

# Parcel Bundler

# ES Modules

Are small modular containers of code.  
That give JS developers a standard  
format importing and exporting  
reusable pieces of code functionality.

Modules are reusable you can use them anywhere in your code again and again with no increase in file size.

We can add additional features that are by adding `node_modules`.

Node Modules are not directly understood by the browser.

You therefore have to use a bundler.

# How It Works



# PARCEL

Blazing fast, zero configuration web application bundler



Takes your **css** files attached to the index.html document and optimizes them before moving them to the build folder

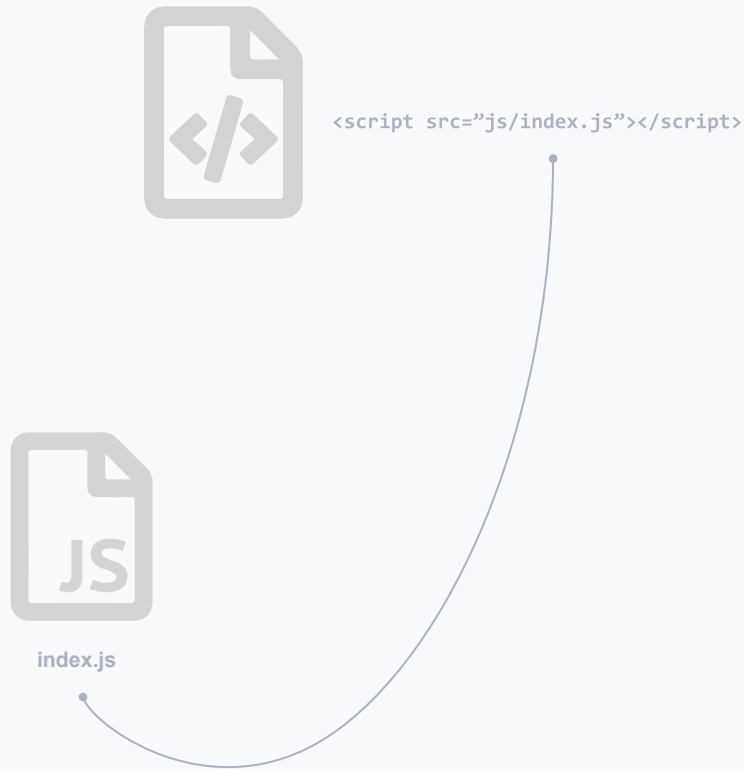


```
<link rel="stylesheet" href="css/styles.css"/>
```

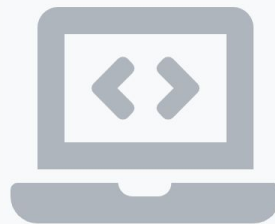


styles.css

Takes your **js** files attached to the index.html document and optimizes them before moving them to the build folder



It looks through all your import statements and adds your modules and any modules that you have imported from the node\_modules directory



index.js

Import \$ from 'jQuery'



data-view.js

Import \$ from 'jQuery'



dataFetcher.js

Import \$ from 'jQuery'

# Parcel

only includes one copy of jQuery in the final bundle.



`import $ from 'jQuery'`



`index.js`



`data-view.js`



`dataFetcher.js`

# Parcel

Keeps a record of your project as you build and add new modules or make changes.

# Parcel

When your finished you run your **package.json npm run build** script to create the production bundle.