Theory

Static Code Analysis

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

Linters

Dynamic Code Analysis
aka "testing"

wunder

Traditionally:

Static Code Analysis

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

Linters

Dynamic Code Analysis
aka "testing"

CODE SMELL

PATTERNS
BEST PRACTICES
SECURITY FLAWS

STYLE

wunder

JSLint

JSHint

ESLint

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 Node.js
  + Keeps up with ES evolution
  + In 2014 adds support for JSX

wunder

```
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)

wunder

Example

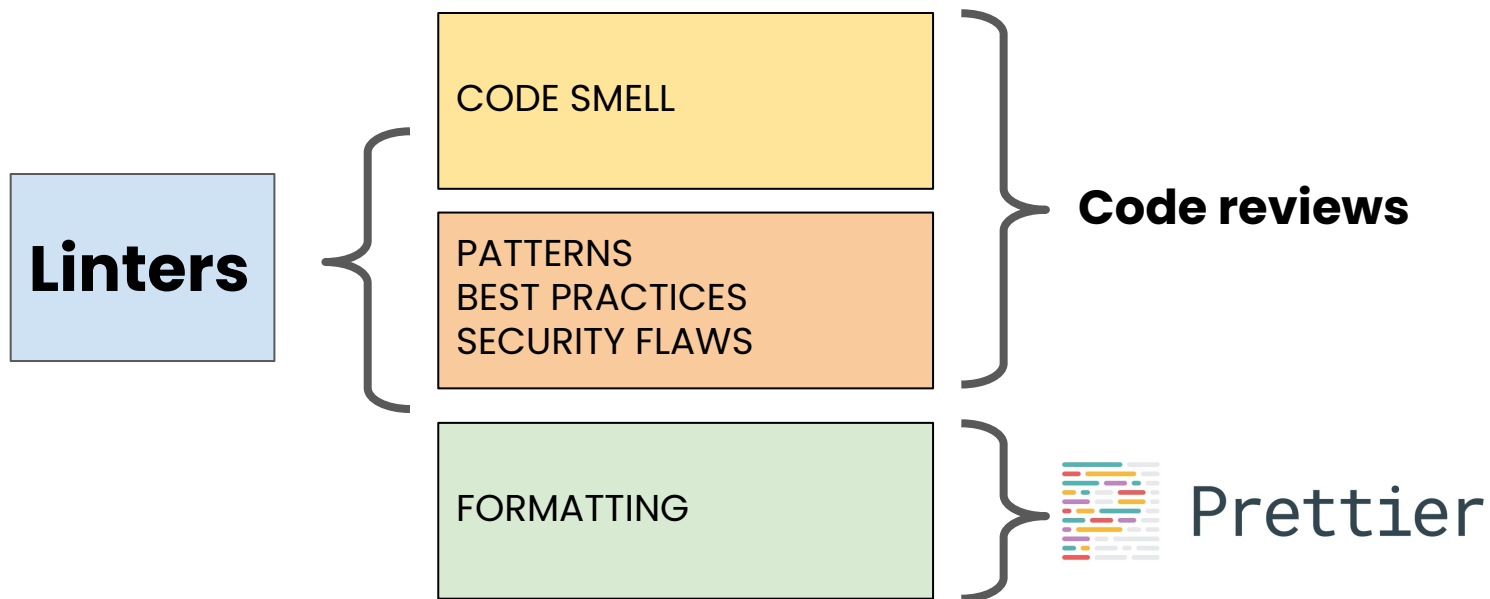| Preset	Projects	% of all presets
eslint:recommended	2,226	32.0%
airbnb	1,166	16.7%
standard	627	9.0%
airbnb-base[11]	557	8.0%
google	104	1.5%
standard-react	81	1.2%
prettier	58	0.8%
rackt	49	0.7%
react-app	48	0.7%
semistandard	44	0.6%
Total	4,960	71.2%

Source: The Adoption of JavaScript Linters in Practice: A Case Study on ESLint

wunder

Enters Prettier:

prettier.io

With Prettier:

Why Prettier?

Print Width

Tab Width

Tabs

Semicolons

Quotes

Trailing Commas

Bracket Spacing

JSX Brackets

Arrow Function Parentheses

wunder

Why Prettier?

Prettier allows you to focus on what's important

wunder

Why Prettier?

frontend/tests/unit/models/assignment-test.js · Hide outdated

```
@@ -1,18 +1,20 @@
1    1    import { moduleForModel, test } from 'ember-qunit'
2    2
     3    +
```
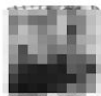
Minor style nitpick, this blank line is not needed

wunder

Why Prettier?

👁 ▓▓▓▓▓ 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backend/routes/api/v1/task.js   ⚡ Hide outdated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

wunder

Why Prettier?

Why Prettier?

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

wunder

+ ### **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.

wunder

Why Prettier? In Conclusion:

- 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
- Faster project onboarding

wunder

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

How to combine ESLint + Prettier
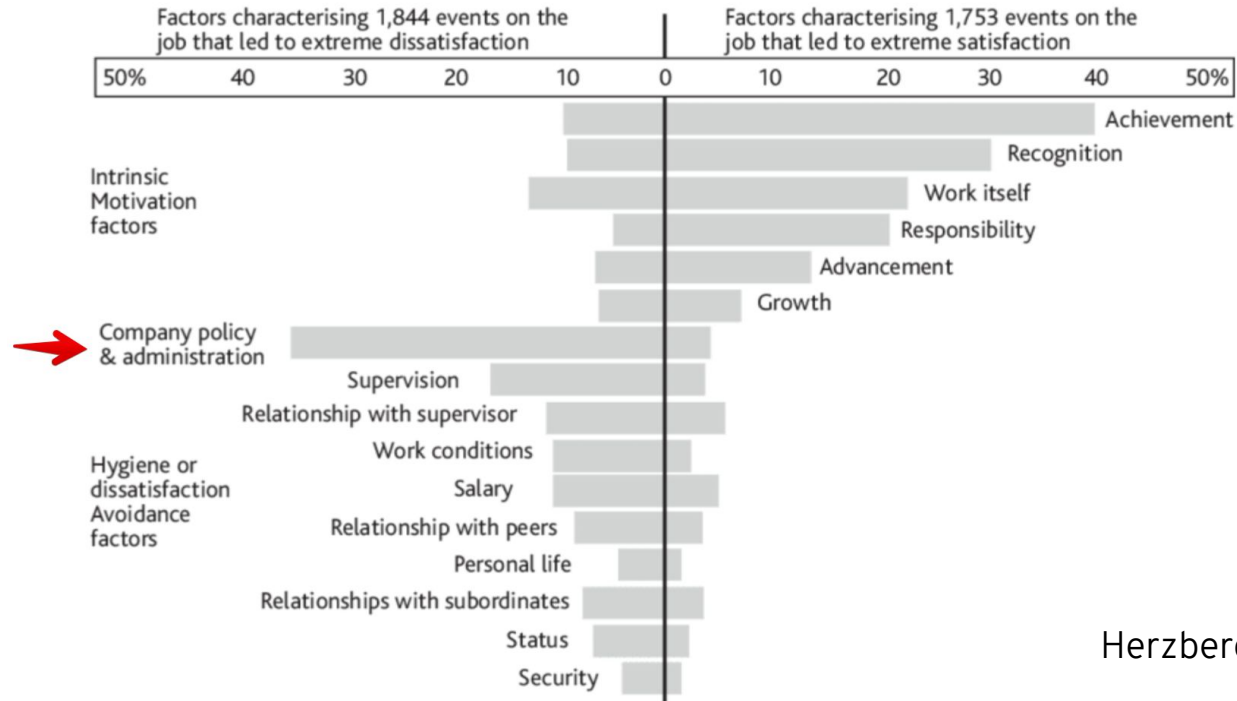
Step 1:

Use **eslint-config-prettier** to disable linter rules that conflicts with prettier

Step 2:

✅ Setup your editor to apply Prettier **formatOnSave**

✅ (Also optionally) with a pre-commit hook (**lint-staged with husky**)

❌ In the past I used to recommend (and there are many blog posts) about using **eslint-plugin-prettier.** This used to be best practice. Not any longer.
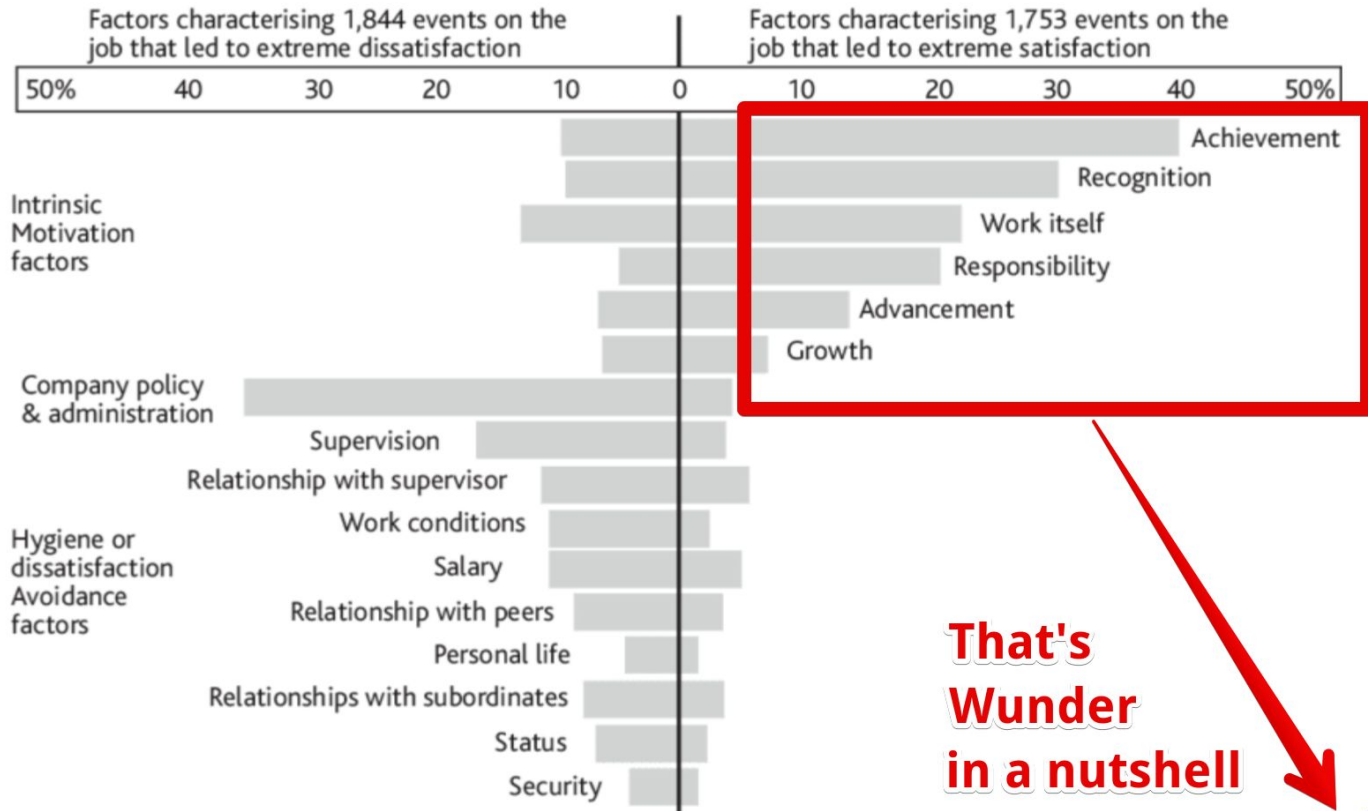
wunder

Story time: Conflicts in the team

wunder

Herzberg et al, 1959

Factors characterising 1,844 events on the job that led to extreme dissatisfaction

Factors characterising 1,753 events on the job that led to extreme satisfaction

| 50% | 40 | 30 | 20 | 10 | 0 | 10 | 20 | 30 | 40 | 50% |

Intrinsic Motivation factors
- Achievement
- Recognition
- Work itself
- Responsibility
- Advancement
- Growth

Company policy & administration
- Supervision
- Relationship with supervisor
- Work conditions
- Salary
- Relationship with peers
- Personal life
- Relationships with subordinates
- Status
- Security

Hygiene or dissatisfaction Avoidance factors

That's Wunder in a nutshell Join us

wunder

Thank you

:-)

wunder

wunder

Shaping the Digital Experience

wunder.io