Contents

[Data structures: Objects and Arrays 2](#_Toc63938716)

[1.1. Properties(eigenschappen) 2](#_Toc63938717)

[1.2. Objects 2](#_Toc63938718)

[JavaScript and the browser 4](#_Toc63938719)

[2.1 The web 4](#_Toc63938720)

[The Document Object Model (DOM) 6](#_Toc63938721)

[3.1. Structuur 6](#_Toc63938722)

[Bronnenlijst 7](#_Toc63938723)

# Data structures: Objects and Arrays

Javascript biedt de optie aan om een reeks van waardes op te slaan in een lijst. In programmeertermen beter bekend als een ‘array’.

Het wordt geschreven als een lijst, met waardes die gescheiden worden door een komma. Het is belangrijk dat je brackets gebruikt om hier de waardes in op te slaan. Hier volgt een kleine demonstratie:



Bij een array is het belangrijk dat je weet dat 0 ook een waarde heeft. Het telt namelijk vanaf 0, 1, 2 enzovoort. Dus als we de bovenstaande code zouden runnen zou de log ‘6’ aan moeten geven. Ook is het belangrijk om te weten dat je hier meerdere data types in op kunt slaan en niet alleen cijfers.

## Properties(eigenschappen)

Variabele.length = verkrijg de lengte van een string (een van de vele eigenschappen)

Bijna alle javascript waardes hebben eigenschappen. De uitzonderingen zijn ‘Null’ en ’Undefined’. Als je bij die 2 waardes een eigenschap probeert te bereiken krijg je een TypeError. Maar wat zijn properties eigenlijk?

De 2 voornaamste manieren op de eigenschappen te bereiken zijn met een ‘.’ En met ‘[]’.

### Objects

Een object maak je met ‘{}’. Hier voeg je allemaal waardes aan toe en deze waardes definiëren zo’n object. Dus wat zijn objects? Bijna alles. Laten we hier een voorbeeld van een kat gebruiken.

Var kat = {

Gedrag: “irritant”,  
Kleur: “zwart”,  
leeftijd: 6,  
poten: 4,  
miauwDecibelNiveau: 783  
}

We hebben zojuist een object aangemaakt met de eigenschappen van een gemiddelde kat. Dus objects zijn een verzameling van verschillend eigenschappen. Om deze objecten uit de lezen kun je de ‘.’ Gebruiken.

Bijvoorbeeld “Mijn kat is “ + kat.kleur;

Als we dit zouden loggen zouden we krijgen: Mijn kat is zwart.

# JavaScript and the browser

~ Without web browsers, there would be no JavaScript.

Het computernetwerk is rond de jaren ’50 ontstaan en heeft een compleet nieuwe wereld geopend voor vele van ons. Door meerdere computers met elkaar te laten verbinden ontstaat de mogelijkheid om enorme hoeveelheid data heen en weer te sturen. In de jaren ’50 was dit nog lokaal en gebeurde dit door middel van kabels die je met meerdere computers verbond en zodoende informatie naar elkaar stuurde. Dit leverde toentertijd al mooie resultaten op, dus kwamen zij (de ontwikkelaars) met de visie om dit wereldwijd te kunnen doen voor geweldige resultaten…Dit is het internet.

Dit wordt gedaan door naar elkaar bits te versturen. Beide computers moeten weten daar de bits voor staan om zo de bits te ontcijferen en er zo duidelijke data ontstaat. Dit wordt bepaald door het netwerkprotocol, denk aan de welbekende http-protocol, dit staat voor ‘Hypertext Transfer Protocol’.

Dit protocol wordt gebruikt om grote delen informatie te ontvangen en te ontcijferen en te laten zien. Denk aan webpagina’s en afbeeldingen. Het werkt met POST en GET. De namen zeggen eigenlijk al genoeg, als je een GET request uitvoert dan haalt hij data op en als je een POST request uitvoert dan stuur je data. Dit wordt vaak gebruikt om data op te slaan zoals een ingevuld formulier of een registratie pagina, maar het houdt hier niet op. Je kan vele soorten informatie POST (en).

Hier een voorbeeld van hoe zo’n request er uit kan zien:

GET/ index.html HTTP/1.1.

Breakdown:

GET: Haalt informatie op  
/ index.html: Wat moet hij ophalen? In dit geval het index.html bestand.  
HTTP/1.1: Dit is de versie van het protocol wat gebruikt wordt.

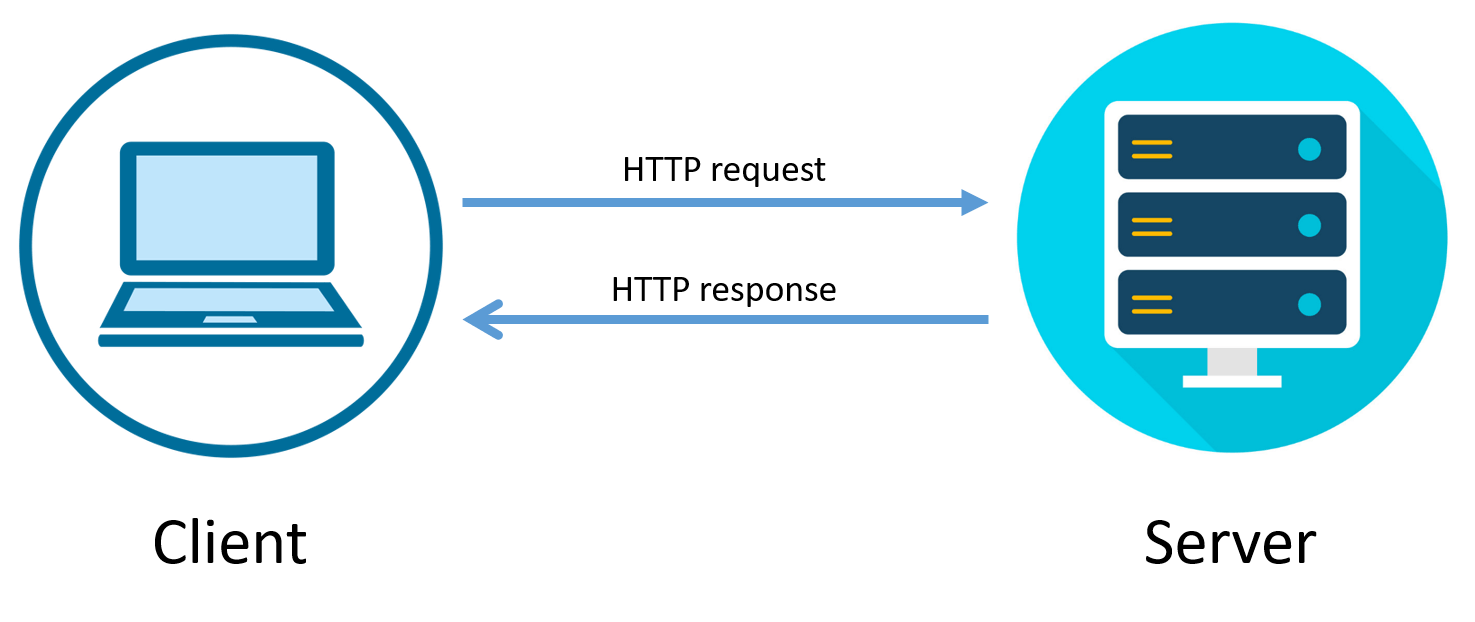
## 2.1 The web

WWW staat voor World Wide Web, dit wordt vaak verward met het internet. Het wereldwijde web bestaat uit een meerdelige set van protocollen die ervoor zorgen dat wij webpagina’s kunnen bezichtigen via een browser.

Hoe zorg je er eigenlijk voor om deel uit te maken van het WWW? Om deel uit te maken van het web moet je een machine hebben die je kan verbinden met het internet en dan de port veranderen naar 80 en het HTTP-protocol selecteren. Nu kan je machine communiceren met andere apparaten die op het internet/web zijn aangesloten. Elk document zou dan een eigen adres moeten hebben om bezocht te kunnen worden. Als metafoor zou je je eigen adres kunnen gebruiken. Het heeft een adres, postcode en nummer. Zo gaat dat ook via het web.

Breakdown:

Protocol: HTTP://  
Server: 24kitchen.nl/  
Path: de\_lekkerste\_brownies.html



https://bytesofgigabytes.com/networking/how-http-request-and-response-works/

# The Document Object Model (DOM)

Het Document Object Model staat voor hoe een pagina is opgebouwd. De browser bouwt een model van de structuur en bouwt hiermee de pagina op het scherm. Het DOM kan worden gemanipuleerd door middel van het gebruik van Javascript. Dit kan worden gedaan omdat het een live datastructuur is en zodra deze worden bewerkt zie je gelijk het resultaat hiervan.

## 3.1. Structuur

Een HTML-document bestaat uit een gelaagde set van elementen. Je begint natuurlijk eerst met een HTML-tag en deze krijgt ‘children’ en ook deze ‘children’ krijgen ‘childrens’. Je snapt het idee, het bestaat uit allemaal verschillende elementen die op een bepaalde volgorde in elkaar zitten.

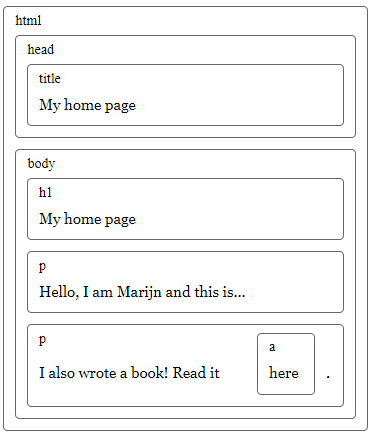


Image by: <https://eloquentjavascript.net/14_dom.html>

De browser gebruikt het DOM om een bepaalde datastructuur te laten zien. Voor elk element is er een object waar de browser mee communiceert om uit te zoeken om welk HTML-tag het gaat en wat de inhoudt hiervan is. De weergave hiervan wordt het DOM genoemd.

# Bronnenlijst

* *Data Structures: Objects and Arrays :: Eloquent JavaScript*. (z.d.). eloquentjavascript. Geraadpleegd op 11 februari 2021, van <https://eloquentjavascript.net/04_data.html>
* *JavaScript Objects*. (z.d.). W3Schools. Geraadpleegd op 11 februari 2021, van <https://www.w3schools.com/js/js_objects.asp>
* Dwyers, S. (2009, 22 april). *Progressive Enhancement: What It Is, And How To Use It?* Smashing Magazine. <https://www.smashingmagazine.com/2009/04/progressive-enhancement-what-it-is-and-how-to-use-it/>
* *The Document Object Model :: Eloquent JavaScript*. (z.d.). Eloquent Javascript. Geraadpleegd op 11 februari 2021, van <https://eloquentjavascript.net/14_dom.html>
* *JavaScript and the Browser :: Eloquent JavaScript*. (z.d.). Eloquent Javascript. Geraadpleegd op 11 februari 2021, van https://eloquentjavascript.net/13\_browser.html