

# Documentatie

## Steen, Papier, Schaar

Julian Roost

## Contents

Inleiding.....	3
Javascript.....	4
HTML .....	7
CSS .....	8

## Inleiding

In dit document staat beschreven hij je het spel steen, papier, schaar moet maken met behulp van javascript.

## Javascript

```
var user = document.querySelector('.userchoice'),
    computer = document.querySelector('.computerchoice'),
    startbtn = document.querySelector('.startbtn'),
    result = document.querySelector('.result'),
```

De variable user slaat de keuze van de gebruiker op. Ik gebruik hiervoor de document.querySelector methode in ik sla hem in de class userchoice op. Hetzelfde doe ik met de computer keuze bij de variable computer. De variable startbtn gebruik ik om het spel te starten en om te kijken of hij nog bezig is. De result variable gebruik om te laten zien of je verloren, gelijk of gewonnen hebt.

```
userchoice,
winStates = { Rock: 'Scissors', Paper: 'Rock', Scissors: 'Paper'};
```

We moeten weten wat, wat verslaat in het spel. Daarom maak ik hier een object aan. Ik maak een key aan en daar stop ik een value in die die beter is dan de key. Dus Rock verslaat Scissors

Nu heb ik alle variabelen die ik wil gaan gebruiken.

```
startbtn.addEventListener('click', function(){
```

Hier voeg ik een EventListener toe aan de start button ik refereer weer terug naar de variable startbtn.

```
this.disabled = true;
```

Ik kan "this" hier gebruiken om te refereren naar startbtn omdat ik in de startbtn variable aan het werk ben.

```
result.innerHTML = '';
```

Hier zorg ik ervoor dat de HTML van de result division leeg is. Hier komt de tekst in wie gewonnen heeft en wat het resultaat is.

```
user.className = "userchoice Rock count-in";
```

Hier zorg ik ervoor dat de animation begint. De rock is hier een placeholder en de count-in gebruik ik om de animation te starten.

```
computer.className = "computerchoice Rock count-in"
```

Hier doe ik exact hetzelfde alleen dan voor de computer.

```
document.addEventListener('animationend', function(){
```

Als de count-in animation klaar is, begint dit event.

```
if (!userchoice) { userchoice = getRandomChoice(); }
```

Hier kijkt die of de userchoice wel een value heeft, als hij dat niet heeft komt er een random keuze.

```
startbtn.disabled = false;
```

Hier zorg ik ervoor dat de startbtn weer aangeklikt kan worden.

```
var computerchoice = getRandomChoice();
```

De cuntion getRandomChoice voor de computerchoice

```
result.innerHTML = getWinner(userchoice, computerchoice);
```

Hier laat ik in mijn HTML zien wie gewonnen heeft

```
function getRandomChoice(){
    return Object.keys(winStates)[Math.floor(Math.random() * 3)];
}
```

De random keuze uit het winStates object. Hier gebruik ik Math.floor om te voorkomen dat ik een decimaal terug krijg.

```
user.className = "userchoice " + userchoice;
computer.className = "computerchoice " + computerchoice;
```

Omdat ik de classes precies hetzelfde genoemd hebt als de WinStates kan ik deze direct gebruiken bij de user en computer elementen.

```
function getWinner(userchoice, computerchoice){
```

Ik maak een nieuwe functie aan zodat ik kan zien wie heeft gewonnen. Ik geef deze 2 values mee de userchoice en de computerchoice. Dit zijn de strings met Rock, Paper of Scissors.

```
if (userchoice === computerchoice){
    return "Gelijkspel!";
```

Hier geef ik aan wanneer het spel gelijkspel is. Dus als userchoice gelijk is aan computerchoice dan is het gelijkspel. Ik return een string om de division te updaten.

```
} else if ( userchoice === winStates[computerchoice] ){
    return "Computer Wint! " + computerchoice + " verslaat " +
userchoice;
```

Hier geef ik aan wanneer de computer heeft gewonnen. Dus als de computerchoice beter is dan de userchoice. Ik kijk naar de value in het winStates object dus als de key die gelijk is aan de computer keuze dan return ik de value in de winStates object. Als de userchoice Rock is en de userchoice is paper dan kijkt hij in het object naar de value die gelijk is aan de paper key en dan return hij dus de value van Rock

```
}
    return "Jij wint! " + userchoice + " verslaat " + computerchoice;
```

Hier geef ik aan wanneer de user heeft gewonnen. Dus als de user choice beter is dan de computer choice. Als het niet gelijkspel is of de computer heeft niet gewonnen dan wint de user dus kan ik deze string returnen.

```
document.addEventListener('keydown', function(event){
```

Dit is voor de user input, ik ga hier ervoor zorgen dat de user een key kan indrukken op zijn keyboard en dat hij zo zijn keuze maakt. Ik geef een function mee met de event paramater zodat ik toegang heb tot alle event informatie die gebruikt wordt zodra de keydown event start.

```
if(event.keyCode === 82){
    userchoice = 'Rock';
```

keycode nummer 82 is "R" op het toetsenbord dus zodra de user R indrukt wordt de userchoice variable "rock"

```
} else if (event.keyCode === 80){
    userchoice = 'Paper';
```

keycode nummer 80 is "P" op het toetsenbord dus zodra de user P indrukt wordt de userchoice variable "paper"

```
} else{  
    userchoice = 'Scissors';  
}
```

Zodra de user niks indrukt wordt zijn keuze automatisch “scissors”.

## HTML

```
<div class="container">
  <h1>Rock, Paper, Siccors</h1>
  <h3>Instructie:</h3>
  <p>Om te spelen druk: "R", "S" of "P" op je toetsenbord.
</p>

  <div class="player">
    <div class="userchoice Rock"></div>
    <p>Speler 1</p>
  </div>

  <div class="player">
    <div class="computerchoice Rock"></div>
    <p>Computer</p>
  </div>

  <div class="result"></div>

  <button class="startbtn">Speel</button>
</div>
```

## CSS

```
body{
  font-family: 'Roboto', sans-serif;
  background-color: #1e1e1e;
  color: white;
}

.container{
  width: 650px;
  margin: 0 auto;
  text-align: center;
  padding: 25px;
  background-color: peru;
  box-shadow: 0 0 34px 8px rgba(0, 0, 0, 0.4);
}

.player{
  width: 150px;
  height: 150px;
  margin: 100px 50px;
  display: inline-block;
}

.player p{
  font-weight: 700;
}

.player div{
  width: 100%;
  height: 100%;
}

.Rock{
  background-image: url('rock.png');
}

.Paper{
  background-image: url('paper.png')
}

.Scissors{
  background-image: url('scissors.png')
}
```



```
.Rock, .Paper, .Scissors{
  background-size: contain;
  background-repeat: no-repeat;
}

.startbtn{
  background-color: white;
  color: black;
  border: none;
  padding: 10px 100px;
}

.startbtn:disabled{
  background-color: gray;
}

.result{
  margin-bottom: 25px;
  font-weight: 600;
  height: 50px;
  font-size: 2em;
}

@keyframes count-in{
  0%{
    transform: translateY(0)
  }
  60%{
    transform: translateY(-100px)
  }
  100%{
    transform: translateY(0)
  }
}

.count-in{
  animation: .7s count-in 3;
}
```