

Les 3: For-loop, Continue - en Break statement

Tijdens les drie gaan we aan de slag met for-loops en leren we de *continue* - en *break* statement gebruiken.

For-loop

Tijdens les 2 hebben we kennis gemaakt met de for-loop. Hierbij hebben we de syntax van de for-loop behandeld en hebben we gezien dat de for-loop uit drie onderdelen bestaat, namelijk:

- Een variabele met beginwaarde
- Een vergelijking, deze wordt ook wel eens een conditie genoemd
- Een toenamefactor.

Let op! De toenamefactor kan ook negatief zijn. Dan is er sprake van een afname!

Syntax van for-loop

De for-loop wordt aangeduid door een *for* gevolgd door ronde haakjes en een accolade **[B1]** die de for-loop opent. De syntax ziet er dan ook zo uit:

```
for(variabele met beginwaarde; conditie; toenamefactor){  
    // hier komt de code die de for-loop moet uitvoeren.  
}
```

Laten we de for-loop uitwerken aan de hand van een voorbeeld.

Stel dat we een for-loop hebben met een variabele *i*. Deze variabele heeft een beginwaarde van 0. De for-loop gaat 10 keer itereren **[B2]** en print bij elke iteratie de waarde van variabele *i*. Na elke iteratie neemt de variabele *i* met 1 toe:

```
for(var i = 0; i < 10; i++){  
    document.write("i is "+i);
```

```
}
```

De bovenstaande for-loop eindigt wanneer variabele *i* gelijk is aan 9. Want met 0 erbij, heeft de for-loop dan in totaal 10 keer geloopt. **Let op!** De conditie is $i < 10$ en niet $i \leq 10$, dus *i* kan in de bovenstaande for-loop nooit 10 worden!

Continue statement

Soms komt het voor dat je binnen een for-loop een if-statement gebruikt. Deze kan, afhankelijk van de conditie, een bepaalde stap uitvoeren. Hier kan een continue statement wel handig zijn. Deze zorgt er namelijk voor dat de for-loop door blijft gaan tot de aangegeven conditie van de for-loop gelijk is aan *false*.

We leggen de continue statement uit aan de hand van het volgende voorbeeld:

Stel dat we een for-loop hebben met een variabele *x*. Deze variabele heeft een beginwaarde van 1 en blijft itereren tot en met 10. Bij elke iteratie neemt *x* met 1 toe. Deze for-loop bevat een if-statement. Deze checkt of *x* gelijk is aan de waarde van de gebruikers input. Indien deze gelijk zijn aan elkaar, wordt een *continue* statement gebruikt om de waarde over te slaan. In alle andere gevallen is de waarde van *x* ongelijk aan *userInput* en wordt deze waarde afgedrukt. Bijvoorbeeld:

```
/*
Vraag de gebruiker om een willekeurig getal op te noemen en vertaal deze van data type string naar number.
Sla dit getal op in een variabele genaamd userInput.
*/
var userInput = parseInt(prompt("Noem een willekeurig nummer tussen de 1 en de 10?"));

for(var x = 1; x <=10; x++){

    /* Vergelijk of x gelijk is aan het ingevoerde nummer van de gebruiker.
    □ De waarde van x moet "overgeslagen" worden als deze gelijk is aan de invoer van de gebruiker.
    In alle andere gevallen moet i geprint worden.
    */
    □ if(x == userInput){
        □ continue;
    } else {
        □ document.write("De waarde van x is "+x);
    }
}
```

Break statement

Soms wil je tijdens een for-loop uit de loop stappen door deze te stoppen. Dit kun je doen met behulp van de *break* statement.

Na de *break* statement houdt de loop dus op.

Stel dat we een for-loop hebben met een variabele *y*. Deze variabele heeft een beginwaarde van 0 en een eindwaarde van 10. Deze for-loop bevat een if-statement. Deze checkt of *y* gelijk is aan de waarde van de gebruikers input. Indien deze gelijk zijn aan elkaar, wordt een *break* statement gebruikt om de for-loop te stoppen. In alle andere gevallen print de for-loop de waarde van variabele *y*. Bijvoorbeeld:

```
var leeftijd = parseInt(prompt("Wat is je leeftijd"));

for(var y = 1; y <= 10; y++){
  // De for-loop stopt met lopen zodra y gelijk is aan de ingevoerde leeftijd. Deze "stop" wordt veroorzaakt door de break statement.
  if(y == leeftijd){
    break;
  }else{
    document.write("Je leeftijd is niet gelijk aan " + y);
  }
}
```

Samenvatting continue - en break statement

1. De continue statement zorgt ervoor dat een gegeven wordt overgeslagen tijdens het lopen.
2. De break statement zorgt ervoor dat een for-loop stopt met itereren.

Les - en huiswerk opdrachten

Les - en huiswerk opdracht 1: Schrijf een for-loop die bij 5 begint en tot en met 20 telt. Bij elke iteratie moet de for-loop het cijfer printen op een nieuwe regel.

Les - en huiswerk opdracht 2: Schrijf een for-loop die bij 10 begint en 20 keer loopt. Deze for-loop moet alleen even getallen printen. Gebruik continue om de oneven getallen over te slaan.

Les - en huiswerk opdracht 3: Als de onderstaande stappen uitgewerkt zijn, heb je een stukje code geschreven die om user-input vraagt en deze als nummer (int) opslaat in een variabele genaamd *leeftijd*. Deze variabele wordt gebruikt in een for-loop. De for-loop moet stoppen als het getal van de iteratie gelijk is aan de leeftijd die is ingevoerd door de gebruiker.

- Schrijf een script die vraagt om een user input (gebruik *prompt()*) en deze waarde opslaat in variabele *leeftijd*.

- Bepaal met behulp van de *typeof()* functie wat het datatype van variabele *leeftijd* is. Print de datatype in de console van je browser met behulp van *console.log()*.

- In de vorige stap heb je met behulp van de *typeof()* functie de datatype van variabele *leeftijd* achterhaald. Schrijf deze als een comment achter de *console.log()* functie die je hebt geschreven in stap 2.

- Zet *parseInt()* om de *prompt()* functie uit stap 1. Check met behulp van de *typeof()* functie wat de datatype van variabele *leeftijd* nu is. Comment deze datatype aan het einde van de regel waar ook je variabele *leeftijd* staat.

- Laat de code van de voorgaande stappen staan. Druk 2x op enter om een ruimte te creëren. We gaan nu een for-loop schrijven die bij 0 begint met tellen en tot 30 telt.

- Schrijf, binnen de for-loop uit stap 5, een if-statement die checkt of het getal van de huidige iteratie gelijk is aan de waarde die is opgeslagen in de *leeftijd* variabele. Als deze conditie *true* is, print je met behulp van *document.write("De ingevoerde leeftijd is gelijk aan" + i)* de loop ma deze print stoppen met tellen. Dit doe je met behulp van een *break*-statement.

Bronnen

[B1] Accolade is een gekruist haakje { en }. Onder accolade openen verstaan we het haakje { en andersom, met accolade sluiten bedoelen we }.

[B2] Itereren is een ander woord voor lopen.