

- Webpack, Motivation, Node.js & NPM

조합해서 하나의 결과물을 만든다.

• webpack?

(Module Bundler)

웹 어플리케이션을 구성하는 자원

- HTML
- CSS
- JS
- IMAGE

각각을 모듈로 뭉쳐

• Module? 프로그래밍 환경에서 특정 기능을 갖는 작은 코드 단위 의미.

↳ 사실 제때로 공부는 하지 않았지만 실제로 쓰고 있었네?

• Module Bundling?

↳ • Web application을 구성하는 자원들을 하나의 파일로 병합/압축.

1. 파일 단위의 JS module 관리의 필요성 → 각각은 범위가 정해져서.

2. Web Task Manager

↳ 전체 파일의 실행을 계획해야됨...

3. Load speed / performance

- HTML, CSS, JS 압축
- IMG 압축
- CSS 전처리기 실행

↳ 브라우저 → 서버 요청 파일 숫자 줄이기. 2번에서 파일 압축/병합

+ lazy loading. / 필요한 자원 그때 로드 (import) 하라.

해결하려는 문제

· JS 변수 용량 범위 --- ①

· 브라우저별 HTTP 요청 수의 제약 --- ② code splitting

· 사용하지 않는 코드의 관리 --- ③

· Dynamic Loading & Lazy Loading 지원 --- ④

① ES6 Modules 문법 / 웹팩의 모듈 번들링

no code executes until requested modules are available and processed. export/import, export default, export *

Dynamic loading - `System.import("_____").then(() => { })`

② 브라우저별 HTTP 요청 숫자의 제약

한 번에 시발로 보낼 수 있는 HTTP 요청 숫자는 제약되어 있음.
→ HTTP 요청 숫자를 줄이는 것이 중요.

Node.js / NPM

- Node.js

- 브라우저 밖에서도 JS 실행할 수 있는 환경.

- NPM

- Node Package Manager
- JS 라이브러리 설치 / 관리할 수 있는 패키지 매니저.

- NPM install

npm 명령어로 package.json 파일 생성.

local

npm install _____ --save --prod

npm i _____

⇒ node_modules 에
설치. (dependencies)
package.json

npm 전역 설치

(npm install _____ --global

npm i _____ -g

→ 전역 JS.
usr/local/lib/node_modules

이제 보실

개발용 / 배포용 라이브러리 구분

(npm i _____

npm i _____ -D

개발용

package.json

dependencies

- NPM 키스텝 명령어

package.json

```
{  
  "scripts": {  
    "hello": "echo hi"  
  }  
}
```

안에 명령어들 정의.

npm run hello

expo로 만든 프로젝트에서

```
express  
demon  
;
```

이 때만 긴 명령어보다 짧게 쓰기 좋거나 포함해서 쓰라.