**GULP**

**Первоначальная настройка**

1. ставим Node.JS c оф.сайта
2. проверить версию **npm -v**

Далее ставим локально в корень проекта:

1. Создаем package.json **npm init -y**
2. Ставим локально последнюю версию gulp: **npm i gulp@latest --save-dev**   
   Проверить загруженную версию gulp можно в файле package.json в св-ве "devDependencies":

"devDependencies": {

"gulp": "^4.0.0"

}

Восстановление папки **node\_modules** : **npm i**

**\*\*\* Мой npm пакет от 16.10.2019 \*\*\***

**npm i gulp-htmlmin gulp-notify gulp-sourcemaps gulp-concat gulp-babel @babel/core @babel/preset-env gulp-rename gulp-sass gulp-autoprefixer gulp-clean-css del browser-sync --save-dev**

**Создаем структуру проекта из корня проекта (структуры могут быть разными):**

1. создаем две папки: **mkdir src build**
2. создаем два файла: **touch gulpfile.js .gitignore**
3. сразу заполним .gitignore:

* **echo node\_modules>.gitignore**
* **echo build>>.gitignore** //если оставить одну > то она перезапишет прошлый результат, поэтому используем >>

1. переходим в папку src (команда cd)
2. Создаем в ней две папки pug и static, в static так же еще вкладываем папки js, stylus, fonts, img :  
   **mkdir -p pug static/{js, stylus, fonts, img}** // так как папка static еще не создана необходимо поставить флаг -p
3. Переходим в папку static
4. Создаем файл main.js в папке js: **touch js/main.js**
5. Создаем файл main.styl в папке stylus: **touch stylus/main.styl**

12 и 13 можно объединить в одну команду: **touch js/main.js && touch stylus/main.styl**

**Курс от JavaScript.ru**

<https://www.youtube.com/watch?v=uPk6lQoTThE&index=1&list=PLDyvV36pndZFLTE13V4qNWTZbeipNhCgQ>

**FrontCoder (gulp 4.0)**

**npm i gulp-sass gulp-sourcemaps gulp-clean-css gulp-autoprefixer gulp-notify gulp-rename browser-sync --save-dev**

**Удалить пакет:**

npm uninstall <name> удаляет модуль из node\_modules, но не package.json

npm uninstall <name> --save также удаляет его из dependencies в package.json

npm uninstall <name> --save-dev также удаляет его из devDependencies в package.json

npm -g uninstall <name> --save также удаляет его глобально

**npm i browser-sync del gulp-autoprefixer gulp-concat gulp-csso gulp-htmlmin gulp-rename gulp-terser gulp-sass**

**npm i browser-sync del gulp-autoprefixer gulp-concat gulp-csso gulp-htmlmin gulp-rename gulp-babel gulp-uglify-es gulp-sass --save-dev**

**npm i gulp-babel gulp-uglify-es --save-dev**

**gulp-uglify**  - Не работает с JS6 - альтернатива **npm i gulp-terser**

Структура проекта от Игоря:

app - сырой проект

dist - скомпелированный проект

далее в app создаем:

mkdir html js sass

del - чистит директорию от файлов.

gulp-concat - склеивает файлы.

gulp-csso - миниыикация css

gulp-htmlmin - минификация html

gulp rename - ...

gulp-uglifyjs - ...

**Установка gulp-плагинов к GULP**

**Плагин для обработки html - 'gulp-pug'**

в корень проекта ставим плагин: npm i gulp-pug --save-dev // Так как ставим локально необходимо добавить флаг --save-dev, флаг -dev так как используем плагины для разработки.

проверяем package.json в "devDependencies" должен появится "gulp-pug": "^4.0.1"

в папку pug добавляем папку layout и pages

создаем таск:

'use strict';

// Подключаем плагины

var gulp = require('gulp'),

pug = require('gulp-pug');

// Таски для плагинов

gulp.task('pug', function() {

return gulp.src('src/pug/pages/\*.pug') // Указываем путь к файлам которые будем обрабатывать, путь указываем относительно файла gulpfile.js

.pipe(pug({ // запускаем плагин pug/, при помощи pipe() - Обработчик gulp, запускает плагины

pretty: true // Если не указать данное св-во, то весь html скомпилится в одну строку

}))

.pipe(gulp.dest('build'))

});

Создаем страничку index.pug в папке pages

Проверяем, запустив, через консоль: gulp pug

Если все ок, то на экране появиться:

[08:20:25] Working directory changed to D:\GEEK\_BRAINS\GULP\FrontCoderTest

[08:20:25] Using gulpfile D:\GEEK\_BRAINS\GULP\FrontCoderTest\gulpfile.js

[08:20:25] Starting 'pug'...

[08:20:26] Finished 'pug' after 55 ms

и в папке build появится файл index.html

**gulp-load-plugins**

Данный плагин позволяет не подключать в общий список плагинов название плагина в переменную (работает долго, мне не оч. понравилось) см. Урок 3

**var = require('gulp-load-plugins')();**

**Плагин для обработки css - 'gulp-stylus'**

**+ csso**

**npm i gulp-stylus gulp-csso --save-dev**

var gulp = require('gulp'),

stylus = require('gulp-stylus'),

csso = require('gulp-csso');

gulp.task('stylus', function() {

return gulp.src('app/scss/\*\*/\*.scss')

.pipe(stylus())

.pipe(csso())

.pipe(gulp.dest('app/css/'));

});

так же в этот таск можно добавить autoprefixer и sourcemaps.

**Watcher task**

**npm i gulp-notify --save-dev**

gulp.task('stylus', function() {

return gulp.src('app/scss/\*\*/\*.scss')

.pipe(stylus())

.on("error", notify.onError({

title: "style error"

}))

.pipe(csso())

.pipe(gulp.dest('app/css/'));

});

**browsersync**

https://www.browsersync.io/

**npm i browser-sync --save-dev**

Важно! ставим без -g, локально в проект, с --save-dev

var bs = require("browser-sync").create();

gulp.task('server', function() {

bs.init({

server: 'app/' // Указываем директорию откуда поднимать сервер

});

bs.watch('app/').on('change', bs.reload); // Автопрезегрузка при изменении в стилях

});

gulp.task('default', gulp.series(

'stylus',

gulp.parallel('watch', 'server')

));

Курс Gulp 4.0: <https://www.youtube.com/watch?v=7Y_PLpOZC0I&list=PLDmWuBfh49Z6gotPQGTIrbNwIDIQzZMVA&index=1>

скринкаст по шаблонизатору HTML - Pug/Jade: <https://www.youtube.com/watch?v=kKrYAdB8XJ4&list=PLDmWuBfh49Z7nZr7FF4eE88PZXWQLIFF0>

Заметки:

npm info gulp dist-tags

======================================

# [Why do we need to install gulp globally and locally?](https://stackoverflow.com/questions/22115400/why-do-we-need-to-install-gulp-globally-and-locally)

When installing a tool globally it's to be used by a user as a command line utility anywhere, including outside of node projects. Global installs for a node project are bad because they make deployment more difficult.

## npm 5.2+

The [npx](https://medium.com/@maybekatz/introducing-npx-an-npm-package-runner-55f7d4bd282b) utility bundled with npm 5.2 solves this problem. With it you can invoke locally installed utilities like globally installed utilities (but you must begin the command with npx). For example, if you want to invoke a locally installed eslint, you can do:

npx eslint .

## npm < 5.2

When used in a script field of your package.json, npm searches node\_modules for the tool as well as globally installed modules, so the local install is sufficient.

So, if you are happy with (in your package.json):

"devDependencies": {

"gulp": "3.5.2"

}

"scripts": {

"test": "gulp test"

}

etc. and running with npm run test then you shouldn't need the global install at all.

Both methods are useful for getting people set up with your project since sudo isn't needed. It also means that gulp will be updated when the version is bumped in the package.json, so everyone will be using the same version of gulp when developing with your project.

### Addendum:

It appears that gulp has some unusual behaviour when used globally. When used as a global install, gulp looks for a locally installed gulp to pass control to. Therefore a gulp global install requires a gulp local install to work. The answer above still stands though. Local installs are always preferable to global installs.

источник <https://stackoverflow.com/questions/22115400/why-do-we-need-to-install-gulp-globally-and-locally>

3 урок

[FeranDAq](https://www.youtube.com/user/FeranDAq)[1 месяц назад](https://www.youtube.com/watch?v=75dtVYT7dBw&lc=UgzFjP60ftvbVQNg5WZ4AaABAg)

Это не четвертый галп !!!! Там другой синтаксис, у вас подцепляется старый - 3ей версии﻿

ОТВЕТИТЬ

Скрыть ответы

[FrontCoder](https://www.youtube.com/channel/UCzhKcnzGNhPEZsNSKjug1jA)[1 месяц назад](https://www.youtube.com/watch?v=75dtVYT7dBw&lc=UgzFjP60ftvbVQNg5WZ4AaABAg.8mcSvPDjcpI8mePnIHI8B3)

Добрый день. В самом начале плейлиста видно, что устанавливается 4й галп﻿

[FeranDAq](https://www.youtube.com/user/FeranDAq)[1 месяц назад](https://www.youtube.com/watch?v=75dtVYT7dBw&lc=UgzFjP60ftvbVQNg5WZ4AaABAg.8mcSvPDjcpI8mfvwEzrkvk)

устанавливается может и 4ый но используется синтаксис 3еого. посмотри в офф документации. в 4ой версии галпа очередность тасков пишется только так: gulp.task("taskname", gulp.{series | parallel} ('taskname', etc)﻿

Урок 5

[Roman Zakrevskyi](https://www.youtube.com/user/Romul3003)[2 месяца назад (изменено)](https://www.youtube.com/watch?v=ar3X9dSF7fM&lc=Ugxb-ia1Q3SUiasFTtV4AaABAg)

а не проще сделать так? gulp.task('serve', function() { browserSync.init({ server: 'build' }); browserSync.watch('build/\*\*/\*.\*').on('change', browserSync.reload); }); И не нужно никаких танцев с бубном в отдельных тасках﻿

[Sergo Minichev](https://www.youtube.com/channel/UCOfcjn-nVBBfF-OHpz3Z50Q)[3 недели назад](https://www.youtube.com/watch?v=ar3X9dSF7fM&lc=Ugxb-ia1Q3SUiasFTtV4AaABAg.8luSP29YqXT8o1Cng5YJJz)

Сколько же времени ты мне сэкономил,гребаный волшебник......﻿

Все установленные пакеты

**npm ls -g**

Пакеты установленные мною

**npm ls -g --depth=0**

gtgt

**Live Reload**

1. ставим Node.JS c оф.сайта
2. проверить версию **npm -v**
3. Открываем командную строку **Node.js command prompt**
4. Ставим gulp-cli (как cli скрипт, это важно!) глобально : **npm install gulp-cli -g  
   npm i gulp-cli -g**

Далее поднимаем автоперезагрузку страницы:

1. создаем простейший проект (проект лучше вынести в отдельную папку src)
2. в корень проекта локально ставим gulp (указываем что наш проект нуждается в gulp) если все ок, то в корне проекта появится папка node\_modules : **npm install gulp -D**   
   **npm i gulp --save-dev**

**Версия WebDesignMaster:**

Ставим gulp-sass (для автоматической компиляции scss файлов в css): **npm i gulp-sass --save-dev**

Проверяем установку npm ls -g --depth=0

1. Ставим пакет gulp Livereload: **npm install gulp-livereload**
2. Ставим расширение Livereload для браузера. (Не забудьте, что для локальных файлов (если не с вебсервера запускаете проект), нужно в расширениях у livereload поставить галочку "Разрешить открывать файлы по ссылкам". В хроме на любую иконку расширения ПКМ -> Управление расширениями, ищите Livereload в списке и ставите эту галочку﻿).
3. Ставим пакет LiveReload в Sublime через установщик пакетов:

* Preference -> Package Control -> Install Package -> LiveReload
* Затем Preference -> Package Settings -> LiveReload -> Plugins -> Enable/Disable plugins -> Выбираем Enable simple reload.

1. Затем в корне проекта создаем файл gulpfile.js в него пишем:

**var**

**gulp = require("gulp"),**

**livereload = require("gulp-livereload");**

**gulp.task("reload-css", function() {**

**gulp.src('./src/\*.css')**

**.pipe(livereload());**

**});**

**gulp.task("default", function() {**

**livereload.listen();**

**gulp.watch('./src/\*.css', gulp.series('reload-css'));**

**});**

1. Открываем корень файла через консоль и запускаем gulp: **gulp**

если прошло без ошибок, то все ок, gulp-livereload для текущего проекта запущен.

1. Открываем страницу проекта в браузере и кликаем на иконку расширения gulp-livereload, если все ок, то в центре появится черный микро-точка.
2. Так же в Sublime имеет смысл настроить autoSave.

**gulp-sass плагин (Предварительная автокомпеляция из SCSS(или SASS) в CSS)**

ставим **npm install gulp-sass**

так как gulp выкидывает автокомпеляцию css файла из scss, надо создать папку css в папке src.

меняем содержание gulpfile.js на

**var**

**gulp = require("gulp"),**

**livereload = require("gulp-livereload");**

**sass = require("gulp-sass");**

**gulp.task("reload-css", function() {**

**gulp.src('./src/\*.scss')**

**.pipe(sass().on('error, sass.logError'))**

**.pipe(gulp.dest('./src/css/'))** - В эту папку сохраняется скомпелированный css файл

**.pipe(livereload());**

**});**

**gulp.task("default", function() {**

**livereload.listen();**

**gulp.watch('./src/\*.scss', gulp.series('reload-css'));**

**});**

LiveReload in Sublime

https://www.youtube.com/watch?v=\_cbgjtUhjyY

Шаблон программирования Glob

loremru50 (рыба в sublime)

https://toster.ru/q/443895