Как внедрить стандарты разработки, чтобы никто не пострадал

Александра Шинкевич (@neesoglasnaja)

Кто я?

- ~~Глава отдела фронтенда, LOVATA~~
- Full-stack Node.js разработчик
- Счастливая безработная

Minsk CSS

MinskJS

CSS MINSK JS

О чем буду говорить?

- Что такое и зачем нужны стандарты разработки
- Как унифицировать code style (HTML, CSS, JS)
- Как автоматизировать его проверку
- Как внедрить проверку в CI

Стандарты разработки

- Guidelines
- Styleguide
- Code style

Guidelines VS Styleguide

Guidelines

1.2 Trademarks

The following marks are applicable for materials created for use in North America and for global distribution.

RIM Registered Marks and Trademarks

When referencing BlackBerry products and services, use RIM Marks as adjectives. Avoid using the RIM Marks generically, as nouns or verbs and do not use them in the plural or possessive form. Also, be sure to follow the additional rules set out in the BlackBerry Trademark Rules section on this page.

RIM Marks that appear in the following list with a registered trademark symbol (®) are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries.

The absence of a RIM mark from this list does not mean that RIM does not use the mark, that the mark is not a registered trademark or trademark of RIM or that the

BlackBerry Trademark Rules

1. **Use the RIM Marks as adjectives, not nouns.**
   Avoid using the RIM Marks as nouns. A trademark is an adjective to be used with the noun it modifies. The BlackBerry wordmark should never be used alone.
   ✓ The BlackBerry® smartphone is...
   ✗ ~~The BlackBerry® is...~~

2. **Use the RIM Marks as adjectives, not verbs.**
   Avoid using the RIM Marks to describe the performance of an act.
   ✓ I will respond to your email using my BlackBerry® smartphones
   ✗ ~~I will BlackBerry you.~~

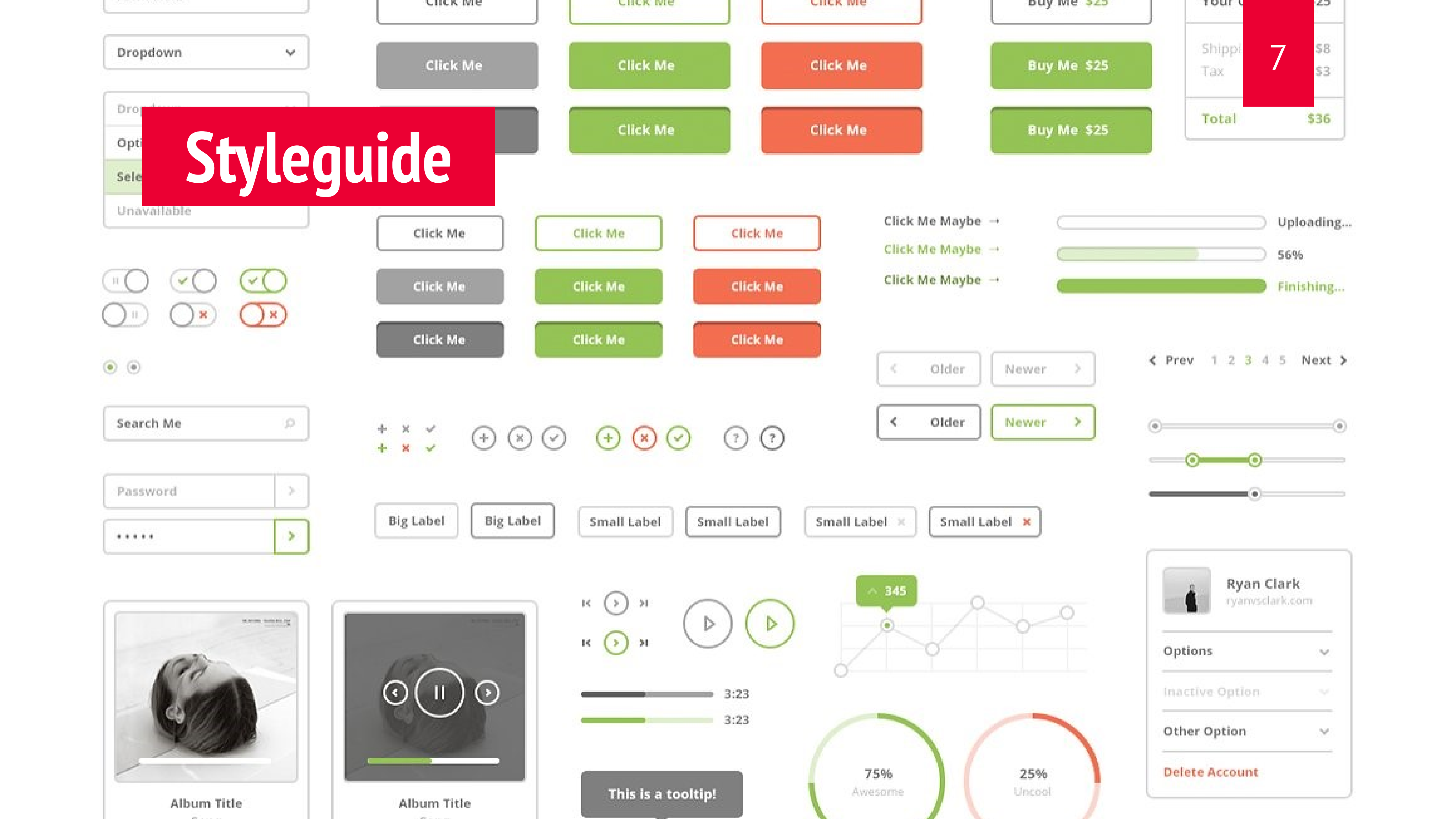
3. **Do not use the RIM Marks in plural or possessive form.**
   ✓ BlackBerry® smartphones
   ✗ ~~BlackBerrys, BlackBerries, BlackBerry's~~

7. **Use a Notice of Ownership and Disclaimer.** A prominent notice should be used when any of the RIM Marks appear on materials or web sites. See the "Trademark Disclaimer" section for the appropriate trademark notice.

8. **Do not alter the RIM or BlackBerry logos.**
   Incorrect use of the RIM or BlackBerry logos compromises the integrity and effectiveness of the logos. To ensure accurate and consistent reproduction of the logos, never alter, add to or attempt to recreate the logos. Always use the approved digital work available from RIM Marketing Communications (marcomm@rim.com) You may only use the RIM or BlackBerry logos or RIM Marks if you have obtained prior approval from RIM Marketing Communications and your use complies with these guidelines.

Trademark Disclaimer

Styleguide


Dropdown
Unavailable
Click Me
Buy Me $25
Shipping $8
Tax $3
Total $36
Click Me Maybe →
Uploading...
56%
Finishing...
Older
Newer
Prev 1 2 3 4 5 Next
Search Me
Password
Big Label
Small Label
345
Ryan Clark
ryanvsclark.com
Options
Inactive Option
Other Option
Delete Account
3:23
This is a tooltip!
75%
Awesome
25%
Uncool
Album Title

Code Style

Правила, по которым надо писать код

- Быстрое восприятие кода
- Предотвращение ошибок
- Подсказки при написании кода

Linters

Автоматическая проверка кода

Linters

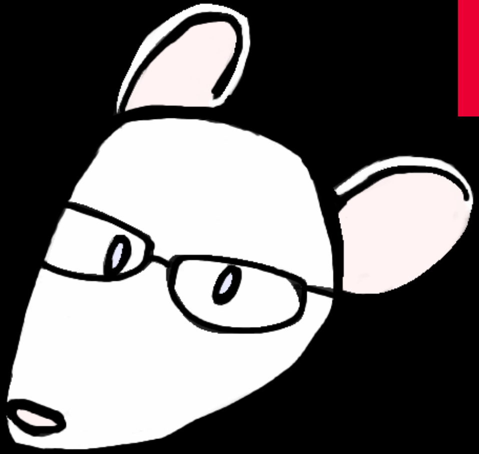
Работают, только если…

- Правила хранятся в репозитории проекта
- Они внедрены в процесс сборки
- Ошибки видно в редакторе
- Невалидный код не попадает в общую кодовую базу

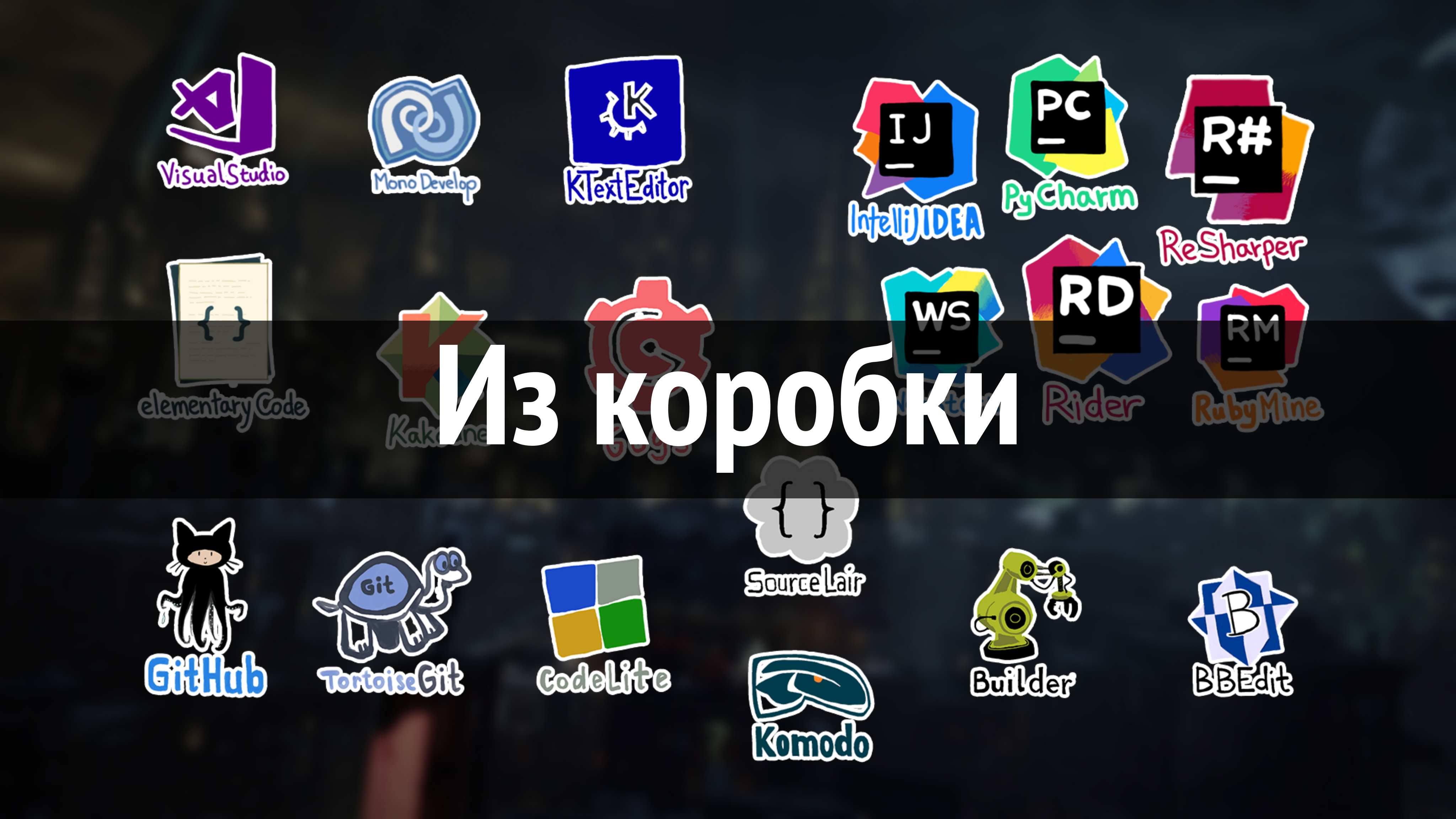
С чего начать?

С настройки редактора

.editorconfig

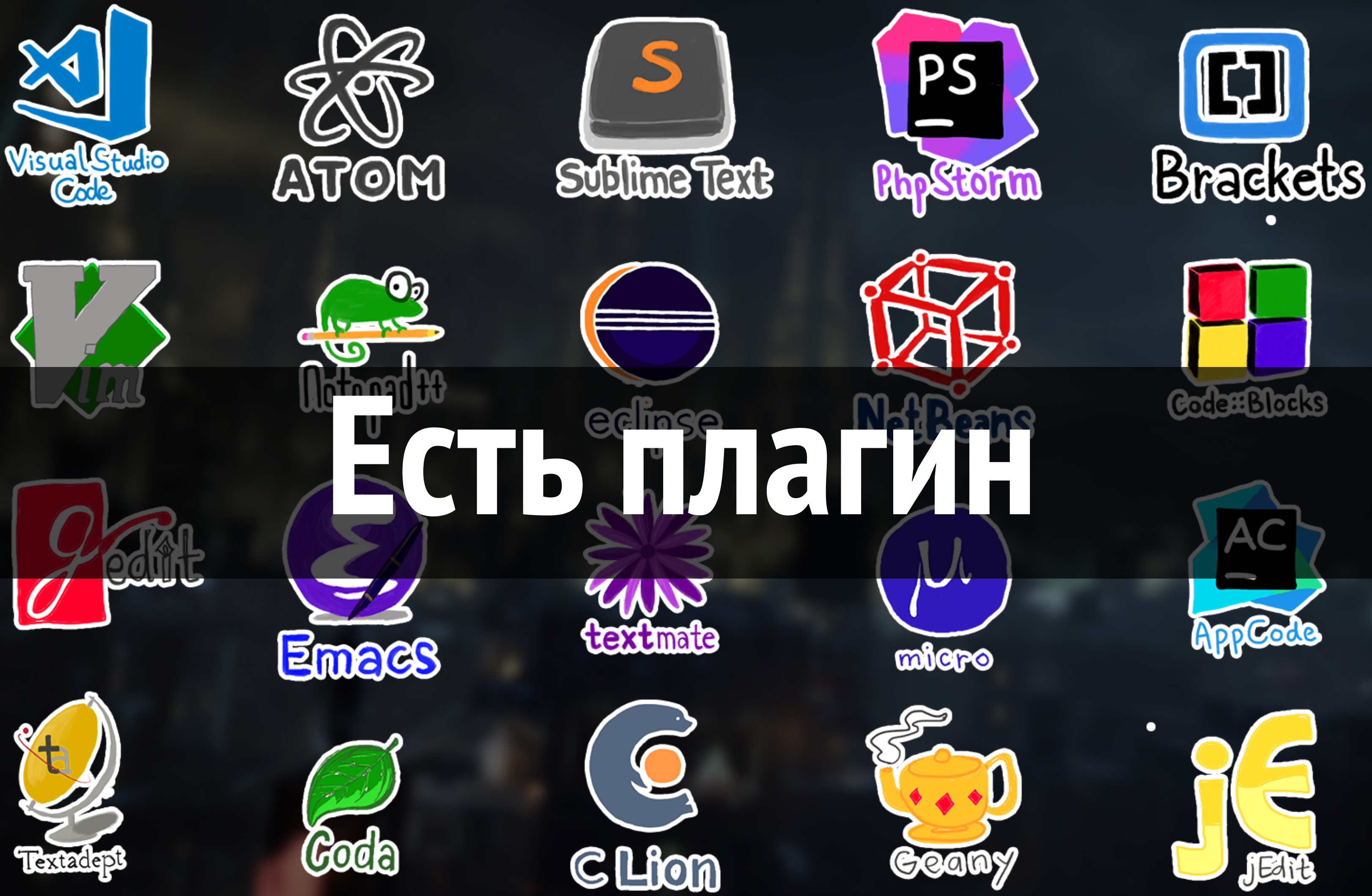


- Единый формат настроек для всех IDE
- ~10 правил
- Разные настройки для разных форматов файлов
- Конец войны «табы или пробелы»



Visual Studio
Mono Develop
KTextEditor
IntelliJ IDEA
PyCharm
ReSharper
elementary Code
Rider
RubyMine
GitHub
TortoiseGit
CodeLite
Source Lair
Komodo
Builder
BBEdit

Из коробки


Visual Studio Code
ATOM
Sublime Text
PhpStorm
Brackets
Vim
Notepad++
eclipse
NetBeans
Code::Blocks
gedit
Emacs
textmate
micro
AppCode
Textadept
Coda
CLion
Geany
jEdit

Есть плагин

Формат .editorconfig

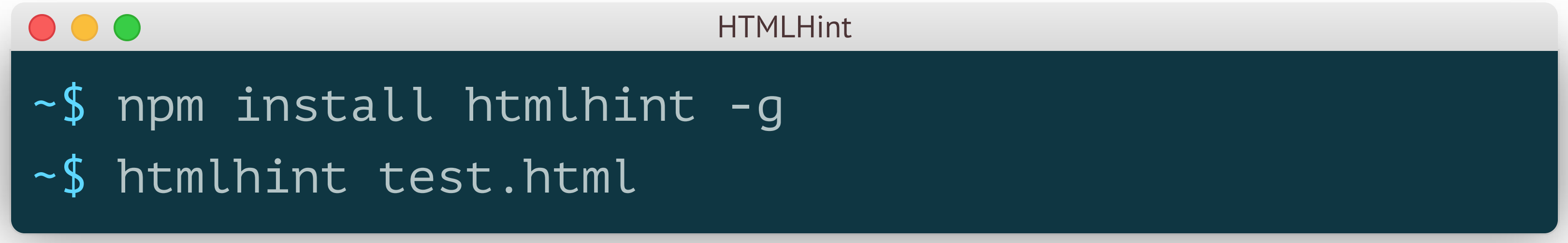
```
root = true

[*]
end_of_line = lf
insert_final_newline = true
charset = utf-8
indent_style = space
indent_size = 4

[*.{md,jade}]
indent_style = tab
```

HTML

HTML

HTMLHint

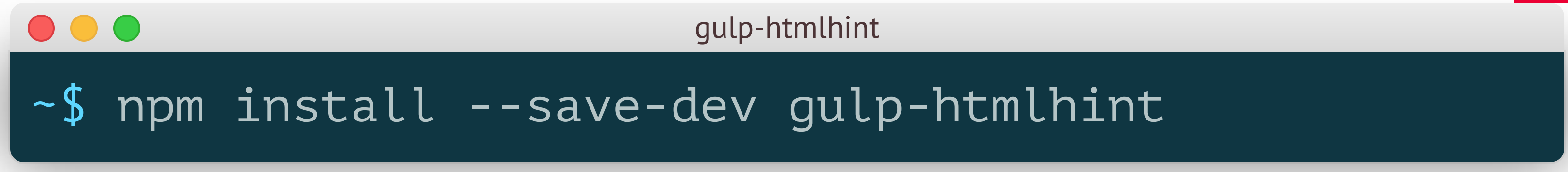
yaniswang/HTMLHint


HTMLHint

```
~$ npm install htmlhint -g
~$ htmlhint test.html
```

.htmlhintrc

```
{
   "tagname-lowercase": true,
   "attr-lowercase": true,
   "attr-value-double-quotes": true,
   "doctype-first": true,
   "tag-pair": true,
   "spec-char-escape": true,
   "id-unique": true,
   "src-not-empty": true,
   "attr-no-duplication": true,
   "title-require": true
}
```



```
~$ npm install --save-dev gulp-htmlhint
```

```
const htmlhint = require('gulp-htmlhint');
gulp
    .src('./src/*.html')
    .pipe(htmlhint());
```

grunt-htmlhint

```
~$ npm install --save-dev grunt-htmlhint
```

```
grunt.loadNpmTasks('grunt-htmlhint');
...
htmlhint: {
  options: {
    htmlhintrc: '.htmlhintrc'
  },
}
```

htmlhint-loader

```
~$ npm install --save-dev htmlhint-loader
```

```
module.exports = {
  module: {
    rules: [{
      enforce: 'pre',
      test: /\.html/,
      loader: 'htmlhint-loader',
      exclude: /node_modules/
    }]
  }
}
```

На любой вкус

- htmllint
- html-lint
- pug-lint
- markdownlint
- ...

CSS

CSS

stylelint

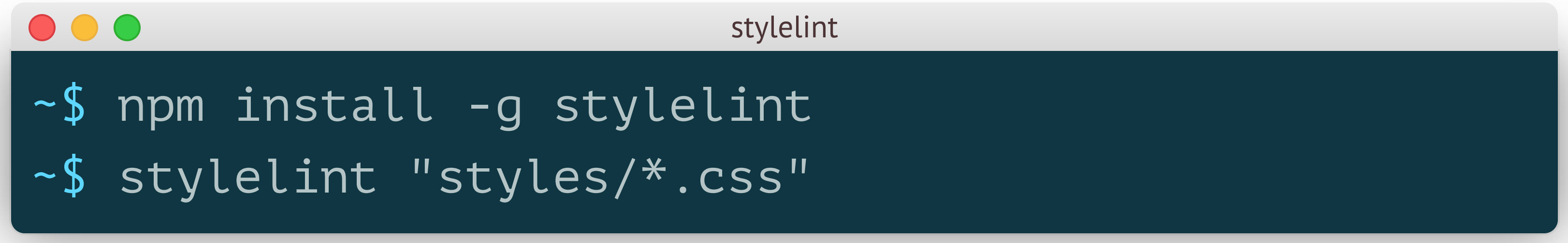
stylelint/stylelint

Stylelint

- 170 встроенных правил
- Возможность написания и подключения плагинов
- Возможность наследования и расширения конфигураций
- Работает с препроцессорными кодом (SCSS, Less, PostCSS)

.stylelintrc

```
{
  "extends": "stylelint-config-recommended",
  "rules": {
    "at-rule-no-unknown": [ true, {
      "ignoreAtRules": ["extends"]
    }],
    "block-no-empty": null,
    "unit-whitelist": ["em", "rem", "s"]
  }
}
```

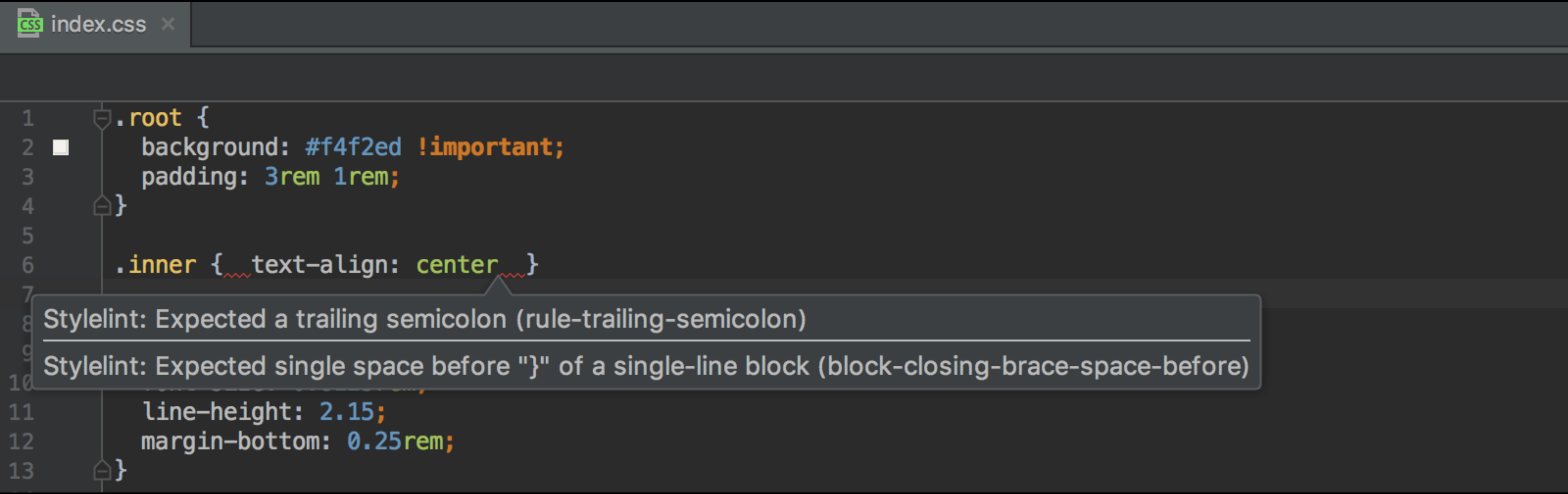

stylelint

```
~$ npm install -g stylelint
~$ stylelint "styles/*.css"
```

Плагины для редакторов

stylelint.io/editor-plugins

WebStorm

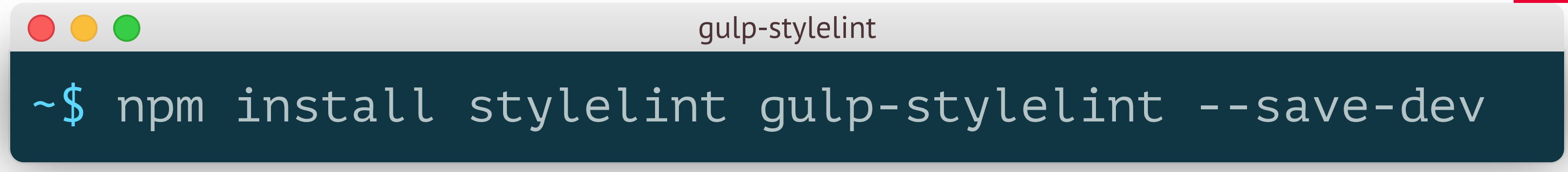


index.css

```
.root {
  background: #f4f2ed !important;
  padding: 3rem 1rem;
}

.inner { text-align: center }

  line-height: 2.15;
  margin-bottom: 0.25rem;
}
```

Stylelint: Expected a trailing semicolon (rule-trailing-semicolon)

Stylelint: Expected single space before "}" of a single-line block (block-closing-brace-space-before)


gulp-stylelint

```
~$ npm install stylelint gulp-stylelint --save-dev
```

```
const stylelint = require('gulp-stylelint');
gulp
    .src('./src/*.css')
    .pipe(gulpStylelint({
        reporters: [
          { formatter: 'string', console: true }
        ]
    }));
```

grunt-stylelint

```
~$ npm install stylelint grunt-stylelint --save-dev
```

```
grunt.loadNpmTasks('grunt-stylelint');
...
grunt.initConfig({
  stylelint: {
    all: ['css/**/*.css', 'sass/**/*.scss']
  }
});
```

stylelint-webpack-plugin

```
~$ npm install stylelint-webpack-plugin --save-dev
```

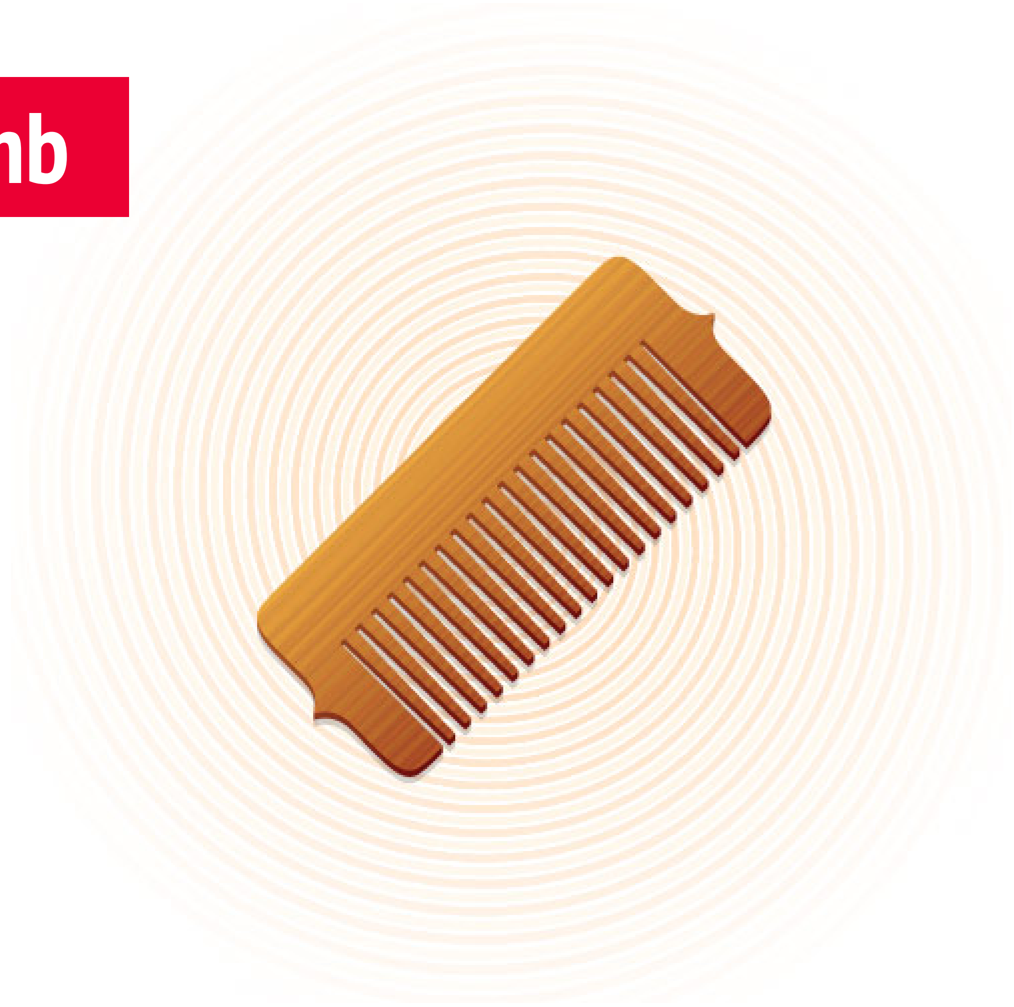
```
const StyleLintPlugin = require('stylelint-webpack-plugin');

module.exports = {
  plugins: [
    new StyleLintPlugin(options),
  ]
}
```

Хочется больше?

- sass-lint
- stylelint-scss
- lesshint
- stylint
- stylelint-selector-bem-pattern

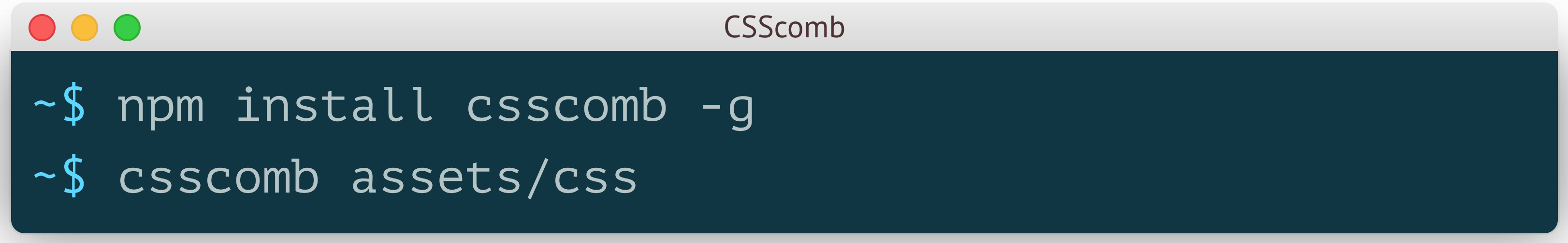
- stylelint.io/plugins

CSScomb



CSSComb

- 24 встроенных правила
- Возможность сортировки любых свойств по блокам
- Встроенные конфигурации
- `(master) Latest commit on 16 Feb 2017`



```
~$ npm install csscomb -g
~$ csscomb assets/css
```

.csscomb.json

```
{
  "always-semicolon": true,
  "eof-newline": true,
  "leading-zero": true,
  "quotes": "single",
  ...,
  "sort-order": [
    [
      "position",
      "z-index"
    ], [
      "display",
      "visibility"
    ]
  ]
}
```

Плагины для редакторов

hudochenkov/postcss-sorting

davidhund/styleguide-generators

JavaScript

JS

ESLint

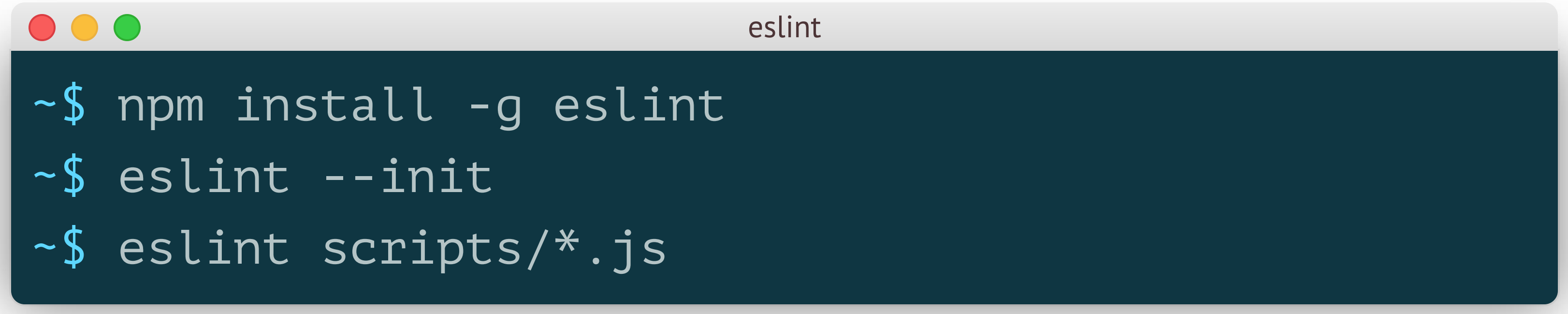
eslint/eslint

ESLint

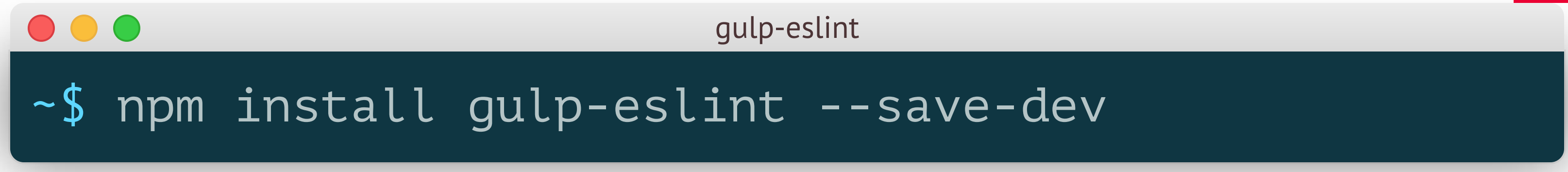
- 250+ встроенных правил
- Возможность написания и подключения плагинов
- Возможность наследования и расширения конфигураций
- Поддерживает различные версии стандартов (ES5/ES6/ESNext)

.eslintrc

```
{
  "parserOptions": {
    "parser": "babel-eslint",
    "sourceType": "module"
  },
  "env": { "browser": true },
  "extends": ["airbnb-base"],
  "plugins": ["vue"],
  "rules": {
    "semi": ["error", "always"],
    "no-debugger": "warn"
  }
}
```



```
~$ npm install -g eslint
~$ eslint --init
~$ eslint scripts/*.js
```



```
~$ npm install gulp-eslint --save-dev
```

```
const eslint = require('gulp-eslint');
gulp
    .src('./scripts/*.js')
    .pipe(eslint())
    .pipe(eslint.format())
    .pipe(eslint.failAfterError());
```

grunt-eslint

```
~$ npm install grunt-eslint --save-dev
```

```
require('load-grunt-tasks')(grunt);

grunt.initConfig({
  eslint: {
    target: ['file.js']
  }
});
```

eslint-loader

```
~$ npm install eslint eslint-loader --save-dev
```

```
module.exports = {
  module: {
    rules: [{
      test: /\.js$/,
      exclude: /node_modules/,
      loader: "eslint-loader",
      options: { emitError: true }
    }]
  }
}
```

Плагины для редакторов

eslint.org/integrations

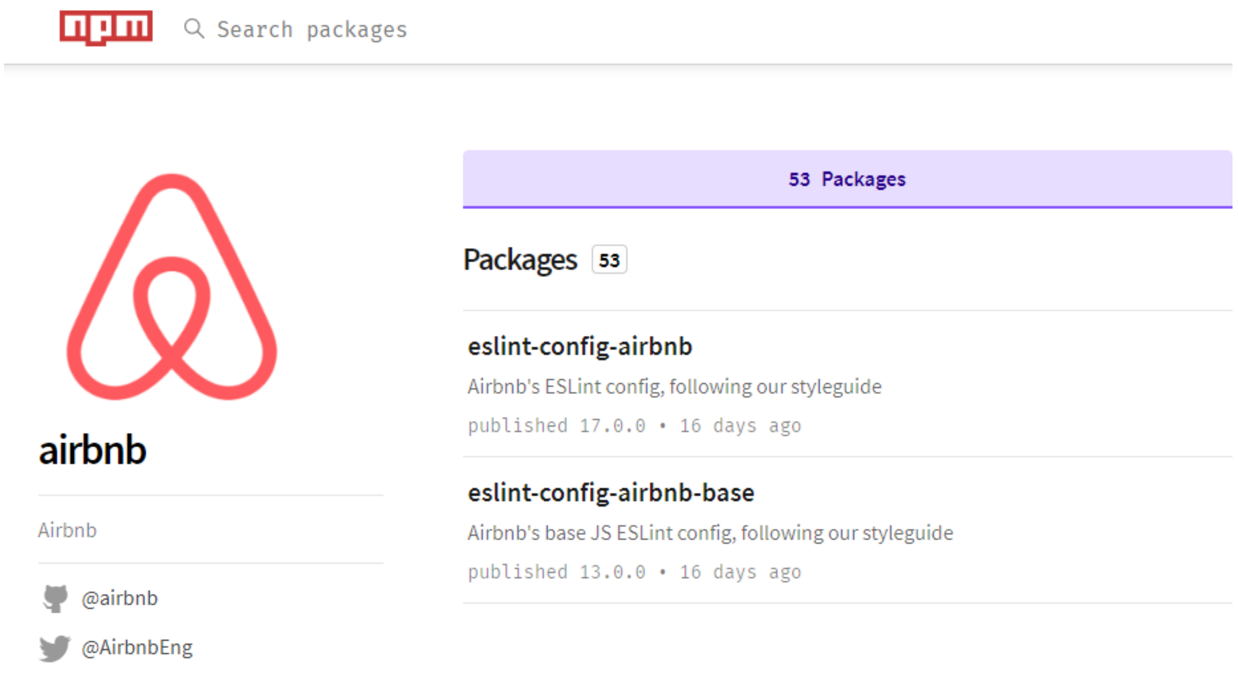
250+ правил?!

dustinspecker/awesome-eslint

StandardJS

standard JS

Airbnb ESLint config



airbnb

Airbnb

@airbnb

@AirbnbEng

Packages 53

eslint-config-airbnb

Airbnb's ESLint config, following our styleguide

published 17.0.0 • 16 days ago

eslint-config-airbnb-base

Airbnb's base JS ESLint config, following our styleguide

published 13.0.0 • 16 days ago

npm eslint-config-lovata

А если не просто JS?

- babel-eslint
- tslint
- eslint-plugin-flowtype

- SonarJS

Prettier

prettier/prettier

JS Doc

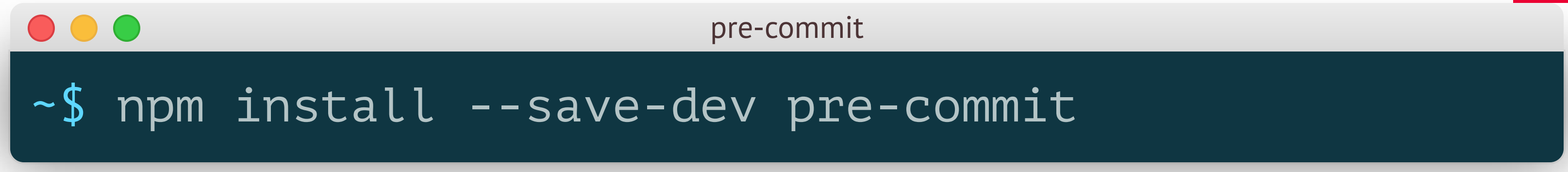
JavaScript

usejsdoc.org

Запрещаем коммитить плохой код

.git/hooks/pre-commit

```
#! /bin/bash
if !(npm run lint:js --silent)
  exit 1;
fi
```



```
~$ npm install --save-dev pre-commit
```

```
{
  "name": "my-project",
  "scripts": {
    "lint:js": "eslint ./js/*.js",
    "test": "echo '🤣 No' && exit 1"
  },
  "pre-commit": [
    "lint:js",
    "test"
  ]
}
```

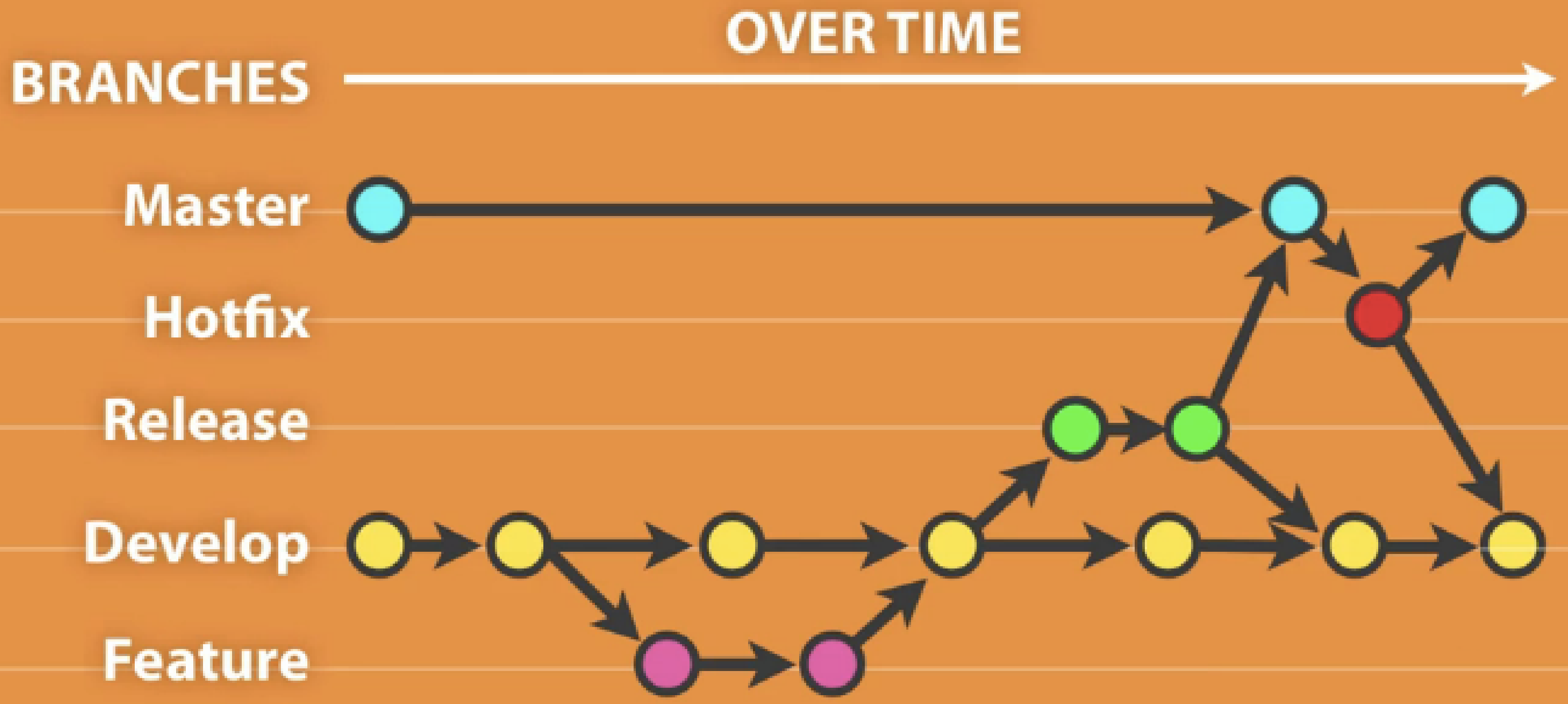
Выбери свой вариант

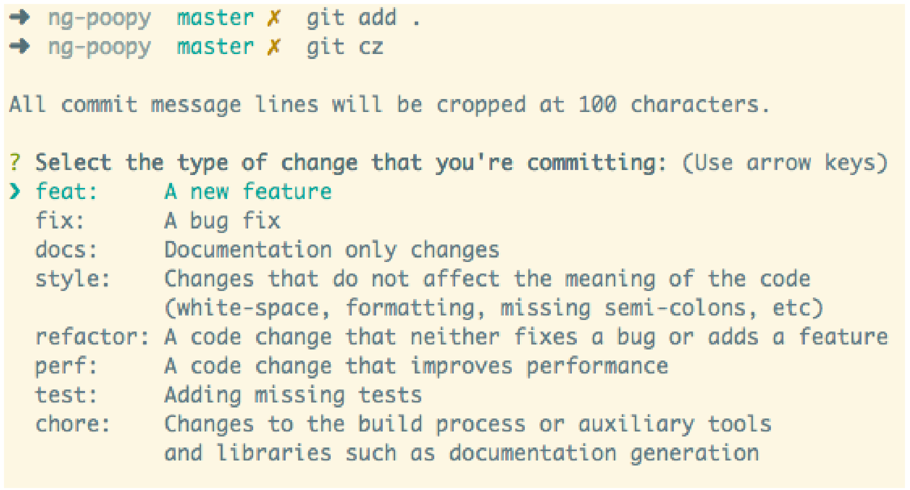
- observing/pre-commit
- okonet/lint-staged
- typicode/husky

Как работать с репозиторием

COMMIT
B MASTER
GIT-FLOW
imgflip.com

“Gitflow”





```
➜ ng-poopy  master ✗  git add .
➜ ng-poopy  master ✗  git cz

All commit message lines will be cropped at 100 characters.

? Select the type of change that you're committing: (Use arrow keys)
❯ feat:     A new feature
  fix:      A bug fix
  docs:     Documentation only changes
  style:    Changes that do not affect the meaning of the code
            (white-space, formatting, missing semi-colons, etc)
  refactor: A code change that neither fixes a bug or adds a feature
  perf:     A code change that improves performance
  test:     Adding missing tests
  chore:    Changes to the build process or auxiliary tools
            and libraries such as documentation generation
```

commitizen/cz-cli

WILL
CRIME FIGHT
FOR
FOOD

Проблема 1.
Слишком строгие правила

Решение: понижать в local-dev-сборке «мешающие» свойства до `warn`

- Создаем еще 1 файл конфигурации `.local-dev.eslintrc`
- В нем — `extend` от существующей конфигурации
- Добавляем правила с уровнем `warn`
- В сборщике по ключу ENV подключаем нужную конфигурацию

Пример для Webpack

```
const isLocal = process.env.ENV_LOCAL === "local";

module.exports = {
  module: {
    rules: [{
      ...
      loader: "eslint-loader",
      options: {
        configFile: isLocal ? ".local-dev.eslintrc" : ".eslintrc"
      }
    }]
  }
}
```

Проблема 2.
Плохой код попадает в репозиторий

Решение: делать проверки в CI

- Bitbucket Pipelines
- Travis
- Jenkins
- TeamCity

Bitbucket Pipelines

```
pipelines:
  default:
    - step:
        caches:
          - node
        name: Source code linting
        script:
          - npm i
          - npm run lint
```

Подводя итог

- Настроить редактор
- Подключить линтеры
- Вести документацию
- Бить по рукам с помощью `git hooks`
- Настроить CI

Материалы по теме

- Кодстайл и насилие — Антон Немцев [Youtube]
- Качество кода — Иван Ботанов [Youtube]
- Как оптимизировать фронтенд — Мария Кабаш [Youtube]

Вопросы?

bit.ly/Standards-ChernivtsiJS

Александра Шинкевич (@neesoglasnaja)