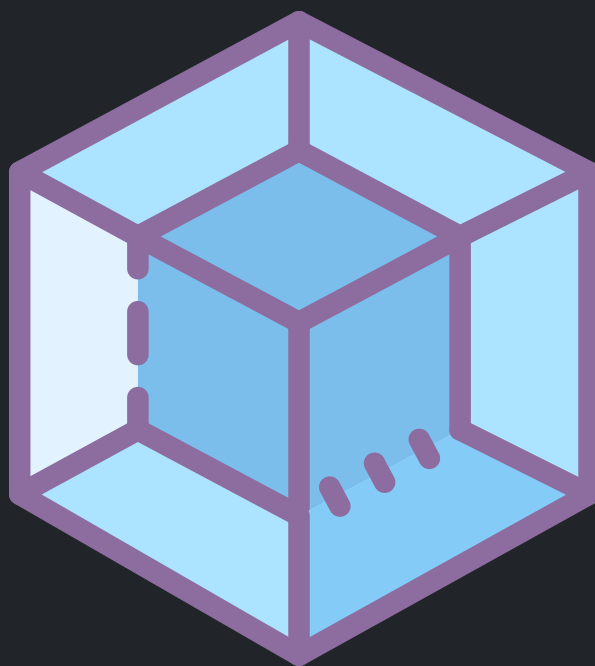




Save it



Like it



Bundle your assets using **Webpack**

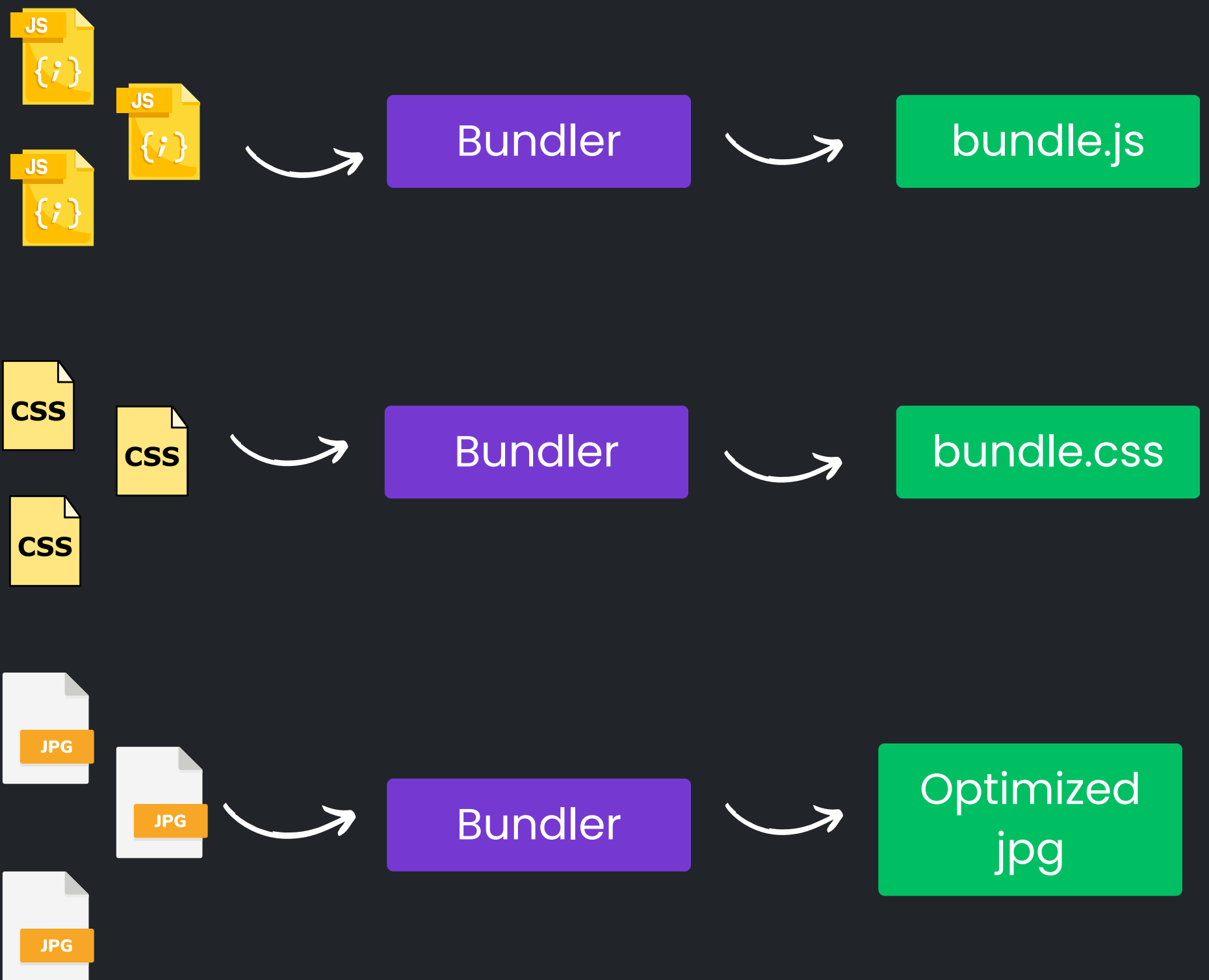


@coder_aishya



What is Webpack ?

Webpack is a module bundler for modern JavaScript applications.



Webpack analyzes connections & Bundles/
Optimizes everything



@coder_aishya



Core concepts of Webpack

Webpack is a module bundler for modern JavaScript applications.

- Entry
- Output
- Loaders
- Plugins
- Mode
- Browser Compatibility



@coder_aishya



When webpack processes your application, it internally builds a dependency graph from one or more entry points and then combines every module your project needs into one or more bundles, which are static assets to serve your content from.

Entry

An entry point indicates which module webpack should use to begin building out its internal dependency graph.

```
● ● ● webpack.config.js  
  
module.exports {  
  entry: './app/index.js'  
}
```

By default its value is `./src/index.js`, but you can specify a different (or multiple) entry points by setting an entry property in the webpack configuration.



@coder_aishya



Output

this configuration parameter simply tells the Webpack where to output the bundle that it creates.

● ● ● [webpack.config.js](#)

```
//Import 'path' to resolve the file path
const path = require('path');

//Add this configuration after module.rules config
output: {
  path: path.resolve(__dirname, 'dist'),
  filename: 'index_bundle.js'
}
```

It defaults to ./dist/main.js for the main output file and to the ./dist folder for any other generated file.



@coder_aishya



Loaders

After specifying the entry point for the application, the next thing to tell our Webpack is what should it do with the files within our project. In other words, what kind of transformations that need to be applied to our files.

To do that, we have something called the 'loaders'. By default, Webpack looks at all the JSON and JS files to build the dependency graph as shown above.

Every loader has 2 attributes that need to be defined:

- use – The name of the loader.
- test – The regex to match the file path.



@coder_aishya



```
● ● ● webpack.config.js  
  
module.exports = {  
  entry: './app/index.js',  
  module: {  
    rules: [  
      { test: /\.css$/, use: 'css-loader' }  
    ]  
  }  
}
```

Plugins

plugins perform a wider range of tasks like bundle optimization, asset management and injection of environment variables.

In order to use a plugin, you need to `require()` it and add it to the plugins array.



@coder_aishya



● ● ● webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const webpack = require('webpack'); //to access built-in plugins

module.exports = {
  module: {
    rules: [{ test: /\.txt$/, use: 'raw-loader' }],
  },
  plugins: [new HtmlWebpackPlugin(
    { template: './src/index.html' }
  )],
};
```

In the example above, the `html-webpack-plugin` generates an HTML file for your application and automatically injects all your generated bundles into this file.



@coder_aishya



Mode

By setting the mode parameter to either development, production or none, you can enable webpack's built-in optimizations that correspond to each environment. The default value is production.

● ● ● [webpack.config.js](#)

```
module.exports = {  
  mode: 'production',  
};
```

Browser Compatibility

Webpack supports all browsers that are ES5-compliant (IE8 and below are not supported). Webpack needs Promise for `import()` and `require.ensure()`. If you want to support older browsers, you will need to load a polyfill before using these expressions.



@coder_aishya



Follow me for more



aishwarya-dhuri



coder_aishya