

PostCSS

▼ Examples

- [Tailwind CSS Example ↗](#)

Default Behavior

Next.js compiles CSS for its [built-in CSS support](#) using PostCSS.

Out of the box, with no configuration, Next.js compiles CSS with the following transformations:

- [Autoprefixer ↗](#) automatically adds vendor prefixes to CSS rules (back to IE11).
- [Cross-browser Flexbox bugs ↗](#) are corrected to behave like [the spec ↗](#).
- New CSS features are automatically compiled for Internet Explorer 11 compatibility:
 - `all` [Property ↗](#)
 - [Break Properties ↗](#)
 - `font-variant` [Property ↗](#)
 - [Gap Properties ↗](#)
 - [Media Query Ranges ↗](#)

By default, [CSS Grid ↗](#) and [Custom Properties ↗](#) (CSS variables) are **not compiled** for IE11 support.

To compile [CSS Grid Layout ↗](#) for IE11, you can place the following comment at the top of your CSS file:

```
/* autoprefixer grid: autoplace */
```

You can also enable IE11 support for [CSS Grid Layout](#) in your entire project by configuring autoprefixer with the configuration shown below (collapsed). See "[Customizing Plugins](#)" below for more information.

► **Click to view the configuration to enable CSS Grid Layout**

CSS variables are not compiled because it is [not possible to safely do so](#). If you must use variables, consider using something like [Sass variables](#) which are compiled away by [Sass](#).

Customizing Target Browsers

Next.js allows you to configure the target browsers (for [Autoprefixer](#) and compiled css features) through [Browserslist](#).

To customize browserslist, create a `browserslist` key in your `package.json` like so:

```
1 {  
2   "browserslist": [ ">0.3%", "not dead", "not op_mini all" ]  
3 }
```

You can use the [browserslist](#) tool to visualize what browsers you are targeting.

CSS Modules

No configuration is needed to support CSS Modules. To enable CSS Modules for a file, rename the file to have the extension `.module.css`.

You can learn more about [Next.js' CSS Module support here](#).

Customizing Plugins

Warning: When you define a custom PostCSS configuration file, Next.js **completely disables** the [default behavior](#). Be sure to manually configure all the features you need compiled, including [Autoprefixer](#). You also need to install any plugins included in your custom configuration manually, i.e.

```
npm install postcss-flexbugs-fixes postcss-preset-env
```

To customize the PostCSS configuration, create a `postcss.config.json` file in the root of your project.

This is the default configuration used by Next.js:

 `postcss.config.json` 

```
1  {
2    "plugins": [
3      "postcss-flexbugs-fixes",
4      [
5        "postcss-preset-env",
6        {
7          "autoprefixer": {
8            "flexbox": "no-2009"
9          },
10         "stage": 3,
11         "features": {
12           "custom-properties": false
13         }
14       ]
15     ]
16   }
17 }
```

Note: Next.js also allows the file to be named `.postcssrc.json`, or, to be read from the `postcss` key in `package.json`.

It is also possible to configure PostCSS with a `postcss.config.js` file, which is useful when you want to conditionally include plugins based on environment:

 `postcss.config.js` 

```
1  module.exports = {
2    plugins:
3      process.env.NODE_ENV === 'production'
4        ? [
5          'postcss-flexbugs-fixes',
6          [
7            'postcss-preset-env',
8            {
9              autoprefixer: {
10                flexbox: 'no-2009',
11              },
12              stage: 3,
13              features: {
14                'custom-properties': false,
15              },
16            ],
17          ],
```

```
18     ]
19     : [
20         // No transformations in development
21     ],
22 };
```

Note: Next.js also allows the file to be named `.postcssrc.js`.

Do **not** use `require()` to import the PostCSS Plugins. Plugins must be provided as strings.

Note: If your `postcss.config.js` needs to support other non-Next.js tools in the same project, you must use the interoperable object-based format instead:

```
1  module.exports = {
2    plugins: {
3      'postcss-flexbugs-fixes': {},
4      'postcss-preset-env': {
5        autoprefixer: {
6          flexbox: 'no-2009',
7        },
8        stage: 3,
9        features: {
10         'custom-properties': false,
11       },
12     },
13   },
14 };
```