

Node.js



Peter Kassenaar
Module 1 - Kennismaking

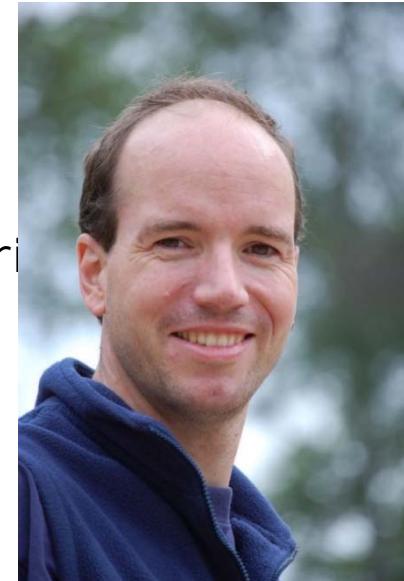
Peter Kassenaar

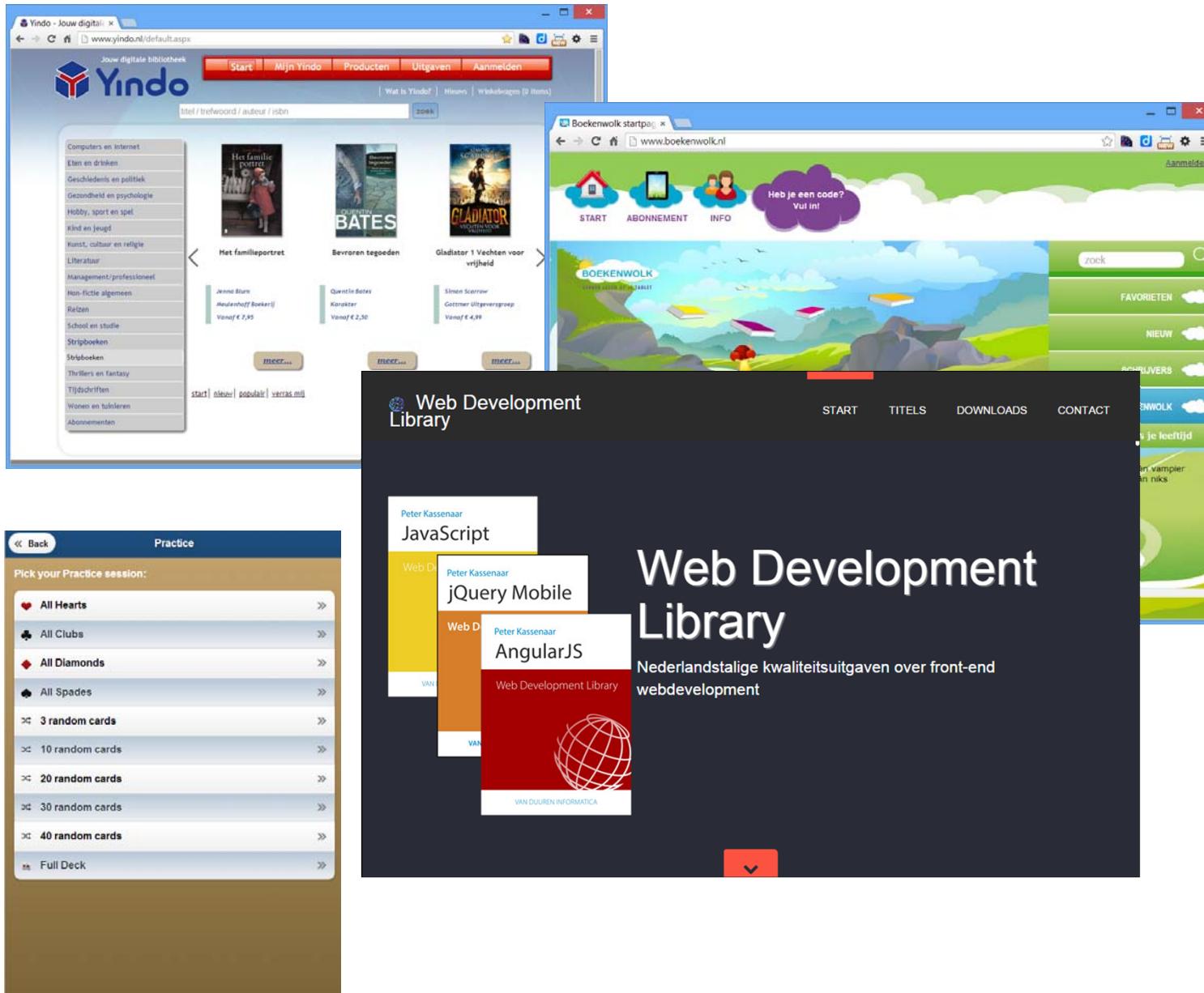
- Trainer, author, developer – since 1996
- Specialty: *“Everything JavaScript”*
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, React, Vue, etc.

www.kassenaar.com

info@kassenaar.com

Twitter: [@PeterKassenaar](https://twitter.com/@PeterKassenaar)





Node.js – Peter Kassenaar

Over jullie...



Voorkennis JavaScript, frameworks?

Voorkennis andere (web)talen?

Verwachtingen van de cursus?

Concrete projecten?

Day 3 – Node.js



- Make a choice...
- Node architecture and concepts
- Relationship with NPM
- Npm init and more npm CLI commands
- Structure and role of package.json
- Creating your own Node.js applications/packages
- Sample Node Core modules (Timers, File System, Http, Path, etc)
- Express + Express Router
- MongoDB
- Deployment
- Publication on NPM or Nexus ...

Werkwijze



Vragen?

Backend – multiple stacks

- **Java** Java 8, Java 9, IntelliJ, Wicket, Maven, JVM, JBoss, etc....
- **.NET** .NET Core, Visual Studio, IIS, ...
- **PHP** PHPStorm, Symfony, Laravel, mySql, ...



Front-end : **one** stack

- HTML, CSS, JavaScript
- But, the JavaScript landscape is

H.U.G.E

HTML



CSS



JS

Some JavaScript libraries and frameworks...



One central tool



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

10.16.0 LTS

Recommended For Most Users

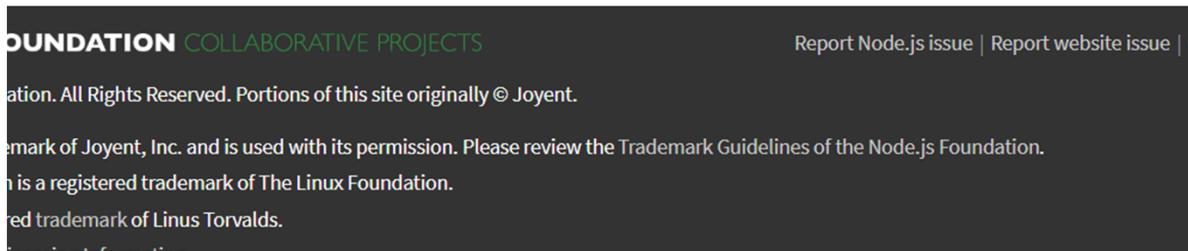
12.6.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.



Wat is NodeJS?

*“JavaScript runtime, designed to
run outside the browser”*

- Chrome V8 Engine als standalone product
- “Node is written *for* JavaScript, it is not written *in* JavaScript”

Node as an application container for...

- Webapplications (Express, MongoDB, Hapi)
- Real Time Communication (chat, messenger, stocks),
- tooling modules
 - TypeScript, Angular, Vue, React, PhoneGap,...
 - WebPack, Gulp, Grunt, Testing frameworks
 - And many, many more...
- Build tool
 - Custom `npm` scripts

<https://www.simplytechnologies.net/blog/2018/4/18/8-top-companies-that-rely-on-nodejs>



Alongside [the rising popularity of JavaScript](#), Node.js has got its own way of attracting many companies worldwide and of becoming a go-to technology for many

TL;DR: Netflix, Uber, Walmart, Paypal, LinkedIn, Medium, eBay, Nasa

Node.js – Peter Kassenaar

Walmart switched over to Node.js on a Black Friday, got more than 200 million visitors that day, and never went above 1% CPU.

LinkedIn rewrote their mobile backend in Node.js, and proceeded to get 20 times the performance out of 1/10 the servers.

Groupon increased page load speed by 50% by switching from Ruby on Rails to Node.js. They also reported being able to launch new features much faster than before.

Paypal did an experiment where two teams built identical apps - one in Java and one in Node.js. The Node.js team built theirs in half the time. The Node.js app had response times that were 50% faster than the Java app.

You can read more about these incredible performance gains (and developer productivity gains) [here ↗](#).

<http://www.roidna.com/blog/node-js-popularity/>

Creator: Ryan Dahl



2009-2012

Nu: Timothy J. Fontaine.

Bedrijf: joyent
<https://www.joyent.com/>

Node is not perfect

“10 things I regret about Node.js”

The screenshot shows a YouTube video player. On the left is a video thumbnail of Ryan Dahl speaking at a podium. The main content area has a white background with a decorative colorful border at the top and bottom. The title 'Regret: Not sticking with Promises' is displayed in bold black text. Below it is a bulleted list of five items. At the bottom of the video player are standard YouTube controls: play/pause, volume, progress bar (5:42 / 26:41), and a set of icons for sharing and settings.

Regret: Not sticking with Promises

- I added promises to Node in [June 2009](#) but foolishly removed them in [February 2010](#).
- Promises are the necessary abstraction for `async/await`.
- It's possible unified usage of promises in Node would have sped the delivery of the eventual standardization and `async/await`.
- Today Node's many async APIs are aging badly due to this.

10 Things I Regret About Node.js - Ryan Dahl - JSConf EU 2018

224.421 weergaven

VIND IK LEUK VIND IK NIET LEUK DELEN OPSLAAN ...

<https://www.youtube.com/watch?v=M3BM9TB-8yA>

Marketing blabla:

“Use NodeJS for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model ”

The screenshot shows the official Node.js website. At the top is a dark header bar with the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the header, a main content area features a paragraph about Node.js's V8 engine and package ecosystem. A green callout box highlights "Important security upgrades for recent V8 vulnerability". Below this, there are download links for Windows (x64) with two options: "v4.4.6 LTS" (Recommended For Most Users) and "v6.2.2 Current" (Latest Features). At the bottom, there are links for "Other Downloads | Changelog | API Docs" and a note about the LTS schedule.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

Important [security upgrades](#) for recent V8 vulnerability

Download for Windows (x64)

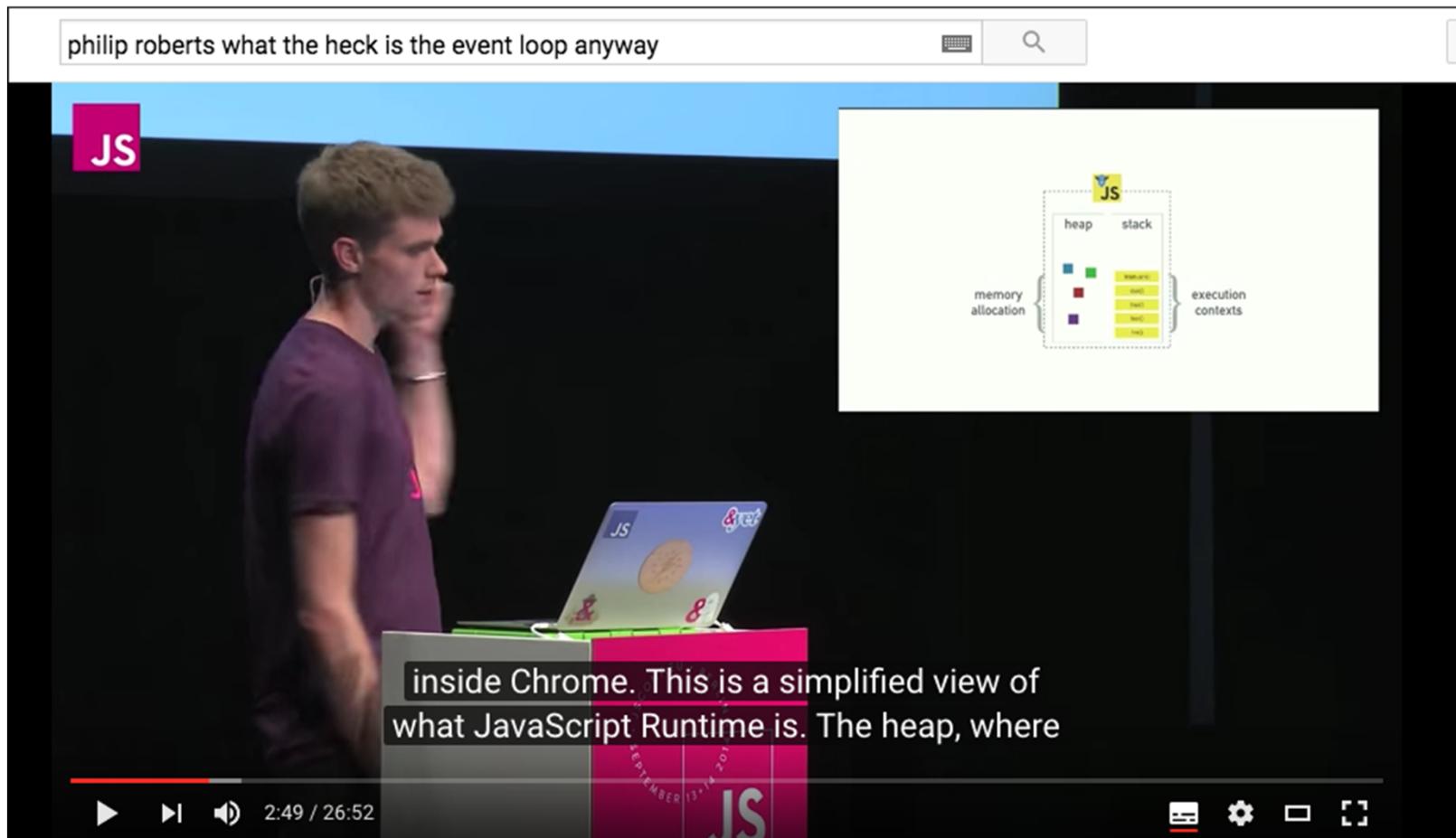
v4.4.6 LTS
Recommended For Most Users

v6.2.2 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [LTS schedule](#).

How does Node actually work?



<https://www.youtube.com/watch?v=8aGhZQkoFbQ>

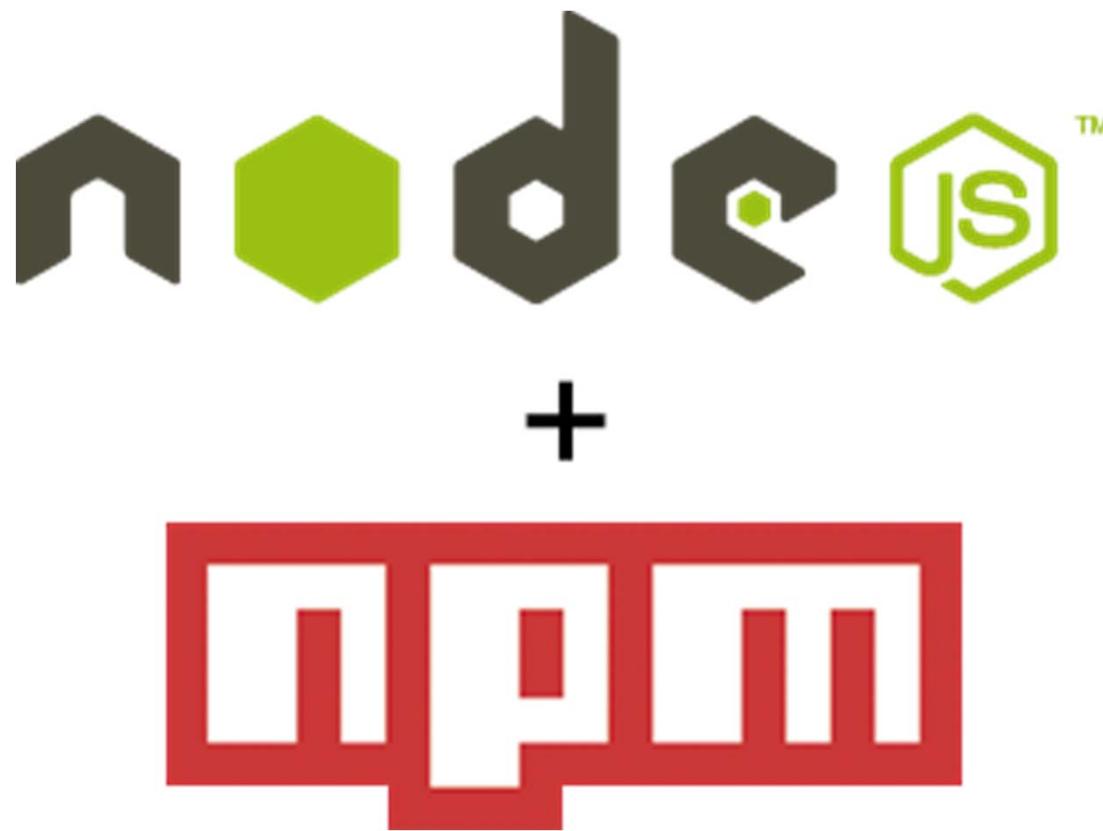
Node.js in het echt:

- Server side JavaScript voor netwerkapplicaties
- Framework Task runner: Vue, React, Angular,...
- Modular JavaScript en dependency management
- Single threaded: non-blocking en asynchronous code
 - Event based
 - Performance!
- Node.js 4.0+ (i.e. **all** modern versions)
 - V8-engine v4.5+
 - New ECMAScript 2015/ES6 - features

Over versienummers

- NodeJS 0.xx (2009 – 2014)
- Fork: *io.js* (versie 1.x, 2.x, 3.x)
- Node 4.x
- Node 6.x – met ES6 features
- Node 8.9.0+ minimal current stable release
- Node 10, 11, ... etc. (“meer van hetzelfde”)





What is NPM ?

- Well, it is actually multiple things..
 - NPM is the Node Package Manager
 - NPM is the worlds largest software registry
 - NPM is a command line tool for Node
- NPM is free
- NPM is installed with NodeJS
 - But it is a separate package
 - NPM can update itself
(`npm install -g npm`)

<https://nodejs.org/en/download/>

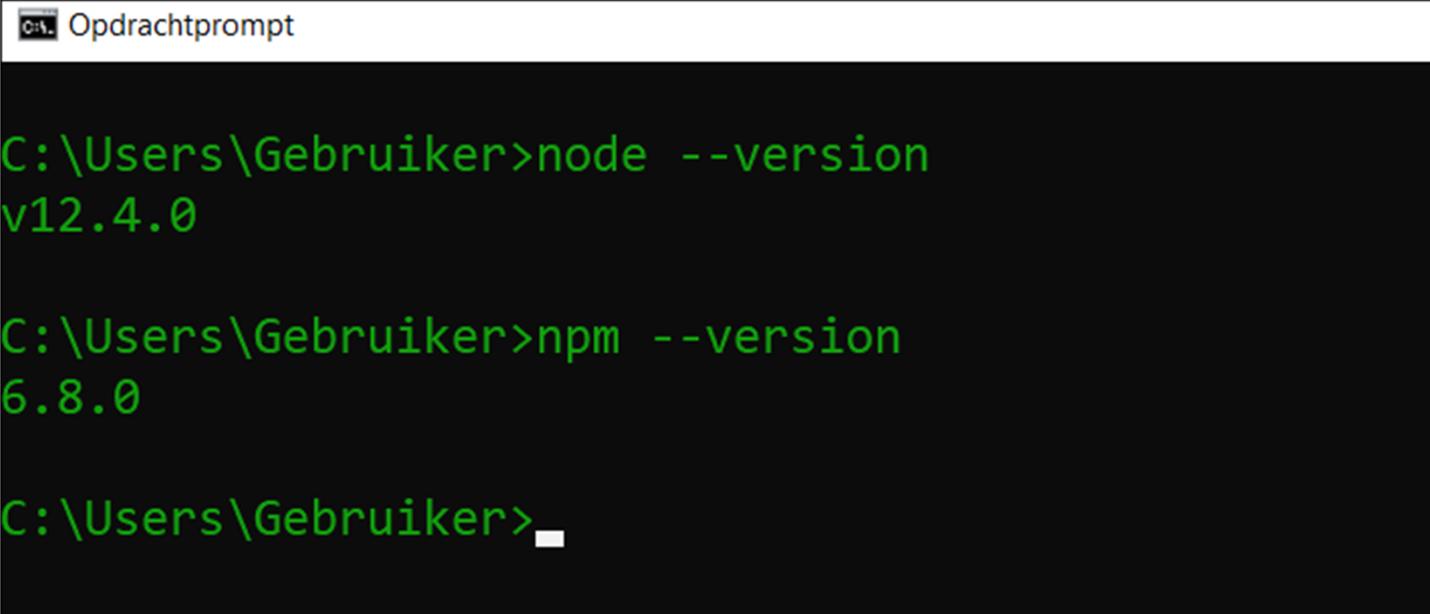
The screenshot shows the Node.js Downloads page. At the top is the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the navigation bar is a section titled "Downloads". It says "Latest LTS Version: 8.9.0 (includes npm 5.5.1)". A sub-section titled "Download the Node.js source code or a pre-built installer for your platform, and start developing today." follows. There are two main sections: "LTS" (Recommended For Most Users) and "Current" (Latest Features). Under "LTS", there are links for "Windows Installer" (node-v8.9.0-x64.msi), "Macintosh Installer" (node-v8.9.0.pkg), and "Source Code" (node-v8.9.0.tar.gz). Under "Current", there are tables for Windows, macOS, and Linux. The Windows table has rows for 32-bit and 64-bit. The macOS table has rows for 32-bit, 64-bit, and ARM. The Linux table has rows for 32-bit, 64-bit, ARMv6, ARMv7, and ARMv8.

	32-bit	64-bit
Windows Installer (.msi)		
Windows Binary (.zip)		
macOS Installer (.pkg)		
macOS Binaries (.tar.gz)		
Linux Binaries (x86/x64)		
Linux Binaries (ARM)		
Source Code		

	32-bit	64-bit
Windows Installer (.msi)		
Windows Binary (.zip)		
macOS Installer (.pkg)		
macOS Binaries (.tar.gz)		
Linux Binaries (x86/x64)		
Linux Binaries (ARM)		
Source Code		

	32-bit	64-bit
Windows Installer (.msi)		
Windows Binary (.zip)		
macOS Installer (.pkg)		
macOS Binaries (.tar.gz)		
Linux Binaries (x86/x64)		
Linux Binaries (ARM)		
Source Code		

Checking the version numbers



```
Opdrachtprompt  
C:\Users\Gebruiker>node --version  
v12.4.0  
  
C:\Users\Gebruiker>npm --version  
6.8.0  
  
C:\Users\Gebruiker>
```

The screenshot shows a Windows Command Prompt window titled "Opdrachtprompt". It contains two command-line entries. The first entry is "node --version" which outputs "v12.4.0". The second entry is "npm --version" which outputs "6.8.0". A red arrow points to the output of the first command, and another red arrow points to the output of the second command.

Synchroon vs asynchroon

```
//*****
// synchronous code:
//*****
var user = User.findById(100);
// ...wait for operation to complete, then do something with user
```



Blocking

```
//*****
// asynchronous code:
//*****
User.findById(100, function(user){
  // do something with user, once operation has completed
});
```



Callback

Modules and npm

CommonJS specificatie:

Elk bestand krijgt een *module.exports* object, zodat functies kunnen worden exposed

Nodejs heeft *geen* global scope!

```
// Maak module HelloWorld.js en exporteer deze

var myFunc = function(input){
    return input + ' world'
};

module.exports.HelloWorld = myFunc;

// in andere file:

var importHello = require('./HelloWorld');

console.log(importHello.HelloWorld('Hello'));

// prints: Hello world
```

Node installeren

The screenshot shows the 'Downloads' section of the Node.js website. At the top, there are two green buttons: 'LTS' (Mature and Dependable) and 'Stable' (Latest Features). Below these are links for 'Windows Installer' (node-v4.2.3-x86.msi), 'Macintosh Installer' (node-v4.2.3.pkg), and 'Source Code' (node-v4.2.3.tar.gz). A large table below lists various binary and source code options for different platforms and architectures. The table includes rows for Windows (.msi), Windows (.exe), Mac OS X (.pkg), Mac OS X Binaries (.tar.gz), Linux Binaries (.tar.gz), SunOS Binaries (.tar.gz), ARM Binaries (.tar.gz), Docker Image, and Source Code. It also shows specific builds for 32-bit and 64-bit architectures, and ARMv6, ARMv7, and ARMv8 for ARM binaries.

	32-bit	64-bit
Windows Installer (.msi)		
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)		64-bit
Mac OS X Binaries (.tar.gz)		64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
ARM Binaries (.tar.gz)	ARMv6	ARMv7
Docker Image	Official Node.js Docker Image	
Source Code	node-v4.2.3.tar.gz	

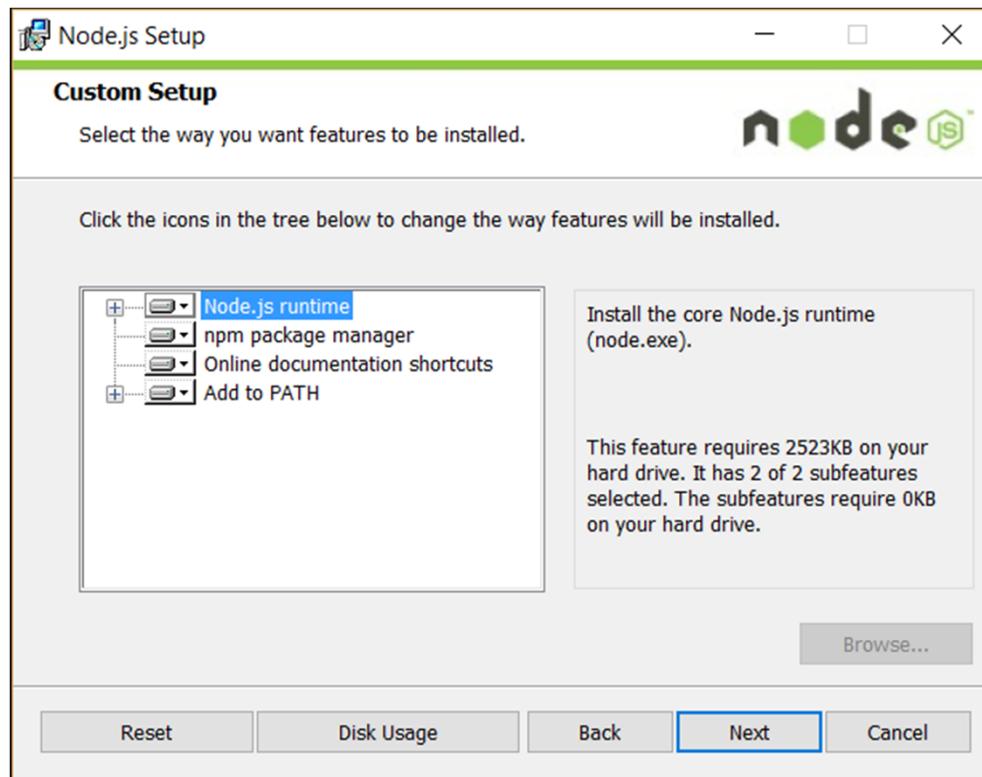
Current version: v4.2.3

- All download options
- Installing Node.js via package manager
- Previous Releases

Runtimes voor:

- Windows
- Mac
- Linux

Standaard installatiewizard

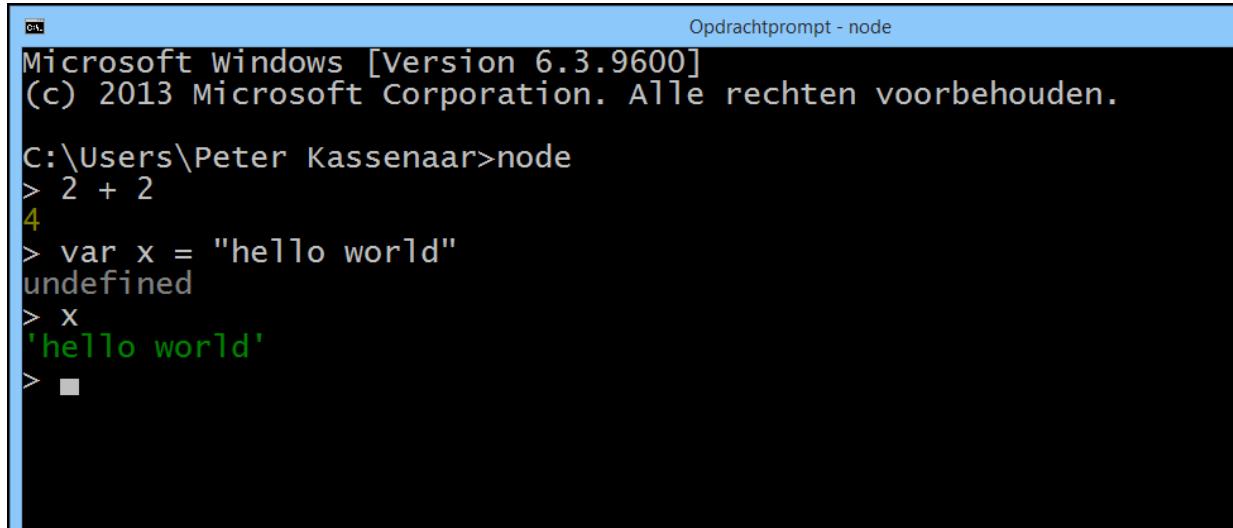


Let op toevoegen aan PATH

Node shell

Type node in een opdrachtprompt

Rechtstreeks JavaScript-opdrachten uitvoeren



```
Opdrachtprompt - node
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. Alle rechten voorbehouden.

C:\Users\Peter Kassenaar>node
> 2 + 2
4
> var x = "hello world"
undefined
> x
'hello world'
> ■
```

2 x Ctrl+C om te beëindigen.

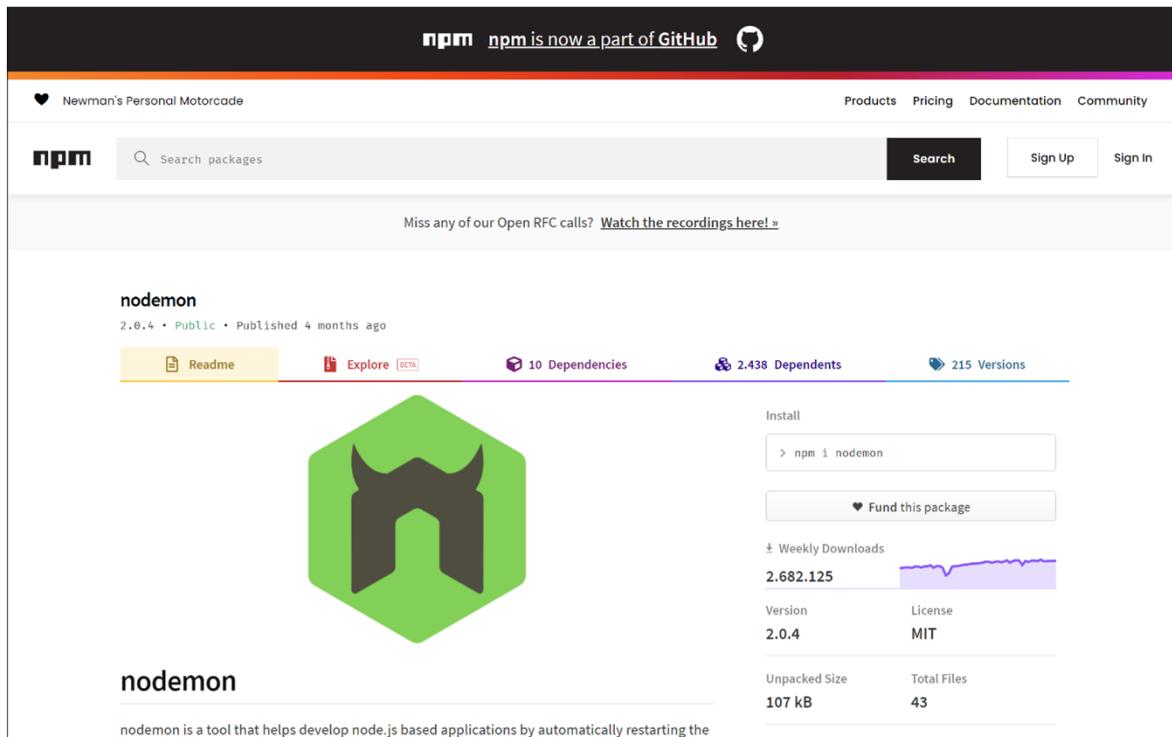
Common usage

- Command prompt: > node bestandsnaam.js
- De opdrachten in bestandsnaam.js worden uitgevoerd

Tip: gebruik nodemon

```
npm install nodemon -g
```

```
nodemon [fileName]
```



Editors voor Node.js

- Elke editor is OK.
- Liefst:
 - Color coding
 - AutoComplete
 - Debugging (!)



ATOM



Visual Studio®



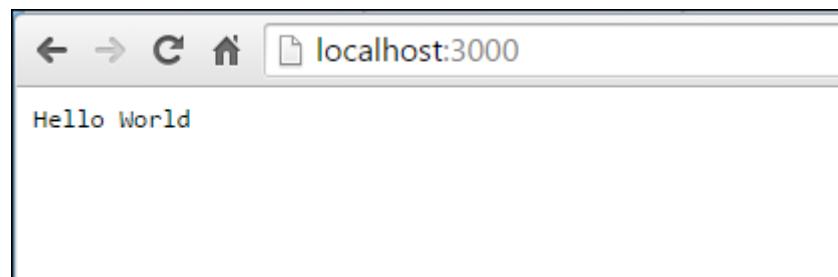
Node Hello World (simple server)

```
// THE most simple NodeJS web server
var http = require('http');

var myServer = http.createServer(function (request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.write('Hello World')
    response.end();
});

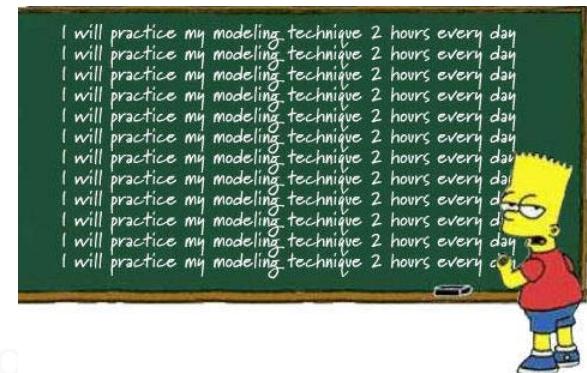
myServer.listen(3000);
console.log('Server running at http://localhost:3000');
```

```
> node 02_simple_server.js
```



Workshop

- Create your own simple web server (like in the previous slide)
- Use the property `request.url`. Show the requested URL in the browser
- Create a JSON-object in the file. Return the JSON-object and show it in the browser.
 - Make sure the correct Content-Type is used
- Create (at least) 2 JSON-objects and return one object or the other, depending on the URL (
 - i.e. `localhost:3000/person` and `.../company`)
- Return a 404-object for all other URLs





Modules en packages

Modulaire code schrijven

Wat is een module

"Een module is een bestand waaruit functionaliteit wordt geëxporteerd"

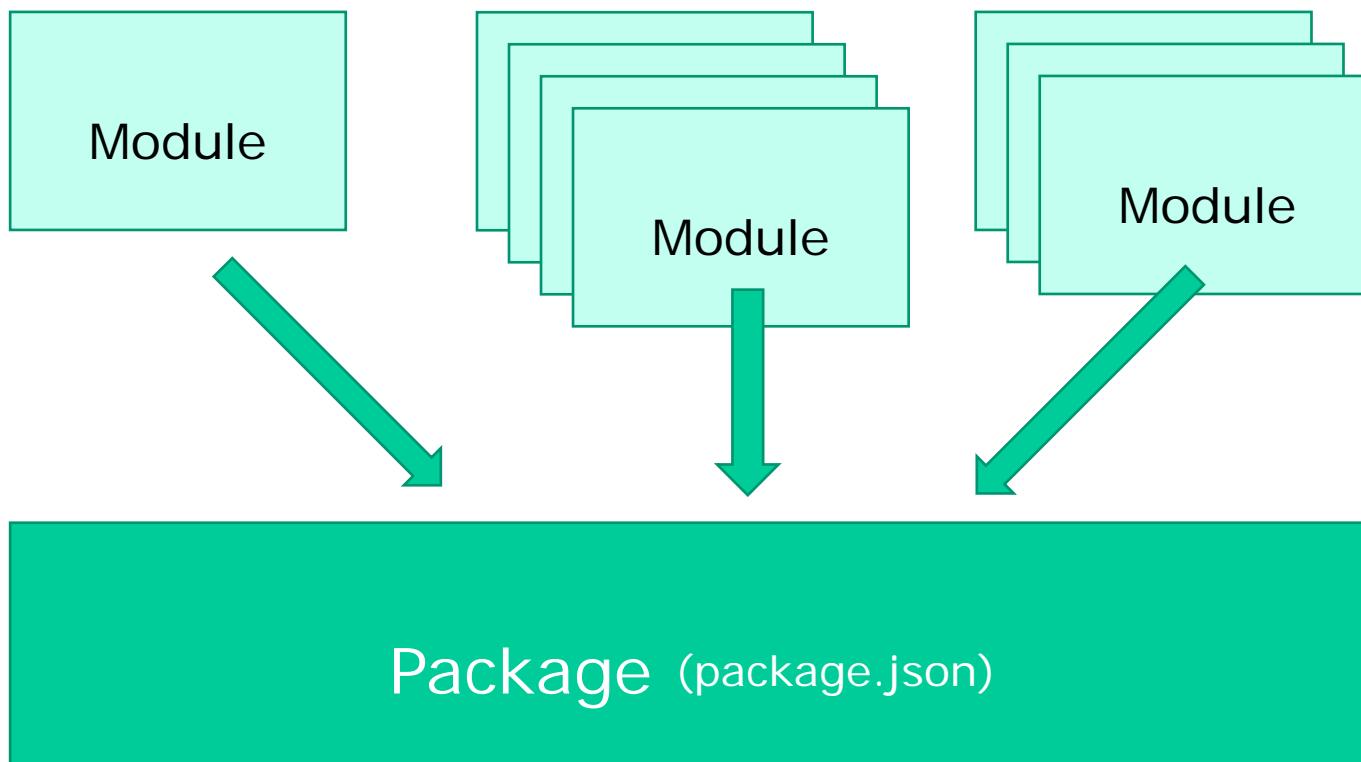
Functie(s), variabele(n), object(en), etc.

Anders dan in ES5/JavaScript in de browser:

- elke module heeft *lokale* scope
- geen iffy's meer nodig (`function() { ... } ()`)

Modules en packages

- Meerdere modules tezamen kunnen een *package* vormen



Eenvoudige module schrijven

```
// Logger.js - Exporteer verschillende algemene Logging-methoden
module.exports.log = function(msg){
    console.log('>> Log: ' + msg);
};

module.exports.info = function(msg){
    console.info('>> Info: ' + msg);
};

module.exports.error = function(msg){
    console.error('>> Error: ' + msg);
};
```

Module gebruiken

```
// app.js - Laadt de Logger en test de drie functies.  
var logger = require('./logger');  
  
logger.log('Algemene melding');  
  
logger.info('Dit is een informatieve melding');  
  
logger.error('Dit is een foutmelding');
```

Andere schrijfwijze, zelfde resultaat

```
// Logger2.js - exporteer verschillende algemene Logging-methoden
// binnen een functie
module.exports = function () {
    this.log = function (msg) {
        console.log('>> Log: ' + msg);
    };

    this.info = function (msg) {
        console.info('>> Info: ' + msg);
    };

    this.error = function (msg) {
        console.error('>> Error: ' + msg);
    };

    return this;
};
```

Dan ook anders includen:

```
// app2.js - Logger.js exporteert nu een class met verschillende methods.  
var logger = require('./logger2');  
  
var myLog = logger(); // functie invoking  
  
myLog.info('Dit is informatie');  
  
myLog.error('Dit is een foutmelding');
```

```
nodemon app2.js  
11 Dec 14:31:41 - [nodemon] v1.3.7  
11 Dec 14:31:41 - [nodemon] to restart at any time, enter `rs`  
11 Dec 14:31:41 - [nodemon] watching: `.*`  
11 Dec 14:31:41 - [nodemon] starting `node app2.js`  
>> Info: Dit is informatie  
>> Error: Dit is een foutmelding  
Algemene logger  
11 Dec 14:31:41 - [nodemon] clean exit - waiting for changes before restart
```

Modules laden in modules

Gebruik require(...)

```
// getTime.js - module een geformateerde tijd (hh:mm:ss) teruggeeft.  
module.exports = function () {  
    var now = new Date();  
    var tijd = now.getHours() + ':' + now.getMinutes() + ':' + now.getSeconds();  
    return tijd;  
};
```

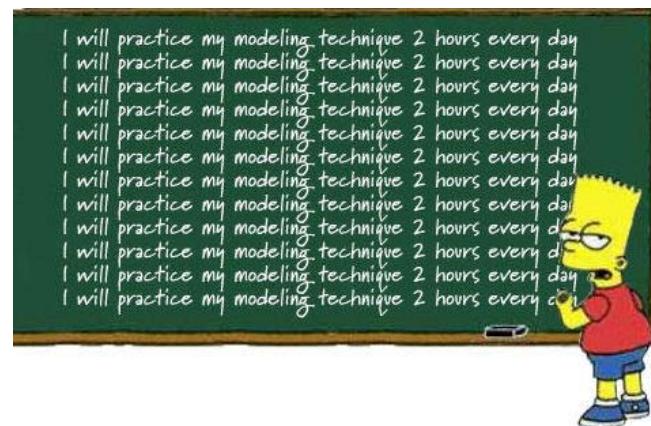
```
// Require een andere module  
var time = require('./getTime');  
// Exporteer verschillende algemene Logging-methoden  
module.exports = function () {  
    this.log = function (msg) {  
        console.log(time() + ', Log >>' + msg);  
    };  
    ...  
    return this;  
};
```

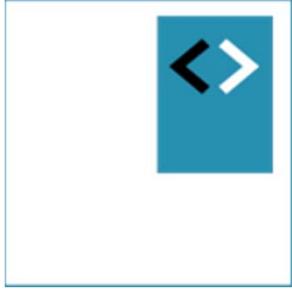
Checkpoint

- Modules zijn JavaScript-bestanden
- Modules hebben *by default* local scope
- Modules worden ingevoegd met `require('...')`
- De extensie `.js` wordt by convention niet gebruikt.

Workshop

- Create a new module, for instance, showing the current date
- Import the module in a new file
- Use the imported module, show the current date in the console from within the new file.
- See previous slide(s) for example





Using NPM

Using NPM to manage your packages

Werken met npm

- Gebruik maken van inspanningen van anderen
 - Wiel niet opnieuw uitvinden
- Installeer via npm packages/modules met gewenste functionaliteit
- Registreer in package.json met flag --save
- Stap 1 : package.json instellen met projectbeschrijving

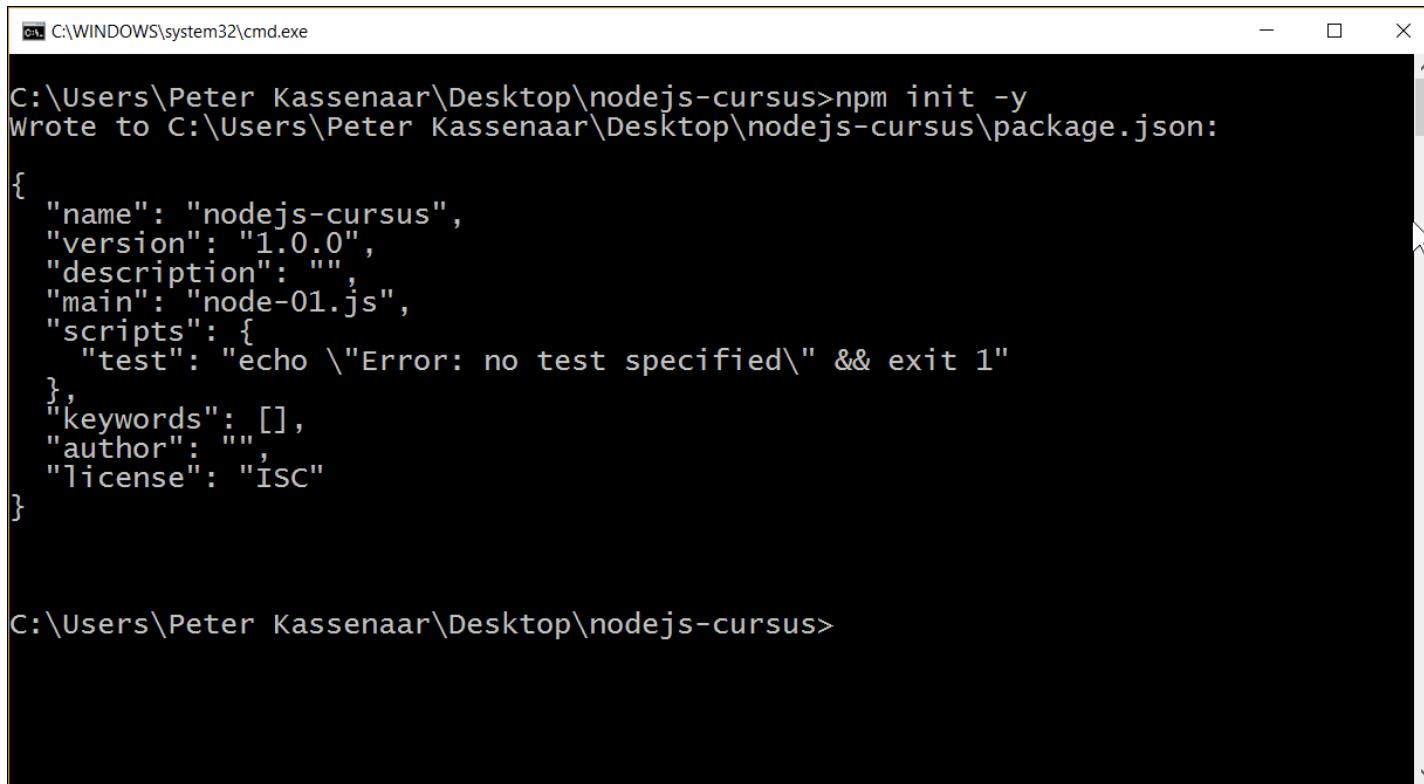
npm init

Typ npm init en beantwoord de vragen

```
{  
  "name": "simple_server",  
  "version": "0.0.0",  
  "description": "Eenvoudige webserver met Express - stap 1",  
  "main": "server.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [  
    "test",  
    "express",  
    "module"  
  ],  
  "author": "Peter Kassenaar",  
  "license": "ISC"  
}
```

Package.json

Shortcut voor standaardwaarden: npm init -y



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Peter Kassenaar\Desktop\nodejs-cursus>npm init -y
Wrote to C:\Users\Peter Kassenaar\Desktop\nodejs-cursus\package.json:
{
  "name": "nodejs-cursus",
  "version": "1.0.0",
  "description": "",
  "main": "node-01.js",
  "scripts": {
    "test": "echo \\"Error: no test specified\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

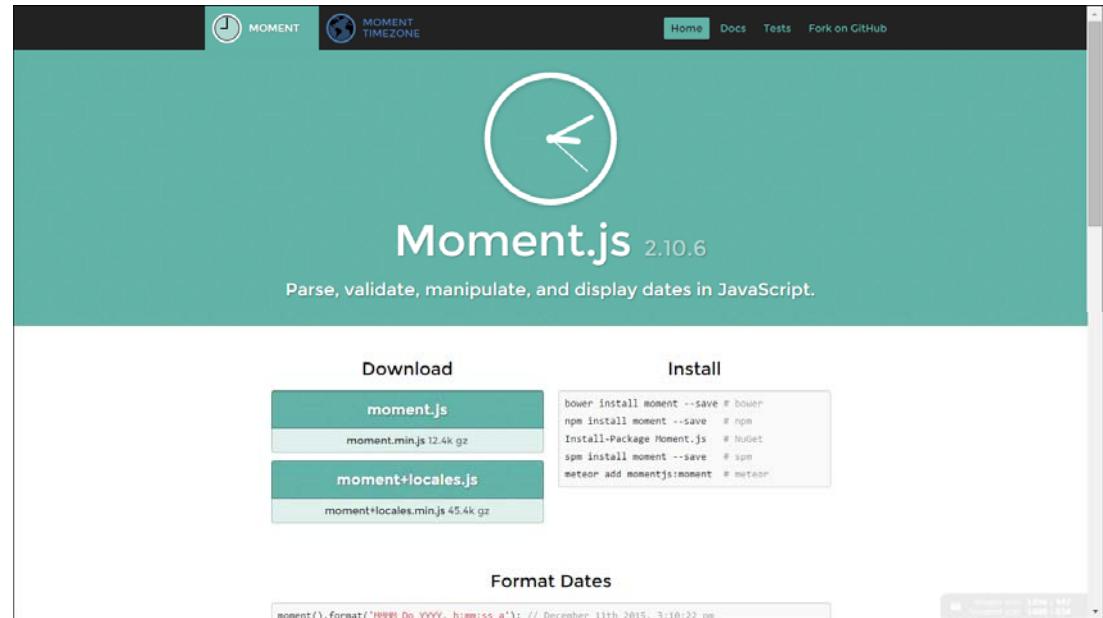
C:\Users\Peter Kassenaar\Desktop\nodejs-cursus>
```

Voorbeeld: de module moment

Werken met datums en tijden

npm install moment OF

npm install moment --save



The screenshot shows the official Moment.js website. At the top, there are navigation links for 'MOMENT' and 'MOMENT TIMEZONE', along with 'Home', 'Docs', 'Tests', and 'Fork on GitHub'. The main header features a large circular arrow icon and the text 'Moment.js 2.10.6'. Below the header, a subtitle reads 'Parse, validate, manipulate, and display dates in JavaScript.' In the center, there are download links for 'moment.js' and 'moment+locales.js', each with its file size: 'moment.min.js 12.4k gz' and 'moment+locales.min.js 45.4k gz'. To the right, there is an 'Install' section with command-line installation instructions for various package managers. At the bottom, there is a 'Format Dates' input field containing the code 'moment().format('YYYY Do YYYY, h:mm:ss a'); // December 11th 2015, 3:10:22 pm'.

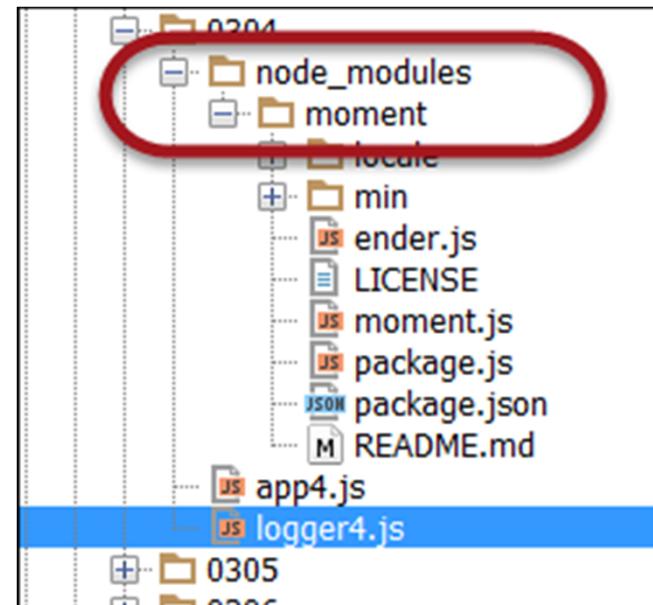
<http://momentjs.com>

Code voor moment

```
// Require de module 'moment'. Let op ontbreken van de pad-informatie.  
// Node.js kijkt dan standaard in \node_modules  
var moment = require('moment');  
module.exports = function () {  
    moment.locale('nl');    // Stel Nederlandse notatie in  
    this.log = function (msg) {  
        console.log(moment().format('lll') + ', Log >> ' + msg);  
    };  
    ...  
    return this;  
};
```

De map node_modules

- Elke package krijgt eigen entry in node_modules
- Ga hier niet handmatig in rommelen
- Npm onderhoudt dependencies



Populaire packages:

- colors
- underscore / lodash
- express
- body-parser
- grunt / gulp
- cordova
- typescript
- less / sass
- ...duizenden...

Node-app instellen

- Draai eerst `npm install` om dependencies te installeren
- Bekende flags:
 - `--save` - installeer required runtime dependency
 - angular/bootstrap/moment/etc
 - `--save-dev` – installeer development dependency
 - typescript/sass/gulp/etc

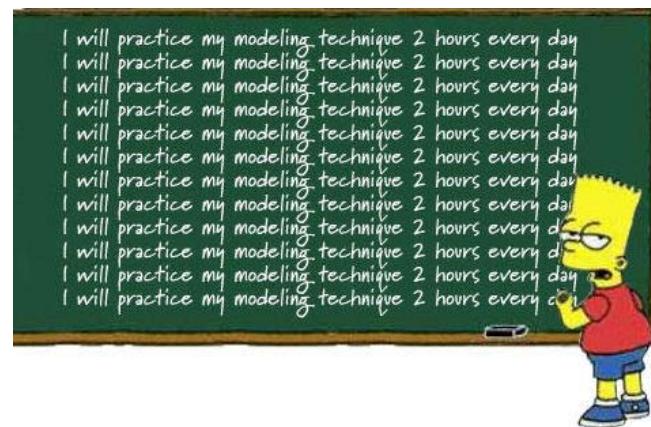
Sinds NPM 5.0.0:

- Niet meer nodig om de flag `--save` toe te voegen.
- Veel developers doen het uit gewoonte nog wel.
- Packages worden *by default* toegevoegd aan `package.json` als ze worden geinstalleerd.
- Zie ook <https://stackoverflow.com/questions/19578796/what-is-the-save-option-for-npm-install>
- Meer achtergrondinfo bij NPM install:
<https://docs.npmjs.com/cli/install>

Checkpoint

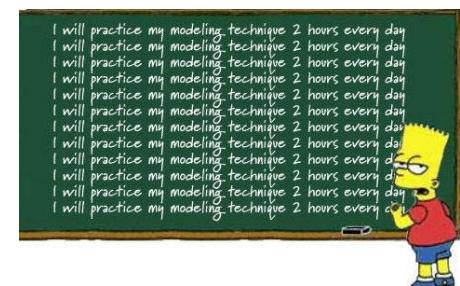
- Met npm installeer je packages van anderen
- Maak een package.json met projectbeschrijving
- Gebruik --save voor runtime dependencies
- Gebruik --save-dev voor development dependencies

Oefening....



Workshop

- Maak een module die het kwadraat van een getal uitrekent
- Sluit de module in met `require(...)`
- Voer hem uit in `app.js`:
 - `console.log(kwadraat(5)); // output: 25`
- Maak een module `Persoon`. Nieuwe personen krijgen properties voornaam, achternaam en email
- Push minimaal 3 personen naar de array
- Toon de array in de console
- Installeer de package `underscore`. Toon hiermee de eerste persoon in de array, de laatste persoon, en sorteer de array (`.sortBy`)
 - Zie eventueel underscorejs.org



Node best practices

The screenshot shows a GitHub repository page for 'Node.js Best Practices'. At the top left is a 'README.md' file icon. Below it, the title 'Node.js Best Practices' is displayed. The main feature is a large, bold, green title 'NODE.JS BEST PRACTICES'. At the bottom, there are three status indicators: 'Item count 86 Best Practices', 'Last update March 12 2020', and 'Updated For Version Node 12.12.0'. A 'Follow us on Twitter! @nodepractices' link is also present.

<https://github.com/goldbergonyi/nodebestpractices#readme>

Node.js – Peter Kassenaar