



Webscripting

# Introductie

## DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](http://www.pxl.be/facebook)



# Doelstellingen

- De student kan een webpagina dynamisch maken met JavaScript (JS) en het Document Object Model (DOM) ,d.m.v. event handling en DOM manipulatie.
- De student kan het OO principe toepassen in een oplossing met JavaScript.
- De student kan een REST API aanspreken gebruik makend van een opgelegde API.
- De student kan JSON integreren in JavaScript toepassingen.



# Leerinhouden

- Inleiding tot functioneel programmeren in JavaScript.
- Event handling voor events die opgeroepen worden vanuit html.
- DOM en Event handling ( W3C scripting ) in script.
- OO programmeren met Javascript + Event handling.
- Rest API aanspreken en integreren van JSON in JavaScript.



# Afspraken

- Online les
- 2 lesblokken van 2 uur.
- Demo's en interactief oefeningen maken.



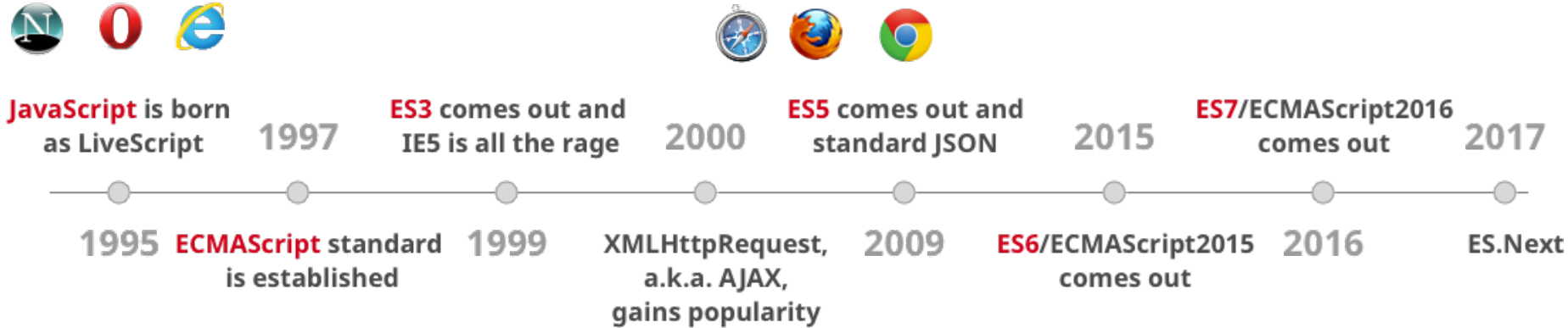
# Evaluatie

- Examen in juni 100% praktijk.
- Schriftelijk open boek laptop examen.
  - => geen copy/paste, tonen dat leerstof beheerst wordt.
  - => enkel werkende code wordt geëvalueerd.

Leertip: veel oefeningen maken!



# Wat is javascript?



- interpreted
  - ↔ C (compiled)
- dynamic typed
  - type checking at runtime
  - ↔ Java (static typed: type checking tijdens compilation)
- weakly typed
  - datatypes mogen door elkaar gebruikt worden:
  - implicit conversion `1+"a" → "1"+"a"`
  - ↔ Python (strong typed, explicit conversion: `str(1)+"a"`)
- ECMAScript standaard



# Installatie

## Webstorm

<https://www.jetbrains.com/webstorm/>  
(student licence)



## Node.js

<https://nodejs.org/en/download/>  
(extra: <https://nodejs.org/en/download/package-manager/>)



## Chrome

<https://www.google.com/chrome/>



# JS in browser (client side scripting)

*index.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>index.html</title>
</head>
<body>
  <script src="demo.js"></script>
</body>
</html>
```

*demo.js*

```
console.log("demo");
// print demo in de console
```





# JS in browser (client side scripting)

[Home](#)[Blog](#)[Docs](#)

## What is V8?

V8 is Google's open source high-performance JavaScript and WebAssembly engine, written in C++. It is used in Google Chrome, the open source browser from Google, and in Node.js, among others. It implements [ECMAScript](#) and [WebAssembly](#), and runs on Windows 7 or later, macOS 10.12+, and Linux systems that use x64, IA-32, ARM, or MIPS processors. V8 can run standalone, or can be embedded into any C++ application.

<https://v8.dev/>



# Node.js



Cross-platform run-time environment

Gebaseerd op V8-engine



Javascript buiten de browser

bv. CLI-application, server-sided scripting

npm: node package manager  
dependencies installeren



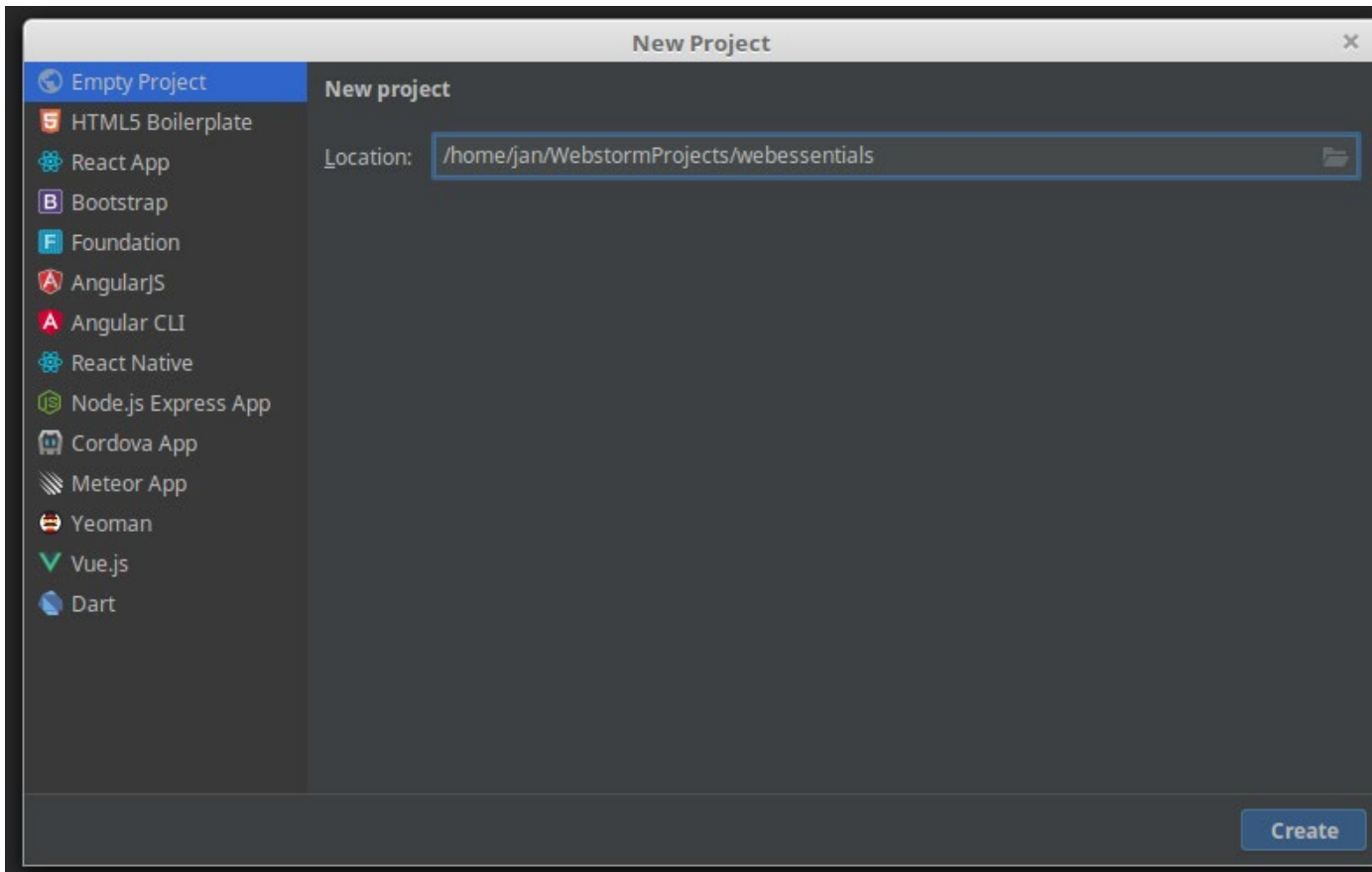
nvm: node version manager



# Webstorm

File > New > Project

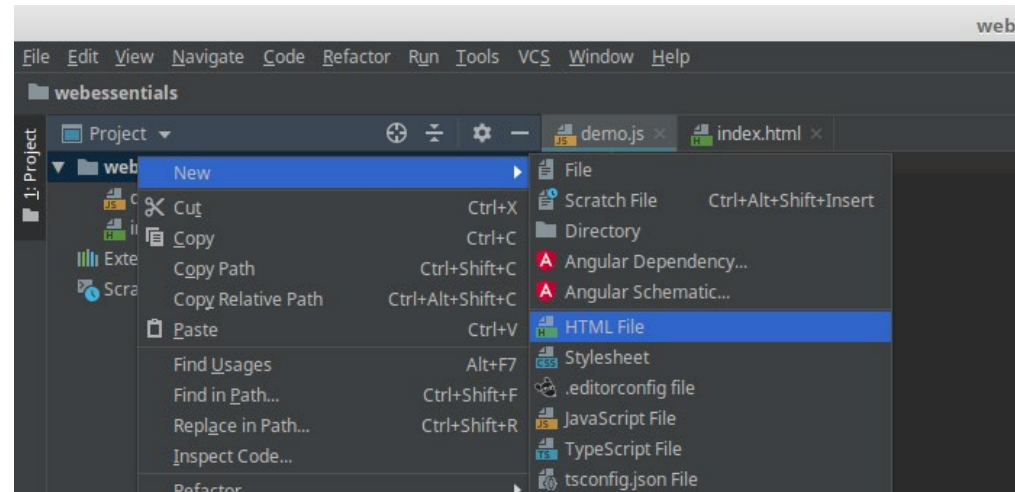
Kies voor Empty project



# Webstorm

RMK op project > New > HTML File

(RMK = rechtermuisknop)



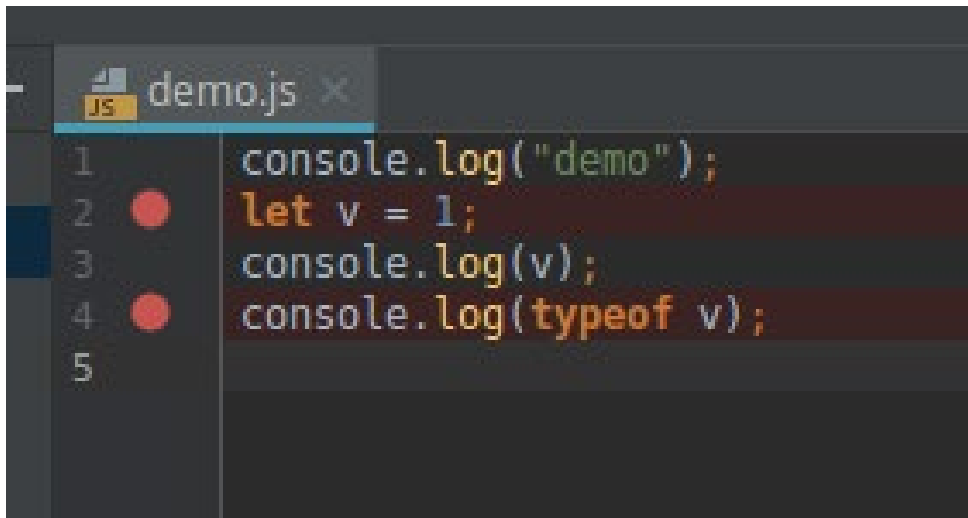
*index.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>index.html</title>
</head>
<body>
  <script src="demo.js"></script>
</body>
</html>
```

# Webstorm

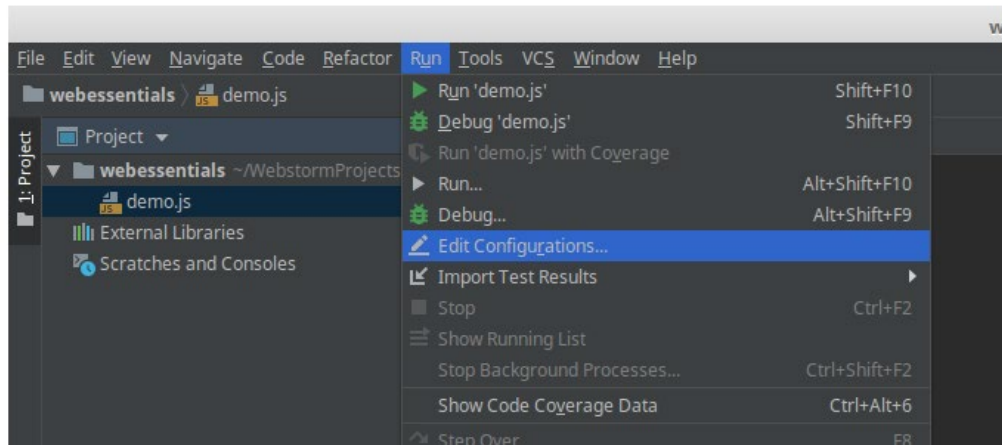
RMK op project > New > Javascript File

Plaats breakpoints door te klikken in de marge van demo.js

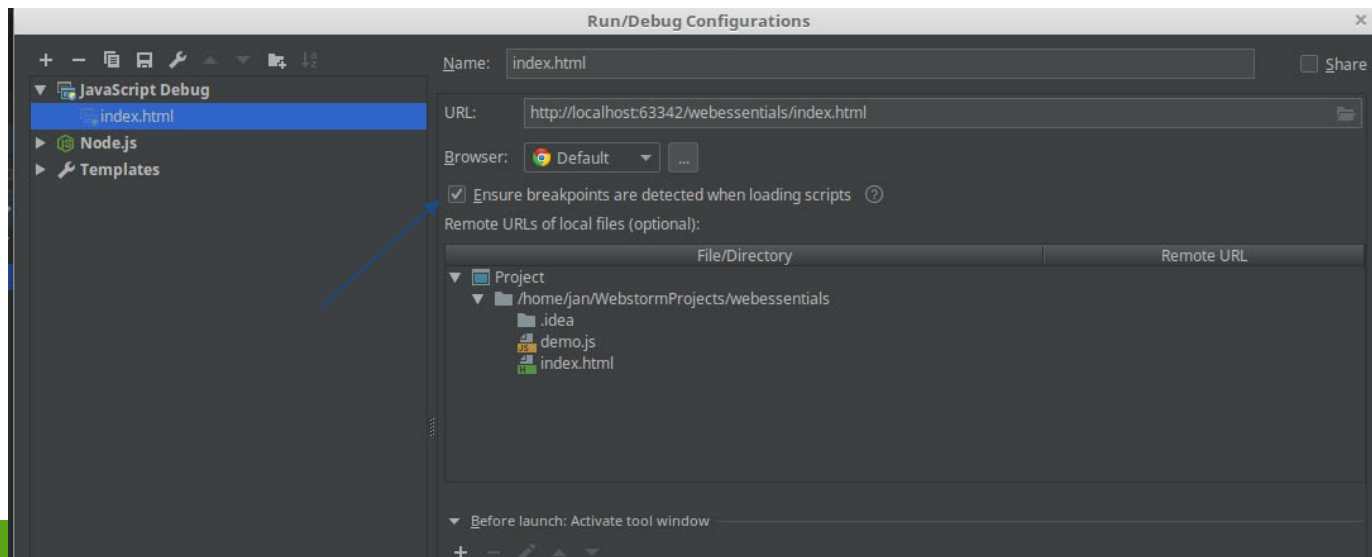


# Webstorm (JS in browser)

Run > Edit Configurations



Click op + en kies JavaScript Debug



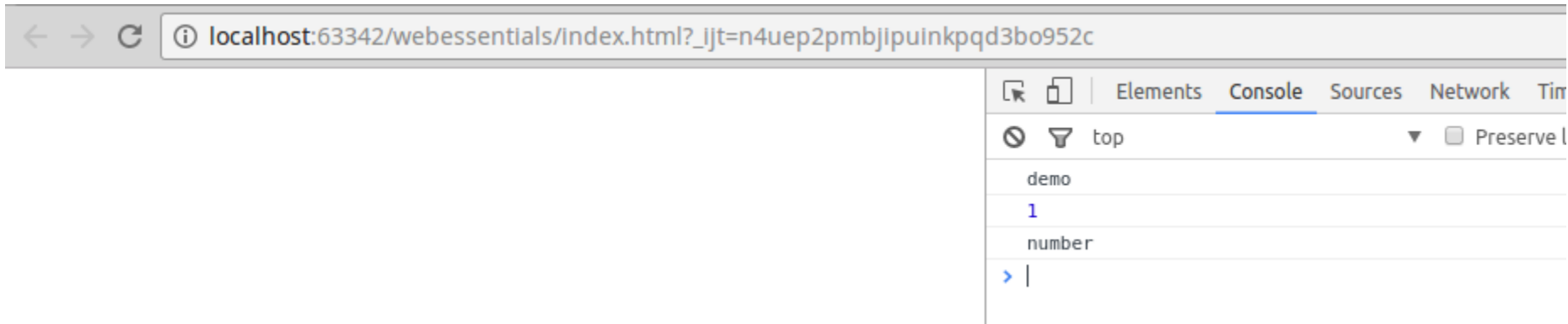
# Webstorm (JS in browser)

Run > Run 'index.html'

Console (in Chrome)

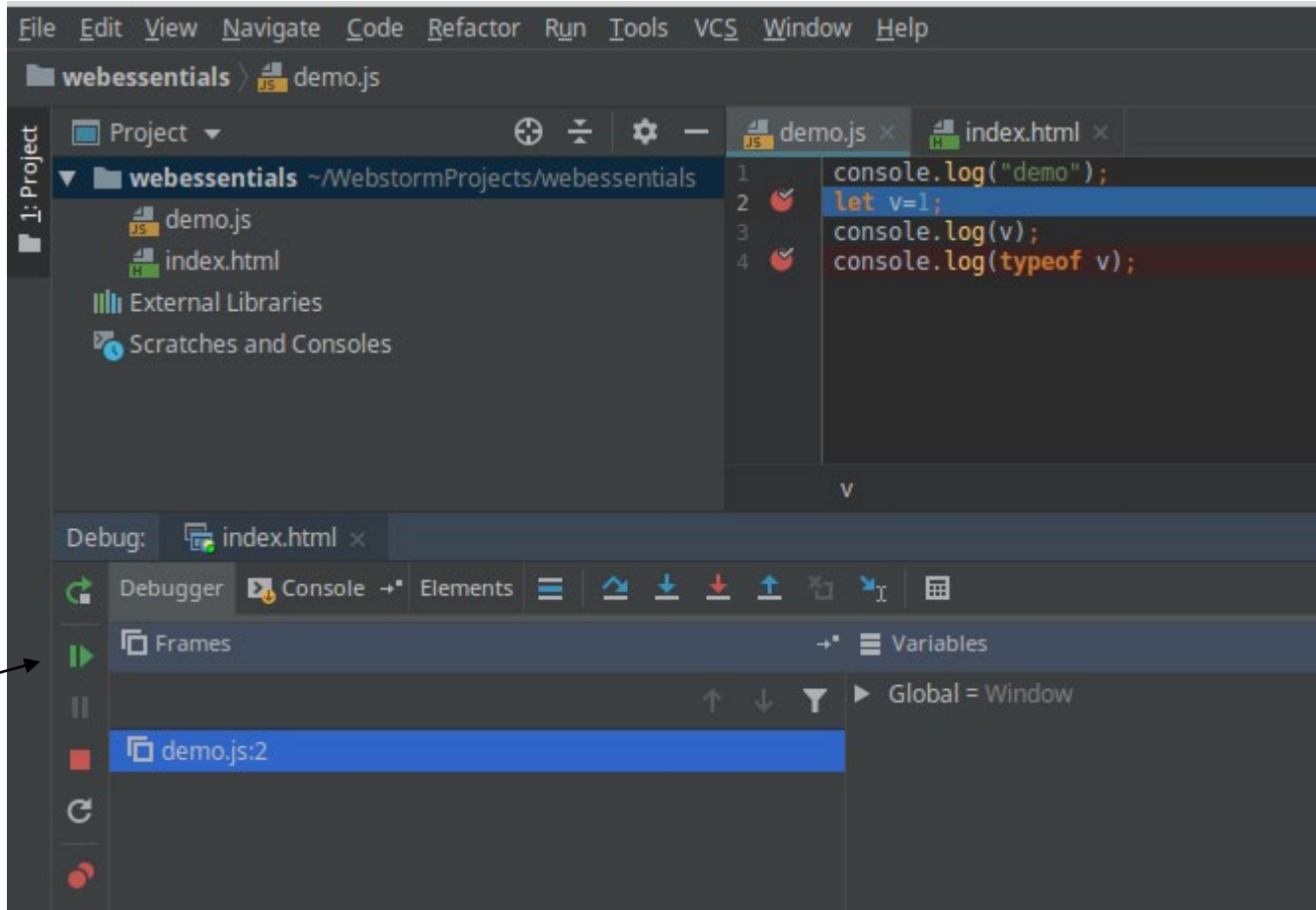
- ctrl-shift-i (or F12)
- klik op Console

15



# Webstorm (JS in browser)

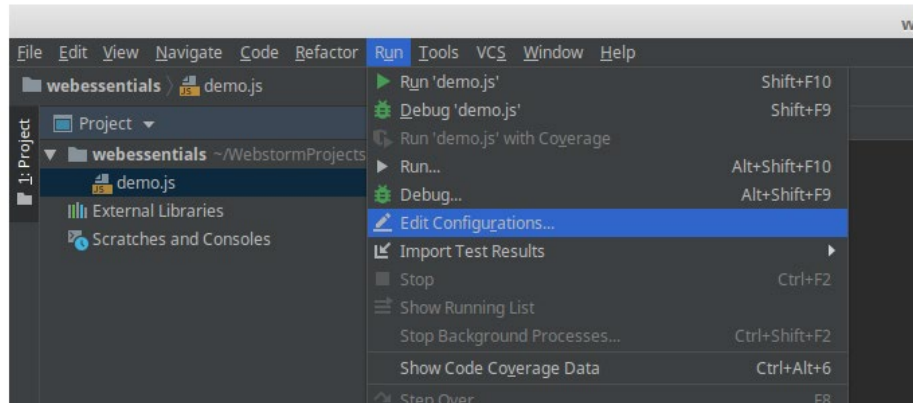
Run > Debug 'index.html'  
(Browser wordt geopend)



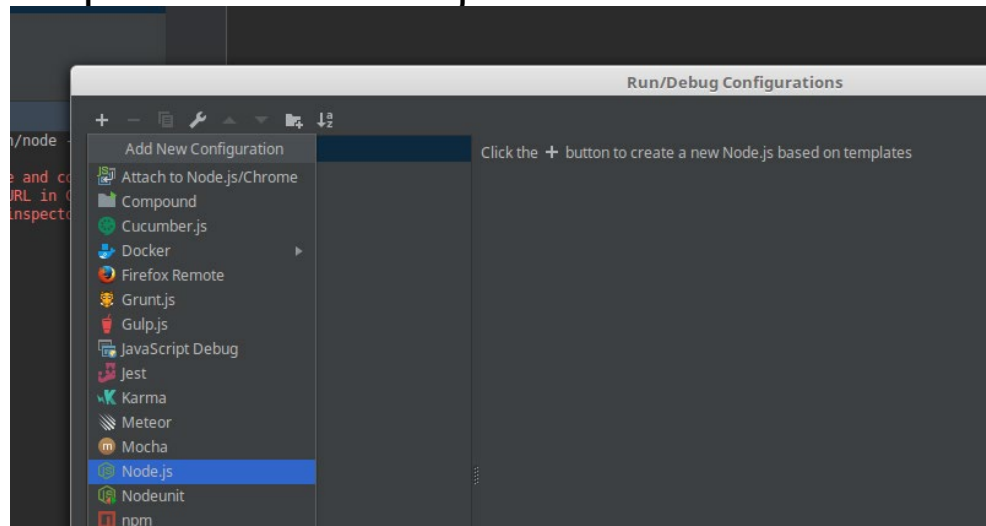


# Webstorm (Node.js)

Run > Edit Configurations

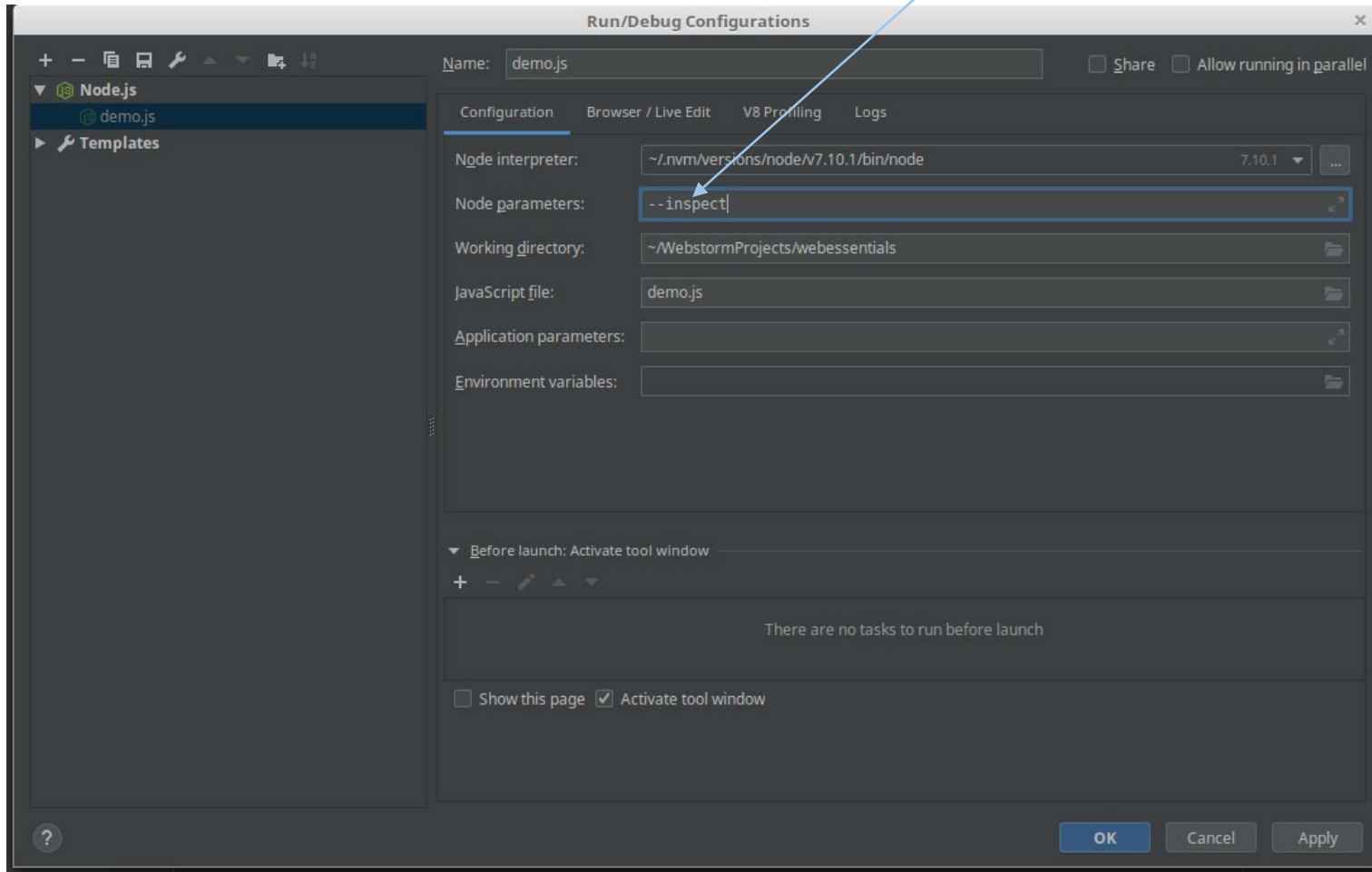


Click op + en kies Node.js



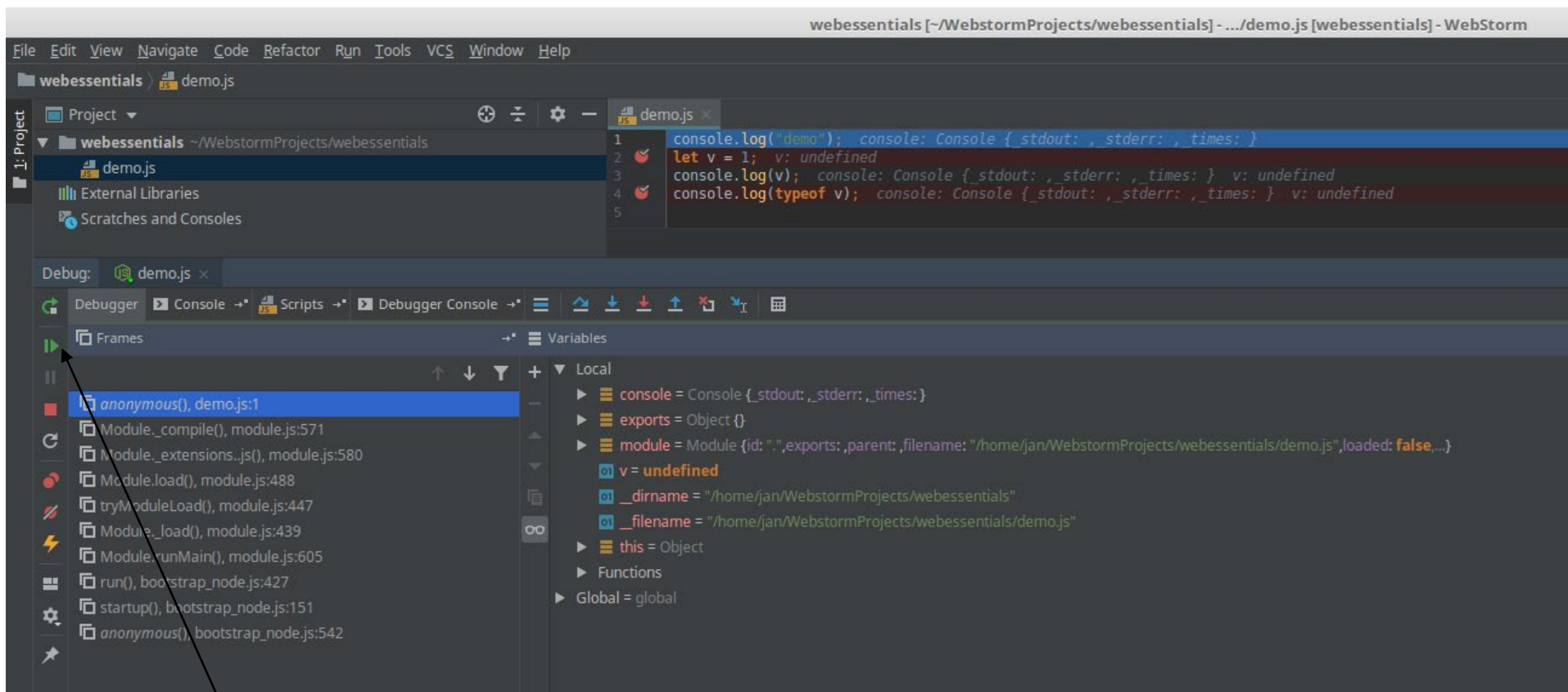
# Webstorm (Node.js)

--inspect



Run > Debug demo.js





Resume (F9)  
Ga naar het volgende breakpoint

