

# Postcss – новая эра фронтенда или почему sass должен умереть



**Postcss** vs **Sass**

# Цитркоп Алексей

frontend-разработчик [7bits](#)



# PostCSS

**Зачем?**

**Что умеет postcss?**

**Ничего**

**Всё делают плагины**

**Что умеет Sass?**



# Переменные: Sass

```
$font-stack: Helvetica, sans-serif;
```

```
$primary-color: #333;
```

```
body {
```

```
    font: 100% $font-stack;
```

```
    color: $primary-color;
```

```
}
```

# Переменные: postcss-simple-vars

\$font-stack: Helvetica, sans-serif;

\$primary-color: #333;

body {

font: 100% \$font-stack;

color: \$primary-color;

}

# Вложенность: Sass

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
  }  
  li { display: inline-block }  
}
```

# Вложенность: postcss-nested

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
  }  
  li { display: inline-block }  
}
```

# Вложенность: postcss-nested

```
.phone {  
  &_title {  
    width: 500px;  
    body.is_dark & { color: white; }  
  }  
  img { width: 500px; }  
}
```

# Вложенность: postcss-nested

```
.phone_title { width: 500px; }
```

```
body.is_dark .phone_title { color: white; }
```

```
.phone img { width: 500px; }
```

# Миксины: Sass

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius; }  
.box { @include border-radius(10px) }
```

# Миксины: postcss-mixins

```
@define-mixin border-radius $radius {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius; }  
.box { @mixin border-radius 10px; }
```



# Миксины: postcss-mixins

```
@define-mixin icon $network, $color: blue {  
  .icon.is-$(network) {  
    color: $color;  
    @mixin-content; }  
  .icon.is-$(network):hover { color: white; }  
}  
  
@mixin icon twitter { background: url(twt.png); }
```

# Миксины: postcss-mixins

```
.icon.is-twitter {  
    color: blue;  
    background: url(twt.png);  
}  
  
.icon.is-twitter:hover {  
    color: white;  
}
```

# Наследование: Sass

```
.message {  
  border: 1px solid #ccc;  
  color: #333; }  
  
.success {  
  @extend .message;  
  border-color: green; }
```

# Наследование: postcss-simple-extend

```
@define-placeholder message {  
  border: 1px solid #ccc;  
  color: #333; }  
  
.success {  
  @extend .message;  
  border-color: green; }
```

# Операторы: Sass

```
.container {  
    width: 600px / 960px * 100%;  
}
```

# Операторы: postcss-calc

```
.container {  
    width: calc(600px / 960px * 100%);  
}
```

# Импорт: Sass

```
// _reset.scss
```

```
html, body { margin: 0; }
```

```
/* base.scss */
```

```
@import 'reset';
```

```
body { background-color: #efefef; }
```

```
...
```

# Импорт: postcss-import

```
// _reset.css
```

```
html, body { margin: 0; }
```

```
/* base.css */
```

```
@import 'reset.css';
```

```
body { background-color: #efefef; }
```

```
...
```



# Импорт: postcss-import

```
@import 'reset';
```

```
@import 'css/foo.css'; (min-width: 25em)
```

```
body { background-color: #efefef; }
```

```
...
```

# Импорт: postcss-import

```
/* content of ./node_modules/foo/index.css */  
@media(min-width: 25em) {  
    /* content of bar.css */  
}  
body { background-color: #efefef; }  
...
```

**что еще?**

# Autoprefixer

```
:fullscreen a {  
  display: flex;  
}
```

# Autoprefixer

```
:-webkit-full-screen a {  
    display: -webkit-box;  
    display: -webkit-flex;  
    display: flex;  
}  
:-moz-full-screen a {  
    display: flex;  
}  
:-ms-fullscreen a {  
    display: -ms-flexbox;  
    display: flex;  
}  
:fullscreen a {  
    display: -webkit-box;  
    display: -webkit-flex;  
    display: -ms-flexbox;  
    display: flex;  
}
```

## postcss-assets

```
var options = { loadPaths:['media/img/', 'images/'] }
```

```
body {  
  background: resolve('foobar.jpg');  
  background: resolve('icons/baz.png');  
}
```

# postcss-assets

```
body {  
  background: resolve('/media/img/foobar.jpg');  
  background: resolve('/images/icons/baz.png');  
}
```

# postcss-assets

```
body {  
  width: width('/images/foobar.png');  
  height: height('/images/foobar.png');  
  background-size: size('/images/foobar.png');  
}
```



# postcss-assets

```
body {  
  width: 320px;  
  height: 240px;  
  background-size: 320px 240px;  
}
```

# postcss-assets

```
body {  
  width: width('images/foobar.png', 2);  
  height: height('images/foobar.png', 2);  
  background-size: size('images/foobar.png', 2);  
}
```

# postcss-assets

```
body {  
  width: 160px;  
  height: 120px;  
  background-size: 160px 120px;  
}
```

# postcss-assets

```
body {  
  background: inline('images/foobar.png');  
}
```

# Отзеркаливание: RTLCSS

This screenshot shows the GitHub profile of Mohammad Younes from a standard desktop perspective. The layout is left-aligned. The profile header on the left includes a profile picture, name, bio, and contact information. The main content area on the right is divided into sections: 'Popular repositories' (listing AlertifyJS, rtlcss, MVC5-MixedAuth, jquery-scrollLock, and OWIN-MixedAuth), 'Repositories contributed to' (listing madskristensen/WebEssentials, Semantic-Org/Semantic-UI, jsdelivr/jsdelivr, twbs/bootstrap, and Semantic-Org/Semantic-UI-Docs), and 'Public contributions' (a calendar heatmap). At the bottom, there are statistics for 'Contributions in the last year' (408 total), 'Longest streak' (10 days), and 'Current streak' (2 days), followed by a 'Contribution activity' section showing recent commits.

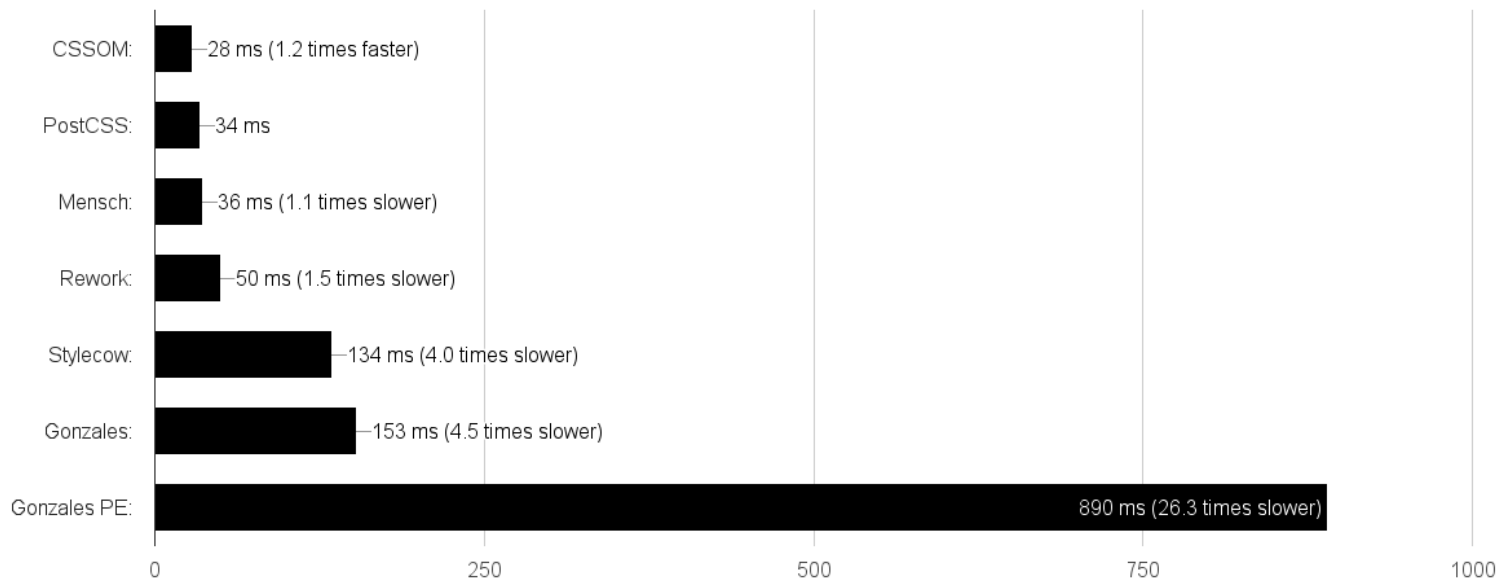
This screenshot shows the same GitHub profile from a right-aligned perspective. The layout is mirrored, with the profile header on the right and the repository and contribution sections on the left. The statistics and contribution activity sections are also mirrored, maintaining the same data as the left view but with the text and layout elements right-aligned.

# PostCss

Более 90 плагинов

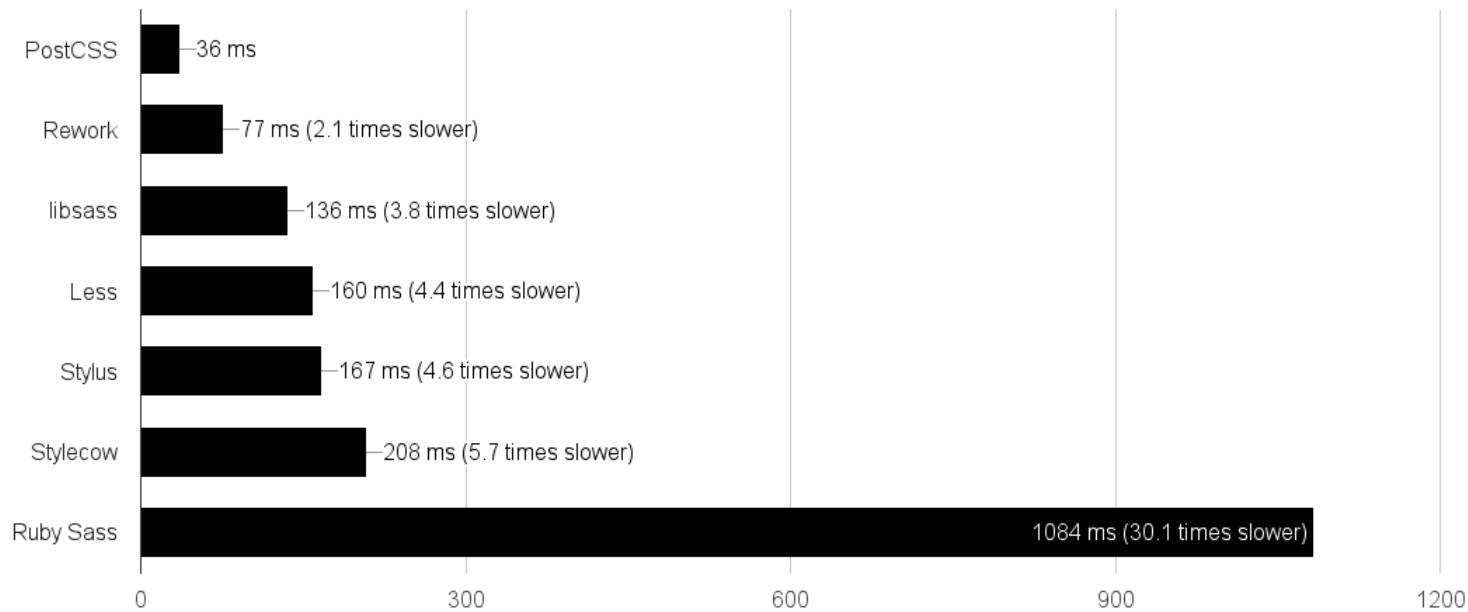
**Скорость?**

# Парсинг CSS





# Парсинг и д.р.



# Плюсы

- Скорость
- Больше функций
- Расширяемость

# Используем в Gulp

```
var postcss = require('gulp-postcss');

gulp.task('compile:css', function () {

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

    .pipe( postcss([plugin1, plugin2, plugin3]) )

    .pipe( gulp.dest('dist') );

});
```

# Свой плагин

```
var postcss = require('postcss');  
module.exports = postcss.plugin('postcss-image-set', function (opts) {  
  opts = opts || {};  
  
  return function (css) {  
    css.eachDecl('background-image', function (decl) {  
      if (!decl.value || decl.value.indexOf('image-set') === -1) { return; }  
      var image = getDefaultImage(decl.value);  
      decl.cloneBefore({  
        prop: 'background-image',  
        value: image  
      });  
    });  
  });  
};
```

**Спасибо за внимание**