Webscripting

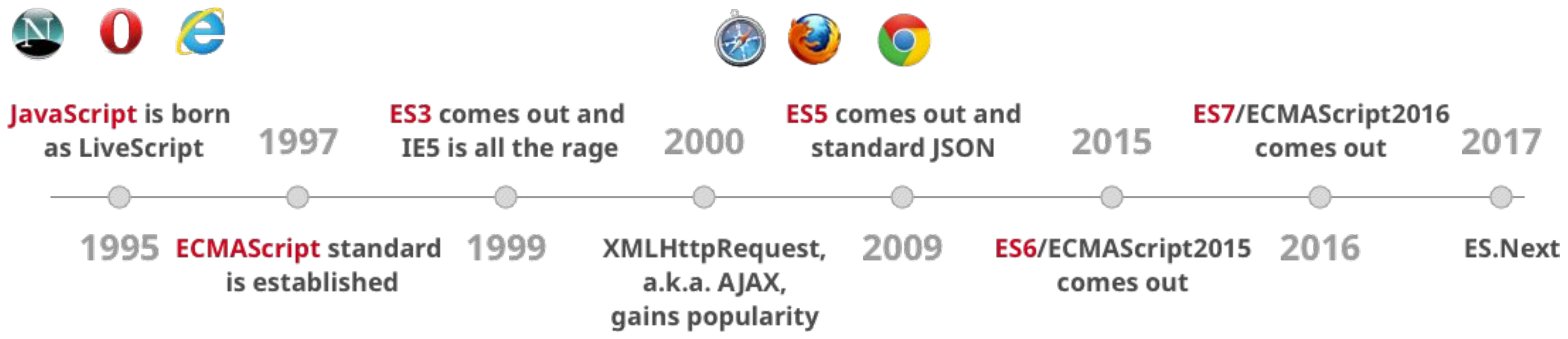
Hoofdstuk 1

Values, types & operators

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Wat is javascript?


JavaScript is born as LiveScript
1995
ECMAScript standard is established
1997
ES3 comes out and IE5 is all the rage
1999
2000
XMLHttpRequest, a.k.a. AJAX, gains popularity
ES5 comes out and standard JSON
2009
2015
ES6/ECMAScript2015 comes out
ES7/ECMAScript2016 comes out
2016
2017
ES.Next

- 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/

Jetbrains IDE support

https://chrome.google.com/webstore/detail/jetbrains-ide-support/hmhgeddbohgjknpmjagkdomcpobmllji

JetBrains IDE Support

Offered by: www.jetbrains.com

(1) JS in browser (client side scripting)

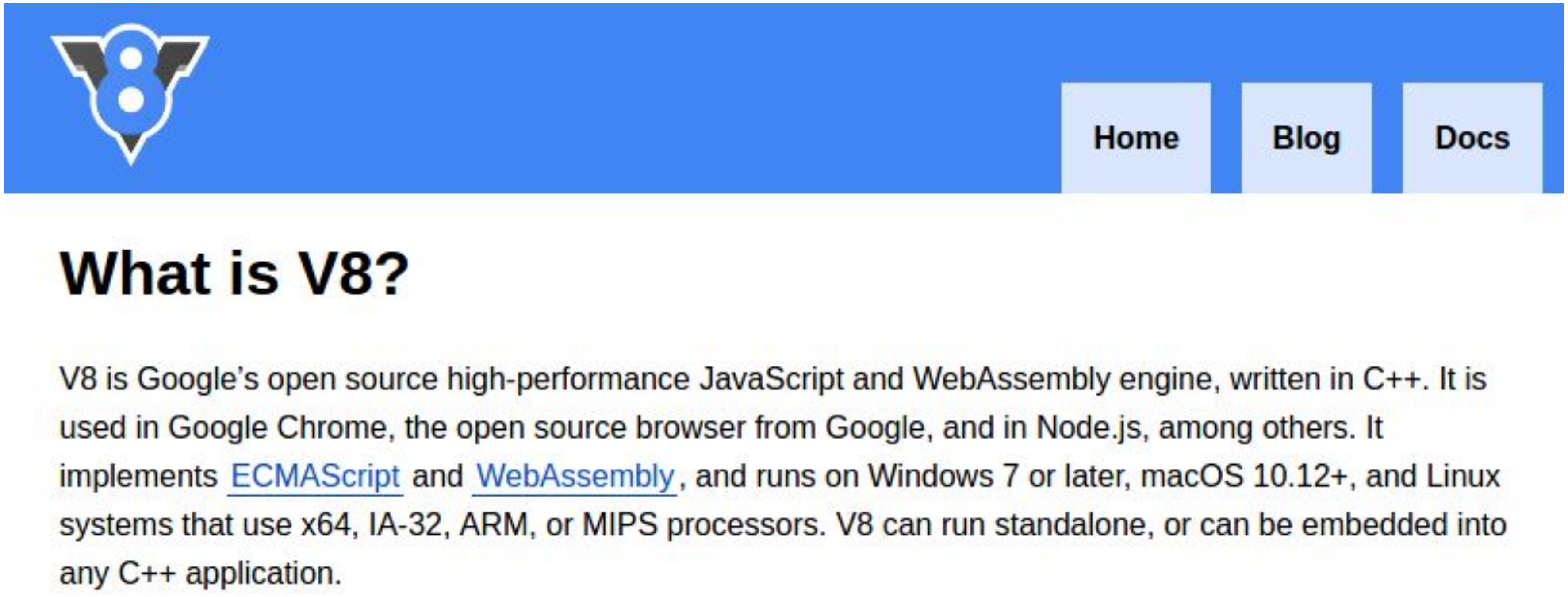
index.html

```
<!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
```

(1) JS in browser (client side scripting)



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/

(2) Node.js

Cross-platform run-time environment

Gebaseerd op V8-engine

Javascript buiten de browser
    bv. CLI-application, server-sided scripting

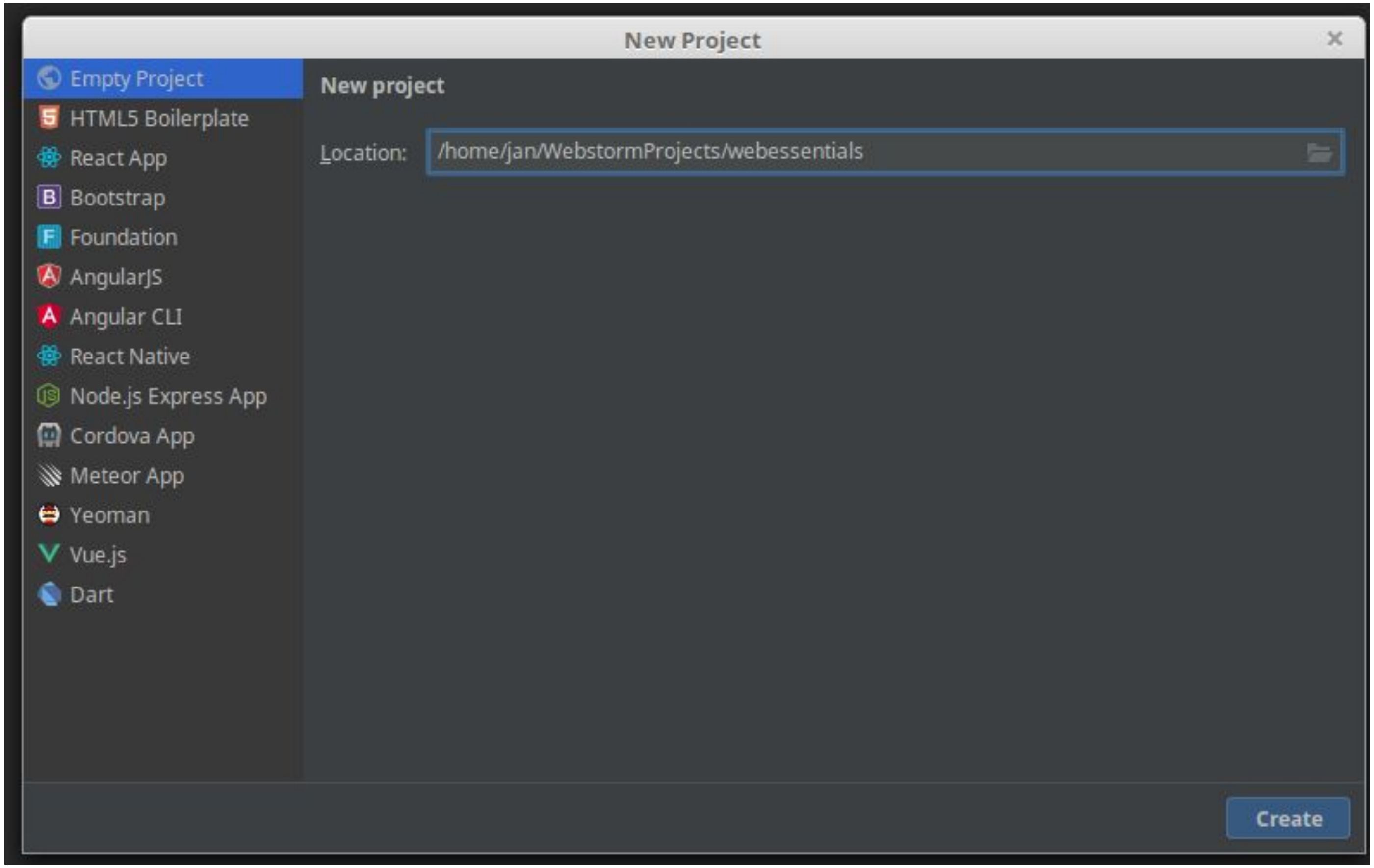
npm: node package manager
    dependencies installeren

nvm: node version manager

Webstorm

File > New > Project

Kies voor Empty project



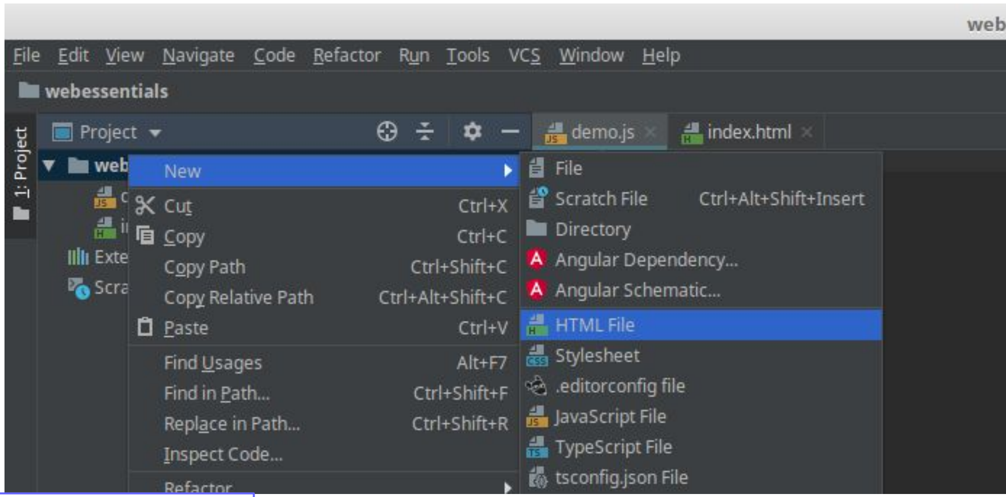
New project

Location: /home/jan/WebstormProjects/webessentials

Webstorm

RMK op project > New > HTML File

(RMK = rechtermuisknop)



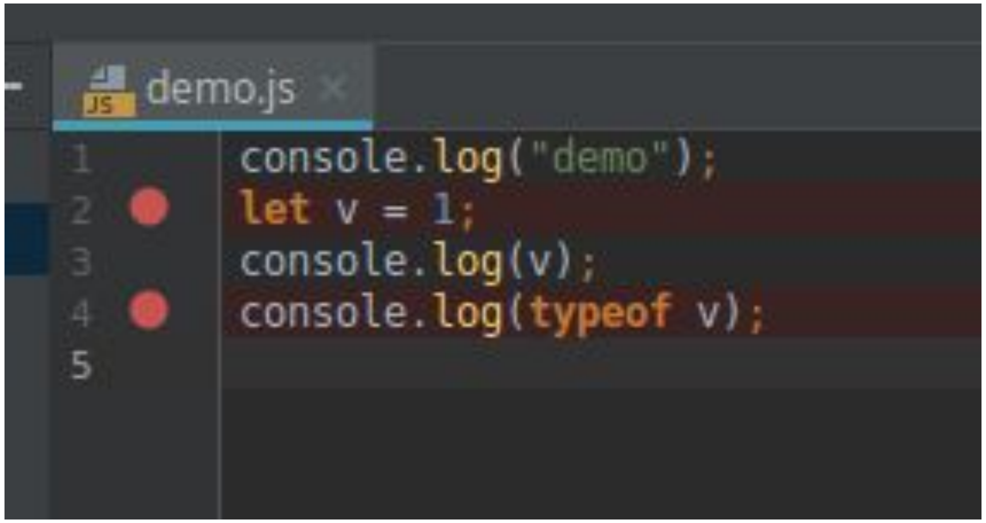
index.html

```
<!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

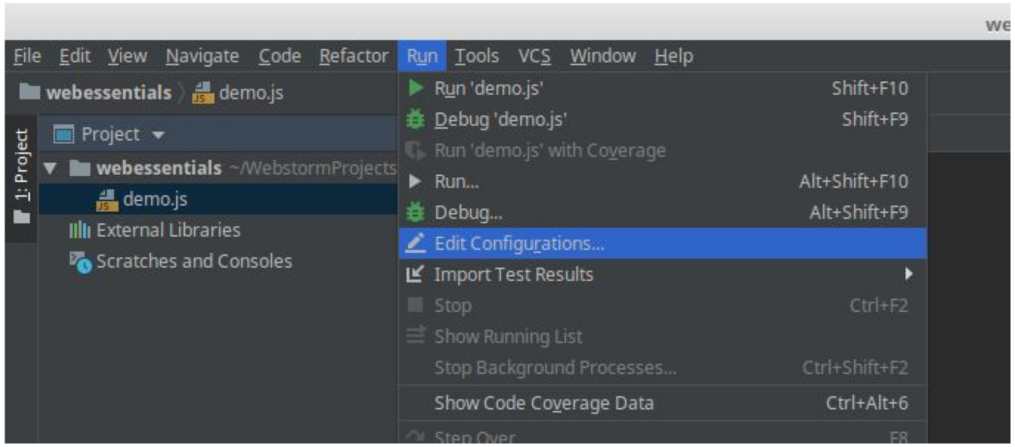
Plaats breakpoints door te klikken in de marge van demo.js



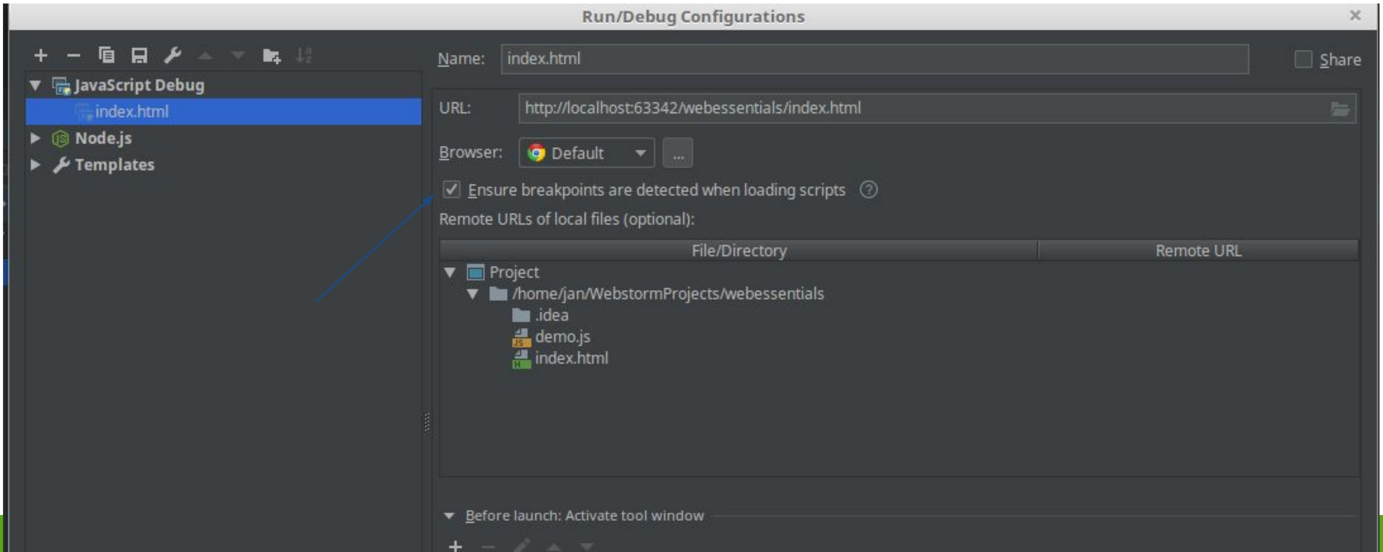
```
console.log("demo");
let v = 1;
console.log(v);
console.log(typeof v);
```

Webstorm (JS in browser)

Run > Edit Configurations



Click op + en kies JavaScript Debug



Name: index.html

URL: http://localhost:63342/webessentials/index.html

Browser: Default

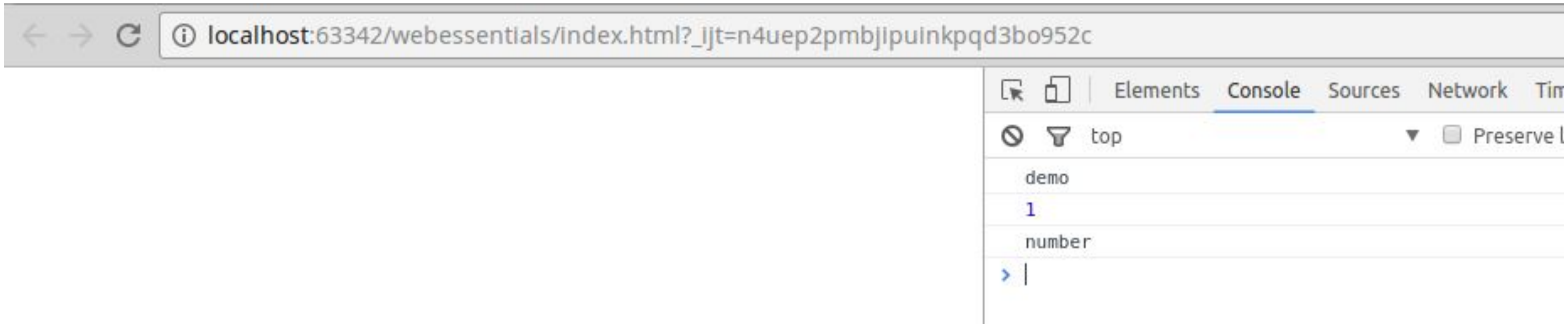
☑ Ensure breakpoints are detected when loading scripts

Remote URLs of local files (optional):

Webstorm (JS in browser)

Run > Run 'index.html'

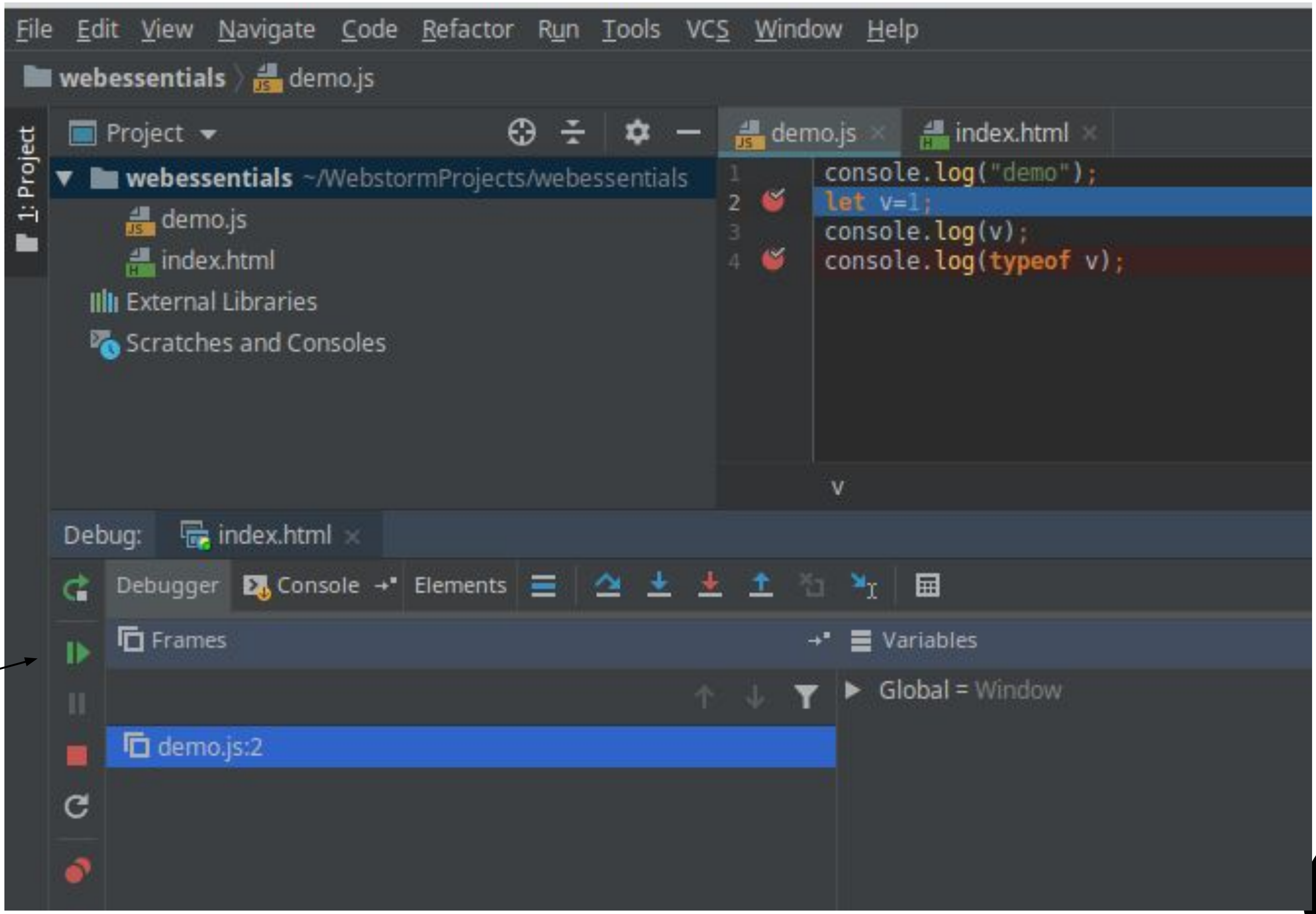
Console (in Chrome)
- ctrl-shift-i
- klik op Console


localhost:63342/webessentials/index.html?_ijt=n4uep2pmbjipuinkpqd3bo952c
Elements
Console
Sources
Network
Tim
top
Preserve l

```
demo
1
number
>
```

Webstorm (JS in browser)

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

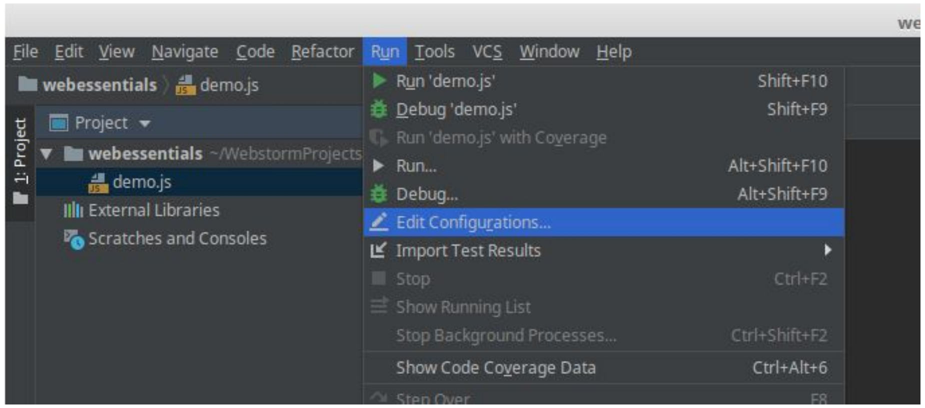


```
console.log("demo");
let v=1;
console.log(v);
console.log(typeof v);
```

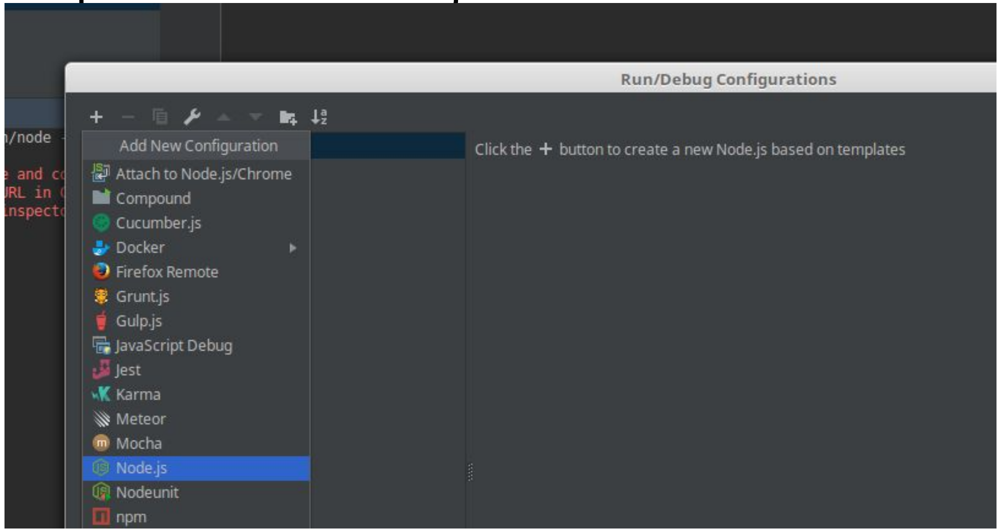
Resume

Webstorm (Node.js)

Run > Edit Configurations



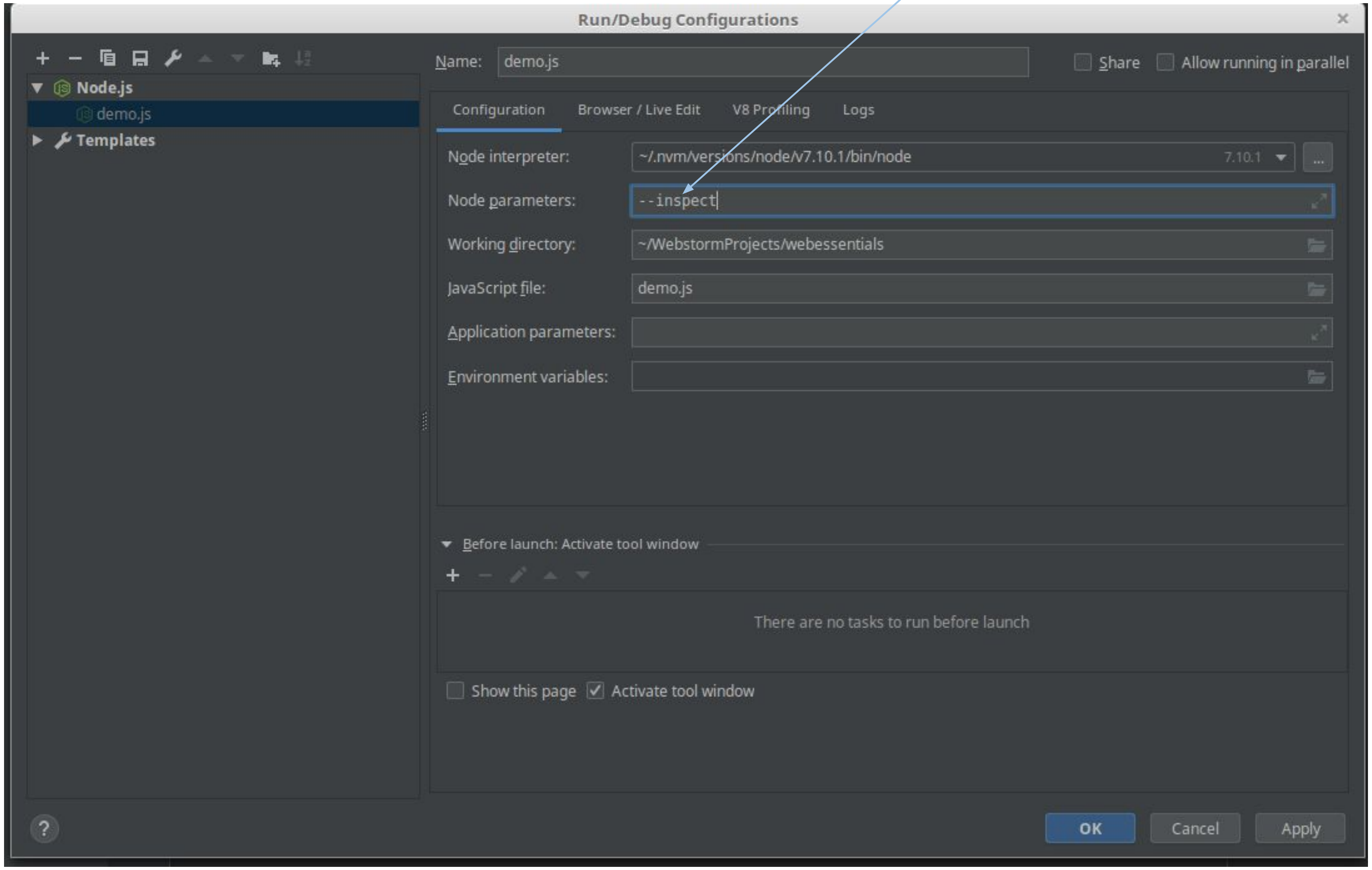
Click op + en kies Node.js



Click the + button to create a new Node.js based on templates

Webstorm (Node.js)

--inspect


Run/Debug Configurations

Name: demo.js

☐ Share ☐ Allow running in parallel

Node.js
demo.js
Templates

Configuration | Browser / Live Edit | V8 Profiling | Logs

Node interpreter: ~/.nvm/versions/node/v7.10.1/bin/node 7.10.1

Node parameters: --inspect

Working directory: ~/WebstormProjects/webessentials

JavaScript file: demo.js

Application parameters:

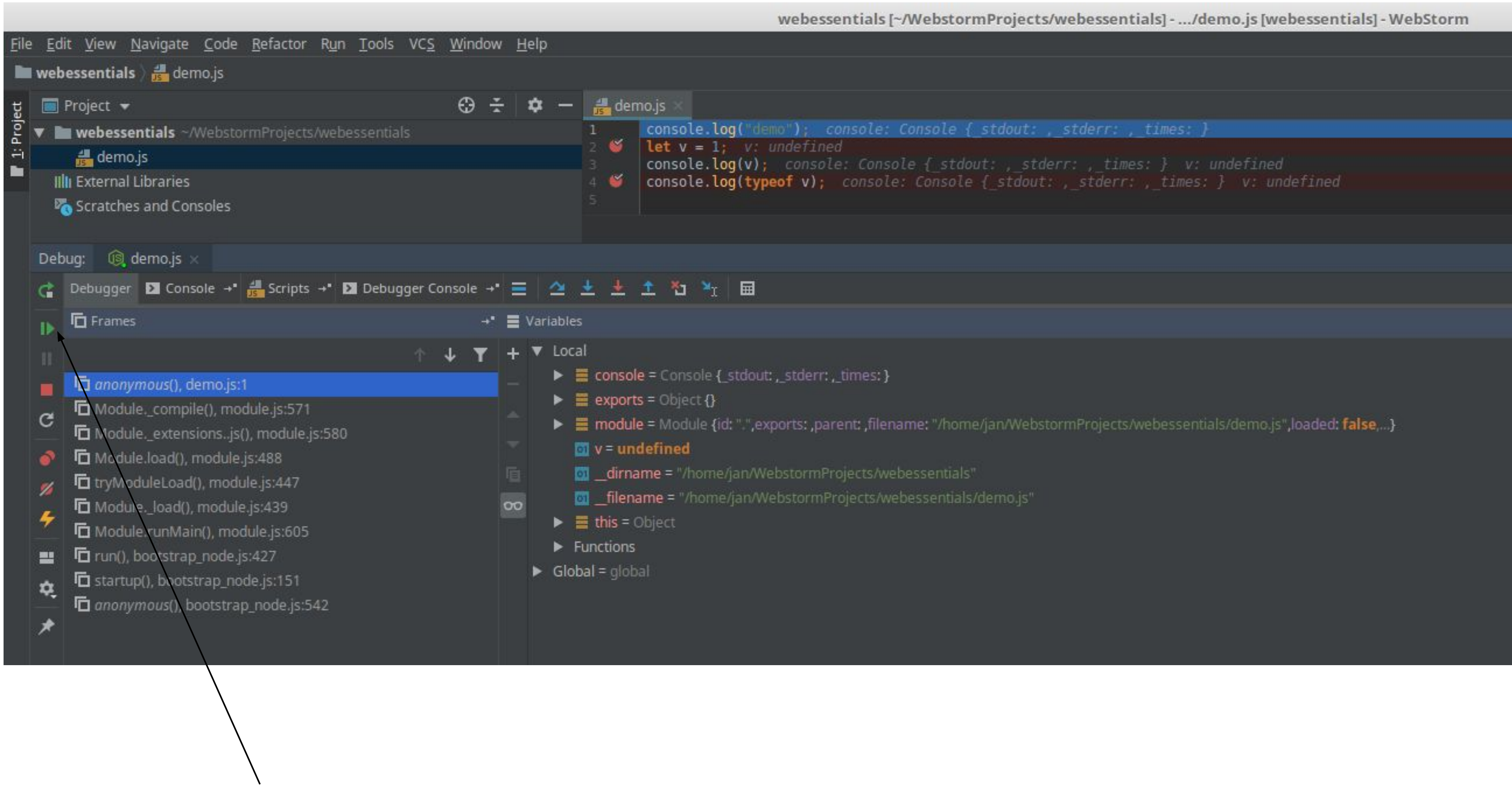
Environment variables:

Before launch: Activate tool window

There are no tasks to run before launch

☐ Show this page ☑ Activate tool window

Run > Debug demo.js


webessentials [~/WebstormProjects/webessentials] - .../demo.js [webessentials] - WebStorm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
webessentials demo.js
Project
webessentials ~/WebstormProjects/webessentials
demo.js
External Libraries
Scratches and Consoles
Debug: demo.js
Debugger Console Scripts Debugger Console
Frames
Variables

```
console.log("demo");  console: Console { _stdout: , _stderr: , _times: }
let v = 1;  v: undefined
console.log(v);  console: Console { _stdout: , _stderr: , _times: }  v: undefined
console.log(typeof v);  console: Console { _stdout: , _stderr: , _times: }  v: undefined
```

```
anonymous(), demo.js:1
Module._compile(), module.js:571
Module._extensions..js(), module.js:580
Module.load(), module.js:488
tryModuleLoad(), module.js:447
Module._load(), module.js:439
Module.runMain(), module.js:605
run(), bootstrap_node.js:427
startup(), bootstrap_node.js:151
anonymous(), bootstrap_node.js:542
```

```
▼ Local
  ▶ console = Console {_stdout: ,_stderr: ,_times: }
  ▶ exports = Object {}
  ▶ module = Module {id: ".",exports: ,parent: ,filename: "/home/jan/WebstormProjects/webessentials/demo.js",loaded: false,...}
    v = undefined
    __dirname = "/home/jan/WebstormProjects/webessentials"
    __filename = "/home/jan/WebstormProjects/webessentials/demo.js"
  ▶ this = Object
  ▶ Functions
▶ Global = global
```

Resume (F9)
Ga naar het volgende breakpoint

Variables / constants

let	variabele

let: variabele

declaratie & initialisatie in 1 keer

```
let aantal = 1;
```

declaratie & initialisatie verspreid

```
let waarde;
waarde = 1;
```

meerdere declaraties / initialisaties op 1 regel

```
let a, b = 1, c;
```

var: variabele

<u>nooit</u> gebruiken

const: constante (niet wijzigbaar na toekenning)

declaratie & initialisatie in 1 keer

```
const een = 1;
```

meerdere declaraties /initialisaties

```
const twee = 2, drie = 2;
```

Types

Primitive datatypes

- number
- string
- boolean
- null
- undefined

Reference datatypes (later)

- array
- function
- object

Types: number

Geen onderscheid tussen komma- en gehele getallen

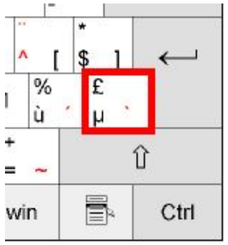
```
let number = 13;
let decimalNumber = 3.14;
let avagadroNumber = 6.0221e23;

let teGrootGetal = 1e1000;  //Infinity

if (teGrootGetal === Infinity) {
   console.log("Infinity!");
}

console.log(1 / teGrootGetal); // 0

console.log(1 / 0); // Infinity
```

Types: number

Bewerkingen: +, -, *, /, %

```
let number = 13;
console.log( number / 2 );  // 6.5
console.log( number % 2 );  // 1
console.log(number * “Jan”) //NaN
```

Types: string



Single & double quotes

```
let welcome = "Hello\nWorld";
```

Backticks

```
let getal=12;
console.log(`${getal} + 1 = ${getal + 1} !`);
// 12 + 1 = 13 !
console.log(`half of 100 is ${100 / 2}`);
// half of 100 is 50
```

Geen char

```
let eersteSymbool = welcome[0];
console.log( eersteSymbool);          //H
console.log(typeof eersteSymbool); // string
```

Concatenatie: +

```
console.log("a"+"bc");   //abc
```

Types: boolean

true, false

```
let goedGekeurd = true;
```

negatie: !

```
if ( !goedGekeurd ) {
```

&&, ||

```
if (name == "tim" && age > 2){
```

Types: undefined

waarde nog niet toegekend

```
let getal;
console.log(typeof getal);  // undefined
```

Types: null

```
let a = null;
```

Wrappers

Primitives: string, number

Reference types: String, Number

```
let name = "Sofie";
console.log(typeof name);      // string

let street = new String("Wetstraat");
console.log(typeof street);    // object

console.log(name.toUpperCase());
   // name wordt omgezet naar String-object
   // toUppercase van String-object wordt
   // uitgevoerd
```

Comparison

```
1 < 2                      // true

1 == 1                     // true

"aardvark" < "zebra"       // true

"a" == "jan"               // false
```

Ternary if / Inline if

```
let age = 19;
let usertype = age < 18 ? 'minor' : 'adult';
```

Type coercion

Bij een operatie tussen één of meerdere datatypes automatische conversie van datatype

```
console.log(8 * null);   // 0 (number)

console.log("5" - 1);    // 4 (number)

console.log("5" + 1);    // 51 (string)

console.log("five" * 2);  // NaN (number)

console.log(false == 0); // true (boolean)

console.log(false === 0); // false (boolean)
```

https://delapouite.com/ramblings/javascript-coercion-rules.html

Type coercion

Truthy / falsy

```
if ( "a" ){
   console.log("de string a is truthy");
}

if ( !null ){
   console.log("null is not truthy");
}
```

http://adripofjavascript.com/blog/drips/truthy-and-falsy-values-in-javascript.html

Type coercion bij && ||

```
let result = a || b;
// als a truthy dan result = a
// anders result = b


function assignName( name ){
    return name || "unknown";
}

let person1 = assignName( null );
let person2 = assignName( "sofie" );
console.log( person1 );  // unknown
console.log( person2 );  // sofie
```

Besluit

primitive: numbers, strings, booleans, null & undefined

binary operators

- arithmetic (+ , - , * , /, %)
- string concatenation (+),
- comparison (== , != , === , !== , < , > , <= , >=)
- logic (&& , ||)

unary operators

- negatie (`getal = -getal;`)
- not (`if (!geslaagd){...}`)
- typeof

ternary operator ? :

- `let usertype = age < 18 ? 'minor' : 'adult';`