웹팩은 무엇일까?

CRA에 숨겨진 비밀

2021.06.17 최진우

웹팩의 등장 배경
모듈화

- 자바스크립트 파일을 분리하자
  - 파일로 분리 시 문제 발생

index.html

```
<html>

<head>
  <title>Webpack demo</title>
  <script src="./a.js"></script>
  <script src="./b.js"></script>
</head>

<body>
</body>
<script>
  alert(number);
</script>

</html>
```

a.js

```
var num = 10;
```

b.js

```
var num = 10;
```

웹팩의 등장 배경
모듈화

- IIFE 방식을 통한 모듈화 성공
- 네임스페이스, 즉시 실행 함수를 통한 모듈화 방식
- CommonJS, AMD
  - NodeJS 진영에서 기본적으로 CommonJS를 사용하여 모듈화 (require)

```
var math = math || {} // math 네임스페이스

;(function () {
  function sum(a, b) {
    return a + b
  }
  math.sum = sum // 네이스페이스에 추가
})()
```

웹팩의 등장 배경

모듈화

- ES6의 **import, export**
- ES6를 지원하지 않는 브라우저

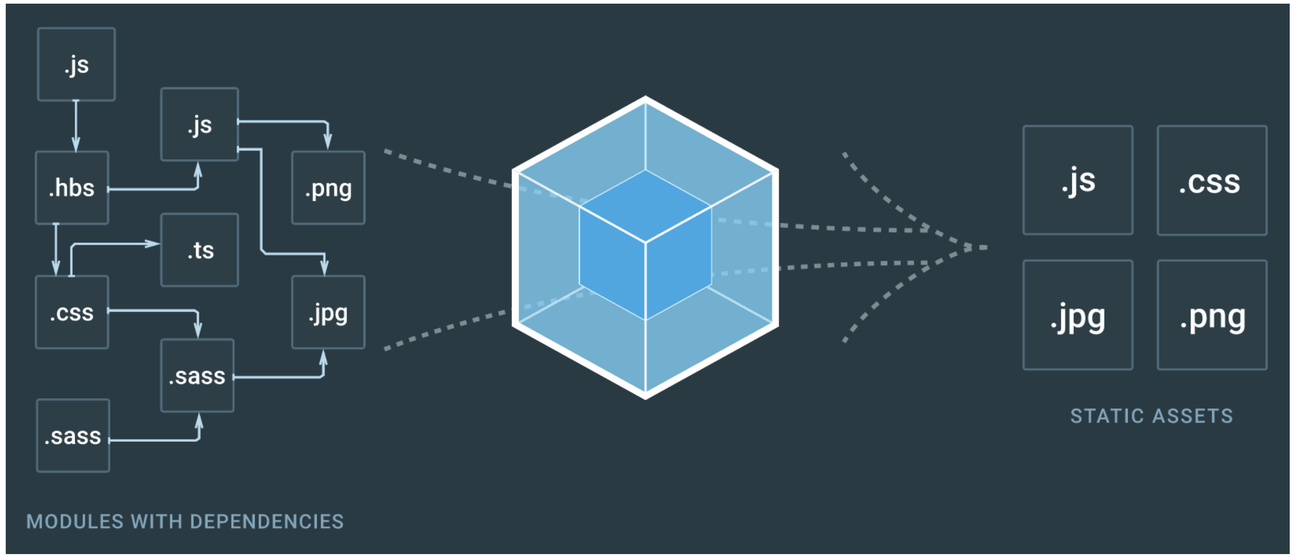
웹팩의 등장 배경

그 외

- HTTP 요청 숫자의 제약 (번들링의 필요성)
- 동적/지연 로딩

웹팩이란?

- 모던 자바스크립트 애플리케이션을 위한 정적 번들러
- 디팬던시 그래프를 생성



웹팩에서의 모듈

- JS 모듈
- HTML
- CSS
- Font
- PNG/JPG/GIF…

웹팩의 주요 컨셉

- 엔트리
  - 디펜던시 그래프가 시작할 부분
- 아웃풋
  - 결과물의 파일 경로
- 로더
  - 웹팩은 JS만 번들링
- 플러그인
  - 번들링된 데이터의 조작

그렇다면 babel은?

- ES6 이전의 문법으로 변경
- JSX를 컴파일
- .babelrc
- @babel/preset-env ES6를 이전 버전으로
- @babel/preset-react JSX

CRA
를 쓰면 안되는 이유?

- eject

React 설정
을 직접 해보자

- 예시