

MODULARBEIT

Modul 165



7. NOVEMBER 2022

SALVATORE SEVERINO

l2b

Inhaltsverzeichnis

Auftrag:.....	2
Erster Schritt 27.09.2022:.....	2
09.10.2022.....	4
25.10.2022.....	5
29.10.2022.....	7
30.10.2022.....	8
31.10.2022.....	9
01.11.2022.....	9
Zusatz 1:.....	10
Zusatz 2:.....	11
Erfolge	12
Misserfolge	12
Fazit	12
Das ist mein fertiger Code!.....	13

Auftrag:

Schreiben Sie ein Programm in Python, bei dem sie Daten aus der Datenbank abfragen können. Die Verbindung mit MongoDB erfolgt durch PyMongo. Die Datenbank soll von Ihnen erstellt und mit Daten gefüllt werden. Es handelt sich um eine Datenbank spiele mit der Collection pcgames, bei dem sie Titel, Ausgabejahr, Verkaufszahlen, Altersgrenze, Art [Array] und Wertung von 0 bis 10 angeben. Das Python-Programm soll nach verschiedenen, von Ihnen bestimmten Kriterien, die Daten aus MongoDB abrufen und in idealer Form darstellen können. Ausserdem sollen Spiele aus dem Python-Programm in die Datenbank eingepflegt und gelöscht werden können.

Erster Schritt 27.09.2022:

Als aller erstes musste ich Python neu installieren. Da hatte ich am Anfang Probleme, weil ich es mit Terminal gemacht habe, dies aber nicht funktioniert hat. Anschliessend habe ich es mit Windows PowerShell probiert und das hat auch nicht funktioniert. Ich habe im Internet recherchiert und mein Laptop Neugestartet und danach hat es funktioniert. Zuerst musste ich eine Datenbank erstellen. Die Datenbank habe ich "spiele" genannt. In dieser Datenbank gibt es eine Collection, welche PC-Games heisst. Ich habe mich für diese 10 Spiele entschieden und jedes Spiel noch ein eigener Rank gegeben. So sieht die Collection aus.

```
{"titel": "Fortnite", "ausgabejahr": "2017", "verkaufszahlen": "129000000", "altersgrenze": "12",
"art": ["Battle Royale", "Onlinespiel"], "wertung": 9}
```

```
{"titel": "Forza Horizon 5", "ausgabejahr": "2021", "verkaufszahlen": "1000000", "altersgrenze": "6",
"art": ["Rennspiel", "simulationsspiel", "Adventure"], "wertung": 8}
```

```
{"titel": "Grand Theft Auto V", "ausgabejahr": "2013", "verkaufszahlen": "165000000",
"altersgrenze": "18", "art": ["Open-World-Spiel", "Action-Adventure", "Ego-Shooter"], "wertung": 9}
```

```
{"titel": "Just Dance 2020", "ausgabejahr": "2017", "verkaufszahlen": "200000", "altersgrenze": "0",
"art": ["Rhythmusspiel", "Actionspiel", "Exergaming"], "wertung": 4}
```

```
{"titel": "Rocket League", "ausgabejahr": "2015", "verkaufszahlen": "4000000", "altersgrenze": "8",
"art": ["Fussball", "Sportsimulation", "Actionspiel"], "wertung": 7}
```

```
{"titel": "Call of Duty 4: Modern Warfare", "ausgabejahr": "2007", "verkaufszahlen": "200000000",
"altersgrenze": "18", "art": ["Taktik-Shooter"], "wertung": 8}
```

```
{"titel": "Fall Guys", "ausgabejahr": "2020", "verkaufszahlen": "2000000", "altersgrenze": "6", "art":
["Battle Royale", "Jump n Run", "Multiplayer Online Game"], "wertung": 5}
```

```
{"titel": "Among Us", "ausgabejahr": "2018", "verkaufszahlen": "264000000", "altersgrenze": "6",
"art": ["Survival-Spiel", "Social deduction game", "Party-Computerspiel"], "wertung": 4}
```

```
{"titel": "Farming Simulator 22", "ausgabejahr": "2021", "verkaufszahlen": "3000000", "altersgrenze":
"0", "art": ["simulationsspiel", "Computer-Strategiespiel"], "wertung": 2}
```

```
{"titel": "Playerunknown's Battlegrounds", "ausgabejahr": "2016", "verkaufszahlen": "15000000",
"altersgrenze": "18", "art": ["Battle Royale", "Shooter-Spiel", "Free-to-play"], "wertung": 6}
```

Nachdem ich die Collection gemacht habe, habe ich eine Verbindung mit Visual Studio Code gemacht. Diese funktioniert mit dem Befehl: "siehe Bild"

```
import pymongo

client=pymongo.MongoClient("mongodb://localhost:27017/")
mydb=client["spiele"]
mycol=mydb["pcgames"]
```

Wie man auch noch sieht, habe ich `mydb=client["spiele"]`, so weiss er, welche Datenbank er nehmen sollte und mit `mycol=mydb["pcgames"]`, so weiss er, welche Collection gemeint ist. Ich habe mir Hilfe von der Webseite w3schools geholt.

Die Collection habe ich in MongoDB Compass importiert. Man muss auf ADD DATA gehen und Import File auswählen, dann öffnet sich dieses Fenster.

Example

Return all documents in the "customers" collection, and print each document:

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["customers"]

for x in mycol.find():
    print(x)
```

Run example >

Import To Collection spiele.pcgames

Select File

Select a file...

Select Input File Type

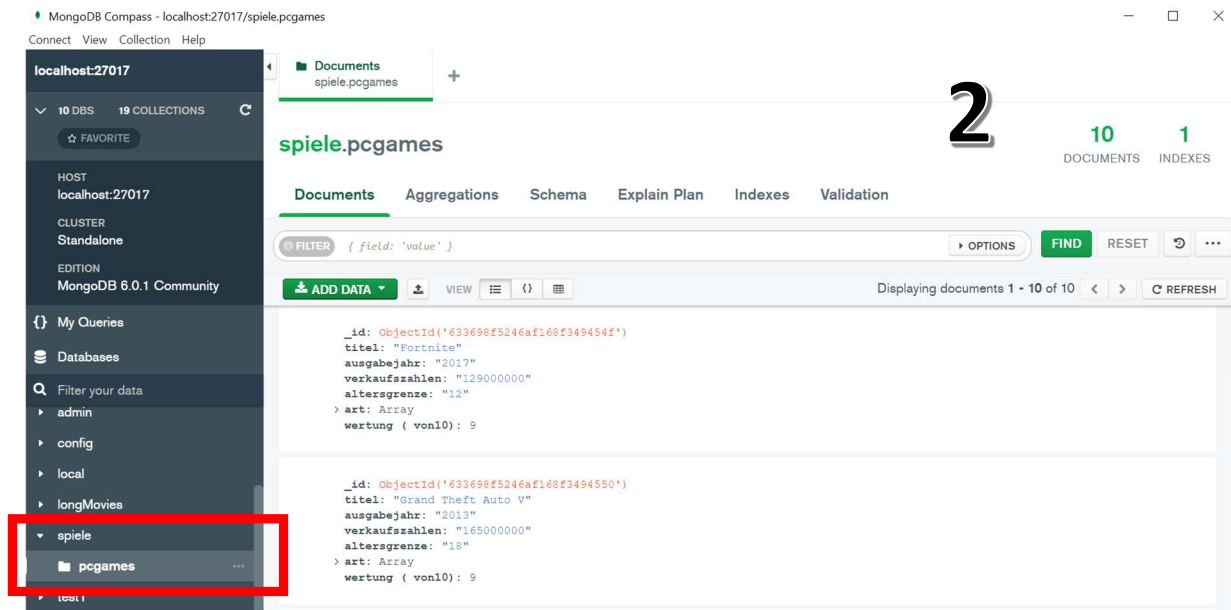
JSON CSV

Options

☐ Stop on errors

CANCEL IMPORT

Zuerst habe ich alles in einer JSON-Datei hineingeschrieben und im Nachhinein habe ich es importiert. Nachdem alles problemlos importiert wurde, sieht es so aus. (siehe Bild 2) Mein Ziel ist es nachher, dass ich sie nicht mehr über das Kompass importieren muss, sondern über mein kleines Programm in Visual Studio Code.



Das habe ich alles am 27. September 2022 während der Informatikstunde gemacht. Ein YouTube Tutorial hat mir sehr geholfen, alle diese Schritte zu machen.

<https://www.youtube.com/watch?v=YbLzV90dksE>

Ich hatte noch eine strenge Woche mit Prüfungen hinter mir und in der ersten Ferienwoche war ich nicht zuhause und konnte somit nicht weiterarbeiten.

09.10.2022

Heute am 09. Oktober 2022 habe ich mich entschieden weiterzuarbeiten. Bevor ich Anfangen konnte, musste ich alles Definieren.

Ich habe zuerst angefangen die Suchen Funktion zu definieren.

```
def findenInt(suchen,titel): #Suchen funktion INT / Für 1
    s = mycol.find_one({suchen:(int(titel))})
    print(s)
```

Diese Funktion sucht nach Zahlen, weil sie INT ist. Ich habe auch noch eine Funktion gemacht, die nicht nach INT sucht, sondern nach einem String sucht.

```
def finden(suchen8,titel8): #Suchen funktion Normal für String / Für 1
    s = mycol.find_one({suchen8:titel8})
    print(s)
```

Diese beiden Befehle suchen immer nach dem ersten Wert in der Collection. Ich muss ebenfalls noch eine Definition erstellen, die nach mehreren Werten sucht. Hier muss ich ebenfalls eine Für INT und eine für String erstellen.

```
def mehreresuchenINT(suchen5,titel5): #Suchen funktion INT / Für mehrere
    mysa = {suchen5:(int(titel5))}
    docsa = mycol.find(mysa,{ "_id": 0})
    for s in docsa:
        print(s)
```

```
def mehreresuchen(suchen3,titel3): #Suchen funktion Normal für String / Für  
mehrere  
    mysa = {suchen3:titel3}  
    docsa = mycol.find(mysa,{ "_id": 0})  
    for s in docsa:  
        print(s)
```

Bei mir hat es am Anfang nicht richtig funktioniert und somit habe ich nochmals recherchiert und die W3schools Seite hat mir sehr weitergeholfen.

Anschliessend wollte ich so ein kleines Menü machen. Das habe ich in " " hineingeschrieben und somit ist es dann ein Text. Damit das aber funktioniert, muss man eine Print Funktion machen.

```
print("WILLKOMMEN IN DER DATENBANK SPIELE. WAS MÖCHTEN SIE MACHEN?\n\n1)Nach  
einem Eintrag suchen\n2)Etwas in die Collection hinzufügen\n3)einen Datensatz  
löschen\nProgramm beenden? (Gebe 'beenden' ein: ")
```

Ich habe noch ein kleiner Titel gemacht. Dieser wird angezeigt, sobald man das Programm startet. Anschliessend wird auch direkt gefragt, was man machen möchte.

Nächste Woche mache ich ein Ferienjob und denke nicht, dass ich die Zeit habe, weiterzuarbeiten. Anschliessend haben wir direkt die Schulkolonie in Fiesch. Weil es nicht sicher gegen Diebstahl ist, haben die Lehrpersonen uns geraten unser Laptop zuhause zu lassen. Somit muss ich nach der Kolonie weiterarbeiten.

25.10.2022

Heute war wieder das erste Mal, wo wir nach 4 Wochen wieder Normalunterricht hatten. Ich musste wieder ein wenig hineinkommen, weil ich nach der Kolonie, mich nicht mehr ganz an alles erinnern konnte.

Heute habe ich mit dem Definieren weitergemacht und anschliessen auch mein Programm geschrieben. Zuerst habe ich eingegeben, ob er nach eins oder mehreren Datensätze suchen sollte. Ich habe 1, 2, 3, beenden als Abkürzung verwendet, weil es sonst zu viel wäre jedes Mal alles auszuschreiben.

So sieht mein geschriebenes Programm aus, welcher man braucht, um etwas in der Collection zu suchen. In der Collection gibt es Dokumente, welche die gleiche Wertung etc. haben. So musste ich mit einer If Funktion anfangen. Wenn ich 1 eingebe, soll er nach 1 Dokument suchen. Wenn es trotzdem mehrere hätte, nimmt es den ersten Wert. In dieser Funktion gibt es nochmal eine If-Funktion wo geprüft wird, ob es eine Zahl ist, oder ein String. Das genau Gleiche habe ich bei der zweiten Funktion gemacht. (Siehe Code auf der nächsten Seite)

```

if eingabe == "1":
    frage=input("eins oder mehrere Datensätze suchen?(1 oder 2)\n--->")
    if frage == "1": #es sucht nur nach 1 Datensatz
        print("Nach was möchten Sie suchen?")
        print("titel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwertung\n--->")
        eingabe8= input("---> ")
        print("Wie heisst Ihr Objekt, nachdem Sie suchen?: ")
        salvi=input("---> ")
        if salvi.isdigit():
            findenInt(eingabe8,salvi)
        else:
            finden(eingabe8,salvi)
    elif frage == "2": #es sucht nach mehreren Datensätzen
        print("Nach was möchten Sie suchen?")
        print("titel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwertung\n--->")
        eingabe8= input("---> ")
        print("Gesuchtes Objekt eingeben: ")
        salvi=input("---> ")
        if salvi.isdigit():
            mehreresuchenINT(eingabe8,salvi)
        else:
            mehreresuchen(eingabe8,salvi)

```

Problem: Ich konnte bei meiner Suchen Funktion nach Buchstaben suchen, aber ich konnte nicht nach wertung etc. suchen. Jetzt habe ich herausgefunden, dass ich bei meiner JSON-Datei ein Fehler gemacht habe und alles in String hinein gemacht hatte. Das habe ich jetzt geändert und jetzt funktioniert es. Ich habe sehr sehr lange nach diesem Fehler gesucht und ich dachte mir, was ist Falsch?

So sieht die JSON-Datei jetzt aus:

```

{"titel": "Fortnite", "ausgabejahr": 2017, "verkaufszahlen": 129000000, "altersgrenze": 12, "art":
["Battle Royale","Onlinespiel"], "wertung": 9}
{"titel": "Forza Horizon 5", "ausgabejahr": 2021, "verkaufszahlen": 1000000, "altersgrenze": 6, "art":
["Rennspiel","simulationsspiel","Adventure"], "wertung": 8}
{"titel": "Grand Theft Auto V", "ausgabejahr": 2013, "verkaufszahlen": 165000000, "altersgrenze": 18,
"art": ["Open-World-Spiel","Action-Adventure","Ego-Shooter"], "wertung": 9}
{"titel": "Just Dance 2020", "ausgabejahr": 2017, "verkaufszahlen": 200000, "altersgrenze": 0, "art":
["Rhythmusspiel","Actionspiel","Exergaming"], "wertung": 4}
{"titel": "Rocket League", "ausgabejahr": 2015, "verkaufszahlen": 4000000, "altersgrenze": 8, "art":
["Fussball","Sportsimulation","Actionspiel"], "wertung": 7}
{"titel": "Call of Duty 4: Modern Warfare", "ausgabejahr": 2007, "verkaufszahlen": 200000000,
"altersgrenze": 18, "art": ["Taktik-Shooter"], "wertung": 8}
{"titel": "Fall Guys", "ausgabejahr": 2020, "verkaufszahlen": 2000000, "altersgrenze": 6, "art":
["Battle Royale","Jump n Run","Multiplayer Online Game"], "wertung": 5}
{"titel": "Among Us", "ausgabejahr": 2018, "verkaufszahlen": 264000000, "altersgrenze": 6, "art":
["Survival-Spiel","Social deduction game"," Party-Computerspiel"], "wertung": 4}
{"titel": "Farming Simulator 22", "ausgabejahr": 2021, "verkaufszahlen": 3000000, "altersgrenze": 0,
"art": ["simulationsspiel"," Computer-Strategiespiel"], "wertung": 2}
{"titel": "Playerunknown's Battlegrounds", "ausgabejahr": 2016, "verkaufszahlen": 15000000,
"altersgrenze": "18", "art": ["Battle Royale","Shooter-Spiel","Free-to-play"], "wertung": 6}

```

So sah es vorher aus und so jetzt.

Vorher	Nachher
<pre> _id: ObjectId('633698f5246af168f349454f') titel: "Fortnite" ausgabejahr: "2017" verkaufszahlen: "1290000000" altersgrenze: "12" > art: Array wertung (von10): 9 </pre>	<pre> _id: ObjectId('6357ee11209a677428cd88bd') titel: "Fortnite" ausgabejahr: 2017 verkaufszahlen: 1290000000 altersgrenze: 12 > art: Array wertung: 9 </pre>

Ich war sehr froh, als mein Suchen Funktion funktioniert hat.

Lösung: Es lag daran, weil ich alles in "." hineingeschrieben habe und es immer danach gesucht hat, obwohl es eine Zahl war.

Es hat mich sehr viel Zeit gekostet, hat sich aber gelohnt, nicht aufzugeben, weil ich jetzt weiter programmieren kann. Jetzt muss ich mich noch an die anderen Funktionen wagen.

29.10.2022

Heute Morgen habe ich zuerst angefangen damit, im Internet zu recherchieren. Ich wollte heute die beenden Funktion vorbereiten. Beenden habe ich in einer While-Schleife hineingetan. Damit diese Schleife unterbrochen wird, muss man ein (!=) vor dem gewünschten Wort schreiben. Das sorgt dafür, das mein Programm dann aus der Schleife heraus fällt und somit das Programm beendet wird.

Das sieht so aus.

```
while eingabe != "beenden":
```

Anhand dieser Funktion konnte ich noch direkt weitermachen, dass wenn etwas gefunden worden ist, ich entscheiden kann, ob man nochmals nach etwas Suchen möchte oder nicht.

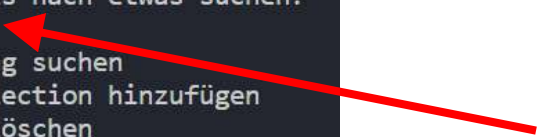
```
Print("\nMöchten Sie nochmals nach etwas suchen?\n\n1)Nach einem Eintrag
suchen\n2)Etwas in die Collection hinzufügen\n3)Einen Datensatz
löschen\nProgramm beenden? (Gebe 'beenden' ein: ")
eingabe=input()
```

Da dieser Befehl in einer Schleife drin ist, wird dieser Vorgang so lange wiederholt, bis man sich dazu entscheidet das Programm zu Beenden. Das (\n) sorgt dafür, dass eine neue Zeile beginnt.

```

Möchten Sie nochmals nach etwas suchen?
1)Nach einem Eintrag suchen
2)Etwas in die Collection hinzufügen
3)Einen Datensatz löschen
Programm beenden? (Gebe 'beenden' ein:

```



1


```
Print("\nMöchten Sie nochmals nach etwas suchen?\n\n1) nach einem Eintrag
suchen\n2) Etwas in die Collection hinzufügen\n3) einen Datensatz
löschen\nProgramm beenden? (Gebe 'beenden' ein: ")
eingabe=input()
```

Wenn man das Zweimal nacheinander schreibt, gibt es so ein grosser Abstand (siehe Pfeil 1).

30.10.2022

Heute habe ich die Funktion gemacht, um Datensätze zu löschen. Zuerst musste ich alles Definieren.

```
def loeschendatensatz(löschenD, wNameloeschen): #Suchen funktion für
Buchstaben (String) / Für mehrere
    loeschenvARIABLE = {löschenD : wNameloeschen}
    x = mycol.delete_many(loeschenvARIABLE)
    print(x.deleted_count, " documents deleted.")

def loeschendatensatzINT(löschenI, wNameloeschenI): #Suchen funktion für INT /
Für mehrere
    loeschenvARIABLE = {löschenI:(int(wNameloeschenI))}
    x = mycol.delete_many(loeschenvARIABLE)
    print(x.deleted_count, " documents deleted.")
```

Ich musste zwei Definitionen machen, damit man sowohl bei Buchstaben (String) als auch bei Zahlen löschen kann. Bei der letzten Print-Line wird gezählt, wie viele Dokumente gelöscht worden sind.

```
elif eingabe == "3": #Löschen, nach was möchte ich löschen
    frage = input("Welchen Datensatz möchtest du
löschen?\ntitel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwertung\n---> ")
    wNameloeschen = input("Geben Sie ein, was Sie Löschen möchten. ---> ")
    if wNameloeschen.isdigit(): # Löschen INT
        loeschendatensatzINT(frage, wNameloeschen)
    else: # Löschen
        loeschendatensatz(frage, wNameloeschen)
```

Zum Dokumente Löschen habe ich mich für den Short-Cut 3 entschieden. Zuerst fragt man, nach was er suchen sollte, damit er es Löschen kann. Da gibt es das zur Auswahl: Titel, Ausgabejahr, Verkaufszahlen, Altersgrenze, Art, Wertung. Wenn es jetzt mehrere Dokumente mit derselben Wertung gibt, werden beide gelöscht, weil er zuerst nach dieser Zahl sucht und anschliessend alles löscht, wegen dem Delete Many.

```
wertung
---> ausgabejahr
Geben Sie ein, was Sie Löschen möchten. ---> 1900
2 documents deleted.
```

31.10.2022

Heute fehlte mir nur noch die Funktion ein neues Dokument erstellen. Ich habe wieder angefangen mit dem Definieren der Insert Funktion. Da die Namen in den Gänsefüßchen nicht gleich sein dürfen, wie die in der Insert Klammer. Die Abkürzung "docsa" steht Dokument Salvi. Der Name kann alles sein. Es braucht wieder eine Variable und hier habe ich i genommen wegen Insert. Am Schluss printet es die Id aus und es steht noch "document successfully created", wenn alles funktioniert hat.

```
def insert(titell,ausgabejahrr,verkaufszahlenn,altersgrenzee,artt,wertungg):
#Einfügen in die Collection / Insert
    docsa = {"titel":titell,"ausgabejahr":ausgabejahrr,"verkaufszahlen":
verkaufszahlenn,"altersgrenze":altersgrenzee,"art":[artt],"wertung":
wertungg}
    i = mycol.insert_one(docsa)
    ausgabeprint = mycol.find_one({"titel":titell})
    print(ausgabeprint, "document successfully created")
```

```
elif eingabe == "2": #Etwas in die Collection einfügen.
    print("Füllen Sie die Untenstehende Informationen aus und der Eintrag wird erstellt.")
    titell = input("titel: ") #String
    ausgabejahrr = int(input("ausgabejahr: ")) #INT
    verkaufszahlenn = int(input("verkaufszahlen: ")) #INT
    altersgrenzee = int(input("altersgrenze: ")) #INT
    artt=input("art: ") #Array
    wertungg = int(input("wertung: ")) #INT
    insert(titell,ausgabejahrr,verkaufszahlenn,altersgrenzee,artt,wertungg) #Mit diesem Befehl kann
```

Ich musste ebenfalls angeben, ob es sich um einen INT handelt (Blauer Pfeil). Beim Array muss man die Werte mit einem Komma trennen und diese werden dann in einem Array gespeichert. Da ich noch Zeit hatte und es schöner gefunden habe, habe ich noch gemacht, dass es beim Printen alle Werte nochmals anzeigt. Das sieht dann so aus.

```
{'_id': ObjectId('6360c1ce509d4776173f0f9d'), 'titel': 'Salvi', 'ausgabejahr': 21, 'verkaufszahlen': 21, 'altersgrenze': 21, 'art': ['shooter'], 'wertung': 1} document successfully created
```

01.11.2022

Ich dachte wir hätten unser Projekt heute abgeben müssen. Ich habe mir alles eingeteilt, damit ich heute nur noch mein Journal verschönern kann und nur Kleinigkeiten ändern muss. Da der Abgabetermin sich jetzt um eine Woche verschoben hat, habe ich jetzt noch Zeit, um eine andere Funktion zu integrieren.

Zusatz 1:

Meine Idee war es jetzt noch eine Funktion mit Regex. Ich möchte auch mit dem Anfangsbuchstaben Suchen können.

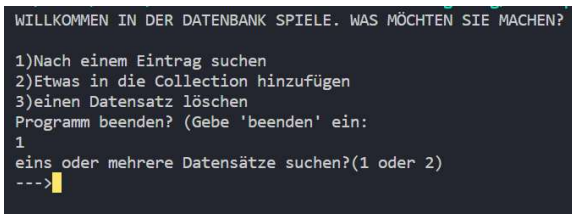
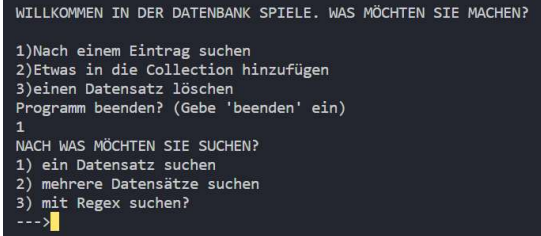
```
def regexsuche(regexx):
    sasa = mycol.find({"titel": {"$regex": regexx}}, {"_id" : 0})
    for s in mycol.find(regexend, {"_id": 0}):
        print(s)
```

Das ist meine Defintion für den Regex Befehl. Ich es ähnlich gemacht, wie bei den anderen Funktionen.

```
elif frage == "3": #Regex
    print("ANFANGSBUCHSTABE EINGEBEN, NACHDEM SIE SUCHEN MÖCHTEN.")
    regexfrage = "titel"
    regexx = input()
    regexend = {"$regex": regexx}
    for s in mycol.find(regexend, {"_id": 0}):
        print(s)
```

Wenn man die Ziffer 3 eingibt, löst es diese Funktion aus. Zuerst printet es, dass man der Anfangsbuchstabe eingeben soll. Die Input Funktion sorgt dafür, dass ich etwas eingeben kann. Am Schluss wird die komplette Collection durchsucht und gibt es mir zurück.

Was ich noch ändern musste, was vorher anders war, ist die Konsole.

Vorher	Nachher
 <pre>WILLKOMMEN IN DER DATENBANK SPIELE. WAS MÖCHTEN SIE MACHEN? 1)Nach einem Eintrag suchen 2)Etwas in die Collection hinzufügen 3)einen Datensatz löschen Programm beenden? (Gebe 'beenden' ein: 1 eins oder mehrere Datensätze suchen?(1 oder 2) --></pre>	 <pre>WILLKOMMEN IN DER DATENBANK SPIELE. WAS MÖCHTEN SIE MACHEN? 1)Nach einem Eintrag suchen 2)Etwas in die Collection hinzufügen 3)einen Datensatz löschen Programm beenden? (Gebe 'beenden' ein) 1 NACH WAS MÖCHTEN SIE SUCHEN? 1) ein Datensatz suchen 2) mehrere Datensätze suchen 3) mit Regex suchen? --></pre>

So sieht der Rückgabewert aus.

```
WILLKOMMEN IN DER DATENBANK SPIELE. WAS MÖCHTEN SIE MACHEN?
1) Nach einem Eintrag suchen
2) Etwas in die Collection hinzufügen
3) einen Datensatz löschen
Programm beenden? (Gebe 'beenden' ein)
--->1
NACH WAS MÖCHTEN SIE SUCHE?
1) ein Datensatz suchen
2) mehrere Datensätze suchen
--->2
ANFANGSBUCHSTABE EINGEBEN, NACHDEM SIE SUCHE MÖCHTEN.
Fortnite
{'titel': 'Fortnite', 'ausgabejahr': 2017, 'verkaufszahlen': 129000000, 'altersgrenze': 12, 'art': ['Battle Royale', 'Onlinespiel'], 'wertung': 9}
```

Zusatz 2:

Da mir diese Darstellung nicht so gefällt, habe ich es noch abgeändert. Ich möchte das geprintete nicht in JSON-Format erhalten.

Ich habe mit Marc zusammen mehrere Stunden probiert und wollten diese Funktion schaffen. Wir haben auf mehrere Internetseiten gesucht und wollten schauen, ob wir etwas finden. Leider haben die nicht weitergeholfen. Wir haben uns von Hand eine Skizze gemacht und probiert es nachzuprogrammieren. Wir haben es schlussendlich geschafft, aber leider konnte man diese Funktion nur mit String benutzen und nicht mit INT oder ARRAY. Somit habe ich mich entschieden es nicht in meinem Programm zu integrieren, sondern es nur in meinem Journal zu schreiben.

```
import pymongo
import json

client=pymongo.MongoClient()
mydb=client["spiele"]
mycol=mydb["pcgames"]

def umwandler(x):
    y = json.loads(x) #array
    umweingabe = "0"
    myarray = ["titel", "ausgabejahr", "verkaufszahlen", "altersgrenze", "art", "wertung"]
    while umweingabe != "3":
        print("1: alle \n2: einzeln")
        umweingabe = input()
        if umweingabe == "1":
            counternw = 0
            for i in y:
                print(myarray[counternw] + ": " + y[i])
                counternw = counternw + 1

        elif umweingabe == "2":
            arber = ""
            menq = "0"
            print("was?")
            coke = input()
            arber = (arber + coke + ": " + y[coke])
```

```
while menq != "2":
    print("1:noch eins 2:ende")
    menq = input()
    if menq == "1":
        print("was?")
        coke = input()
        arber = (arber+ " , " + coke + ": " + y[coke])
    print(arber)

x = '{"titel": "Fortnite", "ausgabejahr": "2017", "verkaufszahlen": "129000000",
"altersgrenze": "12", "art": "yeeet", "wertung": "9"}'
umwandler(x)
```

Erfolge

Ich war jedes Mal sehr Happy, als ich eine Funktion zum Laufen gebracht habe. Das erfreute mich sehr. Mit dieser Arbeit konnte ich mein Python Wissen wieder auffrischen. Ich habe viele neue Funktionen gelernt. Ich habe jetzt auch gelernt, wie ich eine Datenbank in Python importieren kann. Ich habe mein Mongo Wissen wieder aufgefrischt. Während dieser Arbeit habe ich sehr viel neues gelernt, was ich in Zukunft sicher benutzen kann. Jedes gelöste Problem, war ein riesiger Erfolg für mich.

Misserfolge

Ich hatte einige Probleme, beim Schreiben meines Programms. Es gab ein paar Dinge, welche nicht auf anhieb funktioniert haben, deswegen musste ich immer nach Lösungen suchen. Ich habe im Internet recherchiert, oder meine Kollegen gefragt, ob sie wissen, wie ich es machen kann. Bei der Zusatzfunktion, welche Marc und ich machen wollten, mussten wir am Anfang nicht mehr weiter, bis wir uns eine Skizze gemacht haben.

Fazit

Während meinem Projekt habe ich sehr viel neues gelernt. Ich habe sehr viel Spass gehabt. Ich würde gerne nochmal so ein Programm machen, weil man immer wieder vor neuen Herausforderungen gestellt wird. Die 5 Lektionen mehr die wir bekommen haben waren sehr gut, weil ich während diesen Lektionen alles beenden konnte. Die 5 Lektionen vom 8. November konnte ich nicht ausnutzen, weil ich eine Operation habe. Deswegen habe ich auch viel während meiner Freizeit an meiner Arbeit gearbeitet.

Das ist mein fertiger Code!

```
import pymongo

client=pymongo.MongoClient("mongodb://localhost:27017/")
mydb=client["spiele"]
mycol=mydb["pcgames"]

def finden(suchen8,titel8): #Suchen funktion Normal für String / Für 1
    s = mycol.find_one({suchen8:titel8})
    print(s)

def findenInt(suchen,titel): #Suchen funktion INT / Für 1
    s = mycol.find_one({suchen:(int(titel))})
    print(s)

def insert(titell,ausgabejahrr,verkaufszahlenn,altersgrenzee,artt,wertungg):
    #Einfügen in die Collection / Insert
    docsa
    ={"titel":titell,"ausgabejahr":ausgabejahrr,"verkaufszahlen":verkaufszahlenn,"
altersgrenze":altersgrenzee,"art":[artt],"wertung":wertungg}
    i = mycol.insert_one(docsa)
    ausgabeprint = mycol.find_one({"titel":titell})
    print(ausgabeprint, "document successfully created")

def mehreresuchen(suchen3,titel3): #Suchen funktion Normal für String / Für
mehrere
    mysa = {suchen3:titel3}
    docsa = mycol.find(mysa,{ "_id": 0})
    for s in docsa:
        print(s)

def mehreresuchenINT(suchen5,titel5): #Suchen funktion INT / Für mehrere
    mysa = {suchen5:(int(titel5))}
    docsa = mycol.find(mysa,{ "_id": 0})
    for s in docsa:
        print(s)

def loeschendatensatz(löschenD, wNameloeschen): #Suchen funktion für
Buchstaben (String) / Für mehrere
    loeschenvariable = {löschenD : wNameloeschen}
    x = mycol.delete_many(loeschenvariable)
    print(x.deleted_count, " documents deleted.")

def loeschendatensatzINT(löschenI, wNameloeschenI): #Suchen funktion für INT /
Für mehrere
```

```

loeschenvariable = {löschenI:(int(wNameloeschenI))}
x = mycol.delete_many(loeschenvariable)
print(x.deleted_count, " documents deleted.")

def regexsuche(regexx):
    sasa = mycol.find({"titel": {"$regex": regexx}}, {"_id" : 0})
    for s in mycol.find(regexend, {"_id": 0}):
        print(s)

print("\nWILLKOMMEN IN DER DATENBANK SPIELE. WAS MÖCHTEN SIE MACHEN?\n\n1) Nach
einem Eintrag suchen\n2) Etwas in die Collection hinzufügen\n3) einen Datensatz
löschen\nProgramm beenden? (Gebe 'beenden' ein)")
eingabe = input("--->")
#Ab hier Beginnt meine While-Schleife, es beginnt immer wieder von vorne.
while eingabe != "beenden": #Unterbruch, wenn ich dieses Wort(beenden)
    eingebe.
    if eingabe == "1":
        print("NACH WAS MÖCHTEN SIE SUCHEN?")
        frage=input("1) ein Datensatz suchen\n2) mehrere Datensätze suchen\n3)
mit Regex suchen?\n--->")
        if frage == "1": #es sucht nur nach 1 Datensatz
            print("NACH WELCHEN KRITERIEN MÖCHTEN SIE SUCHEN?")
            print("titel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwert
tung")

            eingabe8= input("--> ")
            print("Wie heisst Ihr Objekt, nachdem Sie suchen?: ")
            salvi=input("---> ")
            if salvi.isdigit():
                findenInt(eingabe8,salvi)
            else:
                finden(eingabe8,salvi)
        elif frage == "2": #es sucht nach mehreren Datensätzen
            print("NACH WELCHEN KRITERIEN MÖCHTEN SIE SUCHEN?")
            print("titel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwert
ung")

            eingabe8= input("---> ")
            print("Wie heisst Ihr Objekt, nachdem Sie suchen?: ")
            salvi=input("---> ")
            if salvi.isdigit():
                mehreresuchenINT(eingabe8,salvi)
            else:
                mehreresuchen(eingabe8,salvi)
        elif frage == "3": #Regex
            print("ANFANGSBUCHSTABE EINGEBEN, NACHDEM SIE SUCHEN MÖCHTEN.")
            regexxfrage = "titel"
            regextext = input()
            regexend = {'$regex': regextext}}

```

```
        for s in mycol.find(regexend, {"_id": 0}):
            print(s)

    elif eingabe == "3": #Löschen, nach was möchte ich löschen
        frage = input("Welchen Datensatz möchtest du
löschen?\ntitel\nausgabejahr\nverkaufszahlen\naltersgrenze\nart\nwertung\n--->
")
        wNameloeschen = input("Geben Sie ein, was Sie Löschen möchten. ---> ")
        if wNameloeschen.isdigit(): # Löschen INT
            loeschendatensatzINT(frage, wNameloeschen)
        else: # Löschen
            loeschendatensatz(frage, wNameloeschen)
    elif eingabe == "2": #Etwas in die Collection einfügen.
        print("Füllen Sie die Untenstehende Informationen aus und der Eintrag
wird erstellt.")
        titell = input("titel: ") #String
        ausgabejahr = int(input("ausgabejahr: ")) #INT
        verkaufszahlenn = int(input("verkaufszahlen: ")) #INT
        altersgrenzee = int(input("altersgrenze: ")) #INT
        artt=input("art: ") #Array
        wertungg = int(input("wertung: ")) #INT
        insert(titell,ausgabejahr,verkaufszahlenn,altersgrenzee,artt,wertungg
) #Mit diesem Befehl kann ich sagen, dass es in MongoDBCompass eingefügt wird.

        print("\nMöchten Sie nochmals nach etwas suchen?\n\n1)Nach einem Eintrag
suchen\n2)Etwas in die Collection hinzufügen\n3)Einen Datensatz
löschen\nProgramm beenden? (Gebe 'beenden' ein)")
        eingabe=input()
```