



# Make a CSS sandwich by using postCSS as a bread

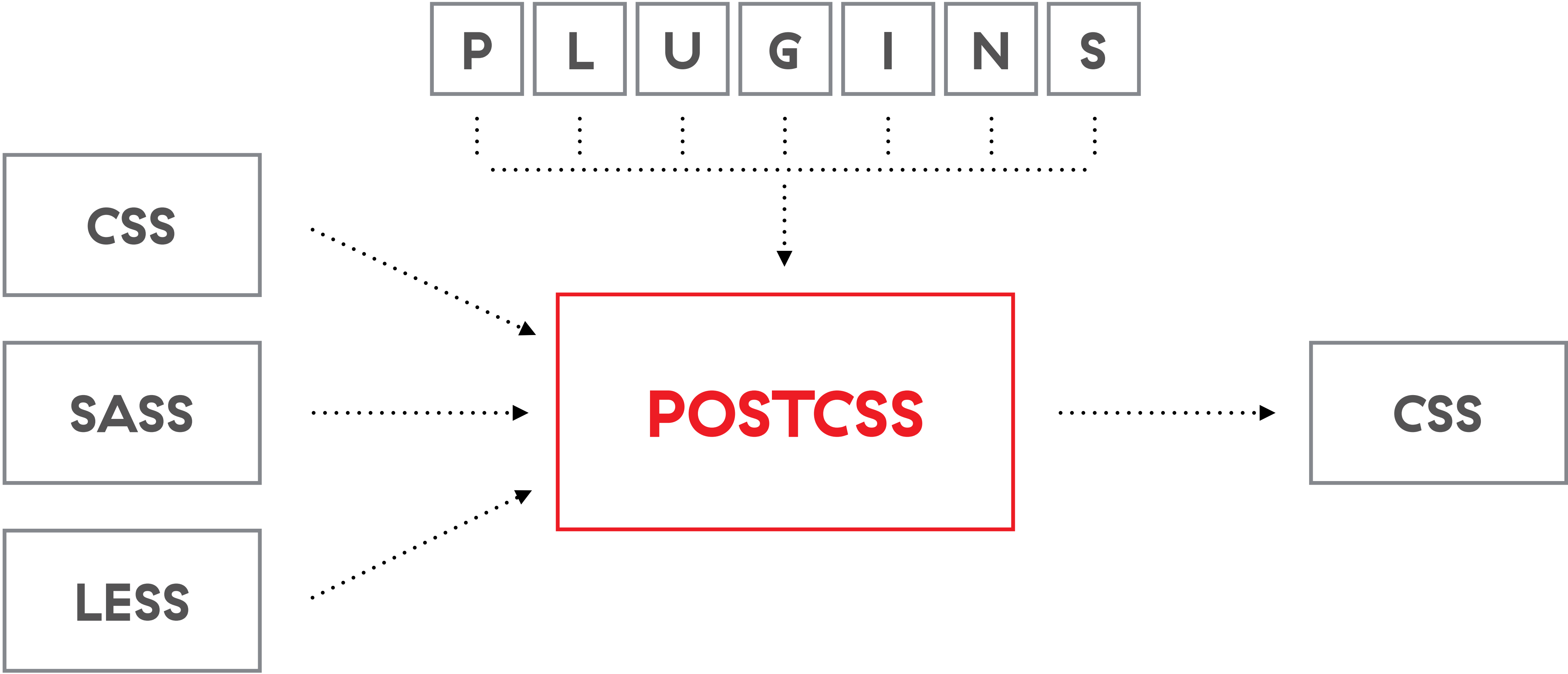
IVAN RUŽEVIĆ

# POSTCSS IN A NUTSHELL

*“PostCSS is a tool for transforming CSS with JS plugins. These plugins can support variables and mixins, transpile future CSS syntax, inline images, and more.”*

– @GITHUB

# LOOK, A COOL DIAGRAM!



# WHAT POSTCSS IS NOT

Not a Pre-processor

Not a Post-processor

Not “Future Syntax”

Not a Clean Up Tool

But it can be!

# WHAT POSTCSS IS

PostCSS is Not Any One Thing but  
think it as a bread to your sandwich.

# WHAT MAKES POSTCSS SPECIAL

Speed

Your plugins

Non dependent libraries

Seamless deployment

and many more...

# SPEED

## UP TO 3 TIMES FASTER

- PostCSS: **36 ms**
- Rework: 77 ms (2.1 times slower)
- libsass: 136 ms (3.8 times slower)
- Less: 160 ms (4.4 times slower)
- Stylus: 167 ms (4.6 times slower)
- Stylecow: 208 ms (5.7 times slower)
- Ruby Sass: **1084 ms (30.1 times slower)**

[HTTPS://GITHUB.COM/POSTCSS/BENCHMARK](https://github.com/postcss/benchmark)



**PLUGINS**

# WHAT ARE POSTCSS PLUGINS

Currently more than 200 active plugins

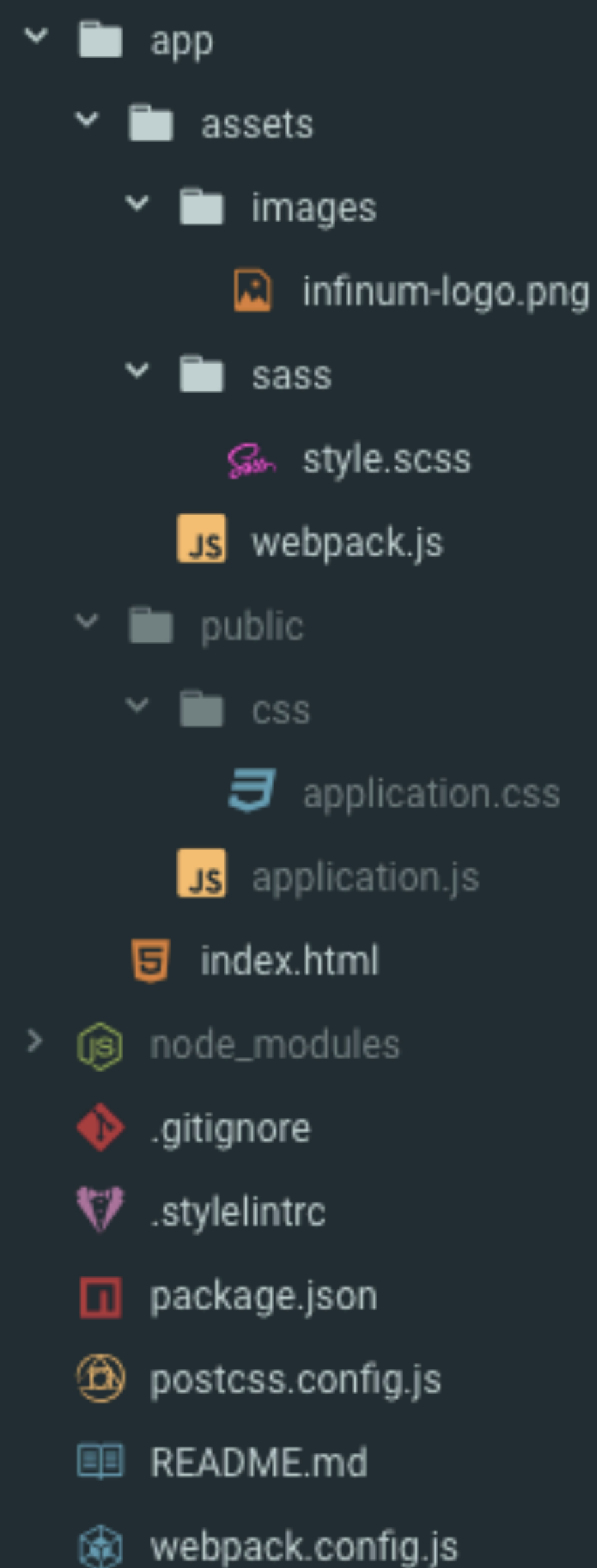
[HTTP://POSTCSS.PARTS/](http://postcss.parts/)

**MAKE YOUR OWN**

**HOW TO SETUP**

# HOW TO SETUP

- Add PostCSS loader to your favourite task runner.
- Create postcss.config.js
- Install plugins that you need via npm installer and add it to the postcss config file.
- Run task and see the magic happen



## WEBPACK.CONFIG.JS

```
use: ExtractTextPlugin.extract({  
  fallback: 'style-loader',  
  use: [{  
    loader: 'css-loader'  
  }, {  
    loader: 'postcss-loader'  
  }, {  
    loader: 'sass-loader'  
  }],  
})
```

## PACKAGE.JSON

```
{
  "name": "PostCSS",
  "devDependencies": {
    "autoprefixer": "^6.7.2",
    "css-loader": "^0.26.1",
    "css-mqpacker": "^5.0.1",
    "cssnano": "^3.10.0",
    "extract-text-webpack-plugin": "^2.0.0-rc.3",
    "file-loader": "^0.9.0",
    "media-blender": "^2.1.0",
    "node-sass": "^4.5.0",
    "postcss-cssnext": "^2.9.0",
    "postcss-font-magician": "^1.6.0",
    "postcss-loader": "^1.2.2",
    "sass-loader": "^4.1.1",
    "style-loader": "^0.13.1",
    "webpack": "^2.2.1"
  },
  "browserslist": [
    "> 1%",
    "last 5 versions"
  ]
}
```

## POSTCSS.CONFIG.JS

```
module.exports = {
  plugins: [

    require('autoprefixer'),

    require('postcss-font-magician')({
      protocol: 'https:',
      variants: {
        'Roboto Condensed': {
          '300': [],
          '400': [],
          '700': []
        }
      },
    },
    foundries: ['google']
  )),

  require('css-mqpacker'),

  require('postcss-cssnext'),

  require('cssnano')
]
}
```





**EXAMPLES**

**AUTOPREFIXER**

## POSTCSS.CONFIG.JS

```
require('autoprefixer')({  
  browserslist: [  
    "> 1%",  
    "last 5 versions"  
  ]  
})
```

## INPUT – SASS

```
.autoprefix {  
  display: flex;  
}
```

## OUTPUT – CSS

```
.autoprefix {  
  display: -webkit-box;  
  display: -ms-flexbox;  
  display: flex;  
}
```

**POSTCSS-FONT-MAGICIAN**

## POSTCSS.CONFIG.JS

```
require('postcss-font-magician')({
  protocol: 'https:',
  variants: {
    'Roboto Condensed': {
      '300': []
    }
  },
  foundries: ['google']
}),
```

## INPUT - SASS

```
.fontFamily {
  font-family: 'Roboto Condensed';
}
```

## OUTPUT - CSS

```
@font-face {
  font-family: "Roboto Condensed";
  font-style: normal;
  font-weight: 300;
  src:
    local("Roboto Condensed Light"),
    local(RobotoCondensed-Light),
    url(https://fonts.gstatic.com/s/robotocondensed/v13/
b9QBgL0iMZfDSpmcXcE8nCrW_eugtb5_efFKynOEdiw.eot?#)
    format("eot"),
    url(https://fonts.gstatic.com/s/robotocondensed/v13/
b9QBgL0iMZfDSpmcXcE8nNeiznanRB67rGZDYkMvKPc.woff)
    format("woff2"),
    url(https://fonts.gstatic.com/s/robotocondensed/v13/
b9QBgL0iMZfDSpmcXcE8nNeiznanRB67rGZDYkMvKPc.woff)
    format("woff");
}

.fontFamily {
  font-family: 'Roboto Condensed';
}
```

**CSS-MQPACKER**

## POSTCSS.CONFIG.JS

```
require( 'css-mqpacker' )
```

## INPUT – SASS

```
@import '~media-blender';
.mqPacker {
  .box1 {
    width: 800px;
    @include media(small) {
      width: 500px;
    }
  }
  .box2 {
    width: 300px;
    @include media(tablet) {
      width: 100px;
    }
    @include media(small) {
      width: 100px;
    }
  }
  .box3 {
    @include media(tablet) {
      width: 50px;
    }

    @include media(small) {
      width: 30px;
    }
  }
}
```

## OUTPUT – CSS

```
.mqPacker .box1 {
  width: 800px;
}
.mqPacker .box2 {
  width: 300px;
}
@media (max-width: 543px) {
  .mqPacker .box1 {
    width: 500px;
  }
  .mqPacker .box2 {
    width: 100px;
  }
  .mqPacker .box3 {
    width: 30px;
  }
}
@media (min-width: 768px) and (max-width: 991px) {
  .mqPacker .box2 {
    width: 100px;
  }
  .mqPacker .box3 {
    width: 50px;
  }
}
```

**POSTCSS – CSSNEXT**



## POSTCSS.CONFIG.JS

```
require('postcss-cssnext')
```

## INPUT - SASS

```
:root {
  --fontSize: 1rem;
  --color: 180;
  --borderColor: 0;
  --bgColor: #F4F4F4;
}
.nextCss {
  padding: calc((var(--fontSize)/2)+10px);
  color: gray(var(--color));
  border: 1px solid gray(var(--borderColor), 30%);
  filter: sepia(0.8);
  display: block;
  width: 300px;
  height: 100px;
  font-size: 30px;
  text-align: center;
  line-height: 100px;
  background-color: var(--bgColor);
  font-weight: bold;
}
```

## OUTPUT - CSS

```
.nextCss {
  padding: calc(0.5rem + 10px);
  color: rgb(180, 180, 180);
  border: 1px solid rgba(0, 0, 0, 0.3);
  filter: url('data:image/
svg+xml;charset=utf-8,<svg xmlns="http://
www.w3.org/2000/svg"><filter
id="filter"><feColorMatrix type="matrix" color-
interpolation-filters="sRGB" values="0.5144
0.61520000000000001 0.1512 0 0 0.2792 0.7488
0.134400000000000002 0 0 0.217600000000000002
0.4272 0.304799999999999996 0 0 0 0 0 1 0" /></
filter></svg>#filter');
  -webkit-filter: sepia(0.8);
  filter: sepia(0.8);
  display: block;
  width: 300px;
  height: 100px;
  font-size: 30px;
  text-align: center;
  line-height: 100px;
  background-color: #F4F4F4;
  font-weight: bold;
}
```

**CSSNANO**

## POSTCSS.CONFIG.JS

```
require('cssnano')
```

## OUTPUT - CSS

```
@font-face{font-family:Roboto Condensed;font-style:normal;font-weight:300;src:local("Roboto Condensed Light"),local(RobotoCondensed-Light),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nCrW_eugtb5_efFKynOEidiw.eot#) format("eot"),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nNeiznanRB67rGZDYkMvKPc.woff) format("woff2"),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nNeiznanRB67rGZDYkMvKPc.woff) format("woff")}@font-face{font-family:Roboto Condensed;font-style:normal;font-weight:400;src:local("Roboto Condensed"),local(RobotoCondensed-Regular),url(https://fonts.gstatic.com/s/robotocondensed/v13/Zd2E9abXLFGSr9G3YK2MsG8ITcfo9NwJpvZiO7_FxEg.eot#) format("eot"),url(https://fonts.gstatic.com/s/robotocondensed/v13/Zd2E9abXLFGSr9G3YK2MsFzqCfRpIA3W6ypxnPISCPA.woff) format("woff2"),url(https://fonts.gstatic.com/s/robotocondensed/v13/Zd2E9abXLFGSr9G3YK2MsFzqCfRpIA3W6ypxnPISCPA.woff) format("woff")}@font-face{font-family:Roboto Condensed;font-style:normal;font-weight:700;src:local("Roboto Condensed Bold"),local(RobotoCondensed-Bold),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nJTpJ4fOLLwmHp8vl0kIMg4.eot#) format("eot"),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nCSLrGe-fkSRw2DeVgOoWcQ.woff) format("woff2"),url(https://fonts.gstatic.com/s/robotocondensed/v13/b9QBgL0iMZfDSpmcXcE8nCSLrGe-fkSRw2DeVgOoWcQ.woff) format("woff")}.autoprefixer{display:-webkit-box;display:-ms-flexbox;display:flex}.fontFamily,body{font-family:Roboto Condensed}.mqPacker .box1{width:800px}.mqPacker .box2,.nextCss{width:300px}.nextCss{padding:calc(0.5rem + 10px);color:#b4b4b4;border:1px solid
```

**CONCLUSION**

# JUST START USING IT

If there's one thing you really need to know about PostCSS, it's that you should learn what it is and how to use it ASAP.



# Any questions?

IVAN.RUZEVIC@INFINUM.CO  
@IRUZEVIC

Visit [infinum.co](https://infinum.co) or find us on social networks:

