",
8     # Mit Teilaufgabe c)
9     "geburtsdatum" : {
10         "tag" : 15,
11         "monat" : 11,
12         "jahr" : 1989
13     },
14     "koerpergroesse" : 181
15 }
16 # Teilaufgabe b)
17 personalausweis["nationalitaet"] = "deutsch"
18 personalausweis["augenfarbe"] = "blau"
19 for key in personalausweis.keys():
20     print(key + ": " + str(personalausweis[key]))
```

Aufgabe 2: Übersetzung des Maus-Intros

```
1 uebersetzung = ""
2 for wort in text_romanian.split(" "):
3     if wort in ro_de:
4         uebersetzung += ro_de[word]
5     else:
6         uebersetzung += wort
7     uebersetzung += " "
8 print(uebersetzung)
```