

# Javascript intro

# Inhoud

- Inleiding Javascript
  - Basisbegrippen: variabelen, primitieve datatypes.
  - Eerste code schrijven
  - Instructies en operatoren.
  - Functies –Hoisting.
  - Arrays.

# Inleiding

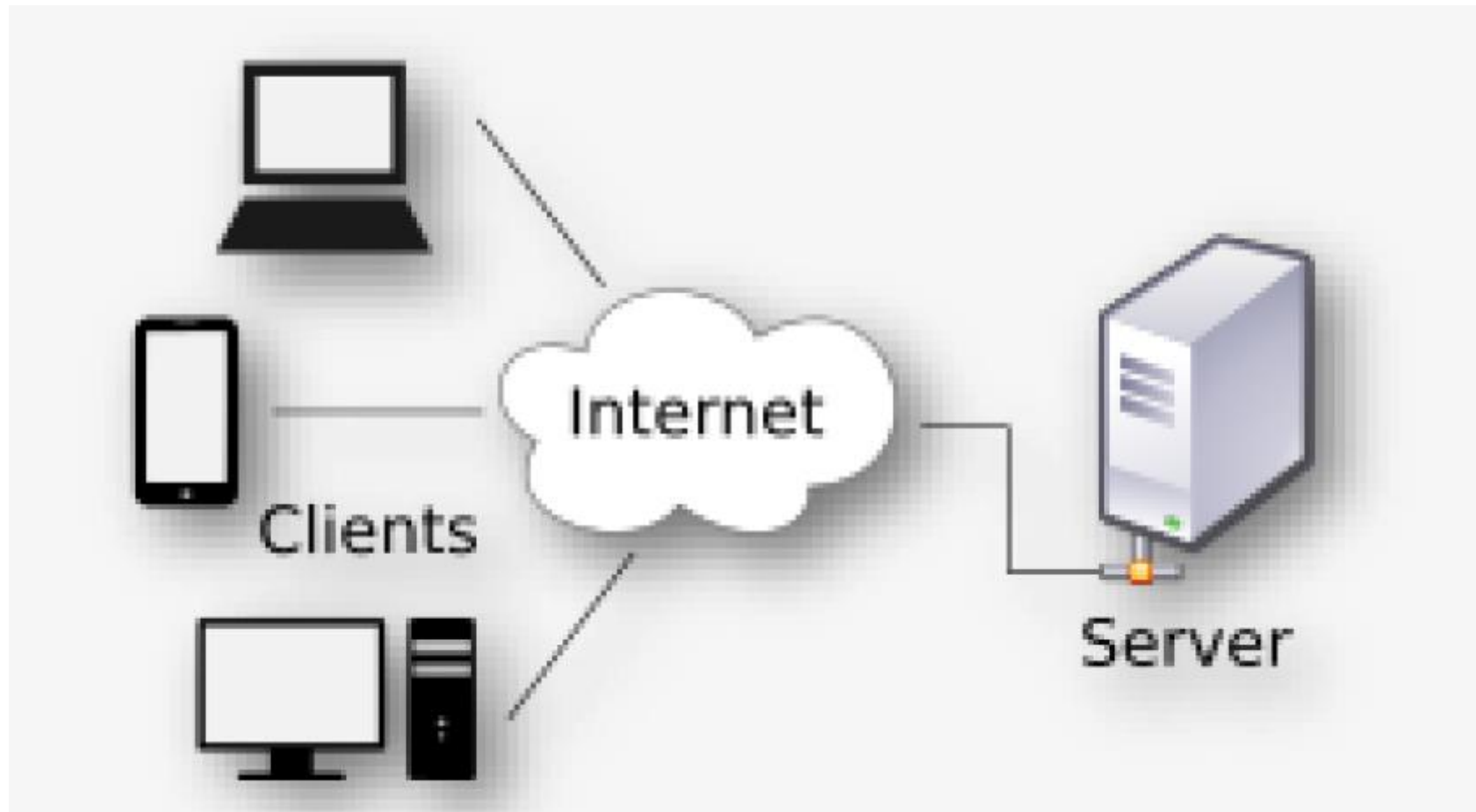
Dagelijks zijn we actief op het Web:

- facebook of andere (a)sociale media raadplegen
- filmpje kijken
- shoppen
- nieuws lezen
- fotootjes kijken
- muziek (il)legaal downloaden
- gamen

# Surfen op het web

- Bij bezoek aan een webpagina, ga je eigenlijk een **verzoek doen aan een server** (uri = adres van server –www.deredactie.be) **om een pagina** naar jouw toestel (laptop –tablet –smartphone) door **te sturen**.
- Als deze pagina op de server staat, zal deze naar jouw toestel doorgestuurd worden.
- Je **browser** (de **client**) op je toestel zal deze pagina opbouwen en weergeven.
- Surfen op het web maakt dus gebruik van een **client-server architectuur**.

# Client-Server Architectuur



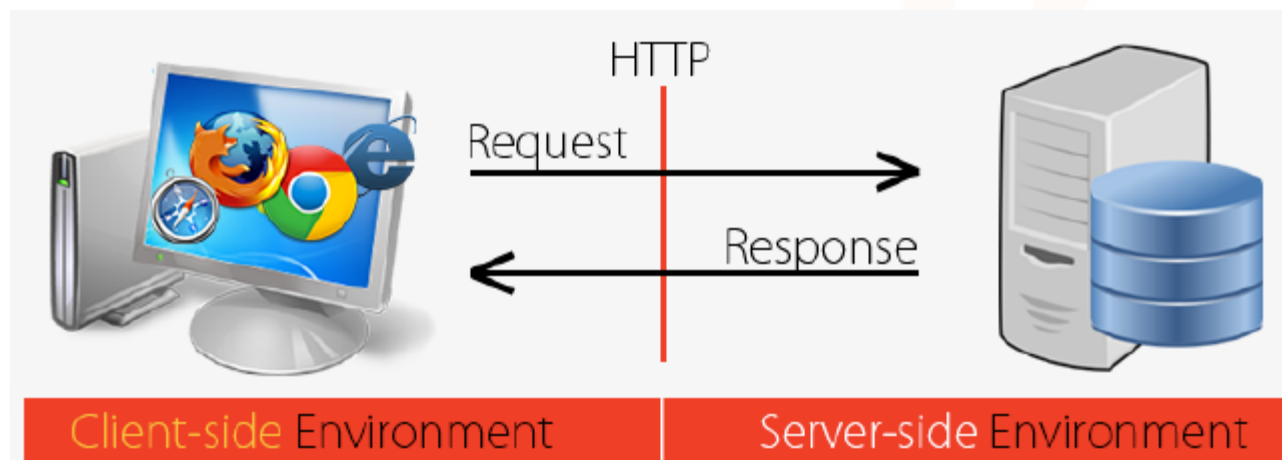
# Client-Server Architectuur

Het **client-servermodel** is een model in de informatica en computertechnologie voor de samenwerking tussen twee of meer programma's, die zich op verschillende computers kunnen bevinden.

Kenmerkend voor de asymmetrie in het model is:

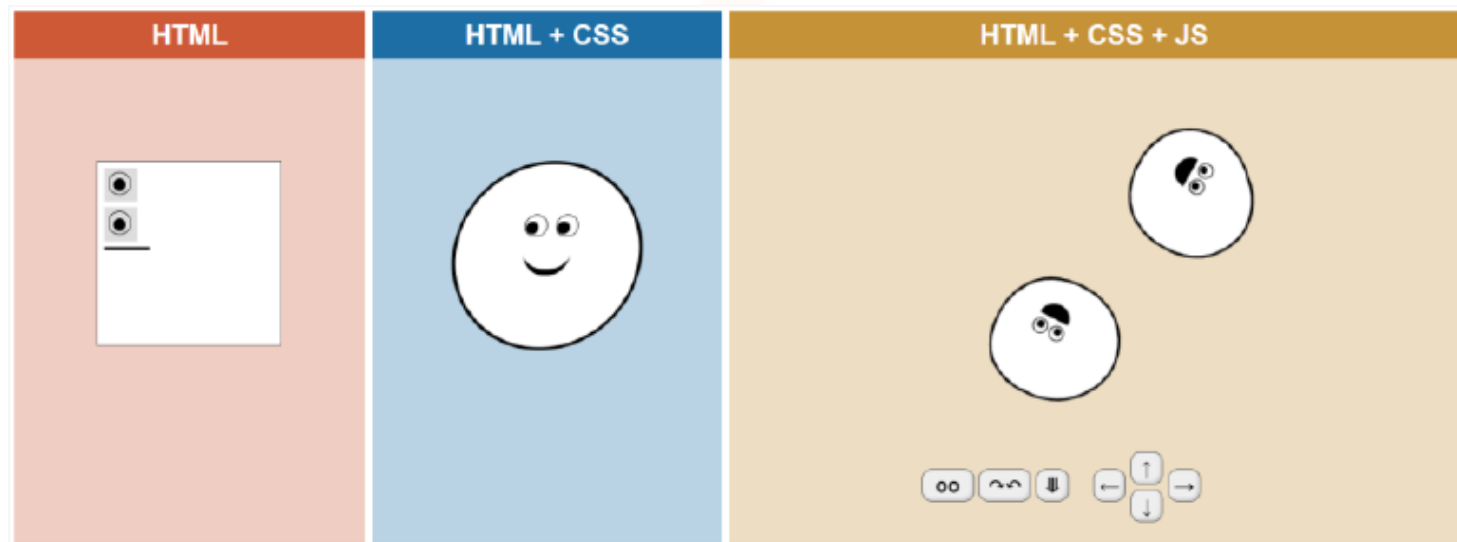
- de server is permanent beschikbaar en is reactief
- een client is bij gelegenheid actief en neemt het initiatief tot communicatie met de server

Een aantal voorbeelden van het client-servermodel zijn e-mail, het www (HTTP), FTP, Telnet, ...



# Client-Server Architectuur: Client

- **De client** (browser) begrijpt drie talen:
  1. **HTML–Hypertext Mark-up Language**: geeft aan je pagina **structuur** en **betekenis** door middel van headings, tabellen, afbeeldingen, lijsten, ...
  2. **CSS–Cascading Stylesheet**: geeft aan je pagina **opmaak** en **layout** door middel van logische indeling en eigenschappen aan de inhoud zoals lettertype –kleur –...
  3. **ECMAScript(= JavaScript)**: geeft interactie aan je pagina zoals het klikken op knoppen, met je cursor bewegen, vraagt data op en geeft deze in de browser weer, kan voor animatie zorgen (games) ...



# Javascript - Geschiedenis

- Een uitgebreide geschiedenis van JavaScript vind je op <https://en.wikipedia.org/wiki/JavaScript#History>
- Een video van de bedenker en ontwikkelaar van JavaScript: <https://www.youtube.com/watch?v=3-9fnjzmXWA>



# Javascript

- De meeste scriptingtalen zijn ontworpen voor ofwel de client side ofwel de server side te worden uitgevoerd.
- JavaScript is oorspronkelijk ontworpen als **client-side(front-end)** technologie
- is sinds enkele jaren verder ontwikkeld als **server-side (back-end)** technologie: **Node.js**



# ECMAScript

ES6 (ook ECMAScript2015 genaamd)

Een overzicht van de nieuwe features: <http://es6-features.org/>

- Verbeterde syntax voor features die reeds bestonden via externe libraries
  - Classes
  - Modules
- •Nieuwe functionaliteiten in de standard bibliotheek
  - Nieuwe methodes voor strings en Arrays
  - Promises
  - Maps, Sets
- •Volledig nieuwe features
  - Generators
  - Proxies
  - WeakMaps
- De browser ondersteuning: <https://kangax.github.io/compat-table/es6/>

# ECMAScript

**ES Next:** code name voor nieuwe versies (totdat het versie nummer gekend is)

- Probleem met ES6: 6 jaar nodig alvorens standaardisatie
- Vanaf ECMAScript2016(ES7): 1 release per jaar, volgens het TC39 process

Meer op: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/New\\_in\\_JavaScript/ECMAScript\\_Next\\_support\\_in\\_Mozilla#See\\_also](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_Next_support_in_Mozilla#See_also)

Browser ondersteuning : <https://kangax.github.io/compat-table/es2016plus/>

# ECMAScript

Een ECMAScript-engine is een programma dat de **broncode uitvoert** die is geschreven in een versie van ECMAScript (Javascript).

Elke browser heeft zo zijn eigen ECMAScript-engine.



**Chakra**



**Nitro**



**V8**



**Carakan**



**SpiderMonkey(s)**

# ECMAScript Transpilers

- Niet alle browsers ondersteunen reeds alle features.
- Transpilars maken het mogelijk om ES versies van morgen reeds vandaag te gebruiken en te runnen in de browser
- Enkele voorbeelden: [Babel](#), [Traceur](#)
- Ook tal van andere talen zoals **TypeScript**, Dart ... maken gebruik van transpilars om naar ES5 om te zetten

## Babel transforms your JavaScript

You put JavaScript in

JavaScript

Try Copy

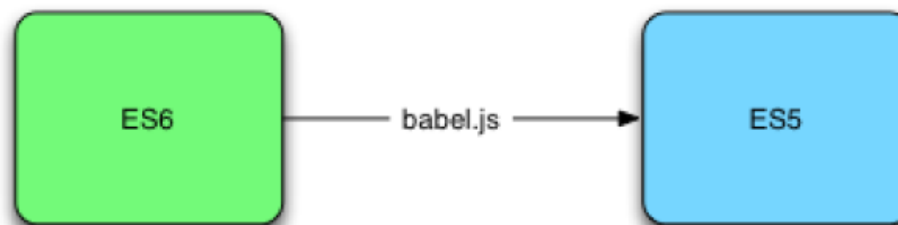
```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

JavaScript

Try Copy

```
[1,2,3].map(function(n) {  
  return n + 1;  
});
```



# Java versus Javascript

- Java is een object geörienteerde programmeertaal.
- JavaScript is sterk beïnvloed door functionele programmeertalen.

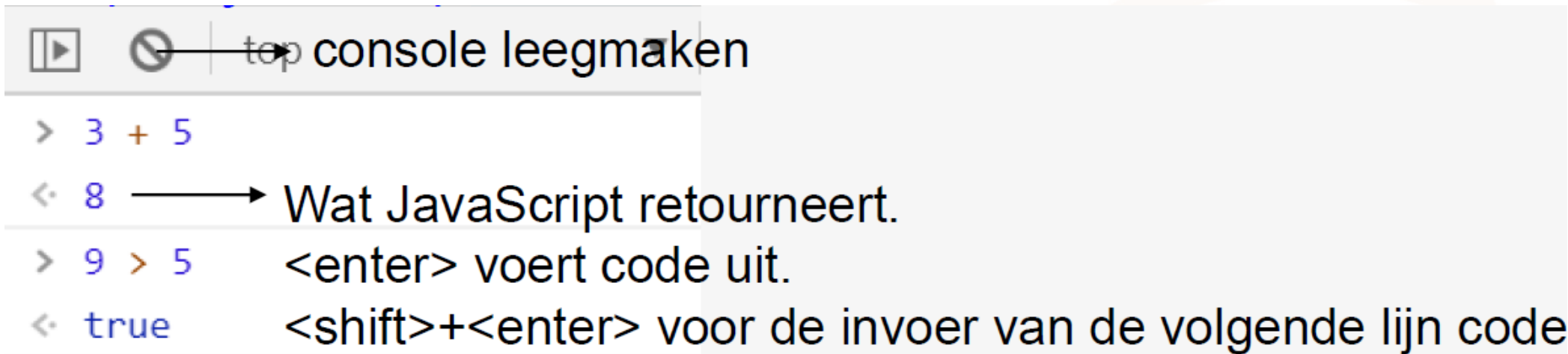


Java	JavaScript
Strongly typed	Losely typed
Static	Dynamic
Classical inheritance	Prototypal
Classes	Functions
Constructors	Functions
Methods	Functions
Compiled	Interpreted
Methods for working with I/O – network / files	Uses the hosting environment

# Javascript console

- Open Chrome
- Ga naar de Developer tools (F12) (Mac: alt-cmd-I)
- Kies de Console tab. Hier kan je JavaScript commando's uitvoeren.
- Instructies voor Safari op Mac

<https://javascript.info/devtools#safari>



# Javascript toevoegen aan html page

Via `<script>` tag voeg je JavaScript toe aan de html pagina

–De JavaScript-code staat ofwel in de webpagina zelf ( bestand met extensie `.html`)

.html

```
<script>  
...  
</script>
```



# Javascript toevoegen aan html page

De JavaScript-code kan ook in een afzonderlijk bestand staan. In de script-tag wordt verwezen naar dit bestand

(Best practice)

- Conventies : extensie bestand **.js**
- Het bestand bevat enkel de JavaScript code, geen script tag
- **src**= verwijzing naar JavaScript bestand, relatief of absolute adressering
- **Voordeel**: herbruikbaarheid, onderhoudbaarheid, caching in browser

.html

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

.js

```
console.log("Hello world");
```

# Javascript toevoegen aan html page

De script-tag plaats je in de `<body>` tag **na** de inhoud (juist voor `</body>`))

–Reden: JavaScript wordt pas gedownload, geparsed, en geïnterpreteerd als pagina al is weergegeven in de browser.

# Referenties

<https://www.learn-js.org/>

<https://www.w3schools.com/js/default.asp>