Статья предназначена для изучения особенностей frontend проектов, как с ними работать и как в них разбираться.

### **Node.Js**

Node обрабатывает JavaScript в машинный код, позволяет подключать различные библиотеки и использовать их в JavaScript коде. Node может работать как на сервере, так и на локальной машине. Локально Node понадобится для установки зависимостей, их работы и локальных серверов. В Node есть встроенный менеджер зависимостей npm, но мы в своей работаем с yarn. Скачать и установить Node можно и нужно на [официальном сайте](https://nodejs.org/en/).

***Node*** *или* ***Node.js*** *— программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода.*

### **Yarn**

Yarn - менеджер зависимостей, который позволяет скачивать и устанавливать в проект сторонние библиотеки.

***Система управления пакетами*** *— набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов программного обеспечения. Системы управления пакетами активно используются в различных дистрибутивах операционной системы Linux и других UNIX-подобных операционных системах.*

Для понимания какие зависимости уже имеются в проекте существует файл package.json и yarn.lock. В первом файле указывается название библиотек и их версии, для того чтобы yarn понимал, что нужно скачать. Во втором, уже сам менеджер записывает все библиотеки, их версии и зависимости этих библиотек которые уже установлены в проекте.

Пример package.json

{

"name": "my-project",

"version": "1.0.0",

"dependencies": {

"graphql": "^0.13.2",

"vue": "^2.1.10"

}

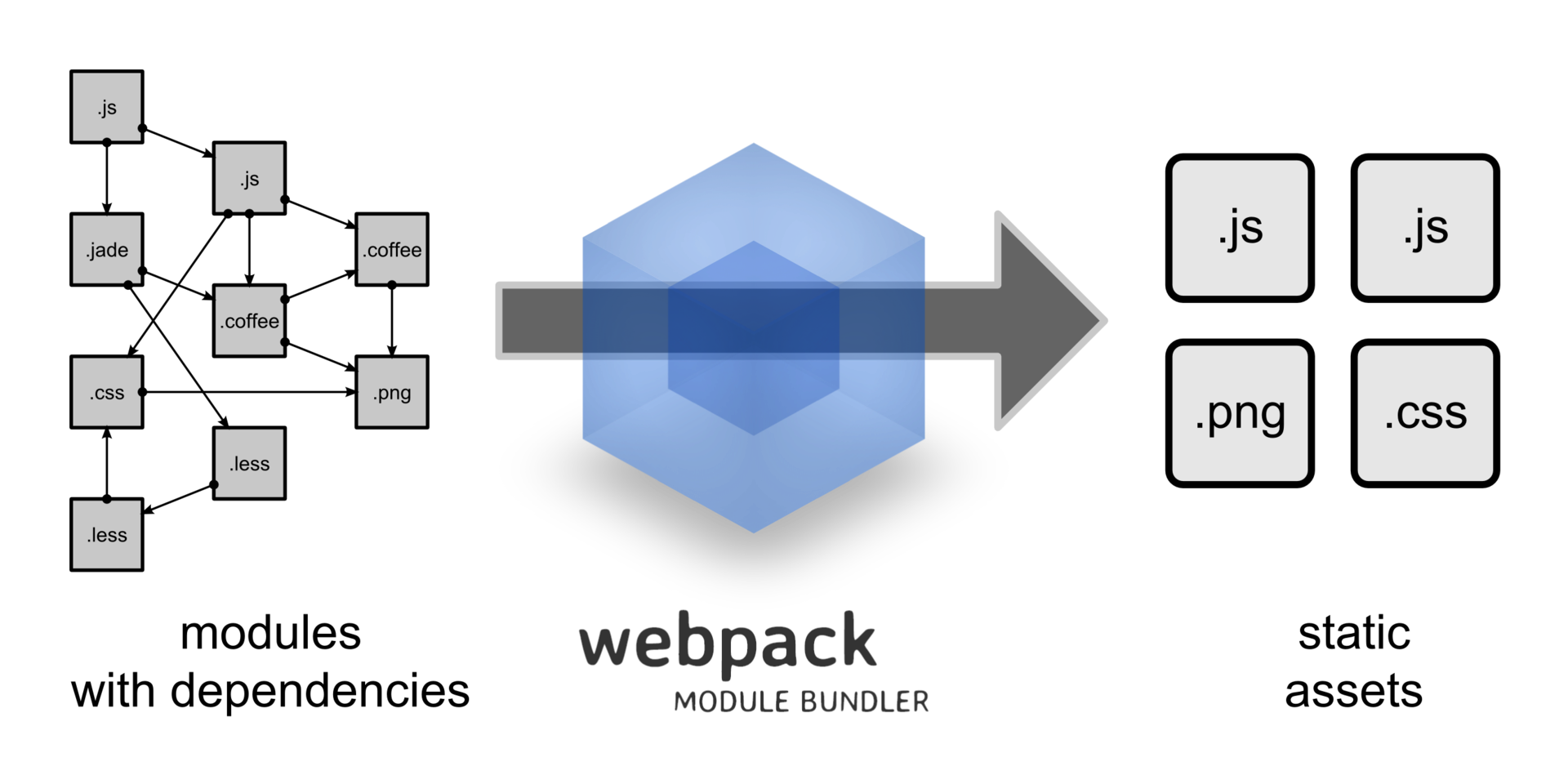
}

В этом файле есть название проекта и версия проекта, используется в том случае если ваш проект будет зависимостью другого проекта, желательно изучить [семантическое версионирование](https://semver.org/lang/ru/#%D1%81%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5-%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-200).

И соответственно [документация](https://yarnpkg.com/en/docs), поможет понять как работать с yarn.

### **Webpack**

Webpack - сборщик JS кода, а так же зависимостей кода, в один файл или несколько, в зависимости от настройки webpack.



Webpack имеет очень интересную [документацию](https://webpack.js.org/concepts/) с большим количеством [примеров](https://webpack.js.org/guides/installation). Посмотреть и изучить, обязательно.

Задание:

Вам было выдано тестовое задание перед собеседованием. А именно:

*Вывод знака зодиака по дате рождения. После ввода в input даты рождения выводить знак зодиака в зависимости от полученных данных. Желательно использовать подход объектно-ориентированного программирования.*

* Реализовать на JS, если было ранее не PHP
* [Скачать и установить NodeJS](https://nodejs.org/en/)
* Установить Yarn глобально с помощью команды npm install yarn -g
* Установить Webpack глобально с помощью командыnpm install webpack -g
* Необходимо собрать ваш js файл с помощью webpack (смотреть [статью на хабре](https://habr.com/ru/post/309306/) и [документацию](https://webpack.js.org/concepts/). В данном случае необходимо знать вкладки документации [getting started](https://webpack.js.org/guides/getting-started/), [entry points](https://webpack.js.org/concepts/entry-points/) и [output](https://webpack.js.org/concepts/output/), но изучить желательно всю документацию). Нужно чтобы код был логически разделен на несколько разных файлов и собран в один общий
* Инициализируйте ваш пакет с помощью команды yarn init и у вас в папке проекта появится файл package.json с введенными вами данными. И выполните команду yarn эта команда скачивает и устанавливает зависимости. В вашем случае просто создаст системные файлы и папку node\_modules с файлом для yarn'a
* Папку node\_modules нужно исключить из git с помощью файла .gitignore, [документация по игнорированию файлов](https://git-scm.com/book/ru/v1/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B-Git-%D0%97%D0%B0%D0%BF%D0%B8%D1%81%D1%8C-%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9-%D0%B2-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9#%D0%98%D0%B3%D0%BD%D0%BE%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2)
* Залить все в ваш репозиторий
* Теперь вам необходимо отправить ваш проект на [npmjs.com](http://npmjs.com/) (хранилище пакетов). Команда npm publish, но для этого вам нужно зарегистрироваться на [сайте](http://npmjs.com/) и затем авторизоваться на вашей машине npm login, изучить [документацию](https://docs.npmjs.com/)
* Отправить ссылку, на ваш пакет npmjs.com мне в [личку](http://t.me/bessamu)

Как пример, мой простенький пакет для utm меток:

<https://www.npmjs.com/package/amo-utm-parser-js> (но тут не использовался webpack, так как необходимо было использовать в legacy проектах). Его можно скачать и посмотреть, что такое зависимости, на примере этого пакета.

* [Официальный сайт NodeJS](https://nodejs.org/en/)
* [Документация Yarn](https://yarnpkg.com/lang/en/docs/)
* [Статья о Webpack на habr.ru](https://habr.com/ru/post/309306/)
* [Документация Webpack](https://webpack.js.org/concepts/)
* [Документация Npm](https://docs.npmjs.com/)