



## Schere-Stein-Papier in der Programmiersprache Python

Stand: 25.03.2021

Jahrgangsstufen	Lernbereich 2: Modul 2.6.1
Fach/Fächer	Informationstechnologie
Übergreifende Bildungs- und Erziehungsziele	Technische Bildung
Zeitraumen	90 Minuten
Benötigtes Material	PC mit Internetanschluss, ggf. Python-Entwicklungsumgebung

## Kompetenzerwartungen

Die Schülerinnen und Schüler...

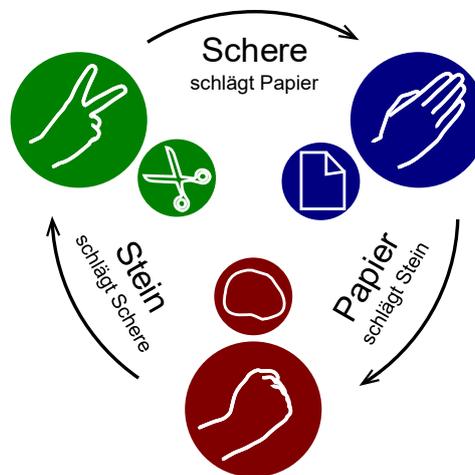
- wählen algorithmische Grundbausteine und Notationsformen, um beschreibbare Abläufe (z. B. bedingte Bewegung eines Roboters) modellhaft und strukturiert darzustellen.
- schreiben auf Basis zuvor erstellter Modelle Quellcode für Programme in einer Entwicklungsumgebung.
- arbeiten mit Variablen, Datentypen, Funktionen und Kommentaren, um Programmcode übersichtlich und besser überprüfbar zu machen.
- überprüfen ihre Programme, um ggf. Fehler zu finden und zu beheben.

## Aufgabe

Es soll das Spiel Schere – Stein – Papier programmiert werden.

### Regeln Schere – Stein – Papier

Beim Spiel Schere – Stein - Papier stehen sich zwei Spieler gegenüber. Sie zählen bis drei (oft mit den Worten „Schnick, Schnack, Schnuck“ oder „Fli, Fla, Flu“) und formen dann ein Zeichen mit ihren Händen. Dabei sind drei Handzeichen möglich: Schere, Stein oder Papier. Jedes Zeichen gewinnt und verliert jeweils gegen eines der anderen Zeichen: Die Schere zerschneidet das Papier (Schere siegt), das Papier wickelt den Stein ein (Papier siegt), der Stein bricht die Schere (Stein siegt). Die Chancen zu gewinnen, sind für jedes Zeichen und jeden Spieler gleich groß.



Quelle: Enzoklop, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=27958795> (Stand: 25.03.2021)

### Testspiel

Bei einem Testspiel zwischen Hannah und Cem wird Hannah per Zufall „Schere“ zugewiesen, Cem bekommt ebenfalls per Zufall „Papier“. Hannah wird als Siegerin ausgegeben, weil Schere Papier schlägt.

### Aufgaben

1. Modelliere das Spiel „Schere, Stein, Papier“ in einer geeigneten Darstellungsform. Dabei müssen folgende Anforderungen erfüllt werden:
  - Abfrage der zwei Spielernamen zu Beginn des Spiels
  - Zufällige Auswahl von Schere, Stein oder Papier für beide Spieler je Spieldurchgang
  - Ausgabe des Siegers je Spieldurchgang in Abhängigkeit der gewählten Zeichen
2. Schreibe anschließend unter Zuhilfenahme deines Modells aus Aufgabe 1 in der Programmiersprache Python ein Programm für das Spiel „Schere, Stein, Papier“.

## Hinweise

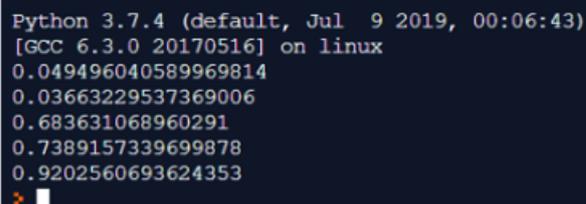
Um das Spiel Schere-Stein-Papier als Programm umsetzen zu können, benötigst du Zufallszahlen. Mit der Bibliothek „random“ können in der Programmiersprache Python Zufallsereignisse hervorgerufen werden. Um damit arbeiten zu können, muss die Bibliothek zuerst importiert werden. Gib dazu am Anfang deines Codes den folgenden Befehl in deine Python-Entwicklungsumgebung ein:

```
import random
```

Es passiert scheinbar nichts – jedoch wurde im Hintergrund die gewünschte Bibliothek mit verschiedenen Befehlen geladen.

Betrachte den folgenden Quellcode:

```
1 import random
2
3 print(random.random())
4 print(random.random())
5 print(random.random())
6 print(random.random())
7 print(random.random())
```

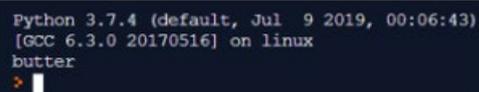


```
Python 3.7.4 (default, Jul 9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
0.049496040589969814
0.03663229537369006
0.683631068960291
0.7389157339699878
0.9202560693624353
>
```

Merke: Der Befehl **print(random.random())** gibt eine Zufallszahl aus, die größer oder gleich 0 und kleiner als 1 ist.

Mit dem Befehl **random.choice** können Elemente aus einer selbst erstellten Liste (=Array) ausgegeben werden:

```
1 import random
2
3 kühlschrank = ["milch", "butter", "eier", "spezi"]
4 print(random.choice(kühlschrank))
5
```



```
Python 3.7.4 (default, Jul 9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
butter
>
```

## Hilfestellung zu Aufgabe 1:

Das Testspiel kann z. B. mit einem Programmablaufplan oder mit Hilfe von Pseudocode modelliert werden.

## Hilfestellung zu Aufgabe 2:

Ein Ausschnitt aus dem Programmcode könnte z. B. so aussehen:



```
import random

spieler1 = input("Bitte Namen von Spieler*in 1 eingeben: ")
spieler2 = input("Bitte Namen von Spieler*in 2 eingeben: ")

zeichen = ["Schere", "Stein", "Papier"]

print (spieler1, "wählt: ")
s1 = (random.choice(zeichen))
print (s1)

print (spieler2, "wählt: ")
s2 = (random.choice(zeichen))
print (s2)
```

## Quellen- und Literaturangaben

Bildquelle zu den Regeln von „Schere – Stein – Papier“: Enzoklop, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=27958795> (Stand: 25.03.2021)

Alle Screenshots wurden vom Autor der Aufgabe (Fabian Pirner) angefertigt.

## Hinweise zum Unterricht

Die Aufgabe greift Kompetenzen und Inhalte aus dem Modul 1.2 auf. Sie eignet sich für eine Verwendung im mittleren Bereich des Moduls 2.6.1. Außerdem sollten die Schülerinnen und Schüler in der Lage sein, Anweisungen, Sequenzen und Wiederholungen mit Hilfe von Modellen anschaulich darzustellen.

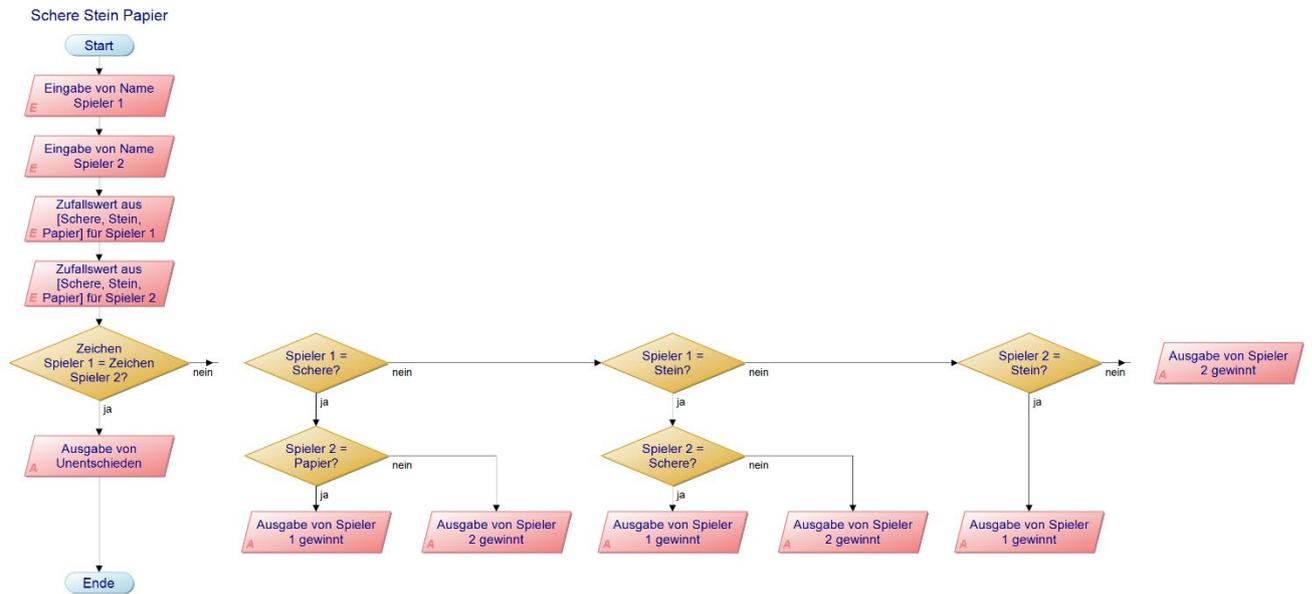
## Beispiele für mögliche Lösungen

Aufgabe 1:

Darstellung des Testspiels mit Hilfe von Pseudocode

```
Eingabe von Spielername 1
Eingabe von Spielername 2
Weise Spieler 1 Zufallswert aus [Schere, Stein, Papier] zu
Weise Spieler 2 Zufallswert aus [Schere, Stein, Papier] zu
Wenn Zeichen Spieler 1 = Zeichen Spieler 2
    Ausgabe von „Unentschieden“
Sonst
    Wenn Spieler 1 = Schere
        Wenn Spieler 2 = Papier
            Ausgabe von „Spieler 1 gewinnt“
        Sonst
            Ausgabe von „Spieler 2 gewinnt“
    Wenn Spieler 1 = Stein
        Wenn Spieler 2 = Schere
            Ausgabe von „Spieler 1 gewinnt“
        Sonst
            Ausgabe von „Spieler 2 gewinnt“
    Sonst
Wenn Spieler 2 = Stein
    Ausgabe von „Spieler 1 gewinnt“
    Sonst
    Ausgabe von „Spieler 2 gewinnt“
```

## Darstellung des Testspiels mit Hilfe eines Programmablaufplans





## Aufgabe 2:

```
import random

spieler1 = input("Bitte Namen von Spieler*in 1 eingeben: ")
spieler2 = input("Bitte Namen von Spieler*in 2 eingeben: ")

zeichen = ["Schere", "Stein", "Papier"]

print (spieler1, "wählt: ")
s1 = (random.choice(zeichen))
print (s1)

print (spieler2, "wählt: ")
s2 = (random.choice(zeichen))
print (s2)

if s1 == s2:
    print ("Unentschieden")

else:
    if s1 == "Schere":
        if s2 == "Papier":
            print (spieler1 + " gewinnt")
        else:
            print (spieler2 + " gewinnt")

    if s1 == "Stein":
        if s2 == "Schere":
            print (spieler1 + " gewinnt")
        else:
            print (spieler2 + " gewinnt")

    else:
        if s2 == "Stein":
            print (spieler1 + " gewinnt")
        else:
            print (spieler2 + " gewinnt")
```

```
Bitte Namen von Spieler*in 1 eingeben: Hannah
Bitte Namen von Spieler*in 2 eingeben: Cem
Hannah wählt:
Stein
Cem wählt:
Stein
Unentschieden
> 
```



```
Bitte Namen von Spieler*in 1 eingeben: Hannah
Bitte Namen von Spieler*in 2 eingeben: Cem
Hannah wählt:
Schere
Cem wählt:
Papier
Hannah gewinnt!
> 
```

### Anregung zum weiteren Lernen

Das fertige Spiel kann z. B. erweitert werden, so dass es erst als gewonnen gilt, wenn ein Spieler seinen Gegner dreimal besiegen konnte.