JS - Herhalingslussen

**In een herhalings lus kan je een statement of een groep statements een aantal keren herhalen.**

for

Om een bepaalde actie verschillende malen uit te voeren gebruiken we een herhalingslus (in het Engels: een "loop"). Dat ziet er als volgt uit:

for (teller = 1; teller <= 1000; teller++) {

console.log("Ik zal braaf zijn");

}

De for-opdracht bevat drie elementen tussen de haakjes. De initialisatie van de tellervariabele, de logische test voor uitvoering van de lus, en een opdracht voor het bijwerken van de tellervariabele. Dan volgt het codeblok dat herhaaldelijk moet worden uitgevoerd.

Voordat de lus wordt uitgevoerd wordt de teller ingesteld op zijn beginwaarde, teller krijgt de waarde 1. Vervolgens wordt gekeken of aan de logische test wordt voldaan, namelijk is teller kleiner dan of gelijk aan 1000? Als dat effectief zo is wordt de codeblok van de lus uitgevoerd. Tenslotte wordt de teller-variabele bijgewerkt, teller++ is een verkorte schrijfwijze voor teller = teller + 1. Na de eerste uitvoering of "iteratie" van de lus krijgt teller de waarde 2. Dat gaat zo door totdat de logische test faalt.

We moeten er wel voor zorgen dat we niet in een eindeloze lus terecht komen. Als tel de beginwaarde 1 krijgt en de voorwaarde luidt: tel > 0, terwijl we na elke iteratie de teller ophogen, dan blijven we altijd aan de voorwaarde voldoen.

In ons voorbeeld is het een lus die een vooraf bekend aantal keren wordt uitgevoerd. Het is ook mogelijk om lussen een variabel aantal keren te laten uitvoeren. Dat kan door in de voorwaarde een variabele te gebruiken, bijvoorbeeld: tel <= maxaantal (waarbij die variabele natuurlijk wel eerst een waarde moet hebben gekregen).

Je kan een for lus voortijdig afbreken met het keyword break:

// 3 kansen om 6 te gooien

for (i=1; i <= 3; i++) {

var worp = Math.round(Math.random() \* 10);

if (worp == 6) {

text = "Bingo! Worp " + i + " is een 6.";

break;

} else {

text = "Goed geprobeerd, je hebt " + i + " keren gegooid, maar geen 6.";

}

}

console.log(text);

while

De while-opdracht bevat maar één element tussen de haakjes (te weten: de logische test); we moeten zelf een beginwaarde aan de teller toekennen en zelf de teller vermeerderen.

We kunnen "voortijdig" een lus beëindigen met behulp van de break-opdracht. Dit gebeurt meestal in combinatie met één van de logische tests

teller = 1

while (telller <= 1000) {

console.log("Ik zal braaf zijn");

teller++

}

Gooien tot je er bij naarvalt:

// gooien tot je erbij neervalt

var failed = true;

i = 1;

while (failed) {

var worp = Math.round(Math.random() \* 10);

if (worp == 6) {

text = "Bingo! Worp " + i + " is een 6.";

failed = false;

}

i++;

}

console.log(text);

Goede gewoonten

1. Het inspringen van de opdrachten is niet verplicht, maar komt de leesbaarheid wel ten goede.
2. Alles op een regel kan, als we de opdrachten van elkaar scheiden door middel van een puntkomma.
3. Zorg ervoor dat je niet in een eindeloze lus terecht komt.