

Les 1: Herhaling JavaScript I

Console

Tijdens de uitleg van JavaScript (JS) maakt de docent gebruik van de console van Google Chrome.

De console kun je openen door te klikken op de drie puntjes naast de adresbalk, deze opent een lijst van mogelijkheden. Click op "Meer hulpprogramma's" -> "Hulpprogramma's voor ontwikkelaars", merk op dat de console - afhankelijk van je instellingen - aan de rechterkant van je browser verschijnt.

De console kan ook geopend worden met de keyboard shortcut: `CTRL + SHIFT + I` of `fn + F12`.

Variabelen

Een variabele verwijst naar een plekje in het computergeheugen. Met behulp van de variabele kun je van een bepaalde waarde uit het geheugen tonen/gebruiken en - als het nodig is - veranderen.

Om verwarring te voorkomen, is het belangrijk dat de variabele naam uniek is. Dit betekent dat de variabelenaam niet mag bestaan.

Waar moet een variabele naam aan voldoen?

1. Het belangrijkste is dat de naam van een variabele uniek moet zijn.
2. De naam van de variabele moet beginnen met een underscore(_), kleine- of hoofdletter.
3. Het eerstvolgende woord in de naam van de variabele mag beginnen met een letter, _ of cijfer.

Variabele namen zijn case sensitive **[B1]**, dit betekent dat er onderscheid wordt gemaakt tussen X en x.

Hoe kan ik een variabele aanmaken en gebruiken?

Het is mogelijk om een variabele eerst aan te maken en dan een waarde toe te kennen.

Bijvoorbeeld:

```
var nummer;  
nummer = 42;
```

Als je weet welke (begin)waarde een variabele heeft, kun je deze gelijk toekennen aan een variabele. Bijvoorbeeld:

```
var nummer = 42;
```

Code

JavaScript code kun je schrijven tussen een script tag, nog netter zou zijn om het in een external JavaScript file te schrijven.

Beide manieren worden hieronder gedemonstreerd:

<script>

De <script> tag wordt gebruikt om een client-side script (JavaScript) te definiëren en kan een script bevatten, of verwijzen naar een script.

Het volgende voorbeeld laat een HTML skeleton zien, met een script tag die JavaScript bevat:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Mijn prachtige titel</title>  
    <script type="text/javascript">  
      var stelling = "Minecraft is leuker dan Fortnite";  
      document.write(stelling);  
    </script>  
  </head>  
  <body>  
  
  </body>  
</html>
```

Het is ook mogelijk om JavaScript in een extern bestand te schrijven. Met behulp van de <script> tag kun je vervolgens verwijzen naar dit bestand. De bovenstaande code zou ook als volgt geschreven kunnen worden:

HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Mijn prachtige titel</title>
    <script src="gameStelling.js"></script>
  </head>
  <body>
    </body>
</html>
```

JavaScript (naam van het bestand is gameStelling.js):

```
var stelling = "Minecraft is leuker dan Fortnite";
document.write(stelling);
```

Je kunt JavaScript dus in je HTML script opnemen, of in een apart JavaScript bestand beschrijven. Afhankelijk van de manier zou je de <script>-tag moeten veranderen. Let hierbij op het volgende verschil:

```
<script type="text/javascript">Hier je script</script>
```

of

```
<script src="JavaScriptBestandNaam.js"></script>
```

Operators

Net als in de meeste programmeertalen, heeft ook JavaScript verschillende categorieën voor operators **[B2]**, namelijk:

- Vergelijkende operatoren
- Rekenkundige operatoren
- Logische operatoren

Vergelijkende operatoren

Vergelijkende operatoren worden gebruikt om waarden met elkaar te vergelijken en returnen een True/False. Deze waarden zijn boolean waarden. Het onderstaande tabel - Tabel 1 - beschrijft vergelijkende operatoren in JavaScript:

Operator	Beschrijving
==	Gelijk aan
!=	Niet gelijk aan
<	Kleiner dan
<=	Kleiner dan of gelijk aan
>	Groter dan
>=	Groter dan of gelijk aan

Tabel 1: Overzicht van vergelijkende operatoren

Rekenkundige operatoren

De rekenkundige operatoren kunnen gebruikt worden voor berekeningen. Voorbeelden van deze berekeningen zijn bijvoorbeeld optellen, aftrekken, vermenigvuldigen, delen, rest berekeningen en exponenten.

Tabel 1 toont verschillende rekenkundige operatoren die je kunt gebruiken wanneer je programmeert in JavaScript. Er zijn twee getallen als voorbeeld genomen om de syntax van een operator uit te leggen. Dit zijn de getallen 4 en 2.

Operator	Beschrijving	Syntax
+	Plus: telt twee waarden bij elkaar op.	4 + 2
-	Min: trekt twee waarden van elkaar af.	4 - 2
*	Vermenigvuldig: telt twee waarden herhaaldelijk bij elkaar op.	4 * 2
/	Delen: trekt twee waarden herhaaldelijk van elkaar af.	4 / 2
%	Modulo: berekend de restwaarde van een deelsom.	4 % 2

Tabel 2: Overzicht van rekenkundige operatoren

Lesopdracht 1: Hieronder staan enkele opgaven. Geef voor iedere regel aan wat het resultaat is en leg in je eigen woorden uit waarom dit zo is.

1) 5 < 3 != 4 > 8

- 2) $18 \% 4$
- 3) $2 + 10 * 4$

Logische operatoren

Logische operatoren worden gebruikt met true/false-waarden en geven - afhankelijk van de gebruikte operator - een true of false waarde terug. Tabel 3 toont verschillende logische operatoren die je kunt gebruiken.

Operator	Beschrijving	Syntax
&&	Logische en	true && false
	Logische of	true false
!	Logische niet	!false

Tabel 3: Overzicht van logische operatoren

Lesopdracht 2: Hieronder staan enkele opgaven. Geef voor iedere regel aan wat het resultaat is (true/false) en leg in je eigen woorden uit waarom dit zo is.

- 4) true && true
- 5) true && false
- 6) true || false
- 7) !true

Datatypes

Datatypes **[B3]** worden ook wel gegevenstypes genoemd en zijn een belangrijk onderdeel van programmeren. Afhankelijk van een datatype kun je operators bijvoorbeeld wel/niet gebruiken.

Tabel 3 toont een overzicht met verschillende datatypes:

Datatype	Syntax	Variabelen van bepaalde datatypen
Strings	""	var str = "";
Numbers	getal (bijv. 1)	var nummer = 24;
Boolean	true/false	var booleanWaarde = true;

Lesopdracht 3: Stel dat we een variabele x hebben, die de som is van een string en nummer, bijvoorbeeld:

8) `var x = "Fifa " + 20`

Voer het bovenstaande stukje code uit in de console en gebruik de functie `typeof()` om de datatype van x te achterhalen. Leg uit welk datatype x heeft.

Comments

Comments kunnen gezien worden als aantekeningen en kunnen gebruikt worden om uit te leggen waarom code op een bepaalde manier wordt geschreven, of waarom een check belangrijk is. Een andere situatie waarin comments gebruikt kunnen worden is om collega's te waarschuwen om een bepaald bestand niet handmatig te veranderen.

JavaScript biedt de mogelijkheid om comments op een regel te schrijven (single line comments), of verdeeld over meerdere regels (multi-line comments). De volgende code snippet demonstreert beide opties:

```
// Dit is een single line comment
// Wanneer ik een dubbele forward-slash (//) gebruik, moet ik dit steeds opnieuw typen om een nieuwe regel als
comment toe te voegen.

/*
Dit is een multi line comment. Ik kan hier zoveel text schrijven als ik wil, zonder dat ik extra tekens moet
toevoegen.
*/
```

Lesopdrachten en huiswerk

De deadline voor de les - en huiswerk opdrachten kun je vinden in Microsoft Teams. De uitwerkingen kan je als Word - of text document inleveren in Microsoft Teams.

Het huiswerk voor deze week kun je vinden op Microsoft Teams; Team "Databases - `${classcode}`". Als je bijvoorbeeld in klas OITAOO9A zit, kun je je huiswerk vinden in Team "JavaScript- OITAOO9A".

Begrippen

[B1] Case sensitive = Hoofdletter gevoeling, met andere woorden, x en X zijn niet hetzelfde.

[B2] Operators = Een andere benaming voor be(werker). Een voorbeeld is bijvoorbeeld een rekenkundige operator +. Deze wordt gebruikt om op te tellen.

[B3] Datatype = Datatype, ook wel een gegevenstype genoemd. Voorbeelden zijn integers (getallen), string (text).

Revision #6

Created 29 January 2020 13:57:08

Updated 22 February 2020 21:41:04