
Übungsblatt 3

1. Betrachten wir folgendes Python3 Programm. Erklären Sie die Bedeutung für jede Zeile des Programms in jeweils einem Satz. Hinweis: itertools ist eine Sammlung von nützlichen Python-Befehlen. Diese ist notwendig für die 5. Zeile.

```
1  import itertools
2  # 1. Grundmenge erstellen:
3  Augenzahlen = {1, 2, 3, 4, 5, 6}
4  #itertools.product liefert das kartesische Produkt:
5  omega = set(itertools.product(Augenzahlen, Augenzahlen))
6  print(omega)
7  #Beachte, das omega als Menge nicht sortiert ist.
8  #Staat "set" kann auch "list" geschrieben werden:
9  #dann wird die Menge schon sortiert.
10
11 #2. Anzahl Günstige und Mögliche bestimmen:
12 günstig = 0
13 möglich = len(omega)
14 for Augenzahl1, Augenzahl2 in omega:
15     if Augenzahl1 + Augenzahl2 == 10:
16         günstig = günstig + 1
17
18 #3. Ausgabe:
19 print(f"{günstig}von{möglich}")
20 print(f"p={günstig/möglich}")
```

Zeile1: Hier wird die Sammlung itertools in unser Programm importiert/verknüpft.

Zeile2: mit # schreibt man einen Kommentar, dies dient zur besseren Erklärung bzw. um einen Überblick zu erhalten. Wir sagen nun „1. Grundmenge erstellen“ (Schritt-für-Schritt)

Zeile3: Wir definieren die Augenzahlen unseres Würfels.

Zeile4: Wieder ein Kommentar

Zeile5: Wir definieren nun die Grundmenge, wenn wir zweimal würfeln.

Zeile6: Ausgabe der Grundmenge.

Zeile7 – 9: Sind wieder Kommentare, in welchen wir erklären das set = unsortiert und list = sortiert ist.

Zeile11: Kommentar zur Erklärung des nächsten Schrittes

Zeile12: Wir definieren die günstigen Fälle, kann man sich als Variable vorstellen beginnend bei 0.

Zeile13: Wir definieren die möglichen Fälle, kann man sich als Variable vorstellen, hier meinen wir die Gesamtanzahl von omega.

Zeile14: nun wollen wir wissen, wenn wir zweimal würfeln (deswegen Augenzahl1 und Augenzahl2) was in Omega liegt.

Zeile 15: Hier wird verlangt das Augenzahl1 und Augenzahl2 addiert 10 ergeben soll

Zeile16: Wir definieren wieder, dass günstige nicht 0 sein darf

Zeile 18: Kommentar

Zeile19: Wir wollen, dass uns die Anzahl an Möglichkeiten ausgegeben wird.

Zeile20: Berechnung der Wahrscheinlichkeit.

```
{ (3, 4), (4, 3), (3, 1), (5, 4), (4, 6), (5, 1), (2, 2), (1, 0),  
  (2, 5), (1, 3), (6, 2), (6, 5), (4, 2), (4, 5), (3, 3), (5, 6),  
  (3, 6), (5, 3), (2, 4), (1, 2), (2, 1), (1, 5), (6, 1), (6, 4),  
  (3, 2), (4, 1), (3, 5), (5, 2), (4, 4), (5, 5), (1, 1), (1, 4),  
  (2, 3), (2, 6), (6, 6), (6, 3) }  
3von36  
p=0.08333333333333333
```

2. Nutzen Sie list statt set in der 5. Zeile. Wie ändert sich das Resultat?

```
[ (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2),  
  (2, 3), (2, 4), (2, 5), (2, 6), (3, 1), (3, 2), (3, 3), (3, 4),  
  (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6),  
  (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (6, 1), (6, 2),  
  (6, 3), (6, 4), (6, 5), (6, 6) ]  
3von36  
p=0.08333333333333333  
[]
```

Es wird die Grundmenge geordnet – sowie es ein fauler Mathematiker aufschreiben würde, damit er keinen Fall vergisst.

3. Zählen Sie die Augenzahlen in der 5. Zeile dreimal auf und geben Sie auch die Variable Augenzahl3 in der 14. Zeile nach einem weiteren Beistrich an. Weiters nutzen Sie Ihre LÖ für die Aufgabe 4 im 2. Übungsblatt und erweitern Sie hier die 15. Zeile, so dass Sie nach dieser Art eine neue Lösungsmethode erhalten.

```
1 import itertools  
2 # 1. Grundmenge erstellen:  
3 Augenzahlen = {1, 2, 3, 4, 5, 6}  
4 #itertools.product liefert das kartesische Produkt:  
5 omega = list(itertools.product(Augenzahlen, Augenzahlen))  
6 print(omega)  
7 #Beachte, das omega als Menge nicht sortiert ist.  
8 #Staat "set" kann auch "list" geschrieben werden:  
9 #dann wird die Menge schon sortiert.  
10  
11 #2. Anzahl Günstige und Mögliche bestimmen:  
12 günstig = 0  
13 möglich = len(omega)  
14 for Augenzahl1, Augenzahl2, Augenzahl3 in omega:
```

```

14- for Augenzahl1, Augenzahl2, Augenzahl3 in omega:
15-     if Augenzahl1 + Augenzahl2 + Augenzahl3 == 10:
16-         günstig = günstig + 1
17-
18- #3. Ausgabe:
19- print(f"{günstig}von{möglich}")
20- print(f"p={günstig/möglich}")

```

```

[(1, 1, 1), (1, 1, 2), (1, 1, 3), (1, 1, 4), (1, 1, 5), (1, 1, 6),
(1, 2, 1), (1, 2, 2), (1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 2, 6),
(1, 3, 1), (1, 3, 2), (1, 3, 3), (1, 3, 4), (1, 3, 5), (1, 3, 6),
(1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 4, 6),
(1, 5, 1), (1, 5, 2), (1, 5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6),
(1, 6, 1), (1, 6, 2), (1, 6, 3), (1, 6, 4), (1, 6, 5), (1, 6, 6),
(2, 1, 1), (2, 1, 2), (2, 1, 3), (2, 1, 4), (2, 1, 5), (2, 1, 6),
(2, 2, 1), (2, 2, 2), (2, 2, 3), (2, 2, 4), (2, 2, 5), (2, 2, 6),
(2, 3, 1), (2, 3, 2), (2, 3, 3), (2, 3, 4), (2, 3, 5), (2, 3, 6),
(2, 4, 1), (2, 4, 2), (2, 4, 3), (2, 4, 4), (2, 4, 5), (2, 4, 6),
(2, 5, 1), (2, 5, 2), (2, 5, 3), (2, 5, 4), (2, 5, 5), (2, 5, 6),
(2, 6, 1), (2, 6, 2), (2, 6, 3), (2, 6, 4), (2, 6, 5), (2, 6, 6),
(3, 1, 1), (3, 1, 2), (3, 1, 3), (3, 1, 4), (3, 1, 5), (3, 1, 6),
(3, 2, 1), (3, 2, 2), (3, 2, 3), (3, 2, 4), (3, 2, 5), (3, 2, 6),
(3, 3, 1), (3, 3, 2), (3, 3, 3),

```

```

(3, 3, 4), (3, 3, 5), (3, 3, 6), (3, 4, 1), (3, 4, 2), (3, 4, 3),
(3, 4, 4), (3, 4, 5), (3, 4, 6), (3, 5, 1), (3, 5, 2), (3, 5, 3),
(3, 5, 4), (3, 5, 5), (3, 5, 6), (3, 6, 1), (3, 6, 2), (3, 6, 3),
(3, 6, 4), (3, 6, 5), (3, 6, 6), (4, 1, 1), (4, 1, 2), (4, 1, 3),
(4, 1, 4), (4, 1, 5), (4, 1, 6), (4, 2, 1), (4, 2, 2), (4, 2, 3),
(4, 2, 4), (4, 2, 5), (4, 2, 6), (4, 3, 1), (4, 3, 2), (4, 3, 3),
(4, 3, 4), (4, 3, 5), (4, 3, 6), (4, 4, 1), (4, 4, 2), (4, 4, 3),
(4, 4, 4), (4, 4, 5), (4, 4, 6), (4, 5, 1), (4, 5, 2), (4, 5, 3),
(4, 5, 4), (4, 5, 5), (4, 5, 6), (4, 6, 1), (4, 6, 2), (4, 6, 3),
(4, 6, 4), (4, 6, 5), (4, 6, 6), (5, 1, 1), (5, 1, 2), (5, 1, 3),
(5, 1, 4), (5, 1, 5), (5, 1, 6), (5, 2, 1), (5, 2, 2), (5, 2, 3),
(5, 2, 4), (5, 2, 5), (5, 2, 6), (5, 3, 1), (5, 3, 2), (5, 3, 3),
(5, 3, 4), (5, 3, 5), (5, 3, 6), (5, 4,

```

```

, (5, 4, 2), (5, 4, 3), (5, 4, 4), (5, 4, 5), (5, 4, 6), (5, 5, 1),
(5, 5, 2), (5, 5, 3), (5, 5, 4), (5, 5, 5), (5, 5, 6), (5, 6, 1),
(5, 6, 2), (5, 6, 3), (5, 6, 4), (5, 6, 5), (5, 6, 6), (6, 1, 1),
(6, 1, 2), (6, 1, 3), (6, 1, 4), (6, 1, 5), (6, 1, 6), (6, 2, 1),
(6, 2, 2), (6, 2, 3), (6, 2, 4), (6, 2, 5), (6, 2, 6), (6, 3, 1),
(6, 3, 2), (6, 3, 3), (6, 3, 4), (6, 3, 5), (6, 3, 6), (6, 4, 1),
(6, 4, 2), (6, 4, 3), (6, 4, 4), (6, 4, 5), (6, 4, 6), (6, 5, 1),
(6, 5, 2), (6, 5, 3), (6, 5, 4), (6, 5, 5), (6, 5, 6), (6, 6, 1),
(6, 6, 2), (6, 6, 3), (6, 6, 4), (6, 6, 5), (6, 6, 6)]
27von216
p=0.125

```

Für das Blatt 2:

```

1 import itertools
2 Augenzahlen = {1, 2, 3, 4, 5, 6}
3 omega = list(itertools.product(Augenzahlen, repeat =3))
4 print(omega)
5 möglich=len(omega)
6 günstig=0
7 for Augenzahl_Würfel_1, Augenzahl_Würfel_2, Augenzahl_Würfel_3
   in omega:
8     if Augenzahl_Würfel_1<Augenzahl_Würfel_2
       <Augenzahl_Würfel_3:
9         günstig = günstig +1
10 print(f"{günstig}von{möglich}")
11 print(f"p={günstig/möglich}")

```

Grundmenge blende ich nicht ein, da oben schon angegeben.

Anzahl der Möglichkeiten: 20 von 216

p=0,09259259...

4. Ändern Sie Ihr Programm in der 5. Zeile nach dieser Art um. Wann kann diese Schreibweise nützlich sein?

Diese Schreibweise kann nützlich sein, wenn man häufiger Würfel zum Beispiel bei 100 würfen.