

**Gulp js**

**URL** : <https://gulpjs.com/docs/en/getting-started/quick-start>

Install Node.js: Gulp.js requires Node.js to be installed on your system. You can download and install Node.js from the official website. Follow the installation instructions provided there.

Initialize your project: Create a new directory for your project and navigate into it using your terminal or command prompt.

Initialize npm: Use npm (Node Package Manager) to initialize a new project. Run the following command:

**npm init**

This command will walk you through creating a package.json file for your project. You can press enter to accept the default options or fill in the details as prompted.

Install Gulp.js globally (optional): While it's not strictly necessary, you can install Gulp.js globally on your system to use the gulp command line tool. Run the following command:

**npm install -g gulp**

Install Gulp.js locally: Install Gulp.js locally in your project directory. This way, you can manage different versions of Gulp.js for different projects. Run the following command:

**npm install gulp --save-dev**

Create a gulpfile.js: In your project directory, create a file named gulpfile.js. This file will contain your Gulp tasks. Here's a simple example to get you started:

**const gulp = require('gulp');**

**// Define a task**

**gulp.task('hello', function() {**

**console.log('Hello, Gulp!');**

**});**

**// Default task**

**gulp.task('default', gulp.series('hello'));**

Run your tasks: You can now run your Gulp tasks using the gulp command followed by the task name. For example, to run the hello task from the example above, you would run:

**gulp hello**

**Gulp**

Gulp js Setup by Mohit

Initialize Node JS

**npm init**

Install Gulp globally and locally

**npm install -g gulp**

**npm install gulp --save-dev**

Install Plugins

**npm install gulp-sass --save-dev**

**npm install gulp-dart-sass --save-dev**

**npm install gulp-autoprefixer --save-dev**

**npm install gulp-clean-css --save-dev**

**npm install gulp-terser --save-dev**

**npm install gulp-imagemin --save-dev**

**npm install imagemin-mozjpeg --save-dev**

**npm install imagemin-optipng --save-dev**

**npm install gulp-webp --save-dev**

**npm install gulp-concat --save-dev**

Add gulpfile.mjs

Create functions, watchtask, & default gulp

Run gulp

In gulpfile.mjs

**import gulp from 'gulp';**

**import sass from 'gulp-dart-sass'; *// Use gulp-dart-sass instead of gulp-sass***

**import prefix from 'gulp-autoprefixer';**

**import minify from 'gulp-clean-css';**

**import terser from 'gulp-terser';**

**import imagemin from 'gulp-imagemin';**

**import mozjpeg from 'imagemin-mozjpeg'; *// Import the MozJPEG plugin***

**import optipng from 'imagemin-optipng'; *// Import the OptiPNG plugin***

**import imagewebp from 'gulp-webp';**

**const { series, src, dest, watch } = gulp;**

***// Create functions...***

***// scss***

**function compilescss() {**

**return src('assets/scss/\*.scss')**

**.pipe(sass())**

**.pipe(prefix())**

**.pipe(minify())**

**.pipe(dest('assets/css/min/'))**

**}**

***// js***

**function jsmin() {**

**return src('assets/js/\*.js')**

**.pipe(terser())**

**.pipe(dest('assets/js/min/'))**

**}**

***// images***

**function optimizeimg() {**

**return src('assets/images/\*.{jpg,png}')**

**.pipe(imagemin([**

**mozjpeg({ quality: 70, progressive: true }), *// Use the imported MozJPEG plugin***

**optipng({ optimizationLevel: 2 }), *// Use the imported OptiPNG plugin***

**]))**

**.on('error', function(*err*) {**

**console.error('Error optimizing images:', *err*);**

**})**

**.pipe(dest('assets/images/'));**

**}**

***// convert from all images to webp images***

**function webpimg() {**

**return src('assets/images/\*.{jpg, png}')**

**.pipe(imagewebp())**

**.pipe(dest('assets/images/optimized/'))**

**}**

***// Create watch task***

**function watchTask() {**

**watch('assets/scss/\*.scss', compilescss);**

**watch('assets/js/\*.js', jsmin);**

**watch('assets/images/\*.{jpg, png}', optimizeimg);**

**watch('assets/images/\*.{jpg, png}', webpimg);**

**}**

***// default gulp***

**export default series(**

**compilescss,**

**jsmin,**

**optimizeimg,**

**webpimg,**

**watchTask**

**)**

👇 In WordPress inside theme folder > assets > gulp > gulpfile.mjs 👇

**import gulp from 'gulp';**

**import imagemin from 'gulp-imagemin';**

**import mozjpeg from 'imagemin-mozjpeg'; *// Import the MozJPEG plugin***

**import optipng from 'imagemin-optipng'; *// Import the OptiPNG plugin***

**const { series, src, dest, watch } = gulp;**

**let date = new Date();**

**let currMonth = date.getMonth();**

**currMonth = currMonth + 1;**

**currMonth = "0" + currMonth;**

**console.log(currMonth);**

***// images***

**function optimizeimg() {**

**return src(`../../../../uploads/2024/${currMonth}/\*.{jpg,png}`)**

**.pipe(imagemin([**

**mozjpeg({ quality: 70, progressive: true }), *// Use the imported MozJPEG plugin***

**optipng({ optimizationLevel: 2 }), *// Use the imported OptiPNG plugin***

**]))**

**.on('error', function(*err*) {**

**console.error('Error optimizing images:', *err*);**

**})**

**.pipe(dest(`../../../../uploads/2024/${currMonth}/`));**

**}**

***// Create watch task***

**function watchTask() {**

**watch(`../../../../uploads/2024/${currMonth}/\*.{jpg,png}`, optimizeimg);**

**}**

***// default gulp***

**export default series(**

**optimizeimg,**

**watchTask**

**)**

👇 In WordPress inside theme folder > assets > gulp > gulpfile.mjs 👇

YEAR DYNAMIC

import gulp from 'gulp';

import imagemin from 'gulp-imagemin';

import mozjpeg from 'imagemin-mozjpeg'; *// Import the MozJPEG plugin*

import optipng from 'imagemin-optipng'; *// Import the OptiPNG plugin*

const { series, src, dest, watch } = gulp;

*// images*

function optimizeimg() {

const date = new Date();

const year = date.getFullYear();

let month = (date.getMonth() + 1).toString().padStart(2, '0'); *// Current month, zero-padded to 2 digits*

const inputDir = `../../../../uploads/${year}/${month}/\*.{jpg,png}`;

const outputDir = `../../../../uploads/${year}/${month}/`;

return src(inputDir)

.pipe(imagemin([

mozjpeg({ quality: 70, progressive: true }), *// Use the imported MozJPEG plugin*

optipng({ optimizationLevel: 2 }), *// Use the imported OptiPNG plugin*

]))

.on('error', function(*err*) {

console.error('Error optimizing images:', *err*);

})

.pipe(dest(outputDir));

}

*// Create watch task*

function watchTask() {

const date = new Date();

const year = date.getFullYear();

let month = (date.getMonth() + 1).toString().padStart(2, '0'); *// Current month, zero-padded to 2 digits*

watch(`../../../../uploads/${year}/${month}/\*.{jpg,png}`, optimizeimg);

}

*// default gulp*

export default series(

optimizeimg,

watchTask

)

Concat all css / js in one file

**npm install gulp-concat --save-dev**

👇 Concat In gulpfile.mjs 👇

import gulp from 'gulp';

import sass from 'gulp-dart-sass'; *// Use gulp-dart-sass instead of gulp-sass*

import prefix from 'gulp-autoprefixer';

import minify from 'gulp-clean-css';

import concat from 'gulp-concat'; *// Import gulp-concat*

import terser from 'gulp-terser';

const { series, src, dest, watch } = gulp;

*// Create functions...*

*// scss*

function compilescss() {

return src('assets/scss/main.scss')

.pipe(sass())

.pipe(prefix())

.pipe(minify())

.pipe(dest('assets/css/min/'))

}

*// Concatenate CSS files*

function concatcss() {

return src('assets/css/min/\*.css') *// Change the source to the directory containing your minified CSS files*

.pipe(concat('all.min.css')) *// Concatenate all CSS files into one file (all.min.css)*

.pipe(dest('assets/css/')); *// Output the concatenated file to the specified directory*

}

function jsmin() {

return src('assets/js/\*.js')

.pipe(concat('all.min.js')) *// Concatenate all JS files into one file (all.min.js)*

.pipe(terser())

.pipe(dest('assets/js/min/'));

}

*// Create watch task*

function watchTask() {

watch('assets/scss/\*.scss', compilescss);

watch('assets/js/\*.js', jsmin);

}

*// default gulp*

export default series(

compilescss,

concatcss,

jsmin,

watchTask

)

**👇 Compile & minify scss from different folder’s path 👇**

*// scss*

function compilescss() {

*// Match SCSS files in 'assets/scss/' and its subdirectories*

return src('assets/scss/main.scss')

.pipe(sass())

.pipe(prefix())

.pipe(minify())

.pipe(dest('assets/css/min/'))

}

function anotherFolderCompilescss() {

*// Match SCSS files in 'assets/scss/' and its subdirectories*

return src('assets/anotherFolder/scss/main.scss')

.pipe(sass())

.pipe(prefix())

.pipe(minify())

.pipe(dest('assets/anotherFolder/css/min/'));

}

*// Create watch task*

function watchTask() {

watch('assets/scss/\*\*/\*.scss', compilescss); *// Use glob pattern to match all SCSS files in 'assets/scss/'*

watch('assets/anotherFolder/scss/\*\*/\*.scss', anotherFolderCompilescss); *// Use glob pattern to match all SCSS files in 'assets/anotherFolder/scss/'*

}

*// default gulp*

export default series(

compilescss,

anotherFolderCompilescss,

watchTask

)