

# تحويل CSS إلى CSS3 باستخدام

الإضافات باستخدام CSS لتحويل أداة هو CSS3. واجهة مشروع في CSS3 استخدام كفيّة سأشرح خطوة دليل إليك المستقبليّة. CSS3 اللغة ودعم والتقليد، التمهيدية، الأوامر إضافة مثل ميّزات وتقدم، CSS3: بخطوة:

## 1. مشروعك إعداد

بإنشاءه: قم كذلك، يكن لم إذا مرفوع. CSS3. مع الاستخدام واجهة مشروع لديك أن من تأكد

```
mkdir my-project
cd my-project
npm init -y
```

## 2. المملوطة والأدوات تثبيت

أو التمهيدية، للأوامر autoprefixer مثل استخدام تريدي التي الإضافات وكل، CSS3، وواجهة، تثبيت بيت قم للتلقليد: CSSnano:

```
npm install --save-dev postcss postcss-cli autoprefixer cssnano
```

## 3. ملف إنشاء

ملف إنشاء، CSS3، مثل المصدر، src/styles.css:

```
/* src/styles.css */
.container {
  display: flex;
  transition: all 0.3s ease;
}
```

## 4. تكويّن CSS3

الإضافات: لتجديد مشروعك جنور في postcss.config.js ملف إنشاء

```
// postcss.config.js
module.exports = {
  plugins: [
    require('autoprefixer'), //
    require('cssnano')({ preset: 'default' }) // CSS
  ]
};
```

## 5. بناء نص إضافة 5.

بإستخدام `package.json` ملف في:

```
"scripts": {
  "build:css": "postcss src/styles.css -o dist/styles.css"
}
```

- src/styles.css: ال إدخال ملف
- dist/styles.css: ال إخراج ملف

## 6. تشغيل 6.

بناء: النص بتشغيل ق م

```
npm run build:css
```

الأوامر `autoprefixer` يضيف قد الأمثال، سبيل على `dist/styles.css` إلى المعالج `src/styles.css` ويخرج `dist/styles.css` يعالج هذا من ه: `cssnano` ويقلل التمهيدية،

```
/* dist/styles.css */
.container{display:-webkit-box;display:-ms-flexbox;display:flex;-webkit-transition:all .3s ease;transition:all
```

## 7. اختياري 7. بناء أداة مع الدمج

بإستخدام `vite.config.js`، `postcss`، `autoprefixer`، `cssnano`، `postcss` مثل وأدوات `postcss` بين بالدمج ق م قوة، أكثر لإعداد

مع `vite.config.js` وتكويته `postcss` بتثبيت ق م، `postcss` تستخدم كنت إذا `postcss` مع

```
// vite.config.js
import postcss from 'postcss';
import autoprefixer from 'autoprefixer';
```

```
export default {
  css: {
    postcss: {
      plugins: [autoprefixer()]
    }
  }
};
```

ملفات استيراد عند تلقائي `postcss` يعالج `postcss`.

مع `postcss-loader` بتثبيت ق م

```
npm install --save-dev postcss-loader
```

بلك: الخاص `webpack.config.js` ملف بتحديث ق م

```
module.exports = {
  module: {
    rules: [
      {
        test: /\.css$/,
        use: ['style-loader', 'css-loader', 'postcss-loader']
      }
    ]
  }
};
```

## 8. اختياري التغيرات مراقبة

لعمل `package.json` في النص بتعديل ق م التطوير، أثناء تلقائيًا

```
"scripts": {
  "watch:css": "postcss src/styles.css -o dist/styles.css --watch"
}
```

الملفات. في التغيرات لمراقبة `npm run watch:css` بتشغيل ق م

## 9. الإضافات

`cssnano` الـمتصفح. دعم على ياس `autoprefixer`: الـأوامر يضيء ل-الإضافة الإضافات بعض هنا `postcss-import` اليوم. الـمستقبلية `postcss-preset-env`: `postcss-preset-env` من يقلل الحاجة: حسب بتثبيتها ق م القواعد. `@import`

```
npm install --save-dev postcss-preset-env postcss-import
```

إلى أضفها `postcss.config.js`:

```
module.exports = {
  plugins: [
    require('postcss-import'),
    require('postcss-preset-env'),
  ]
};
```

