Package Manager

01 패키지 매니저

02 NPM

03 YARN

04 PNPM

05 패키지 매니저 선택

01 패키지 매니저?

page.js

01 패키지 매니저?

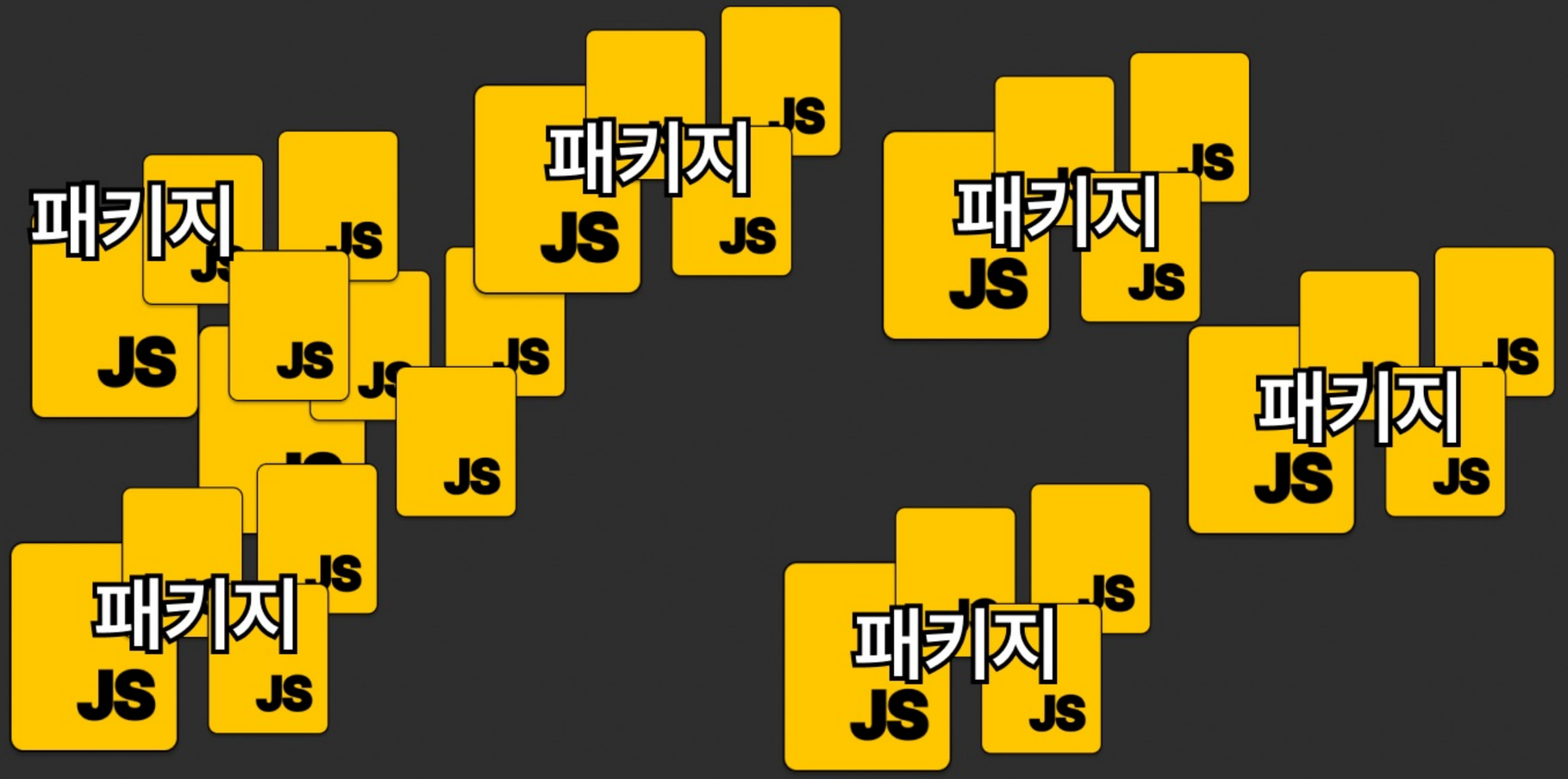
패키지

모듈 덩어리

01 패키지 매니저?

package.json 을 가진
모듈 혹은 모듈의 뭉치를 패키지라고 함

01 패키지 매니저?




THE SWASHBUCKLING MUSKETEERS
ALL FOR ONE, ONE FOR ALL! - WASN'T THAT THE
MOTTO OF THE THREE MUSKETEERS? NOW LOOK
AT THIS FREE-FOR-ALL! CAN YOU SPOT OUR THREE
GALLANT HEROES BATTLING THE RED-COATED
CARDINAL'S GUARDS! WITH THE RED-COATED
ACTION GOING ON, I WONDER HOW THE CAMERAMEN
CAN CAPTURE IT ALL ON FILM!

01 패키지 매니저?

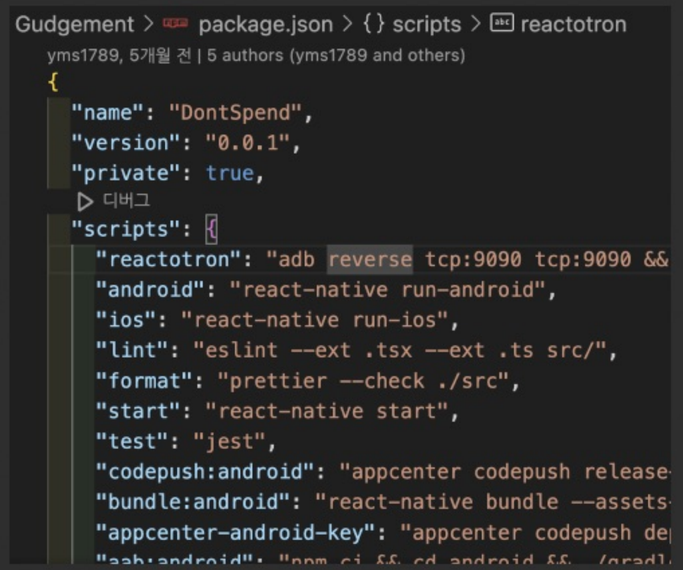
02 NPM

02 NPM

Advantages

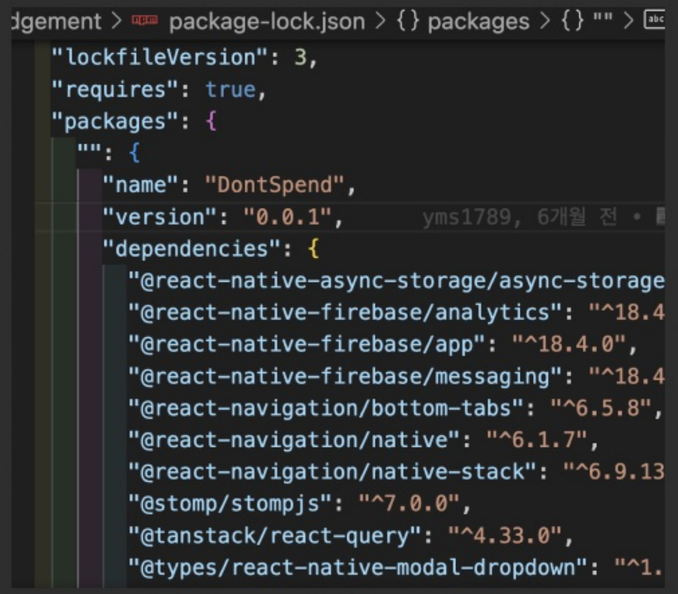
누구나 온라인 배포 가능
설치 방식의 통일
여러 버전의 패키지를 한 프로젝트에서 사용

02 NPM



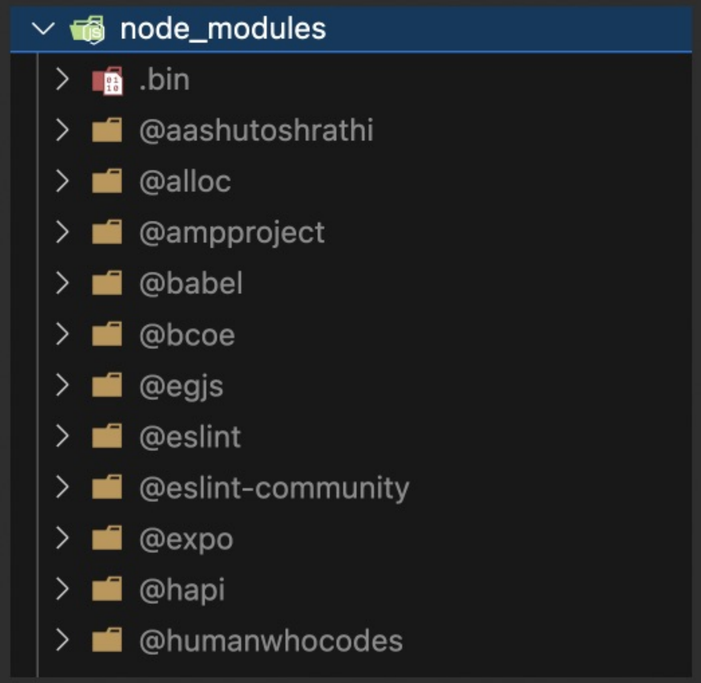
npm의 패키지는 package.json에 정의됨
npm init 명령어를 통하면 최초에 이 파일이 만들어지고
npm 패키지를 설치하게 되면 이 package.json에 명시

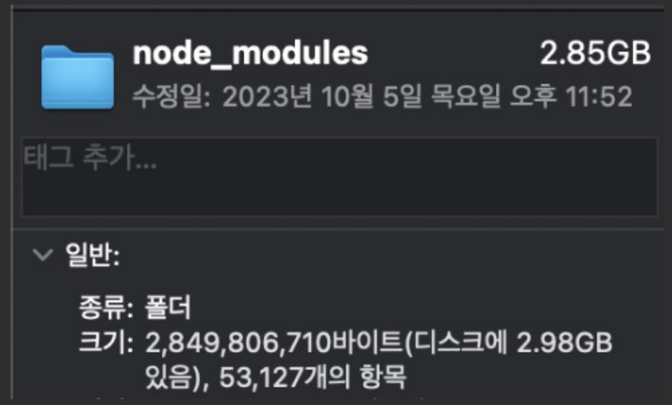
02 NPM



package-lock.json은 최초 npm init을 실행 시 생성되지는 않는다.
package-lock.json은 npm 패키지를 설치하거나 수정, 삭제 등의 작업을 진행할 때 생성
핵심 역할은 각 패키지에 대한 의존성 관리

02 NPM


node_modules
.bin
@aashutoshrathi
@alloc
@ampproject
@babel
@bcoe
@egjs
@eslint
@eslint-community
@expo
@hapi
@humanwhocodes


node_modules
2.85GB
수정일: 2023년 10월 5일 목요일 오후 11:52
태그 추가...
일반:
종류: 폴더
크기: 2,849,806,710바이트(디스크에 2.98GB 있음), 53,127개의 항목

02 NPM

Disadvantage

03 YARN

Yet Another Resource Negotiator

03 YARN

Advantages

의존성 트리 알고리즘 변경
자동화된 lock 생성
캐시 사용

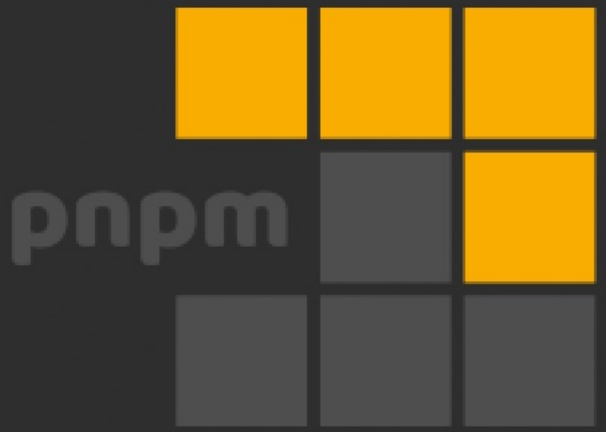
npm은 다운 순서에 따라 의존성이 변할 수 있었음

03 YARN

Disadvantage

yarn berry 등장으로 인한 쇠퇴
2020년 이후 유지보수 단계로 전락

04 PNPM

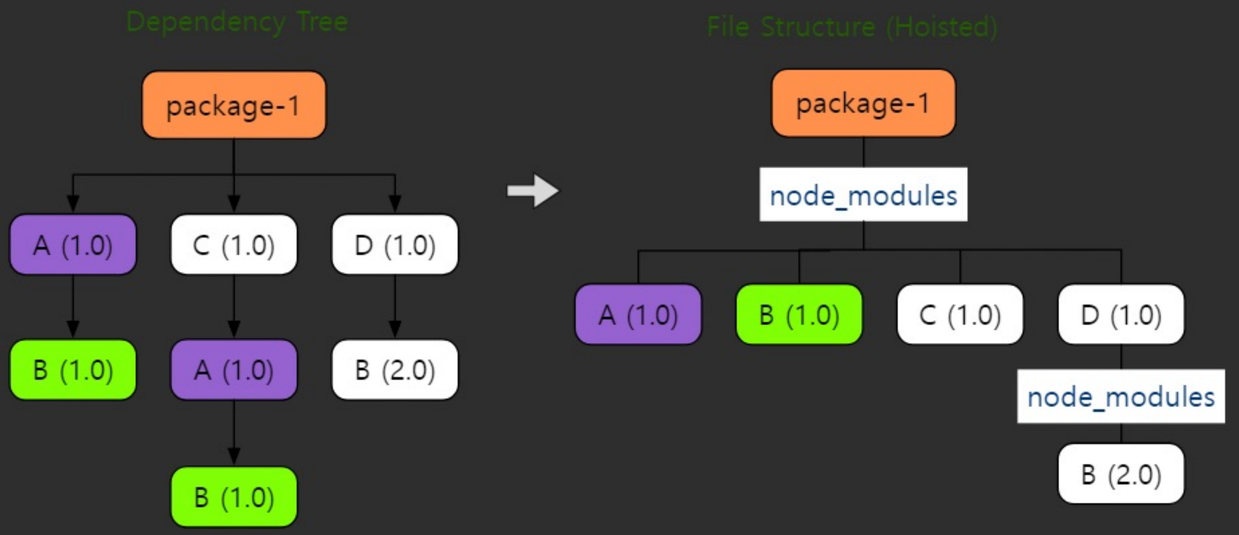


04 PNPM

프로젝트가 100개가 있는 경우,
그만큼의 종속성의 사본들이 디스크에 저장됨
= 노드 모듈스가 100개 생성됨 ㄷㄷ

즉,디스크 공간을 비효율적으로 사용하게 되는 문제가 존재

04 PNPM



npm 또는 Yarn 클래식을 통해 의존성을 설치할 때,
모든 패키지는 모듈 디렉토리의 루트로 호이스트됨

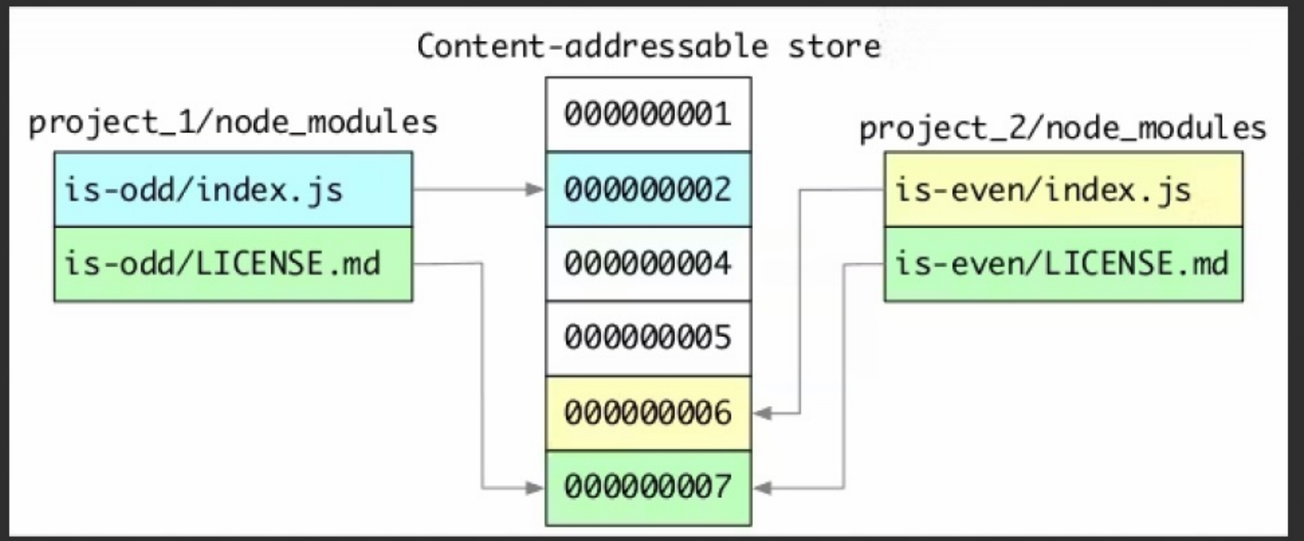
결과적으로, 소스 코드는 프로젝트에 의존성으로
추가되지 않은 의존성에 접근할 수 있음

04 PNPM

Symbolic Link .pnpm store

패키지를 루트 디렉토리의 .pnpm store에 저장하고
node_modules엔 심볼릭 링크만 남김

04 PNPM



패키지를 루트 디렉토리의 .pnpm store에 저장하고
node_modules엔 심볼릭 링크만 남김

05 패키지 매니저 선택

정답은 없다!!