

# Development Tooling

Thanh Pham  
@\_zlatanpham  
thanhpd@d.foundation

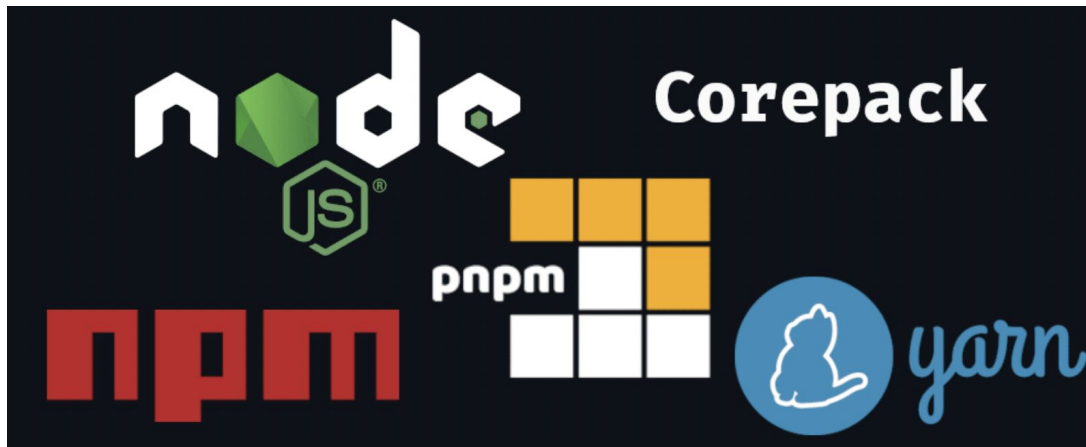


# Agenda

1. Intro to Package Manager (PNPM)
2. Backwards-compatible JS compiler (Babel)
3. Strong-typed complied language (TypeScript)
4. Bundling tool (Webpack)
5. Convention checker & formatter (ESLint vs Prettier)

# Node Package Managers

- Dependency management
- Discovery & sharing
- Versioning
- Scripting





<https://pnpm.io/>

# Backwards-compatible JS compiler (Babel)

*“Convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments”*

- Transform syntax
- Polyfilling
- Source code transformations (codemods)



<https://babeljs.io/>

# Strong-typed complied language (TypeScript)

Offer static typing to JavaScript (Strong typing vs dynamic typing)

- Error detection
- Document as code
- Make IDE more friendly



<https://www.typescriptlang.org/>

# Be simple when using TypeScript for building applications

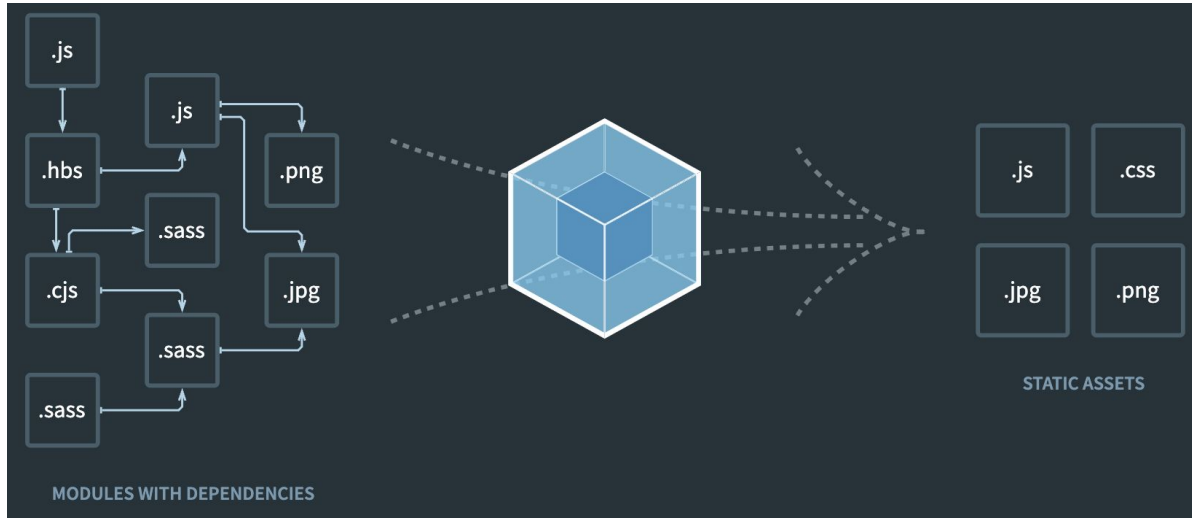
- Type Inference
- Defining Types (Interfaces vs Types)
- Composing Types (| and &)
- Generics
- Structural Type System



<https://www.typescriptlang.org/>

# Bundling tool (Webpack)

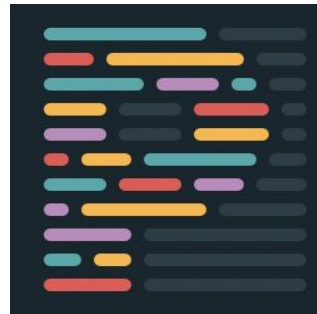
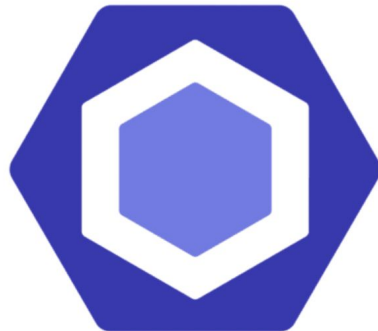
Webpack & most of bundling tools exist to resolve problems in scope, size, readability & maintainability





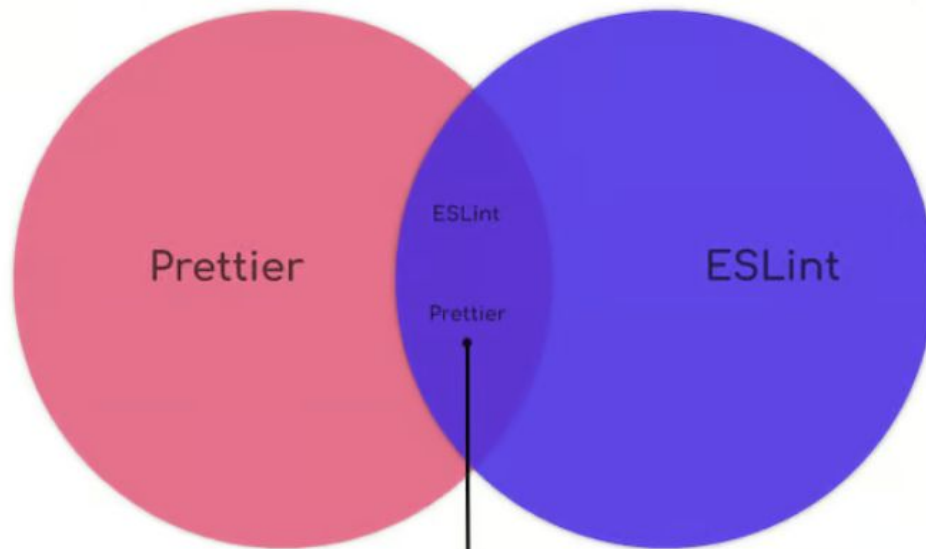
# Convention checker & formatter (ESLint vs Prettier)

- ESLint: Code-quality rules
- Prettier: Formatting rules



Formatting Rules

Code-Quality Rules



This is problematic

<https://blog.logrocket.com/using-prettier-eslint-automate-formatting-fixing-javascript/>

# References

- <https://pnpm.io/motivation>
- <https://www.typescriptlang.org/why-create-typescript>
- <https://webpack.js.org/concepts/why-webpack/>
- <https://babeljs.io/docs/>

# Thank You



# Q&A

