9. Введение в автоматизацию. Node.js

Статьи:

GIT:

GIT Book

Глоссарий терминов для Git и GitHub
Введение в системы контроля версий
Регистрация на Гитхабе. Работа через GitHub Desktop
Следите за концом строки

GitHub Pages:
Размещение на GitHub Pages
Как использовать GitHub Pages? - Изучение веб-разработки | MDN

Онлайн курсы:
Основы командной строки
Введение в Git

Node.js:
Два способа установки Node.js
Скринкаст по Node.JS
Полезные команды для работы с Node.js
JS: Настройка окружения

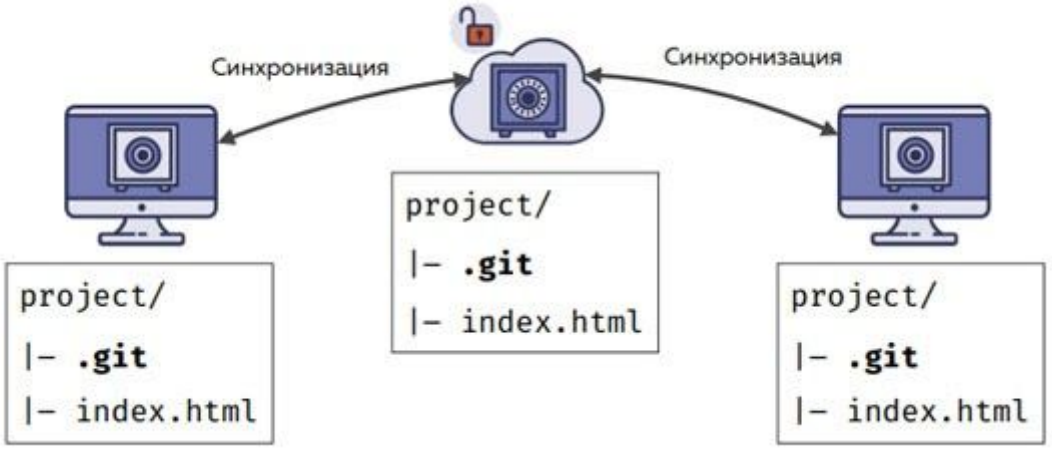
Gulp:
Gulp 4 - простая сборка проекта для вёрстки и javascript

GIT и системы контроля версий

Для решения проблемы с сохранением новой версии файлов удобно использовать **системы контроля версий**. Одна из самых популярных — **Git**.

Проект, в котором была инициализирована система Git, называется **репозиторием**. При инициализации в проект добавляется скрытая папка **.git**. Репозиторий хранит все рабочие файлы и историю их изменений.

Корневая папка проекта — это рабочая область. В ней находятся все файлы и папки, необходимые для его работы.



Локальный репозиторий — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

Удалённый репозиторий — репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.

Как это работает?

1. Репозиторий хранит все версии проекта. В случае передачи этого проекта другому человеку, он увидит всё, что с ним происходило до этого.

2. Ничего не теряется и не удаляется бесследно. При удалении файла в новой версии добавляется запись о том, что файл был удалён.

3. Всегда можно вернуться к любой из версий проекта, загрузив её из хранилища в рабочую область.

Форк (Fork) — копия репозитория. Его также можно рассматривать как внешнюю ветку для текущего репозитория. Копия вашего открытого репозитория на Гитхабе может быть сделана любым пользователем, после чего он может прислать изменения в ваш репозиторий через пулреквест.

Клонирование (Clone) — скачивание репозитория с удалённого сервера на локальный компьютер в определённый каталог для дальнейшей работы с этим каталогом как с репозиторием.

Коммит (Commit) — фиксация изменений или запись изменений в репозиторий. Коммит происходит на локальной машине.

Пул (Pull) — получение последних изменений с удалённого сервера репозитория.

Пуш (Push) — отправка всех неотправленных коммитов на удалённый сервер репозитория.

Пулреквест (Pull Request) — запрос на слияние форка репозитория с основным репозиторием. Пулреквест может быть принят или отклонён вами, как владельцем репозитория.

Мёрдж (Merge) — слияние изменений из какой-либо ветки репозитория с любой веткой этого же репозитория. Чаще всего слияние изменений из ветки репозитория с основной веткой репозитория.

Задание:

1. Зарегистрироваться на https://github.com/
2. Скачать https://desktop.github.com/, добавить учетную запись GitHub
3. Создать папку нового проекта, инициировать репозиторий
4. Создать файл readme.md, записать в него описание проекта
5. Создать коммит
6. Запушить изменения в свой GitHub

Автоматизация

Установка Node.js и инициализация проекта

1. Установить Node.js (LTS версию)
2. Проверить установку в терминале: `node --version`
3. Создать папку проекта, открыть ее в VScode
4. Запустить команду инициализации **npm init** в корне проекта, ответить на вопросы проекта

В итоге создается файл **package.json** - это текстовый файл, внутри которого данные хранятся в JSON формате с информацией о проекте

Зависимости

В коммерческих проектах используются миллионы строк стороннего кода, написанными программистами по всему миру.

Подобный код хранится в библиотеках доступных для установки в проект. В JavaScript мире их принято называть **пакетами**.

Где найти нужные пакеты для работы? https://www.npmjs.com/

Для того чтобы воспользоваться сторонним пакетом, его нужно добавить в проект как **зависимость**.

Установка пакета Gulp с сайта https://www.npmjs.com/package/gulp: **npm i gulp**

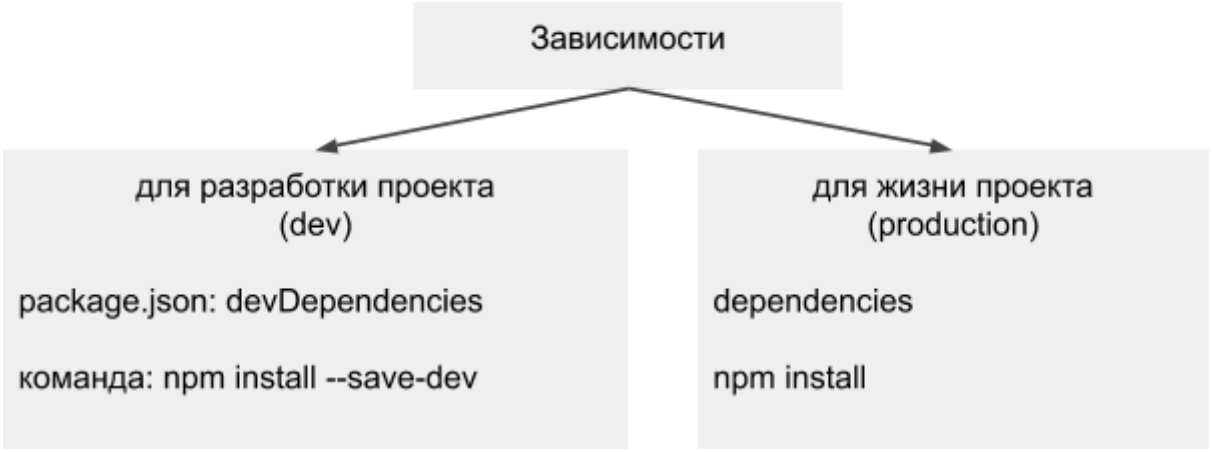
Зависимости в package.json добавляются под ключом **dependencies**. Здесь указаны все пакеты используемые в проекте и не входящие в стандартную библиотеку.

Параллельно создается файл **package-lock.json**. У зависимостей любого проекта есть свои зависимости, а у них, в свою очередь, свои зависимости. Подобная цепочка может быть довольно длинной и на разных её участках возможно появление одних и тех же пакетов, но разных версий.

package-lock.json содержит описание всех пакетов, которые будут поставлены включая всех их зависимости с указанием конкретных версий. Это позволяет получать гарантированно одни и те же версии зависимостей для всех разработчиков проекта. Этот файл создаётся командой **npm install** и потом используется при установке зависимостей. При наличии package-lock.json в проекте, установку зависимостей рекомендуется выполнять командой **npm ci**

Код самих зависимостей не является частью исходного кода. Его всегда добавляют в **.gitignore**. А вот файлы **package.json** и **package-lock.json** являются частью исходного

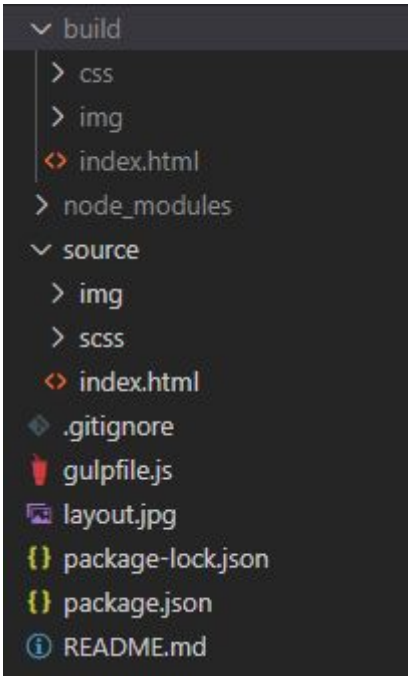
кода и именно через них Node.js узнает о том, какие пакеты нужно поставить в систему.



Для дальнейшей работы проекта нужно установить все необходимые пакеты и начать разработку проекта

Сборка для стандартного проекта делается 1 раз, остальные проекты начинаются с установки зависимостей по списку из **package.json** командой **npm i**

Структура проекта на Gulp + SCSS



source - всё, что нужно для разработки
build - результат для прода

Если в проекте используется Гит, то еще и папка .git

node-modules - папка со всеми зависимостями

Пакеты для автоматизации frontend

Где найти нужные пакеты для работы? https://www.npmjs.com/

```
"devDependencies": {
    "autoprefixer": "^9.8.4",
    "browser-sync": "^2.26.7",
    "del": "^5.1.0",
    "gulp": "^4.0.2",
    "gulp-csso": "^4.0.1",
    "gulp-imagemin": "^7.1.0",
    "gulp-plumber": "^1.2.1",
    "gulp-postcss": "^8.0.0",
    "gulp-rename": "^2.0.0",
    "gulp-sass": "^4.1.0",
    "gulp-sourcemaps": "^2.6.5",
    "gulp-svgstore": "^7.0.1",
    "gulp-webp": "^4.0.1",
     "stylelint": "13.6.1",

  },
```

gulp - сама система сборки)
gulp cli - запускает в терминале команды gulp ...
gulp-clean-css - минимизатор стилей
gulp-uncss - удалить неиспользуемый цсс
gulp-concat - объединение файлов в один
gulp-csso - тоже минификатор
gulp-rename - переименование файлов
gulp-htmlmin - минификатор разметки
gulp-uglify - минификация js
babel - переписывает современный JavaScript-код в предыдущий стандарт

autoprefixer - автопредфиксер
browser-sync - автообновление при изменении файлов

imagemin - оптимизация изображений
postHTML - шаблонизатор разметки

del - удаление файлов

И другие…

Практика

Создать репозиторий для нового проекта, установив сборку урока.

Макеты для проекта/портфолио:
http://psd-html-css.ru/templates/psd-shablon-dlya-veb-studii
http://psd-html-css.ru/templates/waxom-besplatnyy-psd-shablon-lendingovoy-stranicy
http://psd-html-css.ru/templates/besplatnyy-shablon-rezyume-v-formate-psd
http://psd-html-css.ru/templates/psd-shablon-portfolio-v-mini-dizayne