

1.

#1: Wir importieren die „Befehlsliste“ von itertools.

#2: Text

#3: Definition der Variable Augenzahl mit Menge

#4: Text

#5: Definition der Ereignismenge mittels Befehl „set“, welcher das (kartesische) Produkt zweier Augenzahlen liefert.

#6: Die Ereignismenge soll aufgelistet werden. (unsortiert wegen „set“)

#7 bis 11: Text

#12: Definition der Variable günstig

#13: Der Befehl len ist mir unbekannt

#14: Für die Variablen Augenzahl 1 und 2 aus Omega gilt:

#15: Wenn die Summe beider Augenzahlen zehn ergibt gilt:

#16: die Günstigen Fälle nehmen um „1“ zu.

#19 und #20: Drucke Befehle.

2.

Befehl set:

```
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, dass  $\Omega$  als Menge nicht sortiert ist.
8 # Statt "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}")
21
22
```

```
{(3, 4), (4, 3), (3, 1), (5, 4), (4, 6), (5, 1), (2, 2), (1, 6),
(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)}
```

3 von 36
p = 0.08333333333333333
❏

Befehl List:

```

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, dass  $\Omega$  als Menge nicht sortiert ist.
8 # Statt "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}")
21
22

```

```

[(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)]
3 von 36
p = 0.08333333333333333

```

Die Menge wird sortiert.

3.

Das war mein Output:

```
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                               Augenzahlen))
7  # Beachte, dass  $\Omega$  als Menge nicht sortiert ist.
8  # Statt "set" kann auch "list" geschrieben werden:
9  # dann wird die Menge schon sortiert.
10 # 2. Anzahl Günstige und Mögliche bestimmen:
11 günstig = 0
12 möglich = len( $\Omega$ )
13 for Augenzahl1, Augenzahl2, Augenzahl3 in  $\Omega$ :
14     if Augenzahl1 + Augenzahl2 == 10:
15         günstig = günstig + 1
16
17 # 3. Ausgabe:
18 print(f"{günstig} von {möglich}")
19 print(f"p = {günstig/möglich}")
```

18 von 216
p = 0.08333333333333333

Hier wollte ich mir die Liste anschauen:

```

1  import itertools
2  # 1. Grundmenge erstellen:
3  Augenzahlen = {1, 2, 3, 4, 5, 6}
4  # itertools.product liefert das kartesische Produkt:
5  Ω = list(itertools.product(Augenzahlen, Augenzahlen,
6                             Augenzahlen))
7  # Beachte, dass Ω als Menge nicht sortiert ist.
8  # Statt "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(Ω)
14 #for Augenzahl1, Augenzahl2, Augenzahl3 in Ω:
15 #    if Augenzahl1<=Augenzahl2<=Augenzahl3:
16 #        günstig = günstig + 2
17
18 # 3. Ausgabe:
19 print(f"{günstig} von {möglich}")
20 print(f"p = {günstig/möglich}")

```

```

[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1),
0 von 216
p = 0.0
]

```

Das wäre schon mal das richtige Ergebnis:

```

1  import itertools
2  # 1. Grundmenge erstellen:
3  Augenzahlen = {1, 2, 3, 4, 5, 6}
4  # itertools.product liefert das kartesische Produkt:
5  Ω = list(itertools.product(Augenzahlen, Augenzahlen,
6                             Augenzahlen))
7  #Beachte, dass Ω als Menge nicht sortiert ist.
8  # Statt "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(Ω)
14 for Augenzahl1, Augenzahl2, Augenzahl3 in Ω:
15     if Augenzahl1 <= Augenzahl2 <= Augenzahl3:
16         günstig = günstig + 1
17
18 # 3. Ausgabe:
19 print(f"{günstig} von {möglich}")
20 print(f"p = {günstig/möglich}")

```

```

[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1),
56 von 216
p = 0.25925925925925924

```

4.

Dadurch kann man recht effizient (schreiben und rechnen) auch die Wahrscheinlichkeit von größeren monoton wachsenden Folgen berechnen.

```
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, repeat=5))
6
7 #Beachte, dass  $\Omega$  als Menge nicht sortiert ist.
8 # Statt "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, Augenzahl4, Augenzahl5
15     in  $\Omega$ :
16     if Augenzahl1<=Augenzahl2<=Augenzahl3<=Augenzahl4
17         <=Augenzahl5:
18         günstig = günstig + 1
19
20 # 3. Ausgabe:
21 print(f"{günstig} von {möglich}")
22 print(f"p = {günstig/möglich}")
```

```
[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1),
252 von 7776
p = 0.032407407407407406
[]
```