* Why do we need webpack ?

***Asset Modules***

Different asset modules:

**Asset/resource**

Emits a separate file and exports the URL.

Previous: file-loader

**asset/inline**

Exports a data URI of the asset ( attaches jpeg to the bundle )

Previous: url-loader

**asset/source**

Exports the source code of the asset

Previous: raw-loader

**asset**

Automatically chooses between exporting a data URI and emitting a separate file

based on filesize.

Previous: url-loader with asset size limit

**ToPlay**

Go to webpack.config 🡪 inside rules change type to one of the asset modules. 🡪 npm run build 🡪 take a look at the console output, plus at asset/inline, take a look at the bundle.js. ( base64)

**Loaders:**

Loaders help you to import all other kind of files.

show how to import css / scss.

explanation of order in array and each of the loaders: ( Lecture 17, minute 2 )

babel. (lecture 18 )

Thank you for your question!

Both Lodash and React are needed for the application to run in the browser. In other words, the application **can not run without** Lodash or React. That's why these 2 packages are installed as normal dependencies using --save

However, there are some packages that are **not necessary for the application to run in the browser** (in production). For example, html-webpack-plugin or node-sass. html-webpack-plugin generates the resulting html files **during the build process** (not while running in the browser). When your application is running in the browser, it does not need html-webpack-plugin. Actually, your application even doesn't know what html-webpack-plugin is. This plugin is part of the build process and is only required when Webpack runs. The same situation is with node-sass: it generates the css bundle during the build process. Then you would upload these generated files to production servers (as part of your deployment process).

**Plugins**

Plugins are additional javascript libraries that do everything that loaders cannot do.

plugins can also modify how the bundles themselves are created.

You can define global constants across the whole app, minify the bundle, generate other files

**difference between production and dev modes**

* **devServer ( lecture 32 )**