



# PostCSS

**LoftSchool**  
от мыслителя к создателю

## Оглавление

PostCSS + Gulp	3
Подключение плагинов	5
nested	5
short	6
assets	7
stylelint + reporter	8
autoprefixer	10
CSSComb    CSSReader	11

## PostCSS + Gulp

Это все, что вам нужно!



Устанавливаем локально необходимые пакеты:

```
npm i --save gulp-postcss gulp concat
```

Подключаем их в сборку в **gulpfile.js**:

```
const concat = require('gulp-concat');
const gulp = require('gulp');
const postcss = require('gulp-postcss');

gulp.task('styles', function() {
  const processors = [

    return gulp.src('./css/**/*.css')
      .pipe(postcss(processors))
      .pipe(concat('bundle.min.css'))
      .pipe(gulp.dest('./public'));
  });

  gulp.task('watch', function() {
    gulp.watch('./css/**/*.css', function() {
      gulp.run('styles');
    })
  });

  gulp.task('default', ['styles', 'watch']);
```

## Подключение плагинов

Весь перечень плагинов с примерами и инструкциями можно найти на сайте <http://postcss.parts/>

Для подключения плагина в сборку необходимо:

- Скачать и установить локально с помощью команды

```
npm i --save *название плагина*
```

- Подключить в сборку в файл **gulpfile.js** с помощью команды **require**
- Добавить в массив плагинов передаваемый в postcss – в нашем, частном примере, это переменная **processors** типа массив

### nested

Возможности:

Преобразовывает из:	В:
<pre>.phone {   &amp;_title {     width: 500px;     @media (max-width: 500px) {       width: auto;     }     body.is_dark &amp; {       color: white;     }   }   img {     display: block;   } }</pre>	<pre>.phone_title {   width: 500px; } @media (max-width: 500px) {   .phone_title {     width: auto;   } } body.is_dark .phone_title {   color: white; } .phone img {   display: block; }</pre>

## Установка:

```
npm i --save postcss-nested
```

## short

## Возможности:

Преобразовывает из:	В:
<pre>.icon {   size: 48px; }</pre>	<pre>.icon {   width: 48px;   height: 48px; }</pre>
<pre>.frame {   margin: * auto; }</pre>	<pre>.frame {   margin-right: auto;   margin-left: auto; }</pre>
<pre>.banner {   position: fixed 0 0 *; }</pre>	<pre>.banner {   position: fixed;   top: 0;   right: 0;   left: 0; }</pre>
<pre>.canvas {   color: #abccfc #212231; }</pre>	<pre>.canvas {   color: #abccfc;   background-color: #212231; }</pre>
<pre>.container {   border: 1px 2px #343434; }</pre>	<pre>.container {   border-width: 1px 2px;   border-color: #343434; }</pre>
<pre>.heading {   font-size: 1.25em 2; }</pre>	<pre>.heading {   font-size: 1.25em;   line-height: 2; }</pre>
<pre>p {   font-weight: light; }</pre>	<pre>p {   font-weight: 300; }</pre>

<pre>.section {   text: dimgrey bold center uppercase 1.25em 1.5 .05em; }</pre>	<pre>.section {   color: dimgrey;   font-weight: 700;   text-align: center;   text-transform: uppercase;   font-size: 1.25em;   line-height: 1.5;   letter-spacing: .05em; }</pre>
<pre>.menu-item[-active] {   background: #f3f3f3; }</pre>	<pre>.menu-item[data-active] {   background: #f3f3f3; }</pre>

Установка:

```
npm i --save postcss-short
```

assets

Основные возможности:

Что:	Как:
Инлайнить изображения в стили:	<pre>body {   background: inline('images/foobar.png'); }</pre>
Узнавать размеры изображения:	<pre>body {   width: width('images/foobar.png', 2); /* 160px */   height: height('images/foobar.png', 2); /* 120px */   background-size: size('images/foobar.png', 2); /* 160px 120px */ }</pre>

## Установка:

```
npm i --save postcss-assets
```

### stylelint + reporter

**StyleLint** – проверяет ваш код css на стиль кода. Например: отступы, пробелы, единицы переменных после нуля.

В аргументы StyleLint нужно передать конфигурацию или файл с конфигурацией. **В зависимости от версии плагина меняется формат конфигурации.** Будьте осторожны!

Я рекомендую использовать уже готовую мою конфигурацию на первых порах: <https://github.com/6thSence/gulp-postcss/blob/master/stylelintrc.json>

Она сконфигурирована специально для версии StyleLint 1.2.1. Будьте внимательны!

Для того, чтобы установить именно эту версию плагина, нужно добавить в ваши зависимости в **package.json** строку:

```
"stylelint": "1.2.1"
```

Пример package.json:

```
{
  "name": "gulp-postcss",
  "version": "1.0.0",
  "description": "",
  "main": "gulpfile.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
```




```


    "url": "git+https://github.com/6thSence/gulp-postcss.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/6thSence/gulp-postcss/issues"
  },
  "homepage": "https://github.com/6thSence/gulp-postcss#readme",
  "dependencies": {
    "autoprefixer": "6.3.6",
    "gulp": "3.9.1",
    "gulp-concat": "2.6.0",
    "postcss-assets": "4.1.0",
    "postcss-browser-reporter": "0.5.0",
    "postcss-nested": "1.0.0",
    "postcss-short": "1.4.0",
    "stylelint": "1.2.1",
    "gulp-postcss": "6.1.0"
  }
}

```

После подключения StyleLint все ошибки будут выводиться в консоль, для вывода ошибки в браузер вам нужен плагин **postcss-browser-reporter**. Достаточно подключить его после StyleLint в массиве плагинов postcss.

Так выглядит стандартная ошибка в браузере:


autoprefixer: menu.css:4:4: Outdated direction syntax in gradient  
postcss-important: header.css:10:4: Avoid !important declaration



**WIKIPEDIA**  
The Free Encyclopedia


[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)

## Warning

From Wikipedia, the free encyclopedia

*For templates used to warn users who vandalize Wikipedia articles, see [Wikipedia:Template messages/User talk namespace/Warnings](#).*

**Warning** may refer to:



Look up ***warning*** in  
Wiktionary, the free

## autoprefixer

### Возможности:

Преобразовывает из:	В:
<pre>:fullscreen a {   display: flex }</pre>	<pre>:-webkit-full-screen a {   display: -webkit-box;   display: flex } :-moz-full-screen a {   display: flex } :-ms-fullscreen a {   display: -ms-flexbox;   display: flex } :fullscreen a {   display: -webkit-box;   display: -ms-flexbox;   display: flex }</pre>
<pre>a {   display: flex; }</pre>	<pre>a {   display: -webkit-box;   display: -webkit-flex;   display: -ms-flexbox;   display: flex }</pre>

### Установка:

```
npm i --save autoprefixer
```

## CSSComb || CSSReader

Сортировка порядка свойств во имя стиля кода.

Для Sublime:

<https://github.com/csscomb/sublime-csscomb>

Для Atome:

<https://atom.io/packages/atom-csscomb>

Для Brackets:

<https://github.com/i-akhmadullin/brackets-csscomb>

Для WebStorm (or PhpStorm):

<https://github.com/anton-rudeshko/CSSReorder/>