**Webpack**

**Bundlers**, или сборщики, предназначены для интеграции отдельных частей приложения в единое целое.

Благодаря сборщикам происходит обработка кода в понятный браузеру код.

Например, less/sass код преобразуется в обычный css код.

Ecmascript6 преобразуется в ecmascript5.

Так же код из множества файлов собирается в один большой (или несколько) файл(ов).

Код может минифицироваться.

Код может проверяться на ошибки.

После сборки на выходе получается чистый html, css и js.

В настоящее время существует несколько сборщиков, вот наиболее популярные из них:

**Browserify** - позволяет использовать в браузере те же инструменты, что и в node.js.

**Grunt** - один из первых сборщиков, оперирует настройками.

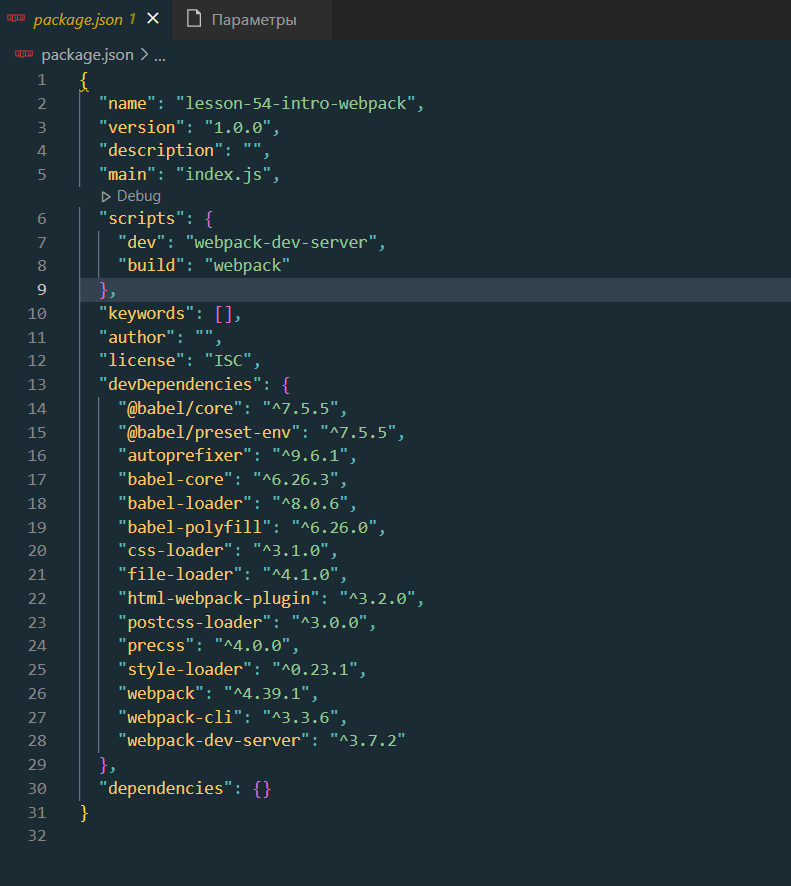
**Gulp** - следующий (по времени выхода) после grunt-а бандлер, работающий с понятием потока.

**Webpack** - инструмент для построения модулей в js приложении.

**Rollup.js** - сборщик модулей для JS, который компилирует маленькие кусочки кода во что-то более емкое и сложное.

При настройке сборщиков обычно внутри директории приложения создается папка, которая хранит актуальный код приложения со всеми исходными файлами. Также создается папка, в которую Bundler будет собирать конечные файлы.

[Webpack](https://webpack.js.org/) - сборщик модулей для современных javascript приложений.



Package.json - создается когда инициализуруется npm

В нем содержится инфо об пакетах



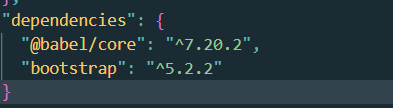
Удаление пакета



Установка пакета



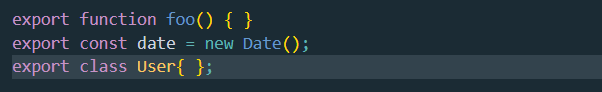
Установка пакета в dependencies(исп в продакшине)



**Модули в webpack**

Модули javascript являются обычными \*.js файлами.  
  
Для того, чтобы экспортировать, то есть предоставить наружу доступ к какому-либо содержимому файла, нужно воспользоваться ключевым словом **export**.  
  
Чтобы импортировать некий код, то есть потребовать функционал из другого файла, нужно воспользоваться ключевым словом **import**

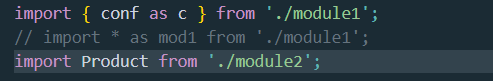
Ключевое слово export используется для экспорта переменных, функций или классов. Любое экспортируемое значение должно быть именованным (нельзя, например, экспортировать безымянные функции).



Если экспортировать нужно с другими именами, можно использовать синтаксис as:



Для того, чтобы использовать код из модулей, нужно импортировать его с помощью ключевого слова import и указать путь к импортируемому файлу (файлу js, из которого производился экспорт):



Импортируемые значения должны совпадать с именем экспортируемой переменной.   
Импортируемые значение заключаются в фигурные скобки.  
При импорте не обязательно указывать расширение .js в имени импортируемого файла.