

Coding guidelines

Tab sizes

Bij voorkeur gebruiken we een specifiek aantal **spaties** (**Eng. spaces**) bij de volgende codeertalen:

| Taal | Aantal spaties |
|------------|----------------|
| CSS | 2 |
| Go | 4 |
| HTML | 4 |
| Java | 4 |
| JavaScript | 2 |
| PHP | 4 |
| Python | 4 |
| ... | ... |

Dit zijn aanbevelingen niet een gouden regel. Jullie toekomstige werkplek hebben eigen regels betreffende **inspringen** (*Eng. indentation*) of het gebruik van spaties. In verschillende code-editors kunnen we het inspringen instellen per taal.

Instellingen in Visual Studio Code:

Ga naar het menu “Code > Preferences > Settings” (Win: File > Preferences > Settings) om de voorkeuren in te stellen voor de Microsoft Visual Code Editor.

Bij de instelling “Detect Indentation - Controls whether Editor: Tab Size and Editor: Insert Spaces will be automatically detected when a file is opened based on the file contents.” verwijderen we het “vinkje”, dat betekent dat deze setting

`false` is.

Ga in het instellingspaneel op zoek naar de link `settings.json` bestand. Druk vervolgens op deze link. Er opent een nieuw venster waarin dit bestand wordt getoond. Voeg de volgende instellingen toe:

```
1  ...
2  "[javascript]": {
3      "editor.insertSpaces": true,
4      "editor.tabSize": 2
5  },
6  "[css]": {
7      "editor.insertSpaces": true,
8      "editor.tabSize": 2
9  },
10 "[html]": {
11     "editor.insertSpaces": true,
12     "editor.tabSize": 4
13 },
14 ...
```

Strict mode

Strict mode activeren

Strict mode kan toegepast worden op volledige scripts of binnen functies.

```
1  // Whole-script strict mode syntax
2  'use strict';
3
4  let v = "Hi! I'm a strict mode script!";
```

Laten we het `let` keyword achterwege, dan leidt dit tot de fout `ReferenceError: assignment to undeclared variable v`:

```
1  // Whole-script strict mode syntax
2  'use strict';
3
4  v = "Hi! I'm a strict mode script!"; //ReferenceError: assignment to undeclared variable
```

Strict mode kan ook ingesteld worden per functie:

```
1 function strict() {  
2   // Function-level strict mode syntax  
3   'use strict';  
4  
5   function nested() { return 'And so am I!'; }  
6   return "Hi! I'm a strict mode function! " + nested();  
7 }
```

Wat wordt beschouwd als een fout in strict mode?



Tip

Mozilla Developer Network (MDN) geeft een duidelijk overzicht omtrent de [regels in strict mode](#).

JavaScript stijlgidsen

Een JavaScript stijlgids is een handleiding met conventies om kwalitatieve JavaScript te schrijven. Deze conventies gaan meestal over naamgeving van variabelen, regels omtrent declaratie van variabelen en functies, regels omtrent het gebruik van witruimte en inspringen, het schrijven van goede commentaar, programmeerprincipes ...

Het volgen van conventies bevordert de leesbaarheid en de JavaScript-code is beter beheersbaar. In de wereld van Front-End Development worden de volgende stijlgidsen frequent gebruikt:

- [Airbnb JavaScript Style Guide](#)
- [JavaScript Standard Style](#)
- [Google JavaScript Style Guide](#)
- [Principles of Writing Consistent, Idiomatic JavaScript](#)

Binnen het Graduaat Programmeren hanteren we de stijlgids van **Airbnb** en de **JavaScript standard**.

Nakijken van JavaScript-code d.m.v. ESLint

ESLint is een statische **codeanalysetool** voor het identificeren van problematische patronen in JavaScript. Het is gemaakt door Nicholas C. Zakas in 2013. **Regels** in ESLint zijn **configureerbaar** en aangepaste regels kunnen worden gedefinieerd en geladen. ESLint behandelt zowel **codekwaliteit** als **codeerstijlproblemen**.

Globale installatie van `eslint-config-airbnb-standard`

Dit pakket combineert ESLint, de stijlguides van **Airbnb** en de stijlguides van **JavaScript standard**. Een aantal regels zullen we overschrijven.

Opgelet

Deze globale installatie is enkel geldig voor de opleidingsonderdelen Programming 1: Front-End (Basic) en @Work 1.

1. Installeer het npm pakket `eslint-config-airbnb-standard` globaal (beschikbaar in alle projecten).

```
1  npm install --global eslint-config-airbnb-standard
```

Op **Windows** krijgen we de volgende output in de command prompt:

```
1  $ npm install --global eslint-config-airbnb-standard
2  C:\Program Files\nodejs\eslint -> C:\Program Files\nodejs\node_modules\eslint-config-airbnb-standard
3  + eslint-config-airbnb-standard@3.1.0
4  added 220 packages from 160 contributors in 23.293s
```

Kopieer het pad `C:\Program Files\nodejs\node_modules\eslint-config-airbnb-standard\node_modules\eslint` naar het klembord.

2. Binnen VSCode ga naar: **View -> Extension**.

3. Ga na of de [ESLint](#) plugin of extensie geïnstalleerd is binnen VSCode (deze plugin is reeds geïnstalleerd tijdens de startdagen). Sluit VSCode en herstart deze vervolgens.

4. Binnen VSCode ga naar: **File -> Preferences -> Settings**. Open vervolgens het `settings.json` bestand. Voeg de volgende ESLint instellingen toe:

```

1  {
2      ...
3      "eslint.nodePath": "/Users/dr dynscript/.nvm/versions/node/v10.16.0/lib/node_modules/
4      "eslint.alwaysShowStatus": true,
5      ...
6  }

```

Pas de waarde van de eigenschap `"eslint.nodePath"` aan die afhankelijk is van de locatie van de installatie van het `eslint-config-airbnb-standard` pakket (zie pad tijdens de installatie van `eslint-config-airbnb-standard`).

Voor **macOS** gebruikers gebeurt deze installatie meestal onder `/Users/**your own user name**/.nvm/versions/node/v10.16.0/lib/node_modules/` .

Voor **Windows** gebruikers wordt het pakket meestal bewaard onder `C:\Users**your own user name**\AppData\Roaming\npm\node_modules\` .

```

1  {
2      ...
3      "eslint.nodePath": "C:\\Users\\phildp\\scoop\\apps\\nvm\\current\\nodejs\\nodejs\\n
4      "eslint.alwaysShowStatus": true,
5      ...
6  }

```

5. Maak een bestand `.eslintrc.js` aan binnen elk JavaScript project, met de volgende inhoud:

```

1  process.chdir(__dirname);
2
3  module.exports = {
4      root: true,
5      parser: 'babel-eslint',
6      parserOptions: {
7          allowImportExportEverywhere: true,
8          codeFrame: false
9      },
10     extends: [
11         'airbnb-standard',
12     ],
13     rules: {
14         'comma-dangle': [
15             'error',
16             'always-multiline'

```

```

17     ],
18     "linebreak-style": "off",
19     "no-console": "off",
20     "no-plusplus": "off",
21     "no-restricted-syntax": [
22         "error",
23         {
24             "selector": "CallExpression[callee.object.name='console'][callee.property.name]",
25             "message": "Unexpected property on console object was called"
26         }
27     ],
28     'no-void': 'off',
29     strict: [
30         'error',
31         'global'
32     ],
33 },
34 };

```

Voorbeeld

Structuur van de website:

- projectfolder (folder)
 - js (folder)
 - main.js
 - css (folder)
 - main.css
 - .eslintrc.js
 - index.html

main.js :

```

1  function fetchData(){
2      var data = [
3          {
4              "firstName": "Philippe"
5          }
6      ]
7      return data

```

js

```
8 }
9
10 console.log(fetchData())
```

Deze code is goedgekeurd indien we geen `.eslinrc.js` bestand in de projectfolder hebben toegevoegd. Voegen we dit bestand wel toe voorzien met de inhoud van het vorige hoofdstuk, dan resulteert dit in een aantal stijlproblemen:

```
1 function fetchData(){
2   var data = [
3     (local var) data: {
4       "firstName": string;
5     }[]
6   ]
7   Missing space before opening brace. eslint(space-before-blocks)
8   Unexpected var, use let or const instead. eslint(no-var)
9   Peek Problem Quick Fix...
10 }
```

We kunnen deze problemen oplossen door met de muis over het bewuste probleem te bewegen, de melding te lezen en te interpreteren en vervolgens de code aan te passen. Dit resulteert in de volgende gevalideerde code:

```
1 function fetchData () {
2   const data = [
3     {
4       firstName: 'Philippe',
5     },
6   ];
7   return data;
8 }
9
10 console.log(fetchData());
11
```

← Time based stuff