



Drupal Europe

Darmstadt, Germany

Sep 10 - 14, 2018



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Drupal + Technology

TRACK SUPPORTED BY

platform.sh 



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Code with style! Prettier + ESLint



David Corbacho Román

<https://www.drupaleurope.org/session/code-style-prettier-eslint>

**We want to deliver software
without flaws.**



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



How to deliver software without flaws.

Ask your team members to review your code



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Code reviews



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



reviewed on 6 May 2015

[View changes](#)

frontend/tests/unit/models/assignment-test.js

Hide outdated

```
...    ...    @@ -1,18 +1,20 @@
1      1      import { moduleForModel, test } from 'ember-qunit'
2      2
3      3      +
```



on 6 May 2015

Owner

Minor style nitpick, this blank line is not needed



Code reviews



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



████████ reviewed on 6 May 2015

[View changes](#)

backend/routes/api/v1/task.js

Hide outdated

```
...    ...    @@ -1,7 +1,7 @@
1      1      import { Router }           from 'express'
2      2      import mongoose         from 'mongoose'
3      3      import { NotFoundError }  from '../.../src/error'
4      - import { Task, Attendance } from '../.../models'
4      + import { Task, Activity } from '../.../models'
```



████████ on 6 May 2015 Owner

Minor style nitpick, align from with other from keywords



Code reviews



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



████████ reviewed on 6 May 2015

[View changes](#)

backend/models/activity.js

 Hide outdated

```
7 + 'military',  
8 + 'school',  
9 + 'civil defence'  
10 + ]}
```



████████ on 6 May 2015 Owner

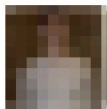
I don't really like this indentation, what about:



Code reviews



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



 reviewed on 12 Jun

public/styles.css

```
360 + font-size: 15px;  
361 + line-height:20px;  
362 + color: #333333;  
363 + }
```



 on 12 Jun Contributor

Watch the indentation and newline here again!
Add a space after colon



friction *noun* [U] (DISAGREEMENT)

★ **disagreement or unfriendliness caused by people having different opinions:**

*There's a lot of friction **between** my wife and my mother.*



Unpopular opinion:

Maybe... perfect formatting is not *that* important



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Unpopular opinion:

Maybe... perfect formatting is not **that** important



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



... and one semicolon here

Following a convention is important



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



How to deliver software without flaws.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Static Code Analysis

Tools that inspect the code **without** executing the software.

Linters

Dynamic Code Analysis

aka “testing”



How to deliver software without flaws.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Static Code Analysis

Tools that inspect the code **without** executing the software.

Linters

CODE SMELL

PATTERNS
BEST PRACTICES
SECURITY FLAWS

STYLE



Linting in JavaScript. Evolution



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



2002 by Douglas
Crockford, JSON
creator.



O'REILLY | YAKOPE PRESS | Douglas Crockford



Linting evolution



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

JSLint

2002 by Douglas
Crockford, JSON
creator.
Too opinionated.



O'REILLY | YAKOOL PRESS | Douglas Crockford



Linting evolution



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



2002 by Douglas
Crockford, JSON
creator.
Too opinionated.



2011. Fork of JSLint

Community driven
Not opinionated
More configurable



Linting evolution



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



2002 by Douglas
Crockford, JSON
creator.

Too opinionated.



2011. Fork of JSLint

Community driven

Not opinionated
More configurable



2013 by Nicholas Zakas.

Not opinionated
Lots of configuration options

Extensible with plugins

- + Supports NodeJS
- + Keeps up with ES evolution
- + In 2014 support for JSX



ESLint demo



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
1 var day;  
2 let week = 5;  
3 console.log(week);  
4
```

Messages

Fixed Code

1:5 - 'day' is defined but never used. (no-unused-vars)

2:5 - 'week' is never reassigned. Use 'const' instead. (prefer-const)

3:1 - Unexpected console statement. (no-console)



Too much configuration



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Secure | <https://eslint.org/docs/rules/>

ESLint Search the docs... User guide - Developer guide - Blog Demo - About

Rules

Rules in ESLint are grouped by category to help you understand their purpose.

No rules are enabled by default. The `"extends": "eslint:recommended"` property in a configuration file enables rules that report common problems, which have a check mark ✓ below.

The `--fix` option on the command line automatically fixes problems (currently mostly whitespace) reported by rules which have a wrench ⚙ below.

Possible Errors

These rules relate to possible syntax or logic errors in JavaScript code:

✓	for-direction	enforce "for" loop update clause moving the counter in the right direction.
✓	getter-return	enforce <code>return</code> statements in getters
	no-async-promise-executor	disallow using an async function as a Promise executor
	no-await-in-loop	disallow <code>await</code> inside of loops
✓	no-compare-neg-zero	disallow comparing against <code>-0</code>
✓	no-cond-assign	disallow assignment operators in conditional expressions
✓	no-console	disallow the use of <code>console</code>
✓	no-constant-condition	disallow constant expressions in conditions
✓	no-control-regex	disallow control characters in regular expressions
✓	no-debugger	disallow the use of <code>debugger</code>
✓	no-dupe-args	disallow duplicate arguments in <code>function</code> definitions
✓	no-dupe-keys	disallow duplicate keys in object literals
✓	no-duplicate-case	disallow duplicate case labels
✓	no-empty	disallow empty block statements
✓	no-empty-character-class	disallow empty character classes in regular expressions



Too much configuration



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Secure | <https://eslint.org/docs/rules/>

ESLint

Search the docs... User guide - Developer guide - Blog Demo - About

Rules

Rules in ESLint are grouped by category to help you understand their purpose.

No rules are enabled by default. The `"extends": "eslint:recommended"` property in a configuration file enables rules that report common problems, which have a check mark ✓ below.

The `--fix` option on the command line automatically fixes problems (currently mostly whitespace) reported by rules which have a wrench ⚙ below.

Possible Errors

These rules relate to possible syntax or logic errors in JavaScript code:

✓	for-direction	enforce "for" loop update clause moving the counter in the right direction.
✓	getter-return	enforce <code>return</code> statements in getters
	no-async-promise-executor	disallow using an async function as a Promise executor
	no-await-in-loop	disallow <code>await</code> inside of loops
✓	no-compare-neg-zero	disallow comparing against <code>-0</code>
✓	no-cond-assign	disallow assignment operators in conditional expressions
✓	no-console	disallow the use of <code>console</code>
✓	no-constant-condition	disallow constant expressions in conditions
✓	no-control-regex	disallow control characters in regular expressions
✓	no-debugger	disallow the use of <code>debugger</code>
✓	no-dupe-args	disallow duplicate arguments in <code>function</code> definitions
✓	no-dupe-keys	disallow duplicate keys in object literals
✓	no-duplicate-case	disallow duplicate case labels
✓	no-empty	disallow empty block statements
✓	no-empty-character-class	disallow empty character classes in regular expressions

ESLint "Shareable configurations"

- [Airbnb](#) - Shareable config for [Airbnb's style guide](#) ←
- [Canonical](#) - Shareable config for [Canonical style guide](#)
- [ESLint](#) - Contains the ESLint configuration used for projects maintain
- [ES](#) - Shareable config for very strict code
- [Facebook](#) - Sharable config for Facebook's style guide.
- [Google](#) - Shareable config for the [Google style](#)
- [Problems](#) - Shareable config that only catches actual problems, and
- [Shopify](#) - Shareable config for [Shopify's style guide](#)
- [Standard](#) - Shareable config for JavaScript [Standard Style](#)
- [Supermind](#) - Shareable config for Supermind style
- [XO](#) - Shareable config for [XO](#)

<https://github.com/dustinspecker/awesome-eslint>



Airbnb JS Style guide



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

2,728,137 repository results Sort: Best match ▾

freeCodeCamp/freeCodeCamp	JavaScript	★ 294k
BSD-3-Clause license Updated a day ago 39 issues need help		
twbs/bootstrap	CSS	★ 127k
MIT license Updated 6 hours ago 10 issues need help		
vuejs/vue	JavaScript	★ 113k
MIT license Updated 3 days ago 2 issues need help		
EbookFoundation/free-programming-books		★ 111k
Updated a day ago 5 issues need help		
facebook/react	JavaScript	★ 111k
MIT license Updated 3 hours ago 3 issues need help		
tensorflow/tensorflow	C++	★ 109k
Apache-2.0 license Updated 2 hours ago		
sindresorhus/awesome		★ 92.1k
Updated 5 days ago 5 issues need help		
getify/You-Dont-Know-JS		★ 88k
Updated 12 days ago		
d3/d3	JavaScript	★ 78.6k
BSD-3-Clause license Updated 10 days ago		
robbyrussell/oh-my-zsh	Shell	★ 76k
Updated 3 hours ago		
airbnb/javascript	JavaScript	★ 75.8k
MIT license Updated 3 days ago		

11th



Airbnb JS Style guide

Excellent opinionated guide to ES6 using ESLint rules.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

References

- 2.1 Use `const` for all of your references; avoid using `var`. eslint: `prefer-const`, `no-const-assign`

Why? This ensures that you can't reassign your references, which can lead to bugs and difficult to comprehend code.

```
// bad
var a = 1;
var b = 2;

// good
const a = 1;
const b = 2;
```

<https://github.com/airbnb/javascript>



Airbnb JS Style guide.



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Too opinionated? Creates friction in the team?

Escape hatch:
Override Airbnb rules

Example
Drupal 8 core

```
{
  "extends": "eslint-config-airbnb",
  "root": true,
  "env": {
    "browser": true,
    "es6": true,
    "node": true
  },
  "globals": {
    "Drupal": true,
    "drupalSettings": true,
    [ ... ]
  },
  "rules": {
    "consistent-return": [0],
    "no-underscore-dangle": [0],
    "max-nested-callbacks": [1, 3],
    "no-mutable-exports": [1],
    "no-plusplus": [1, {
      "allowForLoopAfterthoughts": true
    }],
    "no-param-reassign": [0],
    "no-prototype-builtins": [0],
    "valid-jsdoc": [1, {
      "prefer": {
        "returns": "return",
        "property": "prop"
      }
    }],
    "requireReturn": false
  },
  "brace-style": ["error", "stroustrup"],
  "no-unused-vars": [1]
}
```



How Linters work? AST parser



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Search the docs...

User guide ▾

Developer guide ▾

Blog

Demo ▾

```
var foo = bar;
```

```
{
  "type": "Program",
  "start": 0,
  "end": 14,
  "body": [
    {
      "type": "VariableDeclaration",
      "start": 0,
      "end": 14,
      "declarations": [
        {
          "type": "VariableDeclarator",
          "start": 4,
          "end": 13,
          "id": {
            "type": "Identifier",
            "start": 4,
            "end": 7,
            "name": "foo"
          },
          "init": {
            "type": "Identifier",
```





WHAT IF...

We use the output of the AST parser to **rewrite** the source code?

gofmt was born. 2013
[Go formatter]

Also rfmt
[Reason formatter]

```
/*  
 * vim: set ft=rust:  
 * vim: set ft=reason:  
 */  
  
open React;  
  
exception Impossible of string;  
  
let oneStepRecurse:  
  type p d r.  
  (unit => bool) =>  
  Id.t =>  
  (list (opaqueGraph r) => priorityConstraint => bool) =>  
  componentClass p d r =>  
  option (instaComp p d r) =>  
  instaComp p d r =>  
  priorityConstraint =>  
  option (instaComp p d r, option mutationlist) =  
  fun shouldContinue id flushIf componentClass prevInstaComp reconcilingInstaComp priConstraint =>  
  let {instance: reconcilingInstance, composition: reconcilingComposition, relativeTransform} =  
  let (backwardsComposition, recursiveStepWasTaken) =  
  compositionOneStepAndReverse false shouldContinue [] reconcilingComposition priConstraint;  
  if (flushIf backwardsComposition priConstraint) {  
  let (nextComposition, forwardMutationListList) = dequeueMutationsReverseAndMarkReconciled b  
  let nextInstaComp = {  
    instance: reconcilingInstance,  
    composition: nextComposition,  
    relativeTransform  
  };  
  /* By convention, mutation lists should always be backwards */  
  let antimodularMutations =  
  componentClass.reconcileAntimodularEffects id prevInstaComp nextInstaComp;  
  let backwardsMutationListListAsUsual = List.rev forwardMutationListList;  
  let flushableMutations = List.concat (  
  maybeAppendToMutationListList antimodularMutations backwardsMutationListListAsUsual  
  );  
  /* We know *at least* that a flush acknowledgement step was taken, and  
  NORMAL ~~/Desktop/reconcile.re[+]  
  reason utf-8[unix] 3% 16: 8
```



WHAT IF...

We do it for JavaScript?
using Wadler's algorithm
from "A prettier printer"
and lots of smart tweaks.

Prettier was born

Jan. 2017

by James Long



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



prettier.io

Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

```
render() {  
  return (  
    <div>  
      <Field>  
        <input  
          type="text"  
          value={  
            this.state.mode === 'edit'  
            ? this.state.exp  
            : format(this.state.val)  
          }  
          onFocus={() => this.setState({ mode: 'edit' })}  
          onBlur={this.onBlur.bind(this)}  
          onChange={this.onChange.bind(this)}  
          onKeyDown={this.onKeyDown.bind(this)}  
          style={this.props.style}  
        />  
      </Field>  
    </div>  
  )  
}
```

<https://www.youtube.com/watch?v=hkfBvpEfWdA>

Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

```
}  
  
getValue() {  
  {  
  
  render() {  
    return (  
      <div>  
        <Field>  
          <input  
            type="text"  
            value={getValue()}  
          >  
            onFocus={() => this.setState({ mode: 'edit' })}  
            onBlur={this.onBlur.bind(this)}  
            onChange={this.onChange.bind(this)}  
            onKeyDown={this.onKeyDown.bind(this)}  
            style={this.props.style}
```


Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



Cory House 🏠

@houseecor

Follow



Man, after using [@prettiercode](#) for a month, I'd love to have all the time back that I've wasted over the years manually formatting code!

Prettier is a great example of this maxim: "A luxury, once sampled, becomes a necessity."

8:20 AM - 11 Dec 2017

95 Retweets 353 Likes



Prettier. Not only JS.



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

Sept 2018 supported languages



ES2017
JSX
Flow
TypeScript
Vue
JSON



CSS3+
Less
SCSS
styled-components 🔥
styled-jsx



GraphQL
GraphQL Schemas



CommonMark
GitHub Flavored Markdown
MDX



Work in Progress
Elm (via elm-format)
Java
PHP ←
PostgreSQL
Python
Ruby
Swift



YAML

Prettier. Opinionated.

Only 9 style rules

Print Width

Tab Width

Tabs

Semicolons

Quotes

Trailing Commas

Bracket Spacing

JSX Brackets

Arrow Function Parentheses



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

Prettier. Opinionated.

A hard pill to swallow for artisans



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



... and one semicolon here

Prettier will hurt your feelings

It will not respect columns alignment or other whitespace “decorations”



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

```
const columns = [  
  { name: 'id'      , index: 'id'      , width: 55      },  
  { name: 'invdate', index: 'invdate'  , width: 90      },  
  { name: 'name'    , index: 'name asc, invdate', width: 100     },  
  { name: 'amount' , index: 'amount'  , width: 80 , align: "right" },  
  { name: 'tax'     , index: 'tax'     , width: 80 , align: "right" },  
  { name: 'total'  , index: 'total'  , width: 80 , align: "right" },  
  { name: 'note'   , index: 'note'   , width: 150, sortable: false }  
];
```



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



Posted by u/looneysquash 5 months ago

356



I don't like prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



Posted by u/looneysquash 5 months ago

356



I don't like prettier



[grensley](#) 931 points · 5 months ago



Nobody loves what prettier does to their syntax

Everyone loves what prettier does to their coworkers' syntax

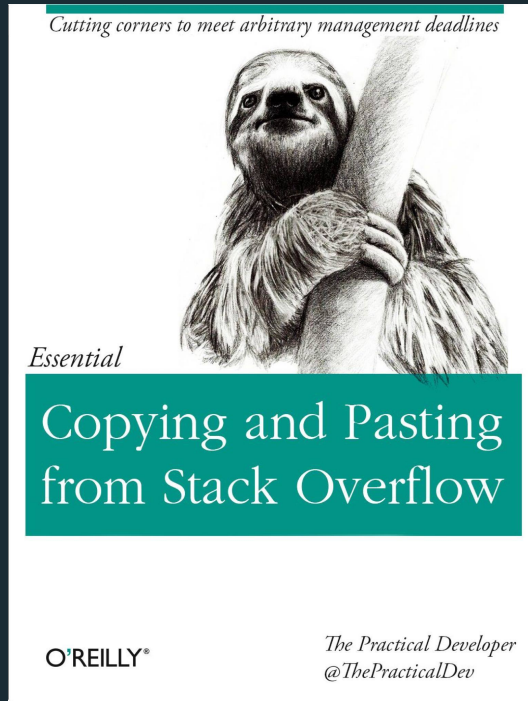
[Share](#) [Save](#)

Prettier. Opinionated.

... but a quick win for juniors



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018



Who uses Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

Facebook is 100% Prettier

Another exciting announcement is that all the JavaScript files in the Facebook codebase have now been converted to Prettier!

Feb. 2018 <https://prettier.io/blog/2018/02/26/1.11.0.html>

Benefits of Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

- Reduces Friction. [Prettier choses formatting for you]
- Speed up Code Reviews
- Instant feedback
- **1 thing less to multi-task while coding**
- Faster copy/pasting and refactoring

Cons of Prettier

- One more tool in the belt to maintain
- No custom formatting possible
- Not for everyone



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

My approach to code formatting is the best.

If it weren't, I wouldn't use it. Formatting is part of my being—an extension of my genetic makeup. And, when I write code, my formatting is the fingerprint that I leave behind. When I'm asked to change my style, I am—quite literally—asked to deny a fundamental Truth of my being.

— Ben Nadel

Prettier



Nicholas C. Zakas ✓

@slicknet

Following



Ok, this needs to be said: Prettier is not a replacement for ESLint. Prettier is a source code formatter. ESLint is a code quality tool.

9:02 AM - 20 Jun 2017

310 Retweets 715 Likes



Prettier



Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

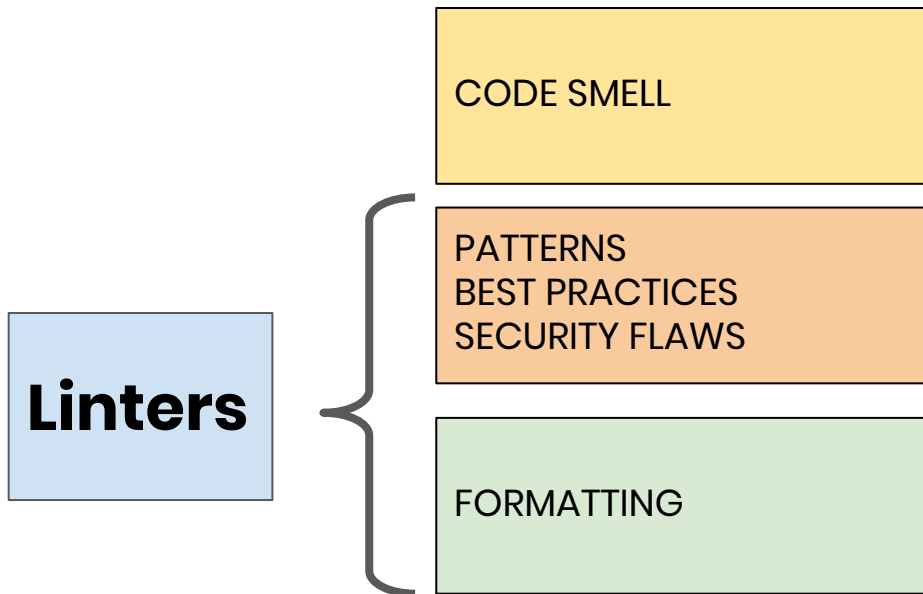
+ **### Does Prettier replace ESLint?**

+ **No**, ESLint does both traditional linting (looking for problematic patterns) and style checking (enforcement of conventions). You can use **ESLint** for **everything**, or you can **combine both** using Prettier to format your code and ESLint to catch possible errors.

Conclusion: Before



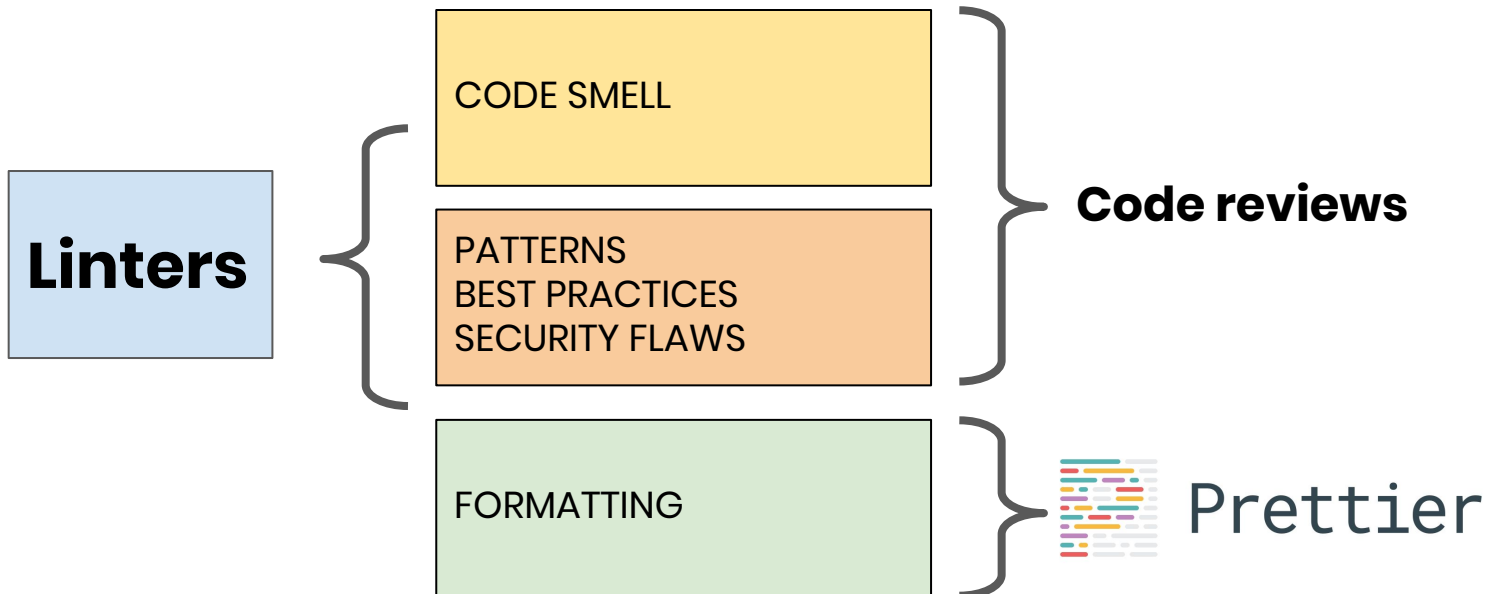
Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Conclusion: Now



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Configure Prettier + ESLint

```
yarn add -D eslint eslint-config-airbnb
```

(you might want just
eslint-config-airbnb-base, if not using React)

```
yarn add -D prettier eslint-plugin-prettier eslint-config-prettier
```


it will override (disable) eslint rules not compatible with prettier
Order matters

.prettierrc.json

```
{  
  "printWidth": 80,  
  "semi": true,  
  "singleQuote": true,  
  "trailingComma": "all"  
}
```

.eslintrc.json

```
{  
  "extends": [  
    "airbnb",  
    "plugin:prettier/recommended"  
  ],  
}
```





Drupal Europe
Darmstadt, Germany
Sep 10 - 14, 2018

Drupal™ +



Drupal 8.6 comes with Prettier. This is an effort to standardize the formatting of our JavaScript in addition to the rules provided by ESLint.

<https://www.drupal.org/node/2986680>

Drupal. PHP Linter

PHP Code Sniffer

Installing Coder Sniffer

<https://www.drupal.org/node/1419988>

Helps to enforce coding standards

<https://www.drupal.org/docs/develop/standards/coding-standards>

Coder module contains the “sniffs”

<https://www.drupal.org/project/coder>



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Using Prettier in your project



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

Option 1. Ask every developer to install Prettier on their editor and auto-save (It adds friction to the team, and could limit editor choice, personal workflow)

Option 2. Run in a pre-commit hook
Use [lint-staged](#) for easily setting up git commit hooks.



Introduce Prettier to an existing project



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

First: Agree with the team!

If only part of the team uses Prettier, it will warranty a lot of friction. Prettier by default formats the whole file, not only your changes. This multiplies the chances of git conflicts for the whole team and lengthy code reviews because of unrelated changes.

Option 1.

Apply Prettier to the whole project (if medium size project)
It will dirt git history

Option 2.

Go back to the past. Rewrite git history like Prettier would have been there from the day one.

<https://medium.com/millennial-falcon-technology/reformatting-your-code-base-using-prettier-or-eslint-without-destroying-git-history-35052f3d853e>



Drupal. CSS linters

In the process of migrating from CSS Lint to STYLE Lint

Drupal stylelint config:



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018

```
1  {
2    "extends": "stylelint-config-standard",
3    "plugins": [
4      "stylelint-no-browser-hacks/lib"
5    ],
6    "rules": {
7      "comment-empty-line-before": null,
8      "function-linear-gradient-no-nonstandard-direction": null,
9      "function-whitespace-after": null,
10     "no-descending-specificity": null,
11     "no-duplicate-selectors": null,
12     "no-unknown-animations": true,
13     "media-feature-name-no-unknown": [true, {
14       "ignoreMediaFeatureNames": ["prefers-reduced-motion"]
15     }],
16     "number-leading-zero": "always",
17     "plugin/no-browser-hacks": [true, {
```





David Corbacho Román

Developer at Wunder

 @dcorbacho



Drupal Europe
Darmstadt, Germany
10 - 14 September 2018



Become a Drupal contributor Friday from 9am

- First timers workshop
- Mentored contribution
- General contribution