WHAT IS IT AND WHAT IS IT FOR?

WHAT IS A LINTER?

A linter is a useful tool that analyzes the code and detects any inconsistencies like syntax errors, code smells or even problems with security.

It also helps maintaining the code and make it more readable to everyone. In theory, a linter will make the code seem it was written by one person, when used in big projects.

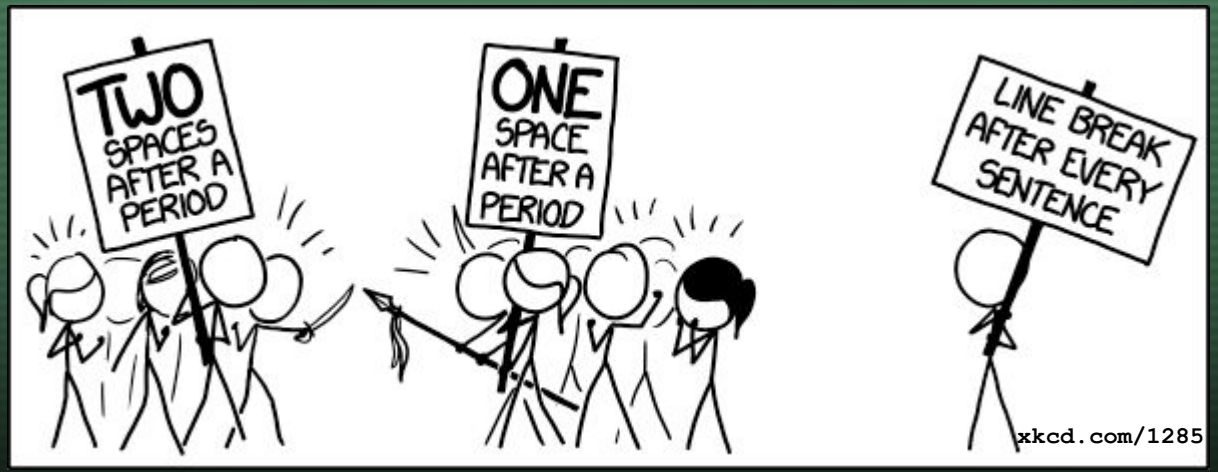
WHAT IS ESLINT?

ESLint is a JavaScript linter that enables you to enforce a set of style, formatting, and coding standards for your codebase. It looks at your code, and tells you when you're not following the standard that you set in place.

There was a TSLint until 2019 because “ESLint exists, and there was a lot of duplicate code between projects with the same intended purpose”.

WHAT IS ESLINT FOR?

Clean code isn't just code that is readable and maintainable, it's also code that is consistent. Having multiple ways of doing the same thing is confusing and time-consuming.



WHAT IS ESLINT FOR?

It’s possible to have a pre-commit hook that lints your code before committing, like Husky.

Since we’re on topic, it’s possible to even have a commit “linter” and make standards in your commit messages, like Commitizen.

Didn’t find a rule for you? No problem, you can also create your own rules.

STEPS FOR IMPLEMENTING ESLINT

1. `npm i eslint --save-dev`
2. `npx eslint --init`

...and that's it!

STEPS FOR IMPLEMENTING ESLINT FOR TS

1. `npm i --save-dev eslint @typescript-eslint/parser @typescript-eslint/eslint-plugin`
2. `npm i eslint --save-dev`
3. `npx eslint --init`

...and that's it!

EXTRA STEPS FOR LINTING

Create a script in package.json with:

```
"lint-fix": "eslint . --ext .js --fix"
```

or

```
"lint-fix": "eslint . --ext .ts --fix"
```

This way, the linter is only run after you wrote everything.
It also fixes some of the problems.

EXTRA STEPS FOR LINTING

It will show something like this:

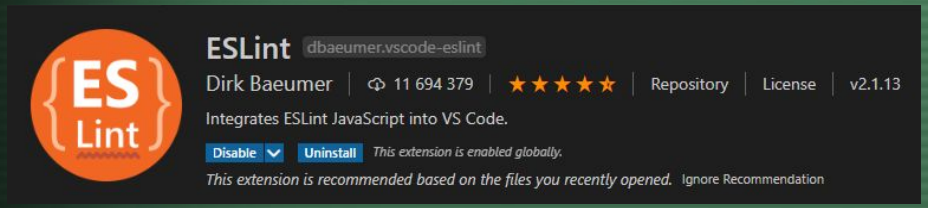
```
C:\Users\filipe.silva\Desktop\lint-examples\app_ts\src\index.ts
   1:15  error    Variable 'coolStuff' should be initialized on declaration  init-declarations
   1:15  warning  'coolStuff' is defined but never used                      @typescript-eslint/no-unused-vars
   1:26  warning  Unexpected any. Specify a different type                   @typescript-eslint/no-explicit-any
  24:10  warning  Identifier 'show_id' is not in camel case                  camelcase
  25:18  warning  Unexpected any. Specify a different type                   @typescript-eslint/no-explicit-any
  26:5   warning  Unexpected console statement                               no-console
  30:7   warning  'user' is assigned a value but never used                  @typescript-eslint/no-unused-vars

✖ 7 problems (1 error, 6 warnings)
```

Making it easier to identify every problem with it.

EXTRA STEPS FOR LINTING

You can also use some extensions and implement a even more powerful linter on your editor, when formatting. Some formatters, like Prettier,
become even more powerful with ESLint configured.



GOOD NEWS! ATD already has (some)

projects with ESLINT!

```
//#region DialogSequenceManager
export interface IDialogItem
//Representa um Dialog
{
    dialogSettings: IDialogOptions,
    btnYes?: IDialogSequenceButton;
```

LINKS AND STUFF

- ESLint: https://eslint.org/docs/user-guide/getting-started
- Various linters: https://github.com/collections/clean-code-linters
- Husky: https://github.com/typicode/husky
- Commitizen : https://github.com/commitizen/cz-cli
- Prettier: https://prettier.io/docs/en/integrating-with-linters.html