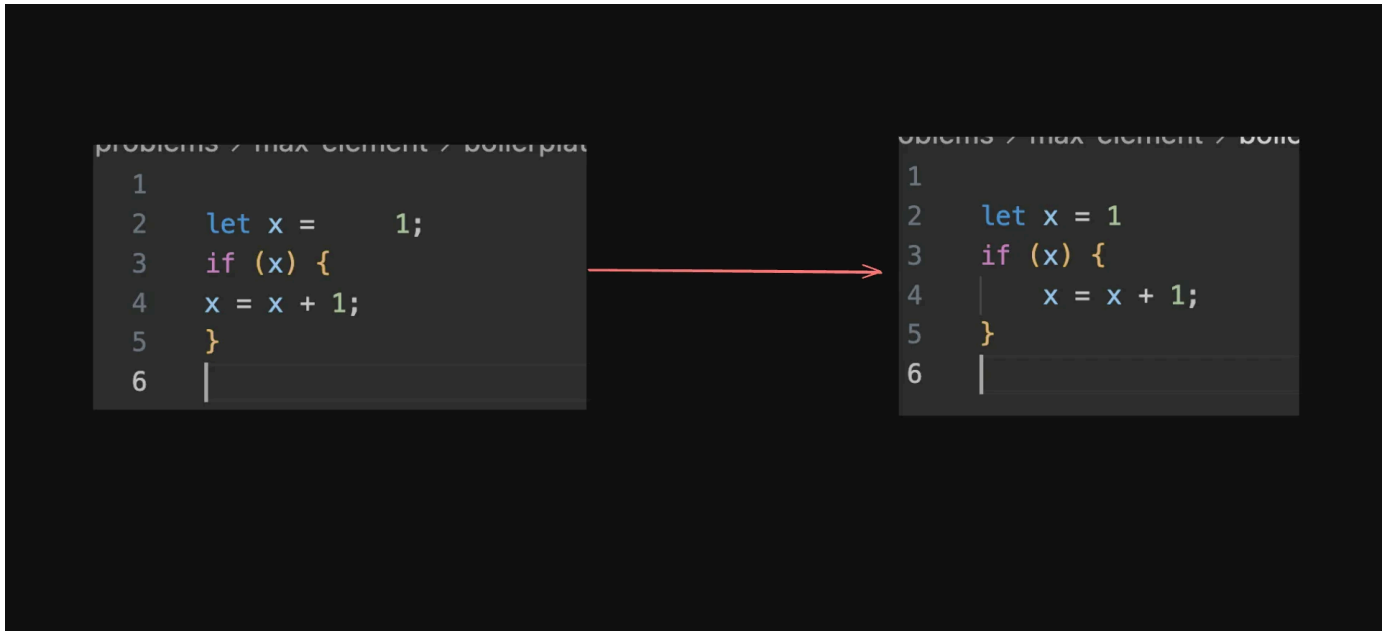
What is Linting

Linting is the process of running a program that analyzes source code to flag programming errors, bugs, stylistic errors



1. Multiple developers in a team can have a similar coding pattern

3. Automatic Linting so developer doesnt have to manually lint their code

Tools

1. ESLint - https://eslint.org/
2. Prettier - https://prettier.io/
3. Husky - https://typicode.github.io/husky/

Two types of lints

1. Code analysis (Similar to TS checks)

Analyzes code to find and fix problems according to a set of rules. These problems can include syntax errors, stylistic issues, and potential bugs.

```
let x = 1;
let y = 1;
```

▼ Can you guess some bad things about this code?

1. x is never used, and hence can be removed
2. can be a const

2. Code formatting (Purely stylistic changes)

```
let x = 1;
let     y = 1
let firstName = "harkirat";
let lastName = 'singh';
```

▼ Can you guess some bad things about this code?

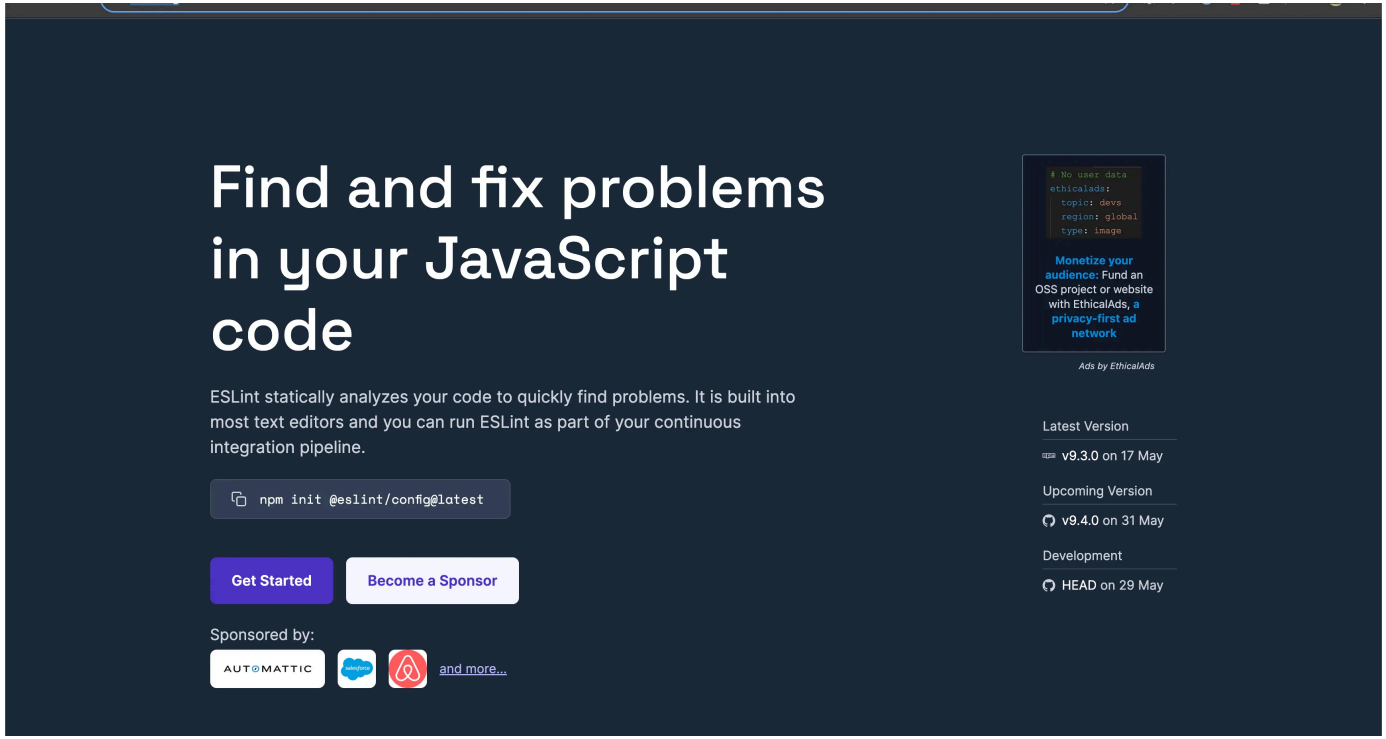
1. One line is missing semi colons
2. Spacing is wrong in second line
3. One string uses double quotes, other uses single quotes

Eslint lets you do both 1 and 2

Prettier is used for 2

ESLint

Ref - https://eslint.org/



Node.js express app


```
import express from "express";
const app = express();
let x = 1;

app.get("/", (req, res) => {
   res.json({
      message: "Hi there"
   });
});
```

- Add ESLint

```
npm init @eslint/config@latest
```

> 💡 Make sure you press `space` to actually select the values. If you face issues, try using `yarn` instead of `npm` (npm i -g yarn)

ESLint, Prettier and Pre-commit hooks 1 of 9

```
➜  1-eslint-express np
                                                t/config@latest
                                                · problems
                                                oject use? · esm
✔ Which framework does your project use? · none
✔ Does your project use TypeScript? · typescript
✔ Where does your code run? · node
The config that you've selected requires the following dependencies:

eslint@9.x, globals, @eslint/js, typescript-eslint
✔ Would you like to install them now? · No / Yes
✔ Which package manager do you want to use? · npm
☕ Installing...
```

- Explore eslint.config.mjs

```
import globals from "globals";
import pluginJs from "@eslint/js";
import tseslint from "typescript-eslint";

export default [
  {languageOptions: { globals: globals.node }},
  pluginJs.configs.recommended,
  ...tseslint.configs.recommended,
];
```

- You might have to install typescript

```
yarn add typescript
```


```
"scripts": {
},
```

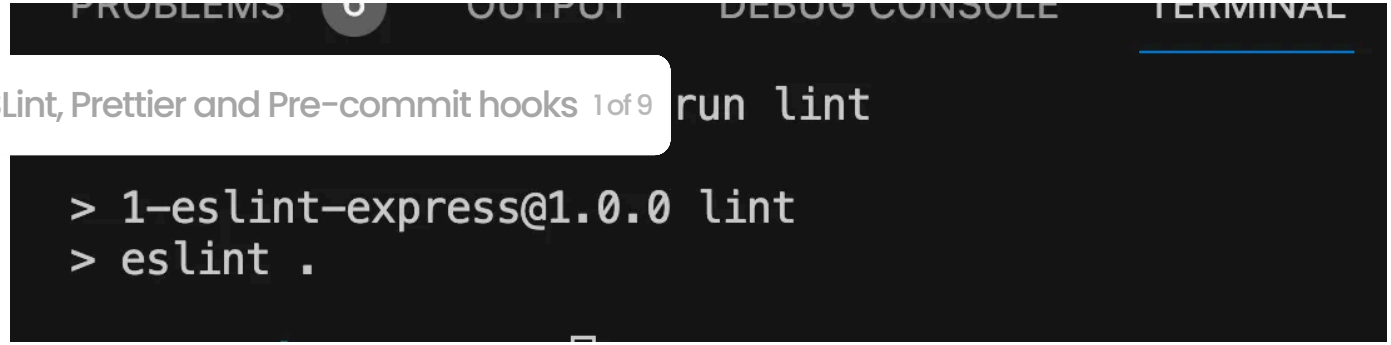
- Run `npm run lint`

```
⊗ → 1-eslint-express npm run lint

> 1-eslint-express@1.0.0 lint
> eslint .


/Users/harkiratsingh/Projects/week-26-eslint-prettier/solution/1-eslint-express/src/index.ts
  5:5  error  'x' is never reassigned. Use 'const' instead  prefer-const
  5:5  error  'x' is assigned a value but never used        @typescript-eslint/no-unused-vars

✖ 2 problems (2 errors, 0 warnings)
  1 error and 0 warnings potentially fixable with the `--fix` option.
```

- Try removing the extra variable

```
import express from "express";

const app = express();

app.get("/", (req, res) => {
    res.json({
        message: "Hi there"
    });
});
```

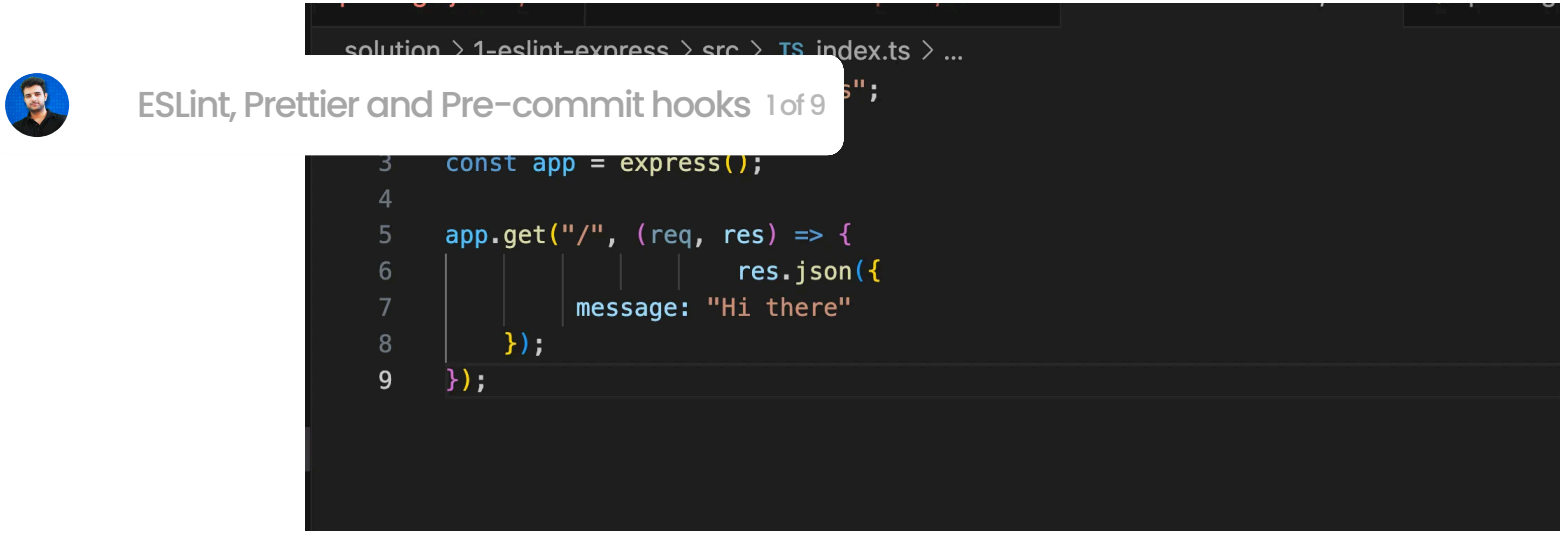
- Now run `npm run lint`



- Try breaking the indendation

```
import express from "express";

const app = express();

app.get("/", (req, res) => {
        res.json({
    message: "Hi there"
  });
});
```

- Notice eslint still DOESNT complain



Code formatting in ESLint

- Update eslint.config.mjs

```
import globals from "globals";
import pluginJs from "@eslint/js";
import tseslint from "typescript-eslint";

export default [
  {
```

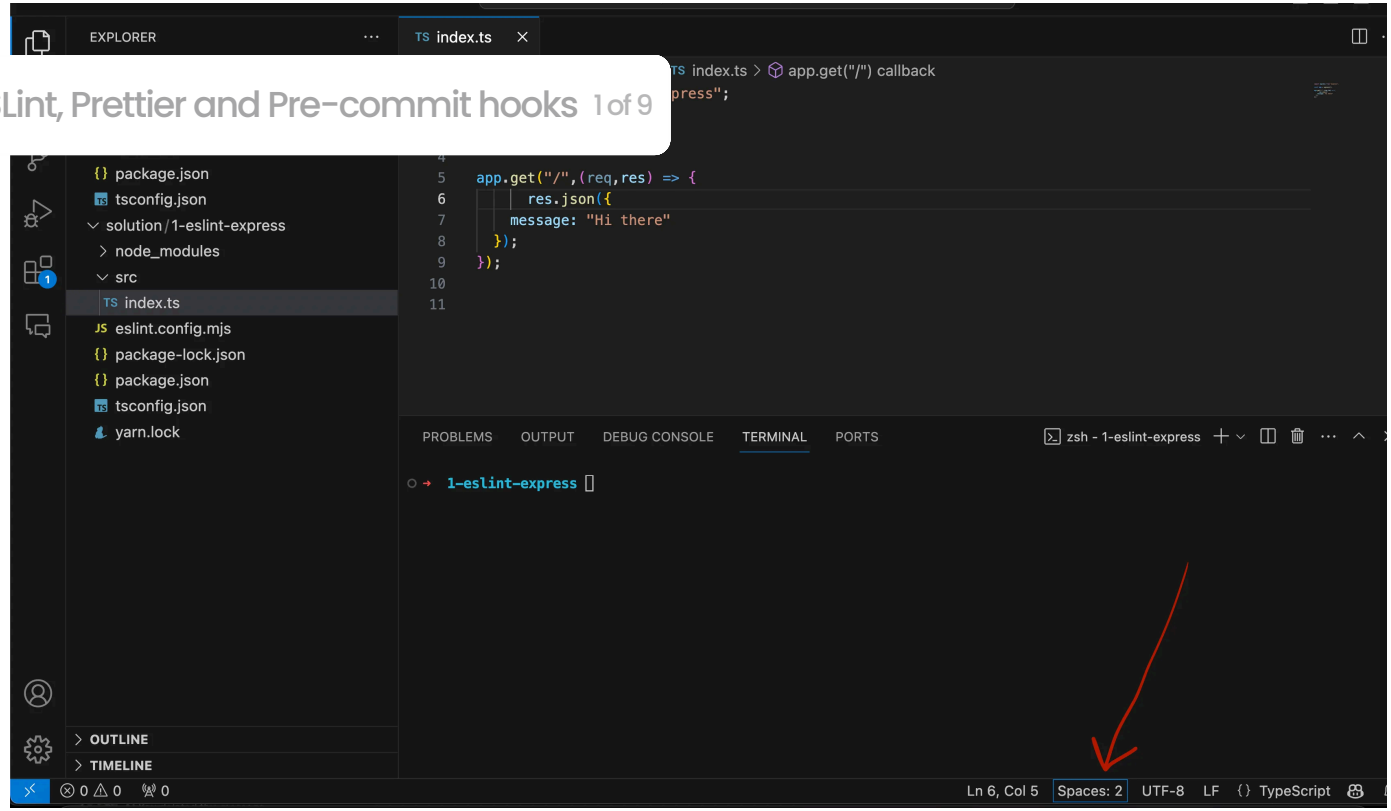
```
// Indentation rule: 2 spaces
```

```
    "indent": ["error", 2],
    ...hesis
    ["error", "always"],
    // No space inside parentheses
    "space-in-parens": ["error", "never"]
  }
 },
 pluginJs.configs.recommended,
 ...tseslint.configs.recommended,
];
```

```
→  1-eslint-express npm run lint

> 1-eslint-express@1.0.0 lint
> eslint .


/Users/harkiratsingh/Projects/week-26-eslint-prettier/solution/1-eslint-express/src/index.ts
  6:1  error  Expected indentation of 2 spaces but found 5 tabs  indent
  8:1  error  Expected indentation of 2 spaces but found 1 tab   indent

✖ 2 problems (2 errors, 0 warnings)
  2 errors and 0 warnings potentially fixable with the `--fix` option.
```

Problem

If you press tab while coding, it adds a tab. You need 2 spaces to be added now that our eslint config says that



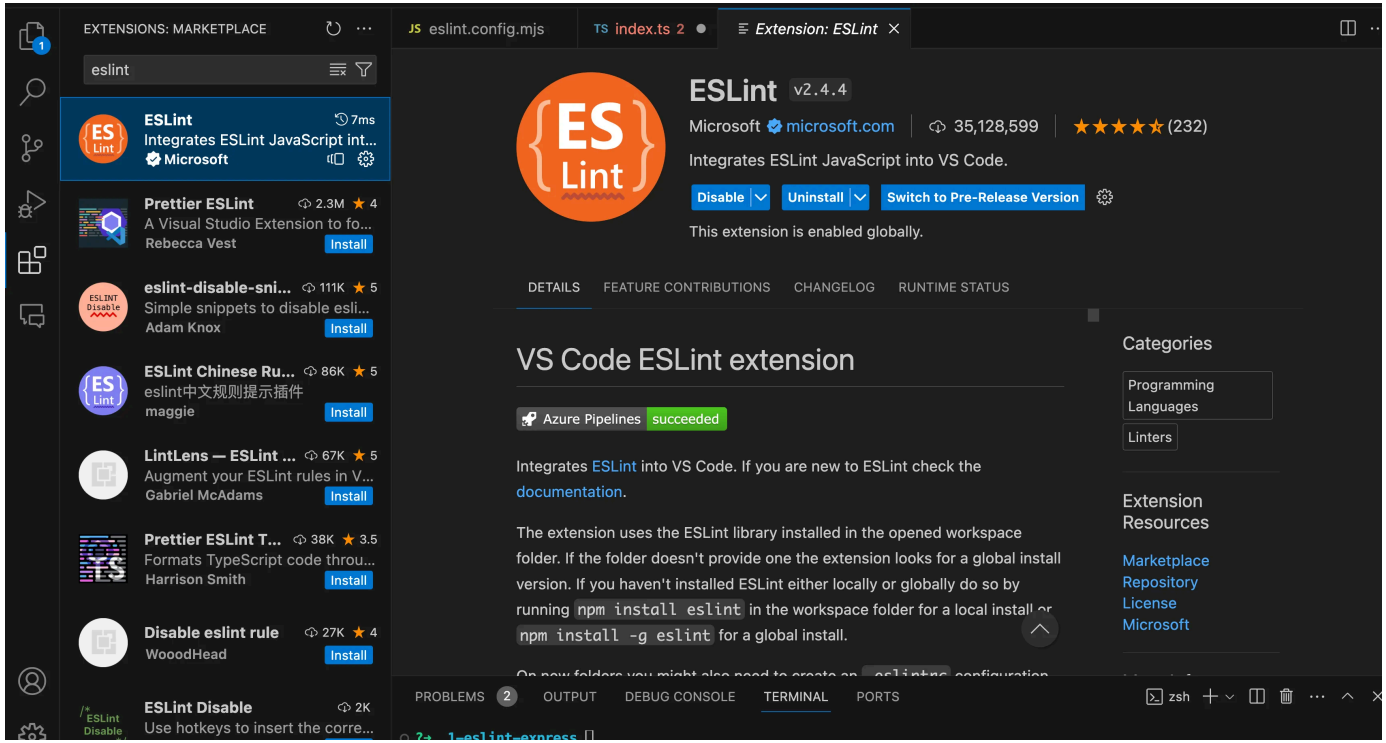
VSCode plugin

Problem

our editor directly. We have to run `npm run lint` to be able to see them.

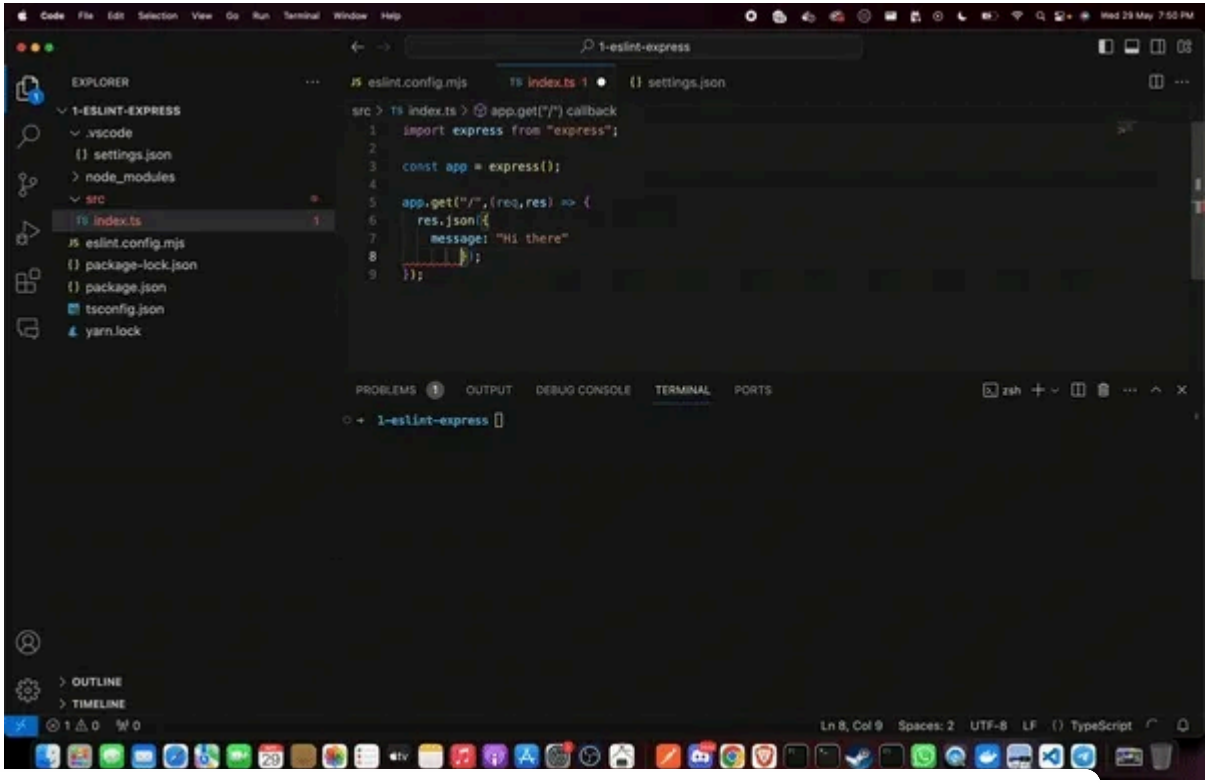
Solution

- Install the ESLint VSCode plugin




```
    source.fixAll.eslint": "explicit"
  },
  "eslint.validate": ["javascript", "typescript"]
}
```

- Notice code gets auto linted every time you save now



ESLint, Prettier and Pre-commit hooks 1 of 9

ESLint in react project

- Initialize a react project

```
npm create vite@latest
```

- vite by default gives you eslint
- Run lint

```
yarn install
yarn lint
```

Popular eslint configurations

You can roll your own config, like we did in the last slide

```
module.exports = {
  root: true,
  env: { browser: true, es2020: true },
  extends: [
    'eslint:recommended',
    'plugin:@typescript-eslint/recommended',
    'plugin:react-hooks/recommended',
  ],
  ignorePatterns: ['dist', '.eslintrc.cjs'],
  parser: '@typescript-eslint/parser',
  plugins: ['react-refresh'],
  rules: {
```

```
    { allowConstantExport: true },
```

```
],
```

ESLint, Prettier and Pre-commit hooks 1 of 9

Or you can use a popular one created by other people, for example

1. https://github.com/airbnb/javascript

1. https://github.com/dustinspecker/awesome-eslint#configs

ESLint, Prettier and Pre-commit hooks 1 of 9

g multiple properties of an object. eslint: `prefer-destructuring`

> Why? Destructuring saves you from creating temporary references for those properties, and from repetitive access of the object. Repeating object access creates more repetitive code, requires more reading, and creates more opportunities for mistakes. Destructuring objects also provides a single site of definition of the object structure that is used in the block, rather than requiring reading the entire block to determine what is used.

```
// bad
function getFullName(user) {
  const firstName = user.firstName;
  const lastName = user.lastName;

  return `${firstName} ${lastName}`;
}

// good
function getFullName(user) {
  const { firstName, lastName } = user;
  return `${firstName} ${lastName}`;
}

// best
function getFullName({ firstName, lastName }) {
  return `${firstName} ${lastName}`;
}
```

Using airbnb config

- Add dependencies

ESLint, Prettier and Pre-commit hooks 1 of 9

```
yarn add eslint-config-airbnb-typescript eslint-plugin-
```

- Update .eslintrc.cjs

```
module.exports = {
  root: true,
  env: { browser: true, es2020: true },
  parserOptions: {
    ecmaVersion: "latest",
    sourceType: "module",
    project: ["./tsconfig.json", "./tsconfig.node.json"],
    tsconfigRootDir: __dirname,
  },
  extends: [
    "eslint:recommended",
    "plugin:@typescript-eslint/recommended",
    "plugin:react-hooks/recommended",
    "airbnb",
    "airbnb/hooks",
  ],
  ignorePatterns: ["dist", ".eslintrc.cjs", "vite.config.ts"],
  parser: "@typescript-eslint/parser",
  plugins: ["react-refresh"],
  rules: {
    "react/jsx-filename-extension": [


    ],
```

```
    "react-refresh/only-export-components": [
      ],
    },
  };
```

💡 If you try doing the same in the new config system (for eg, the node.js project), you'll face some issues.

New config system - ▸

Ref - https://stackoverflow.com/questions/74925642/how-to-use-eslint-config-airbnb-with-the-new-eslint-config-spec

- Update .eslintrc.cjs (read README.md)

```
parserOptions: {
  ecmaVersion: 'latest',
  sourceType: 'module',
  project: ['./tsconfig.json', './tsconfig.node.json'],
  tsconfigRootDir: __dirname,
},
```

- Try running the lint command

ESLint, Prettier and Pre-commit hooks 1 of 9

```
2-eslint-react yarn lint
                                                    ives --max-warnings 0
                                                    /solution/2-eslint-react/src/App.tsx
  1:33  error  Missing semicolon                                                  semi
  2:43  error  Missing semicolon
```

ESLint with nextjs

- Create a next application (Select yes on ESLint)

```
npx create-next-app
```

- Check the default config

```
{
  "extends": ["next/core-web-vitals"]
}
```

- Add .vscode/settings.json

```
{
  "editor.formatOnSave": true,
```

```
}
```

ESLint, Prettier and Pre-commit hooks 1 of 9

- Add airbnb config

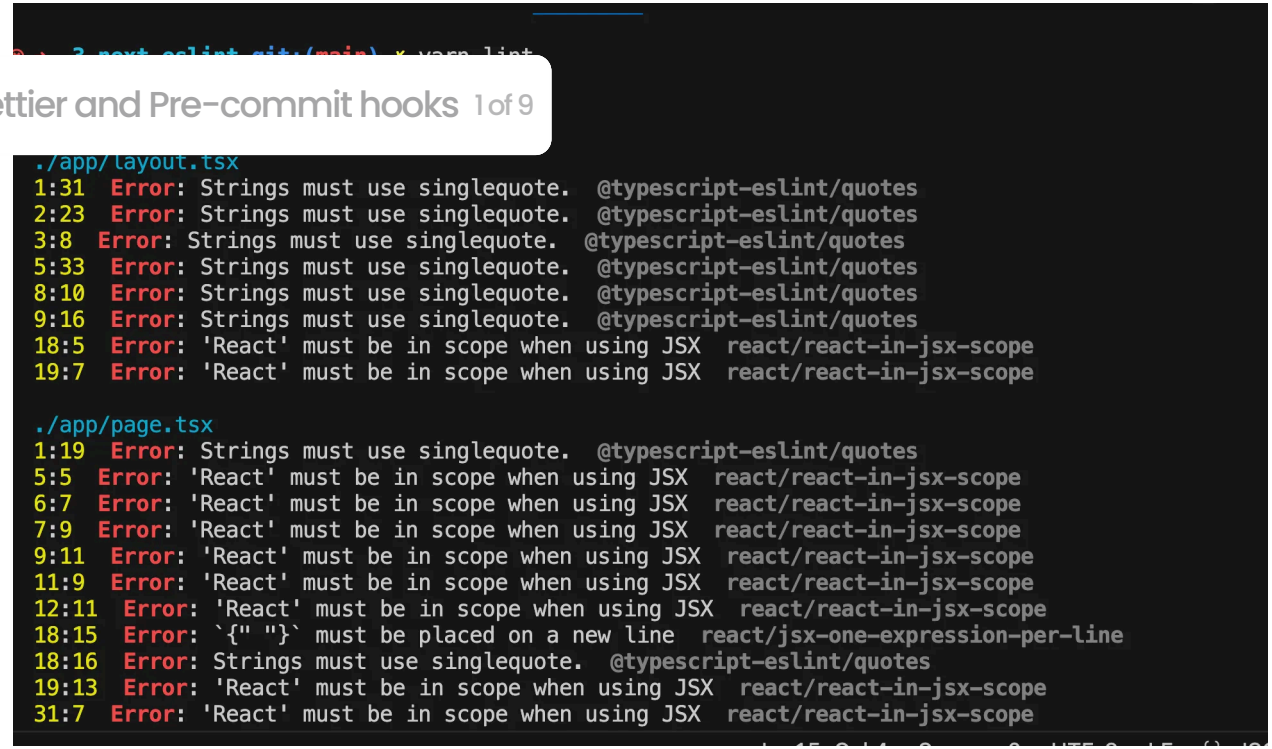
```
yarn add eslint-config-airbnb eslint-config-airbnb-typescript @typescript-e
```

- Update the eslint config

```
{
  "extends": ["next/core-web-vitals", "airbnb", "airbnb/hooks"]
}
```

- Run `npm run lint`

ESLint, Prettier and Pre-commit hooks 1 of 9



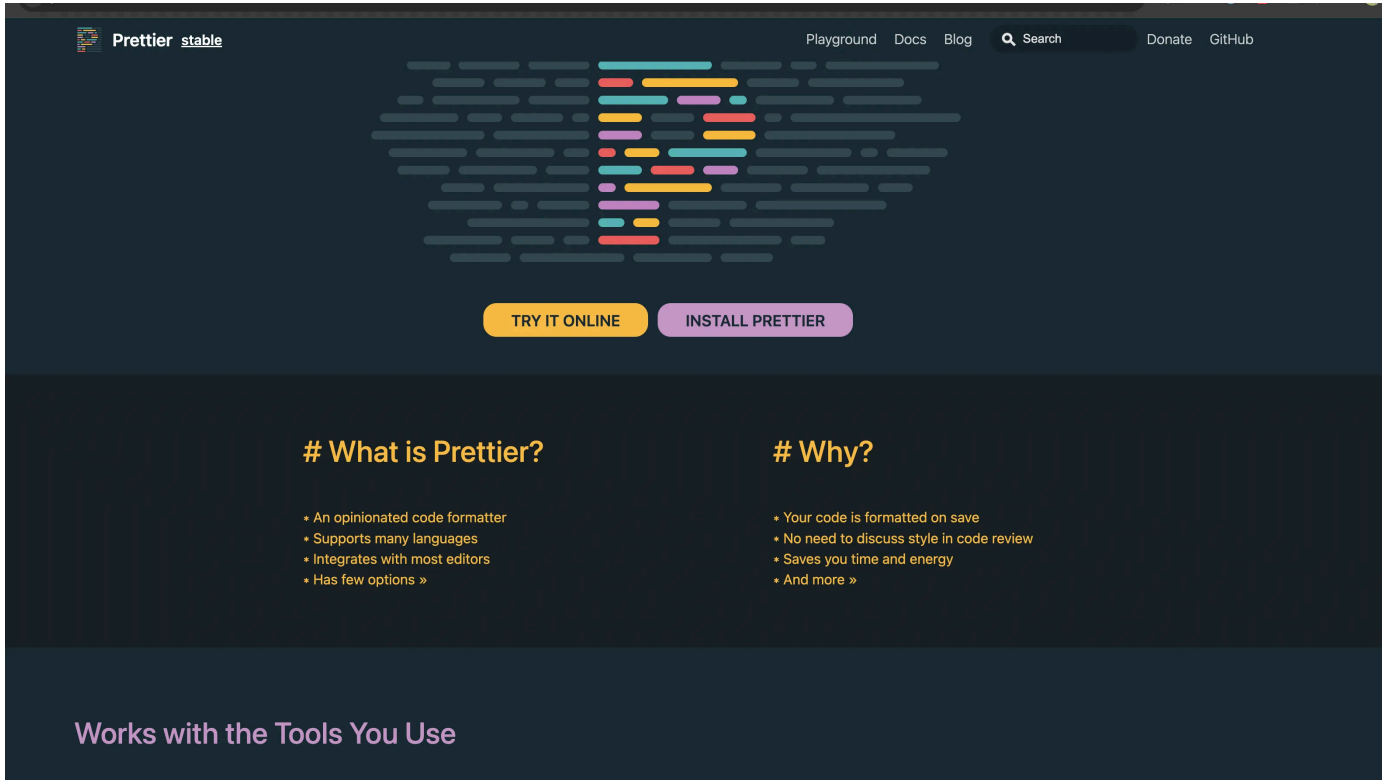
```
./app/layout.tsx
1:31  Error: Strings must use singlequote.  @typescript-eslint/quotes
2:23  Error: Strings must use singlequote.  @typescript-eslint/quotes
3:8  Error: Strings must use singlequote.  @typescript-eslint/quotes
5:33  Error: Strings must use singlequote.  @typescript-eslint/quotes
8:10  Error: Strings must use singlequote.  @typescript-eslint/quotes
9:16  Error: Strings must use singlequote.  @typescript-eslint/quotes
18:5  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
19:7  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope

./app/page.tsx
1:19  Error: Strings must use singlequote.  @typescript-eslint/quotes
5:5  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
6:7  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
7:9  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
9:11  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
11:9  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
12:11  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
18:15  Error: `{" "}` must be placed on a new line  react/jsx-one-expression-per-line
18:16  Error: Strings must use singlequote.  @typescript-eslint/quotes
19:13  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
31:7  Error: 'React' must be in scope when using JSX  react/react-in-jsx-scope
```

- Fix some automatically

```
yarn lint --fix
```

ESLint, Prettier and Pre-commit hooks 1 of 9

Ref - https://prettier.io/



Installing prettier - https://prettier.io/docs/en/install

- Install dependency

- Create a .prettierrc file in the root

ESLint, Prettier and Pre-commit hooks 1 of 9

- Create .prettierignore

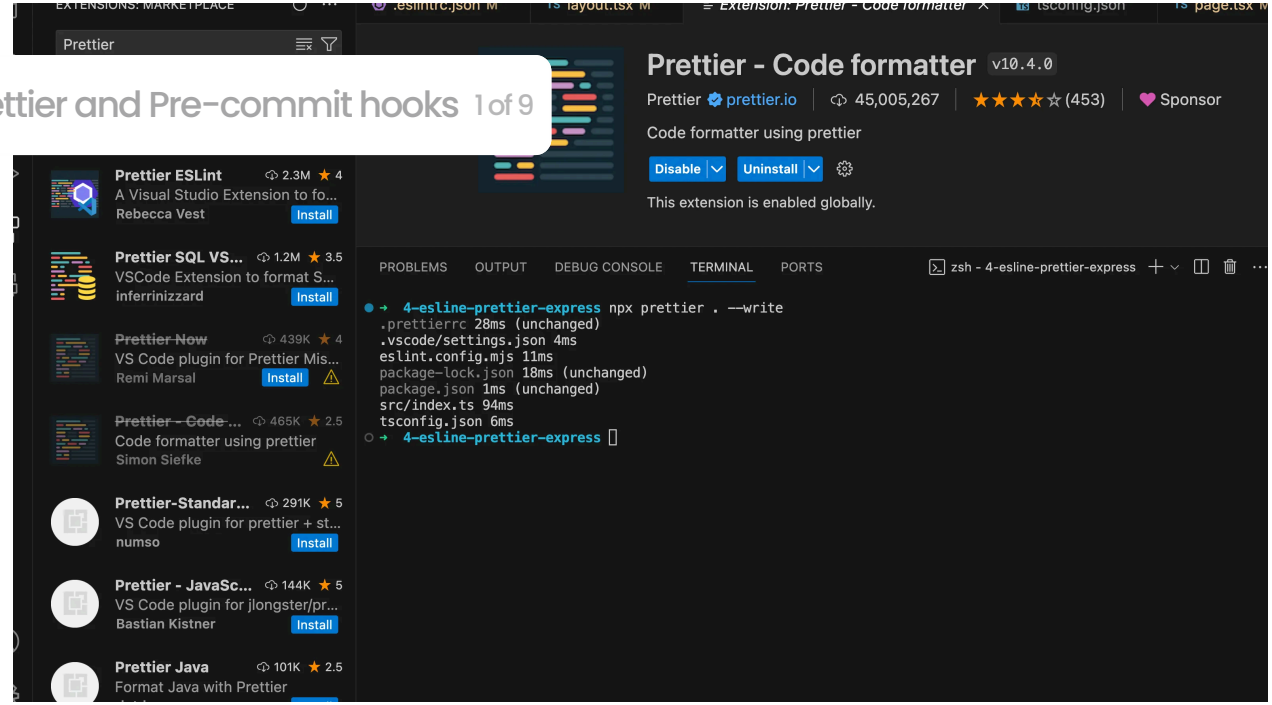
```
build
coverage
dist
```

- Try formatting

```
npx prettier . --write
```

- Add vscode extension

ESLint, Prettier and Pre-commit hooks 1 of 9



- Update .vscode/settings.json

```
{
 "editor.codeActionsOnSave": {
  "source.fixAll.eslint": "explicit"
 },
 "eslint.validate": ["javascript", "typescript"],
 "editor.defaultFormatter": "esbenp.prettier-vscode",
 "[javascript]": {
  "editor.defaultFormatter": "esbenp.prettier-vscode",


 "[typescript]": {
```

```
    "editor.defaultFormatter": "esbenp.prettier-vscode",
```


```
}
```

- Install `eslint-config-prettier` (this is to ensure they dont conflict with each others configs)

```
npm install --save-dev eslint-config-prettier
```

- Extend eslint config to use `prettier`

```
"extends": [
  "next/core-web-vitals",
  "airbnb",
  "airbnb-typescript",
  "airbnb/hooks",
  "prettier"
],
```

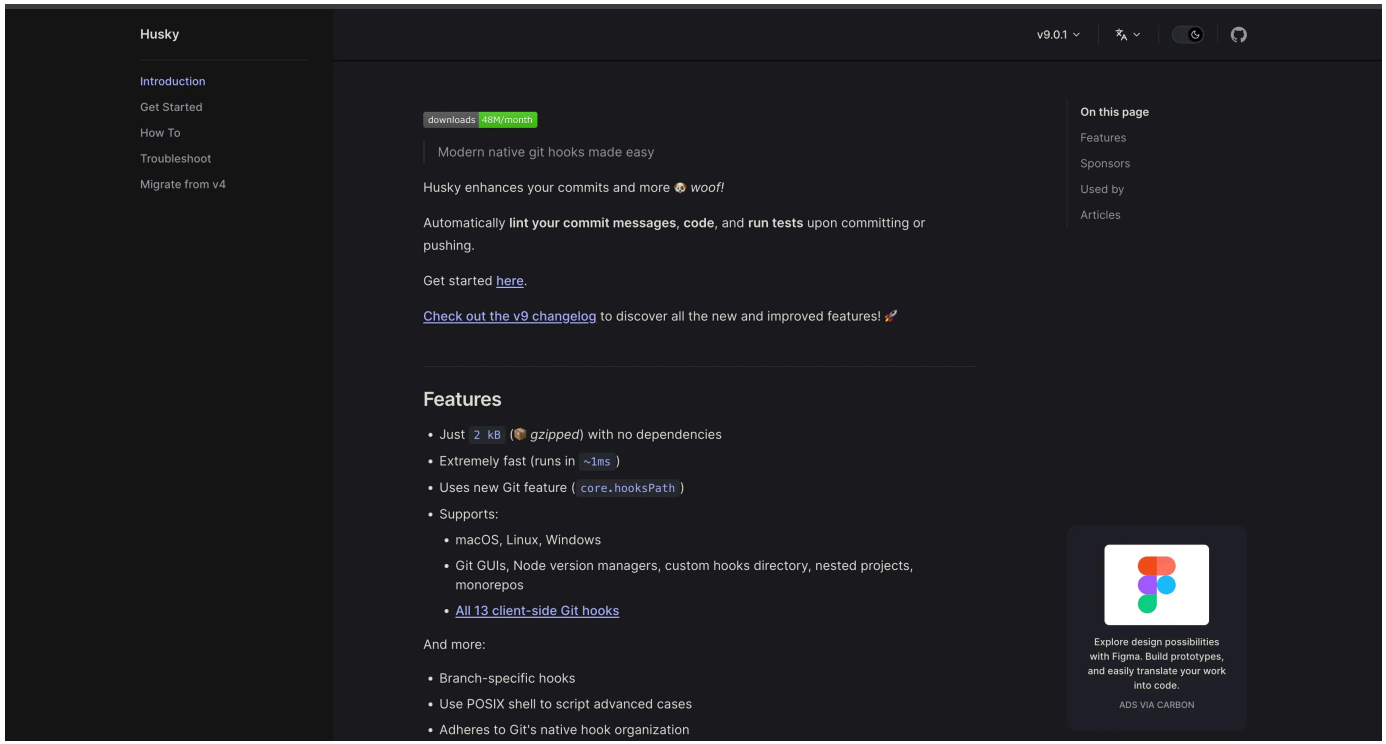
- Add `prettier` script

```
"prettier": "prettier . --write"
```

- Try running `npm run prettier`

Pre commit hooks

Ref - https://typicode.github.io/husky/



How to start

ESLint, Prettier and Pre-commit hooks 1 of 9

- Initialise husky

  npx husky init

- Update .husky/pre-commit

  npm prettier

- Try to commit

  git init
  git add .
  git commit -m "Init"

- You can get past it by using --no-verify

ESLint, Prettier and Pre-commit hooks 1 of 9

git commit -m "init" --no-verify