WEBPACK BABEL

- We will add the **Babel transpiler** to our Webpack setup.
- This will allow Webpack to transpile ES6 Javascript code back to ES5.
- Install the Babel **core package, webpack loader** and **env preset**.
- babel-preset-env replaces the earlier babel-preset-es2015

```
npm install babel-core babel-loader babel-preset-env
```

- This will update package.json

```
  "dependencies": {
    "babel-core": "^6.26.0",
    "babel-loader": "^7.1.2",
    "babel-preset-env": "^1.6.1",
    "webpack": "^3.8.1"
  }
```

- We need to create a Babel configuration file **.babelrc**.
- This tells Babel to use the ENV preset.
- We update webpack.config.js to use **Babel loader**.
- This code also tells Webpack to ignore files in the node_modules folder.

```
module.exports = {
  entry: path.join(paths.SOURCE,'code.js'),
  output: {
    filename: 'code.js',
    path: paths.DEST
  },
  module: {
    rules: [
      {
        test: /\.(js|jsx)$/,
        exclude: /node_modules/,
        use: [
          'babel-loader',
        ],
      },
```

```
    ],
  },
};
```

- Convert the hyphenate function to use ES6 arrow syntax.

```
let hyphenate = words => words.split(" ").join("-");
```

- Run Webpack

```
npm run build
```

- ES6 code in source/code.js be **transpiled** back to ES5 in dest/code.js.