

JavaScript – Schere, Stein, Papier

Zwei Personen suchen sich jeweils ein Symbol aus und nennen es gleichzeitig. Haben beide Spieler das gleiche Symbol, ist die Runde unentschieden. Ansonsten gelten folgende Regeln:

- Schere schneidet Papier und gewinnt
- Papier wickelt Stein ein und gewinnt
- Stein schleift die Schere und gewinnt

Weitere Überlegungen:

- Anzahl der Runden ist nicht festgelegt
- Gewinnt ein Spieler dreimal, aber nicht unbedingt in Folge, dann hat er gewonnen und das Spiel ist aus.

Schritt 1: Festlegen der Vorgangsweise und der Variablen:

Überlegung	Variable
Der Computer denkt sich zu Beginn jeder Runde ein Zahl von 1 bis 3 aus, die in ein Symbol gewandelt wird.	var symbolComputer;
Der Spieler gibt eine Zahl von 1 bis 3 in, die ebenfalls gewandelt wird	var symbolSpieler;
Dann wird verglichen, ob wer gewonnen hat	var anzahlRunden = 0; var gewinneComputer = 0; var gewinneSpieler = 0;
Hat einer dreimal gewonnen, wird das Spiel beendet	var gewinnBedingung = 3;
Es wird ausgegeben, wer wie oft gewonnen hat.	

Am Beginn müssen die **Variablen deklariert werden**. Man könnte alle in eine Zeile schreiben, mit Komma getrennt. Aber zwecks Übersicht kommen sie untereinander.

Bei den Variablen ist es sinnvoll, möglichst sprechende Bezeichnungen zu verwenden. Wenn im Verlauf des Programmierens noch weitere Variablen dazukommen, ist das überhaupt kein Problem. Diese werden einfach an Ort und Stelle deklariert, sie müssen nicht hier am Anfang stehen. Je länger das Programm ist, desto mehr verteilt sich das.

Schritt 2) Schleife mit „do-while“:

Nun folgt der Beginn der „do-while“-Schleife.

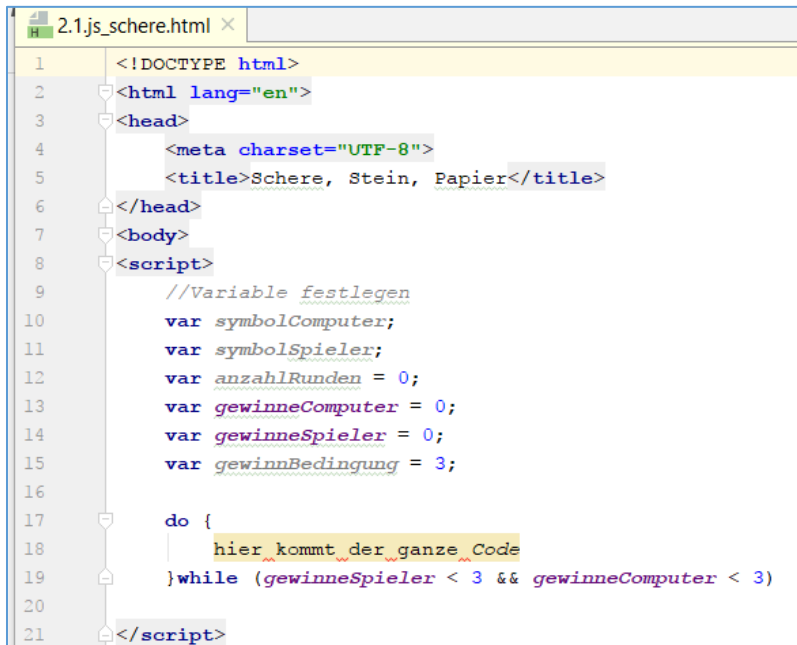
Wenn man das Programm jedes Mal neu startet, wäre das sehr umständlich und außerdem würde der Computer jedes Mal eine neue Zahl errechnen. Es gibt bestimmte Teile im Programm, die nur einmal ausgeführt werden sollen, und Teile die immer wieder ausgeführt werden müssen.

In diesem Beispiel muss folgendes immer wiederholt werden: der Spieler gibt seine Zahl ein, diese wird verglichen und ein Ergebnis ausgegeben.

Daher braucht es hier die „do – while“ – Schleife: es wird solange wiederholt, solange eine angegebene Bedingung erfüllt ist. Ist diese falsch, endet die Schleife sofort. Die Schleife wird auf jeden Fall einmal durchlaufen.

```
do{ frage nach einer Zahl, vergleiche sie und gib das Ergebnis aus  
    }while ( gerateneZahl != ausgedachteZahl );
```

Schritt 3)Erstelle ein neues HTML-Dokument in HTML5:



```
2.1.js_schere.html x  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4 <meta charset="UTF-8">  
5 <title>Schere, Stein, Papier</title>  
6 </head>  
7 <body>  
8 <script>  
9 //Variable festlegen  
10 var symbolComputer;  
11 var symbolSpieler;  
12 var anzahlRunden = 0;  
13 var gewinneComputer = 0;  
14 var gewinneSpieler = 0;  
15 var gewinnBedingung = 3;  
16  
17 do {  
18     hier kommt der ganze Code  
19 }while (gewinneSpieler < 3 && gewinneComputer < 3)  
20  
21 </script>
```

Code:

```
<script>  
    //Variable festlegen  
    var symbolComputer;  
    var symbolSpieler;  
    var anzahlRunden = 0;  
    var gewinneComputer = 0;  
    var gewinneSpieler = 0;  
    var gewinnBedingung = 3;  
  
    do {  
        hier kommt der ganze Code  
    }while (gewinneSpieler < 3 && gewinneComputer < 3)  
</script>
```

Die Variablen „gewinneComputer“ und „gewinneSpieler“ starten bei „Null“ und dienen dazu, mitzuzählen, wer nämlich wie oft gewonnen hat. Das wird am Ende dann ausgegeben. Bei jedem Sieg wird diese Variable um „Eins“ erhöht (siehe Schritt 5).

Auch die Anzahl der Runden wird hier festgelegt und mit „0“ gestartet. Da eine Zählvariable vor der ersten Verwendung bereits einen Wert haben muss, setze man sie hiermit auf 0. Mit dem Hochzählen später will man am Ende erfahren, wie viele Runden insgesamt, bis der erste Gewinner feststeht nötig waren.

Schritt 4)Die Zufallszahl festlegen lassen – von 1 bis 3

```
17 do {
18     //Der Computer denkt sich eine Zahl aus, zu Beginn jeder Runde
19     ausgedachteZahl = Math.random() * 3;
20     ausgedachteZahl = Math.round(ausgedachteZahl + 0.5);
21
22     //Zuordnung für den Computer
23     if(ausgedachteZahl == 1) {
24         symbolComputer = "Schere"
25     };
26     if(ausgedachteZahl == 2) {
27         symbolComputer = "Stein"
28     };
29     if(ausgedachteZahl == 3) {
30         symbolComputer = "Papier"
31     };

```

Man lässt mit „Math.random“ eine Zufallszahl erzeugen, diese liegt immer zwischen 0 und 1. Das reicht nicht, daher multipliziert man es mit der gewünschten Obergrenze. Das Ergebnis hat aber viele Kommastellen.

Daher wird das Ergebnis in der nächsten Zeile mit „Math.round“ gerundet und das Ergebnis wird der Variablen zugewiesen.

Dabei wird aber die ausgedachte Zahl mit 0,5 addiert. Denn durch das kaufmännische Runden könnte man eine „Null“ erhalten, wenn z.B. 0,1 mit 3 multipliziert wird und somit 0,3 ergibt. Das Runden würde somit ein abgerundetes Ergebnis von „0“ ergeben. Null soll aber nicht vorkommen. Daher der kleine Kunstgriff: man sorgt dafür, dass die Zahl immer größer als 0,5 ist, indem man 0,5 addiert. Das Ergebnis liegt nach dem Runden immer im Bereich von 1 bis 3.

Danach wird mit einer „if“ Abfrage die ausgedachte Zahl des Computers als Text dem Symbol vom Computer übergeben.

Schritt 4)Die Zufallszahl einer Möglichkeit zuordnen

Der Spieler wählt anhand der Eingabe der Zahlen 1 bis 3.

Diese drei Zahlen 1, 2 und 3 werden nun einer der drei Möglichkeiten zugeordnet.

Diese Wahl wird als richtiger Text in eine Variable gespeichert.

Die drei Zahlen werden nun zugeordnet: 1 = Schere, 2 = Stein und 3 = Papier.

Das wird dem Spieler vorher noch bekanntgegeben:

```
27 //Der Spieler muss nun eingeben, was er wählt
28 wahlSpieler = prompt("Gib deine Wahl ein", "Schere ist 1, Stein ist 2, Papier ist 3");
29 //Zuordnung für Spieler
30 if(wahlSpieler == 1) {symbolSpieler = "Schere"};
31 if(wahlSpieler == 2) {symbolSpieler = "Stein"};
32 if(wahlSpieler == 3) {symbolSpieler = "Papier"};
33

```

Schritt 5) Vergleich, wer die Runde gewonnen hat

Haben beide, der Computer und der Spieler das gleiche Symbol, ist die Runde unentschieden.

```
//Es wird verglichen, wer diese Runde gewonnen hat
//Hier zuerst, ob ein unentschieden vorliegt
if(symbolComputer == symbolSpieler) {alert("unentschieden"); }
```

Beachte: der Strichpunkt ist innerhalb, gleich nach dem alert. Das „if“ selbst braucht zum Abschluss kein Semikolon.

Nun muss man jede verschiedene Art mit einer IF-Entscheidung anschreiben, die zu einem Gewinn führt. Streng nach der Vorschrift:

- Schere gewinnt gegen Papier
- Papier gewinnt gegen Stein
- Stein gewinnt gegen Schere

Dieses nun für alle Möglichkeiten. Sieht nach viel aus, aber man sollte einfach „copy & paste“ nutzen und dann einige wenige Elemente umschreiben.

Dazwischen wird hochgezählt, wie oft der Computer bzw. der Spieler gewonnen hat. Der Start erfolgte ja bei „null“, als die Variable ganz zu Beginn registriert wurde. Diese Ausgabe erfolgt dann am Ende des Programms.

```
//Nun, jeweils wer gewinnt
//Schere gewinnt gegen Papier, Papier gegen Stein
//Stein gewinnt gegen Schere
if(symbolComputer == "Schere" && symbolSpieler == "Stein") {
    gewinneSpieler++;
    alert("Du gewinnst gegen Schere");
}
```

Code:

```
if(symbolComputer == "Schere" && symbolSpieler == "Stein") {
    gewinneSpieler++;
    alert("Du gewinnst gegen Schere");
}
```

Nun gibt es insgesamt 6 Möglichkeiten, da die „Unentschieden“ bereits verarbeitet sind, siehe oben. Man sollte jedoch keine Kombination auslassen. Da sonst das Spiel nicht richtig funktioniert.

Hier wurde so vorgegangen, dass zuerst alle Möglichkeiten bei „symbolComputer“ durchgearbeitet wurde, Schere 2 xmal dann Stein 2 mal und dann noch Papier 2 mal.

Jeweils mit der richtigen Lösung, wer denn nun gewinnt.

```

}
if(symbolComputer == "Schere" && symbolSpieler == "Papier") {
    gewinneComputer++;
    alert("Computer gewinnt mit Schere");
}
if(symbolComputer == "Stein" && symbolSpieler == "Schere") {
    gewinneComputer++;
    alert("Computer gewinnt mit Stein");
}
if(symbolComputer == "Stein" && symbolSpieler == "Papier") {
    gewinneSpieler++;
    alert("Du gewinnst gegen Stein");
}
if(symbolComputer == "Papier" && symbolSpieler == "Stein") {
    gewinneComputer++;
    alert("Computer gewinnt mit Papier");
}
if(symbolComputer == "Papier" && symbolSpieler == "Schere") {
    gewinneSpieler++;
    alert("Du gewinnst gegen Papier");
}
}

```

Schritt 6)Anzahl der Runden mitzählen

Da man am Ende wissen will, wie viele Runden tatsächlich gespielt wurden, wird jedes Mal die Variable „anzahlRunden“ mit „+“ hochgezählt.

Die Registrierung der Variablen erfolgte ganz zu Beginn.

Schritt 7)Das Spiel beenden, wenn wer dreimal gewonnen hat

Im „while“ wird überprüft, ob beide Spieler immer weniger als 3 Siege haben. Solange dies der Fall ist, beginnt oben bei dem „do“ eine neue Runde. Ansonsten geht es hinter dem „while“ weiter.

Hier kommt nämlich dann die Meldung, wer wie oft gewonnen hat.

```

64     }while (gewinneSpieler < 3 && gewinneComputer < 3)
65
66     //Hier wird ausgegeben, wer wie oft gewonnen hat
67     if(gewinneSpieler >= 3) {
68         alert("Du hast das Spiel gewonnen. Gratuliere!");
69     }else{
70         alert("Der Computer war besser und hat das Spiel gewonnen!");
71     }

```

Wer der Spieler 3 oder mehr Siege verbuchen kann, hat er gewonnen. Das wird im „if“ festgestellt und im „alert“ verkündet.

Schritt 8) Ausgabe, wer wie oft gewonnen hat

```
alert(gewinneSpieler + " zu " + gewinneComputer);
```

Ergebnis:

