

# Bootcamp dag 1

Dit zijn mijn aantekeningen van alle bronnen, ik heb het zo kort mogelijk proberen voor mezelf te verwoorden om het te kunnen begrijpen. Ik ben niet goed in Javascript en kan het een beetje lezen, dus ik probeer er zoveel mogelijk zo goed mogelijk van op te steken.

# Introduction

[https://eloquentjavascript.net/00\\_intro.html](https://eloquentjavascript.net/00_intro.html)

Computers zijn steeds moeilijker geworden, je kunt je mail bekijken je kunt je rekenmachine pakken. Je computer weet wel wat je wil. Maar voor unieke dingen is er niet altijd een applicatie. Programmeren is daarvoor handig, programmeren is het constructueren van een programma, ook wel domme instructies die zeggen wat er moet gebeuren. Computers zijn dom. Programmeren is moeilijk en frustrerend maar wel rewarding. Je kunt je computer dingen laten doen die het eerst niet kon. Een programmeertaal bestaat ook, zo kun je communiceren met de computer. Javascript is er hier een van. Er is veel veranderd wat betreft interface van computers, vroeger had je DOS en BASIC. Dit zorgde voor minder vrijheid maar makkelijker te gebruiken.

Programmeren, je maakt je eigen doolhof waar jezelf in verdwaald. Een computer is een fysieke machine die reageert als een host, maar computers kunnen alleen domme directe dingen doen. Maar ze zijn wel snel en kunnen moeilijke dingen doen. Best practice, ook wel zo min mogelijk met zo goed mogelijke code.

Javascript is ook erg veranderd door de tijd heen. Javascript bestaat sinds 1995 om programma's naar de webpage Netscape navigator browser. Ook kan je interactievere en leuker maken. Er is een document gemaakt hoe je javascript moet gebruiken, dit heet ECMAScript. Problemen komen pas echt in JS als je het niet weet, voor beginners is het soms makkelijker dan tweede rang.

Talen veranderen dus browsers ook, oude browsers vervallen daarom. Databases zoals MongoDB en CouchDB gebruiken ook Javascript.

# The Complete ECMAScript 2015-2019 Guide

<https://flaviocopes.com/ecmascript/>

ECMAScript is de standard script waarop JS is gebaseerd, het woord ook wel ES genoemd. Javascript gebruikt veel termen zoals: ES3, ES5, ES6, ES7, ES8, ES2015, ES2016, ES2017, ECMAScript 2015, ECMAScript 2016, ECMAScript 2017, ECMAScript 2018, ECMAScript 2019.

Dit zijn ECMAScript. Hierbij heb je ook nog ActionScript en Jscript. Actionscript wordt wel minder gebruikt door het vertrek van Flash. JScript was een soort dialect script, soort uitzondering. Javascript is de populairste.

TC39 zorgt dat Javascript verandert en evolueert. Bedrijven die hierbij horen zijn, Mozilla, Google, Facebook, Apple, Microsoft, Intel, Paypal en andere. Dit zijn gewoon de bedrijven die over Javascript beslissen, zo moet je het ongeveer zien. Zo kun je steeds maar 1 script te hoeven leren.

ES11 – ES2020 – Summer 2020?

ES10 – ES2019 – Summer 2019

-  
-  
-  
-

ES1 – ES1 – June 1997

ES.Next is een naam dat altijd de laatste versie van Javascript laat zien.

# You don't know JS yet: Get started

## – 2<sup>nd</sup> edition

<https://github.com/getify/You-Dont-Know-JS/blob/2nd-ed/get-started/ch1.md>

Javascript is een bestemming, geen richting.  
Scope/Closures, Prototypes/objects, types/coercion.

Javascript is an artifact of marketing shenanigans. Toen het bedacht werd door Brendan Eich noemde hij het Mocha. Maar wanneer het live ging naar publiek, hete het JavaScript omdat mensen het zo noemde. Dit omdat op dat moment het meest werd aangesproken door Java users. Script betekent Licht gewicht programmeren. Het had ook WebJava kunnen heten. C/C++ gebruiken net zoals Java “{ }” om te openen en sluiten. Ook gebruiken ze beide “;”. Zeg Javascript of ES2020, zeg niet ES2017 of iets anders. We praten in de tijd van nu, gebruik dat.

# var, let and const – What, why and how – ES6 JavaScript Features

<https://www.youtube.com/watch?v=sjyJBL5fkp8>

Var pakt alles, waar het ook is, het gaat het proberen te zoeken. Let doet het alleen binnen de mapjes waar binnen het staat, dus bijvoorbeeld in een functie. Const is een constante eenheid die niet mag worden veranderd.

Let = blockscope

Var = functionscope

# Value, types and operators

[https://eloquentjavascript.net/01\\_values.html](https://eloquentjavascript.net/01_values.html)

In de computer wereld is er alleen data, data om te lezen, om te maken enz... Dit bestaat uit nullen en enen. Soms zijn value nummers, soms tekst, soms, sommige zijn functies. Deze values moeten ergens zijn opgeslagen. Bij te veel data raakt je geheugen vol. Javascript werkt met 64 bits omdat je zoveel verschillende patronen hiermee kan maken.

Voor getallen met decimalen gebruiken we een punt. (9.81). Voor hele grote getallen dit (2.998e8). Dit betekent  $2.998 \times 10^8 = 299,800,000$ .

+ en \* noem je operators. Computers moet je helpen bij rekenen.

Niet:  $7 + 8 * 3$

Wel:  $(7 + 8) * 3$

Je hebt ook nog het % symbool. Dit is wat je over houdt.  $144 \% 12 = 0$   
 $5 \% 2 = 1$ .

NaN = Not a Number maar is wel een value.

"A newline character is written like `\"\\n\"`."

`"con" + "cat" + "e" + "nate"`

Je hebt verschillende symbolen die je kunt gebruiken voor Boolean.

>groter dan

<kleiner dan

=< kleiner dan of gelijk aan

=>groter dan of gelijk aan

= is dit

== is gelijk aan

=== is gelijk aan value en soort

!= is niet gelijk aan

|| als een van de twee waar is, is het waar

&& als beide waar zijn, is het waar

Null of undefined betekent dat het geen waarde heeft gekregen of er niet is.

```
console.log(8 * null)
```

```
// → 0
```

```
console.log("5" - 1)
```

```
// → 4
console.log("5" + 1)
// → 51
console.log("five" * 2)
// → NaN
console.log(false == 0)
// → true

console.log(null || "user")
// → user
console.log("Agnes" || "user")
// → Agnes
```

# General Javascript guidelines

[https://developer.mozilla.org/en-US/docs/MDN/Contribute/Guidelines/Code\\_guidelines/JavaScript#General JavaScript guidelines](https://developer.mozilla.org/en-US/docs/MDN/Contribute/Guidelines/Code_guidelines/JavaScript#General_JavaScript_guidelines)

Maak je code zo leesbaar mogelijk. Gebruik op de juiste manier spaties en enters. Code moet duidelijk zijn met relevante logische begrijpelijke namen. Gebruiken elementen waar ze voor bedoeld zijn.