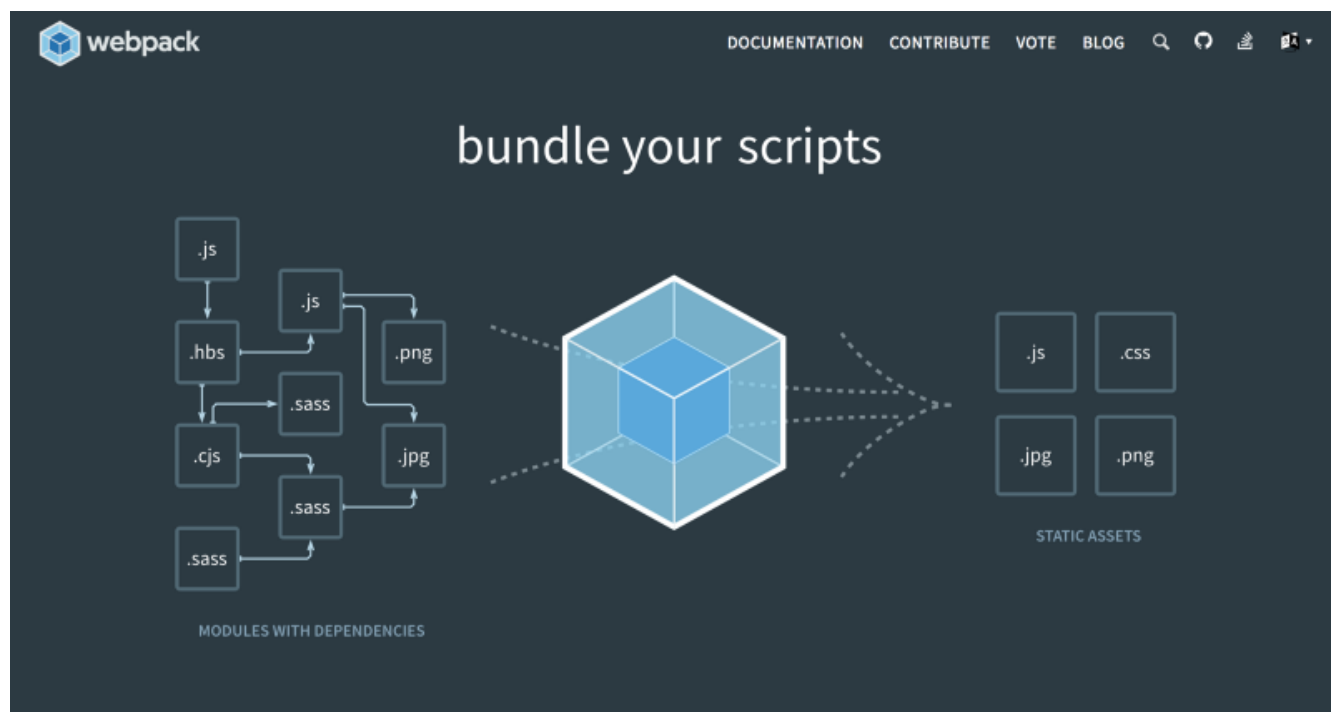


# Webpack

## prerequisite

src：專門放我們Preprocess的檔案，包括es6、pug、sass、vue、jsx等檔案，這個資料夾不會丟上去server部署

dist：經過webpack編譯打包後，產生出瀏覽器看得懂的html、css、js，要部署也是這個資料夾去部署



我們可以從官網首頁的這張圖看到整個webpack的核心概念，「注入」

## 什麼是「注入」？

webpack是一個以自己編寫設定檔然後透過指令去驅動的一個自動化工具，所有的動作都是透過自己寫的規則去做編譯，我們必須透過javascript的引入語法讓webpack知道需要它幫忙編譯什麼東西，例如pug或是sass等等.....

所以我們會有一支俗稱進入點的js專門注入那些Preprocess，讓那些Preprocess可以通過這個進入點的js讓webpack去編譯它，進而打包到dist資料夾。

## demo

```
> npm install webpack webpack-cli --save-dev
```

```
//index.js  
console.log("Hello")
```

```
// webpack.config.js
const path = require('path');
module.exports = {
  entry: './index.js',
  output: {
    filename: 'index.bundle.js',
    path: path.resolve(__dirname, './'),
  }
};
```

```
// package.json
//在package.json之中的script的部分加入 “build”: “webpack”
{
  "name": "webpack-test",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "build": "webpack"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "webpack": "^4.6.0",
    "webpack-cli": "^2.0.15"
  }
}
```

```
> npm run build
```

在index.html 中

```
<body>
  <div id="wrapper"></div>
  <script src="index.bundle.js"></script>
</body>
</html>
```

## Appendix

可以參考 entry使用object的方式來帶入多個js 在 output來輸出的時候用[name]來抓entry的object的key name

```
// webpack.config.json
entry: {
  index: './js/index.js',
  about: './js/about.js'
},
output: {
  path: path.resolve(__dirname, './dist'),
  filename: './js/[name].js',
},
```