

Metodika - Code style JavaScript

ESLint - TypeScript/ JavaScript / svelte linter

ESLint vyhledá najčastejšie chyby v Javascripte automaticky a vyznačí vo vývojovom prostredí. Tiež poskytuje pre niektoré problémy automatické opravy.

<https://github.com/eslint/eslint>

Analýzou kódu nájde chyby ako

- Vytvorená premenná, ktorá nikdy nie je použitá
- Zabudnutý console.log z debugovania
- Nepoužité camelCase pomenovanie premenných
- Implicitné globálne premenné (zabudnuté let / const)
- ...

ESLint - 20k hviezdíčiek na githube, 17 mil stiahnutí za týždeň na npm

Cez inicializáciu (eslint --init) sa dá nakonfigurovať. Napríklad:

- aká verzia syntaxu sa používa (ES6)
- ES6 moduly alebo commonJS (node),
- spaces vs tabs
- typ ukončenia riadku (\n vs \r\n)
- Používať bodkočiarky na konci riadka alebo nie?

Ale **ESLint** plugin pre **Svelte** framework **nemá podporu TypeScriptu**

<https://github.com/sveltejs/eslint-plugin-svelte3>

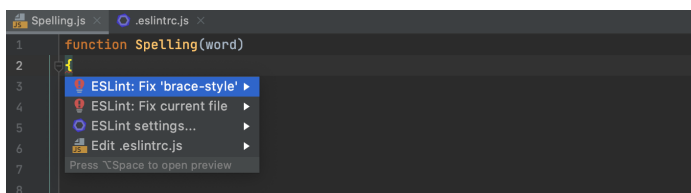
Typescript ESLint:

<https://github.com/typescript-eslint/typescript-eslint>

ESLint podpora pre IntelliJ

https://www.jetbrains.com/help/webstorm/eslint.html#ws_js_linters_eslint_before_you_start

Je možné nastaviť, aby opravil problémy automaticky po uložení.



Prettier - Alternatíva k formátovaniu kódu cez ESLint

Deaktivovaním formátovania cez ESLint a nech to ide cez Prettier

<https://github.com/prettier/eslint-plugin-prettier>

JS - Mozilla guide

Mozilla má dobrý guide:

https://developer.mozilla.org/en-US/docs/MDN/Guidelines/Code_guidelines/JavaScript

Highlights z Mozilla guideline

Na jeden riadok písať vždy len jeden príkaz, nepoužívať “jednoriadkovce”

Správne

```
function myFunc() {  
  console.log('Hello!');  
};
```

Nesprávne

```
function myFunc() { console.log('Hello!'); };
```

Používať **medzery medzi operátormi a parametrami**

Správne

```
if(dayOfWeek === 7 && weather === 'sunny') {  
  goOnTrip('beach', 'car', ['ice cream', 'bucket and spade',  
  'beach towel']);  
}
```

Nesprávne

```
if(dayOfWeek === 7 && weather === 'sunny') {  
  goOnTrip('beach', 'car', ['ice cream', 'bucket and spade',  
  'beach towel']);  
}
```

Nepoužívať `var`, ale **let** / **const**.

Názov **premených** a **funkcií** štýlom **camelCase**. Názvy **tried** štýlom **PascalCase**

Zrozumiteľne nazývať premenné, nie “a”, “b”, ...

```
let playerScore = 0;
```

```
function sayHello() {
```

```
class Person {
```

Používať lepšie pochopiteľnejšie **porovnanie s porovnaním typov**

```
if(name === 'Chris')  
if(age !== 25)
```

Na vkladanie premenných do reťazcov používať **string literals**

```
console.log(`Hi! I'm ${myName}!`);
```

Používať moderný JS (ES6) syntax (arrow funkcie, literals pre tvorbu objektov a polí)

```
let sum = array1.reduce((a, b) =>
```

```
let myObject = { };
```

```
let myArray = [];
```