

Introduction to Webpack

The webpack 4's

- Entry: Tells webpack where to start.
- Output: Where to put the finished code.
- Loaders: Transform files.
- Plugins: Transform the whole transformed code.

Entry

```
module.exports = {  
  entry: './path/to/my/entry/file.js'  
};
```

Output

```
const path = require('path');

module.exports = {
  entry: './path/to/my/entry/file.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'my-first-webpack.bundle.js'
  }
};
```

How loaders work

- Identify which file or files should be transformed by a certain Loader.
- Identify how the file or files should be transformed by a certain Loader (with the use property)

Loaders

```
const path = require('path');

const config = {
  entry: './path/to/my/entry/file.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'my-first-webpack.bundle.js'
  },
  module: {
    rules: [
      { test: /\.txt$/, use: 'raw-loader' }
    ]
  }
};

module.exports = config;
```

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

const config = {
  entry: './path/to/my/entry/file.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'my-first-webpack.bundle.js'
  },
  module: {
    rules: [
      { test: /\.txt$/, use: 'raw-loader' }
    ]
  },
  plugins: [
    new webpack.optimize.UglifyJsPlugin(),
    new HtmlWebpackPlugin({template: './src/index.html'})
  ]
};

module.exports = config;
```

Plugins

Stuff to remember

- Webpack treats every file (.css, .html, .scss, .jpg, etc.) as a module.
- Webpack only understands Javascript
- Webpack can have multiple entry points, but only one output


```
{  
  entry: {  
    app: './src/app.js',  
    search: './src/search.js'  
  },  
  output: {  
    filename: '[name].js',  
    path: __dirname + '/dist'  
  }  
}
```

```
// writes to disk: ./dist/app.js, ./dist/search.js
```

Multiple Entries. One Output

Practice Time

Goals:

- Create a blank Webpack project.
- Write ES6 Javascript.
- Get a bundle.js file with old Javascript code.
- Use a plugin to uglify and compress the code