

# Unit 11a

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es `17_unit11a`! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

## 1 Schulübungen

In diesem Beispiel wird der Umgang mit Dictionaries, Schleifen und Exceptions sowie die formatierte Ausgabe geübt.

Programmiere ein Notenverwaltungsprogramm, das die Eingabe von Schülern und Punkten erlaubt, sowie als Ausgabe die Mittelwerte der Punkte berechnet.

1. Zuerst soll das Programm die Eingabe einer beliebige Anzahl von Schülern sowie deren Punkte in einem Gegenstand ermöglichen. Die Eingabe besteht jeweils aus dem Namen des Schülers sowie der Punkteanzahl.

Beendet kann die Eingabe werden, indem die Steuerung (Control)-Taste und die Taste "D" gedrückt wird (`<CTRL-D>`). Dadurch wird von Python ein `EOFError` als Exception geworfen, wenn das Programm gerade auf eine Eingabe wartet, d.h. ein `input()` ausgeführt wird.

Die Zuordnung von Name zu Punkteanzahl soll in einem Dictionary gespeichert werden. Die Ausgabe soll testweise so aussehen, dass einfach das Dictionary mittels `print()` ausgegeben wird.

Die Ein- bzw. Ausgabe soll vorerst folgendermaßen aussehen:

Die Eingabe kann mit `CTRL-D` abgebrochen werden!

```
Name:  Otto Normalverbraucher
Punkte: 76
```

```
Name:  Mini Musterfrau
Punkte: 100
```

Name: Maxi Mustermann  
Punkte: 1

Name: <CTRL-D>

```
{'Otto Normalverbraucher': 76, 'Mini Musterfrau': 100, 'Maxi Mustermann': 1}
```

2. Ändere jetzt das Programm so ab, dass die Ausgabe der Daten folgendermaßen aussieht:

```
Schüler: Otto Normalverbraucher Punkte: 76  
Schüler: Mini Musterfrau Punkte: 100  
Schüler: Maxi Mustermann Punkte: 1
```

3. Ändere jetzt das Programm weiters ab, sodass die Ausgabe jetzt so aussieht, wobei eine maximale Länge von 30 Zeichen für den Namen und insgesamt 4 Zeichen für die Punktezahl vorgesehen wird:

```
Schüler: Otto Normalverbraucher      Punkte:  76  
Schüler: Mini Musterfrau             Punkte: 100  
Schüler: Maxi Mustermann             Punkte:   1
```

4. Die Ausgabe soll jetzt noch sortiert werden. Verwende dazu die Funktion `sorted(seq)`, die ein iterierbare Datenstruktur (engl. iterable) als Parameter erwartet und eine Liste zurückliefert (Tipp: `keys()` Methode der Dictionaries verwenden):

```
Schüler: Maxi Mustermann             Punkte:   1  
Schüler: Mini Musterfrau             Punkte: 100  
Schüler: Otto Normalverbraucher      Punkte:  76
```

5. Erweitere nun das Programm folgendermaßen, dass auch der Rang hinzugefügt wird, wobei der Rang dreistellig und linksbündig ausgerichtet sein soll:

```
Schüler: Mini Musterfrau             Rang: 1   Punkte: 100  
Schüler: Otto Normalverbraucher      Rang: 2   Punkte:  76  
Schüler: Maxi Mustermann             Rang: 3   Punkte:   1
```

6. Die Ausgabe soll jetzt um die Noten gemäß des folgenden Schlüssels erweitert werden:

Punkte	Note
bis 50	5
bis 62	4
bis 78	3
bis 90	2

Die Ausgabe sieht dann folgendermaßen aus:

Schüler: Mini Musterfrau	Rang: 1	Punkte: 100	Note:1
Schüler: Otto Normalverbraucher	Rang: 2	Punkte: 76	Note:3
Schüler: Maxi Mustermann	Rang: 3	Punkte: 1	Note:5

Schreibe eine eigene Funktion zur Bestimmung der Note aus der Punkteanzahl.

7. Letztendlich soll die Ausgabe noch so abgeändert werden, dass auch der Punktedurchschnitt (Mittelwert) berechnet wird und als zusätzliche Zeile angefügt wird. Es soll keine weitere Schleife programmiert werden!

Schüler: Mini Musterfrau	Rang: 1	Punkte: 100	Note:1
Schüler: Otto Normalverbraucher	Rang: 2	Punkte: 76	Note:3
Schüler: Maxi Mustermann	Rang: 3	Punkte: 1	Note:5

3 Schüler	Schnitt: 59.0
-----------	---------------

```
{'Otto Normalverbraucher': 76, 'Mini Musterfrau': 100, 'Maxi Mustermann': 1}
```

Beachte, dass der Schnitt als Gleitkommazahl ausgegeben wird. Allerdings sollen die Nachkommastellen jetzt auf eine Stelle gerundet werden. Verwende die Funktion `round()` (Hilfe wieder mit `help()`).

1. Jetzt ist das Programm so zu erweitern, dass die Punkte für mehrere Gegenstände ermittelt werden können.
  - a) Dazu sollen **zuerst** die Schüler eingegeben werden können.
  - b) Danach sollen die Gegenstände eingegeben werden.
    - i. Dazu soll zuerst der Gegenstandsname festgelegt werden.
  - #) Dann sollen für jeden erfassten Schüler die Punkte eingegeben werden.
  - c) Im Anschluss erfolgt eine Ausgabe der Ergebnisse je Gegenstand.
  - d) Als Abschluss werden die Gegenstände aufsteigend nach den errechneten Mittelwerten der Noten ausgegeben (schwierig!).

Der Abbruch einer "Eingabeschleife" soll wieder durch `<CTRL-D>` erfolgen.

Das soll folgendermaßen aussehen:

Die Eingabe kann jeweils mit CTRL-D abgebrochen werden!

Bitte die Schüler eingeben:

Schüler: Otto Normalverbraucher

Schüler: Mini Musterfrau

Schüler: Maxi Mustermann

Schüler: <CTRL-D>

Bitte die Gegenstände und die jeweiligen Punkte eingeben:

Gegenstand: POS1

Schüler Otto Normalverbraucher: 78

Schüler Mini Musterfrau: 58

Schüler Maxi Mustermann: 34

Gegenstand: TC4A

Schüler Otto Normalverbraucher: 51

Schüler Mini Musterfrau: 49

Schüler Maxi Mustermann: 47

Gegenstand: AM

Schüler Otto Normalverbraucher: 87

Schüler Mini Musterfrau: 74

Schüler Maxi Mustermann: 55

Gegenstand: <CTRL-D>

Auswertungen

=====

Gegenstand POS1

-----

Schüler: Otto Normalverbraucher	Rang: 1	Punkte: 78	Note:2
Schüler: Mini Musterfrau	Rang: 2	Punkte: 58	Note:4
Schüler: Maxi Mustermann	Rang: 3	Punkte: 34	Note:5

3 Schüler

Schnitt: 56.7

Gegenstand: TC4A

```

-----
Schüler: Otto Normalverbraucher      Rang: 1   Punkte:   51 Note:4
Schüler: Mini Musterfrau             Rang: 2   Punkte:   49 Note:5
Schüler: Maxi Mustermann             Rang: 3   Punkte:   47 Note:5

```

```

3 Schüler                               Schnitt:  49.0

```

```

Gegenstand: AM
-----

```

```

Schüler: Otto Normalverbraucher      Rang: 1   Punkte:   87 Note:2
Schüler: Mini Musterfrau             Rang: 2   Punkte:   74 Note:3
Schüler: Maxi Mustermann             Rang: 3   Punkte:   55 Note:4

```

```

3 Schüler                               Schnitt:  72.0

```

```

Reihung der Gegenstände basierend auf den Schnitt der Punkte
=====

```

1. AM: 72.0
2. POS1: 56.7
3. TC4A: 49

Hilfestellungen:

- a) Es werden "verschachtelte" Schleifen benötigt.
- b) Schreibe eine Funktion `read_students()` zum Erfassen der Schüler, die den Benutzer nach den Schülernamen fragt und eine **Liste** der Schülernamen zurückliefert.
- c) Schreibe jetzt weiters eine Funktion `read_course(students)`, das die Liste der Schüler `students` als Parameter erhält.

Diese Funktion fragt den Benutzer zuerst nach der Gegenstandsbezeichnung und danach für jeden Schüler nach der Punkteanzahl.

Diese Funktion liefert ein **Tupel** bestehend aus Gegenstandsbezeichnung, einem Dictionary und dem Punkteschnitt zurück. Das Dictionary enthält jeweils als Key den Schülernamen und als Value die Punktezahl.

Die Ausgabe des Punktemittelwertes soll ebenfalls von dieser Funktion durchgeführt werden.

Wurde kein Gegenstand zurückgeliefert, d.h. wenn die Tasten `CTRL-D` gedrückt wurde, dann wird eine `EOFError` Exception geworfen. Diese Exception wird von der Funktion **nicht** abgefangen, sondern an die aufrufende Funktion weitergegeben.

- d) Jetzt ist das Hauptprogramm so zugestalten, dass zuerst die Funktion `read_students()` aufgerufen wird und danach so lange die Funktion `read_course()` aufgerufen wird bis diese eine `EOFError` Exception wirft.

Wird keine Exception geworfen, dann soll das zurückgelieferte Dictionary mit den Schülernamen und den Punkten unter dem zurückgelieferten Gegenstand als Key in ein Dictionary `courses` eingefügt werden.

Der Punkteschnitt soll als Value in ein Dictionary `means` unter dem Key der Gegenstandsbezeichnung eingetragen werden.

- e) Sorge jetzt, dass jede Benutzereingabe geprüft wird und das Programm auch durch Fehlbedienung nicht abstürzt.
- f) Als letzten und schwierigsten Schritt ist jetzt noch die sortierte Ausgabe der Mittelwerte ausständig.

Die sortierte Ausgabe der Mittelwerte der Gegenstände kann wieder durch die Funktion `sorted` ermittelt werden. Allerdings müssen jetzt die Einträge im Dictionary `means` gemäß den Values sortiert werden.

D.h. die Keys zu sortieren ist sinnlos. Die Werte einfach so zu sortieren (zu bekommen mittels der `values()` Methode) bringt auch nichts, da man dadurch nicht auf die Keys zugreifen kann.

Was tun? Es gibt noch die Methode `items()` für Dictionaries, die eine Liste (als ein Typ `dict_items`, aber uninteressant) von Key und Value Paaren (Tupel) zurückliefert (**ausprobieren!**). Diese Liste ist jetzt gemäß der Values zu sortieren. Aber wie?

Einfach `sorted()` zu verwenden bringt nichts, da diese Funktion die Liste zuerst nach den Keys und bei gleichen Keys nach den Values sortiert. Man muss der Funktion also mitteilen wie die Sortierung zu erfolgen hat. D.h. man muss `sorted` mitteilen, dass es nach den Values zu sortieren hat. Dies kann mit dem Keyword-Parameter `key` erreicht werden (siehe `help(sorted)`). Aber was ist diesem Parameter mitzugeben? Dieser Keywordparameter erhält eine Funktion, die jeweils einen Wert der Liste erhält und den "Wert" zurückliefert nach dem zu sortieren ist!