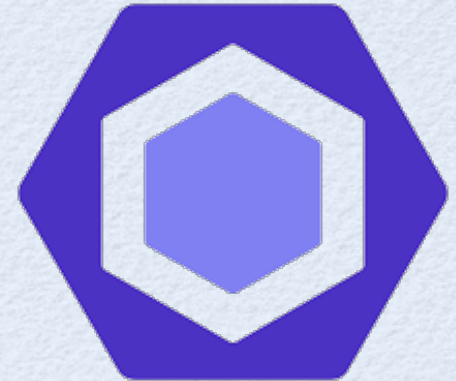# Vue Tools

Tools Commonly Used

- Software tools commonly used with Vue projects include
  - **ESLint** for linting JavaScript, HTML, and CSS
  - **Prettier** for formatting JavaScript, HTML, and CSS
  - **VS Code**

ESLint Overview

- "The pluggable linting utility for JavaScript and JSX"

- Can report many syntax errors and potential run-time errors

- Can report deviations from specified coding guidelines

- Error messages identify violated rules,
  making it easy to adjust them if you disagree

- Has `--fix` mode that can fix violations of many rules

  - modifies source files

- Many ways to configure

  - `.eslintrc.json` file

  - `.eslintrc.js` file

    - allows conditionally setting options based on NODE_ENV environment variable

  - `"eslint"` key in `package.json`

3

ESLint Details

- To install, `npm install -D eslint`

  - Vue CLI projects already have this installed

- Additional recommended installs

  - `npm install -D *` where `*` is
    `eslint-config-prettier`
    `eslint-plugin-html`
    `eslint-plugin-import`
    `eslint-plugin-prettier`
    `@vue/eslint-config-prettier`

- To use from an npm script, add following to `package.json`

  `"lint": "eslint --fix --quiet src --ext .js,.vue",`   `--quiet` only reports errors
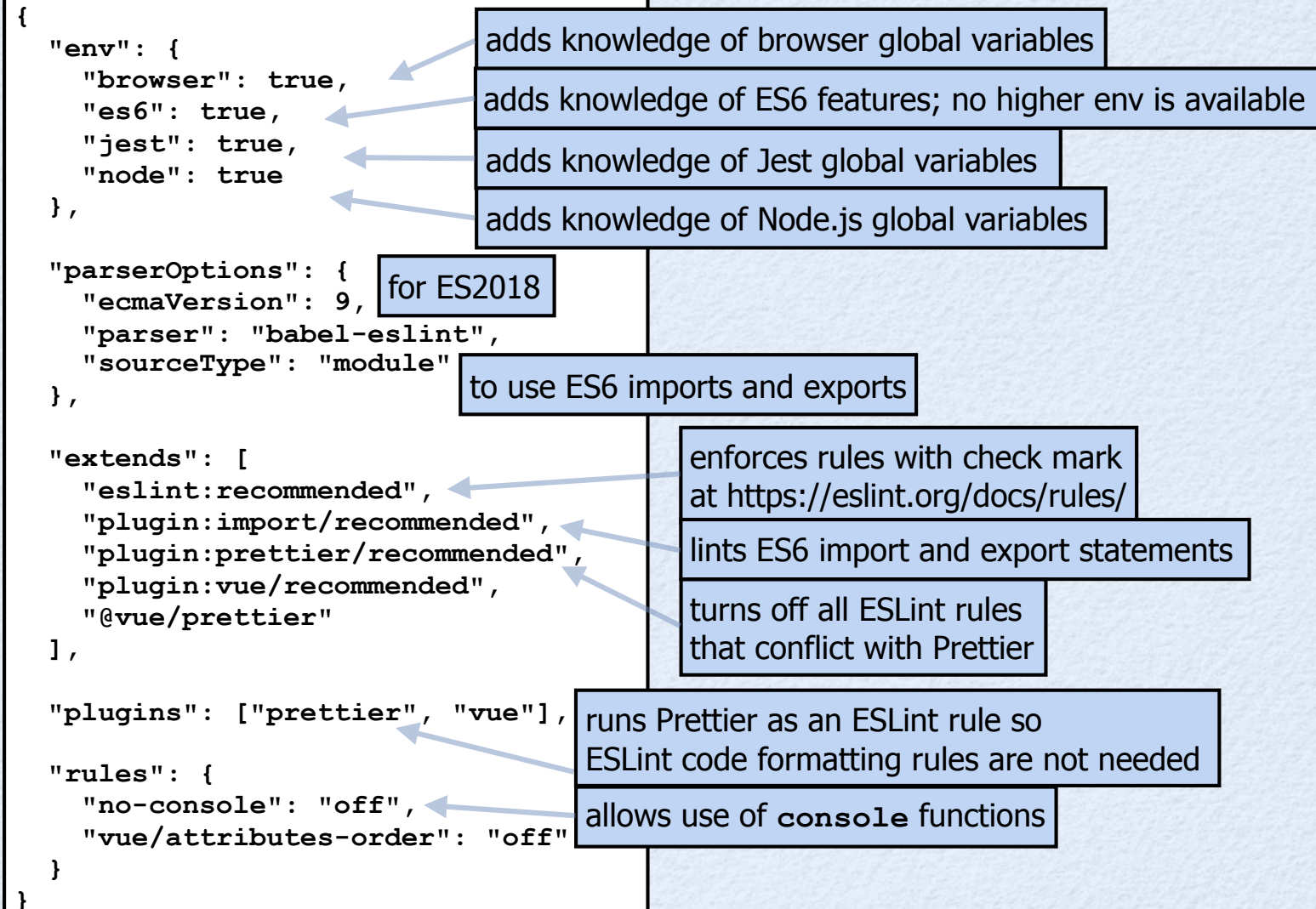
  - Vue CLI projects use this instead `"lint": "vue-cli-service lint",`

- Editor/IDE integrations available

  - Atom, Eclipse, emacs, Intellij IDEA, Sublime, VS Code, Vim, WebStorm
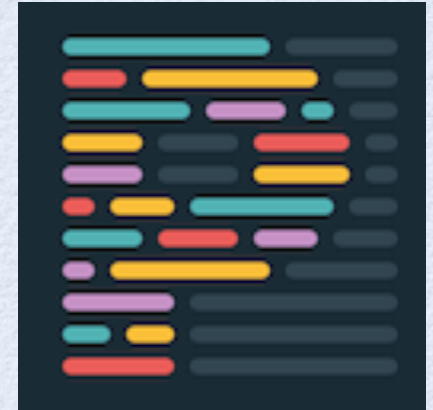
Recommended ESLint Config

for Vue projects

```json
{
  "env": {
    "browser": true,
    "es6": true,
    "jest": true,
    "node": true
  },

  "parserOptions": {
    "ecmaVersion": 9,
    "parser": "babel-eslint",
    "sourceType": "module"
  },

  "extends": [
    "eslint:recommended",
    "plugin:import/recommended",
    "plugin:prettier/recommended",
    "plugin:vue/recommended",
    "@vue/prettier"
  ],

  "plugins": ["prettier", "vue"],

  "rules": {
    "no-console": "off",
    "vue/attributes-order": "off"
  }
}
```

adds knowledge of browser global variables

adds knowledge of ES6 features; no higher env is available

adds knowledge of Jest global variables

adds knowledge of Node.js global variables

for ES2018

to use ES6 imports and exports

enforces rules with check mark at https://eslint.org/docs/rules/

lints ES6 import and export statements

turns off all ESLint rules that conflict with Prettier

runs Prettier as an ESLint rule so ESLint code formatting rules are not needed

allows use of `console` functions

Vue Tools

Prettier Overview

- "An opinionated JavaScript formatter ...
  with advanced support for language features
  from ES2017, JSX, Flow, TypeScript, CSS, LESS, and SCSS"

- "Parses your JavaScript into an AST and pretty-prints the AST,
  completely ignoring any of the original formatting"

  - "Well actually, **some original styling is preserved** when practical -
    see empty lines and multi-line objects."

- Can also format JSON, Markdown, and more

Prettier Details

- To install, `npm install -D prettier`

- To use from an npm script, add following to `package.json`

```
"format": "prettier --no-bracket-spacing --single-quote --write 'src/**/*.{js,vue}'",
```

  - to format all matching files under `src` directory, enter `npm run format`

  - `--write` option overwrites existing files with formatted versions

Must have quotes around glob path!
(see https://prettier.io/docs/en/cli.html)

in Windows use escaped double-quotes

```
--write \"src/**/*.{css,js}\"",
```

- Can also configure in `.prettierrc` file

```
{
  "bracketSpacing": false,
  "singleQuote": true
}
```

- Doesn't run on files under `node_modules` by default

- Editor/IDE integrations available

  - Atom, Emacs, JetBrains, Sublime, Vim, VS Code

Prettier Options

- **--jsx-bracket-same-line**

  - puts closing > of JSX start tags on last line instead of on new line

```
<something
  prop1="value1"
  prop2="value1"
  prop3="value1"
  prop4="value1"
>
  content
</something>
```
VS.
```
<something
  prop1="value1"
  prop2="value1"
  prop3="value1"
  prop4="value1">
  content
</something>
```

- ⭐ **--no-bracket-spacing**

  - omits spaces between brackets in object literals

`{ foo='1' bar=true }` VS. `{foo='1' bar=true}`

- **--no-semi** - omits semicolons

- **--parser** - for use with TypeScript, set to `"typescript"`

- **--print-width** *n* - defaults to 80

- ⭐ **--single-quote**

  > This may become the default. See
  > https://github.com/prettier/prettier/issues/4102

  - uses single quotes instead of double quotes for string delimiters

- **--tab-width** *n* - defaults to 2

- **--trailing-comma**

  - adds trailing commas wherever possible; defaults to none

- **--use-tabs** - uses tabs instead of spaces for indentation

8

VS Code Recommended Extensions

- **ESLint**

- **Prettier**

- **Vetur** for VS Code

  - adds syntax highlighting, snippets, Emmet support, auto-completion, debugging, and more

  - currently uses too much CPU, at least in macOS

  - change `vetur.validation.template` setting to `false` to rely on ESLint and Prettier

  - change `eslint.autoFixOnSave` setting to `true` to run ESLint

  - change `editor.formatOnSave` setting to `true` to run Prettier

- **Vue.js Extension Pack** for VS Code

  - includes several other extensions including Vetur, Vue Peek, ESLint, Prettier, and more

VS Code User Settings

- To auto-fix ESLint issues in `.html`, `.js`, and `.vue` files

```
"eslint.validate": [
  {
    "language": "html",
    "autoFix": true
  },
  {
    "language": "javascript",
    "autoFix": true
  },
  {
    "language": "vue",
    "autoFix": true
  }
],
```

Review

- Install some npm modules

  - `npm install -D eslint-config-prettier`

  - `npm install -D eslint-plugin-html`

  - `npm install -D eslint-plugin-import`

  - `npm install -D eslint-plugin-prettier`

  - `npm install -D @vue/eslint-config-prettier`

  - `npm install -D prettier`

  > in vue-workshop repo
  > enter `. ./tool-installs`

- Create `.eslintrc.json` file with content on slide 5   `copy from vue-workshop repo`

- Create `.prettierrc` file with content on slide 7   `copy from vue-workshop repo`

- Add following script to `package.json`

  - `"format": "prettier --write 'src/**/*.{js,vue}'",`

- To lint all `.js` and `.vue` files in project

  - `npm run lint`

- To format all `.js` and `.vue` files in project

  - `npm run format`

11