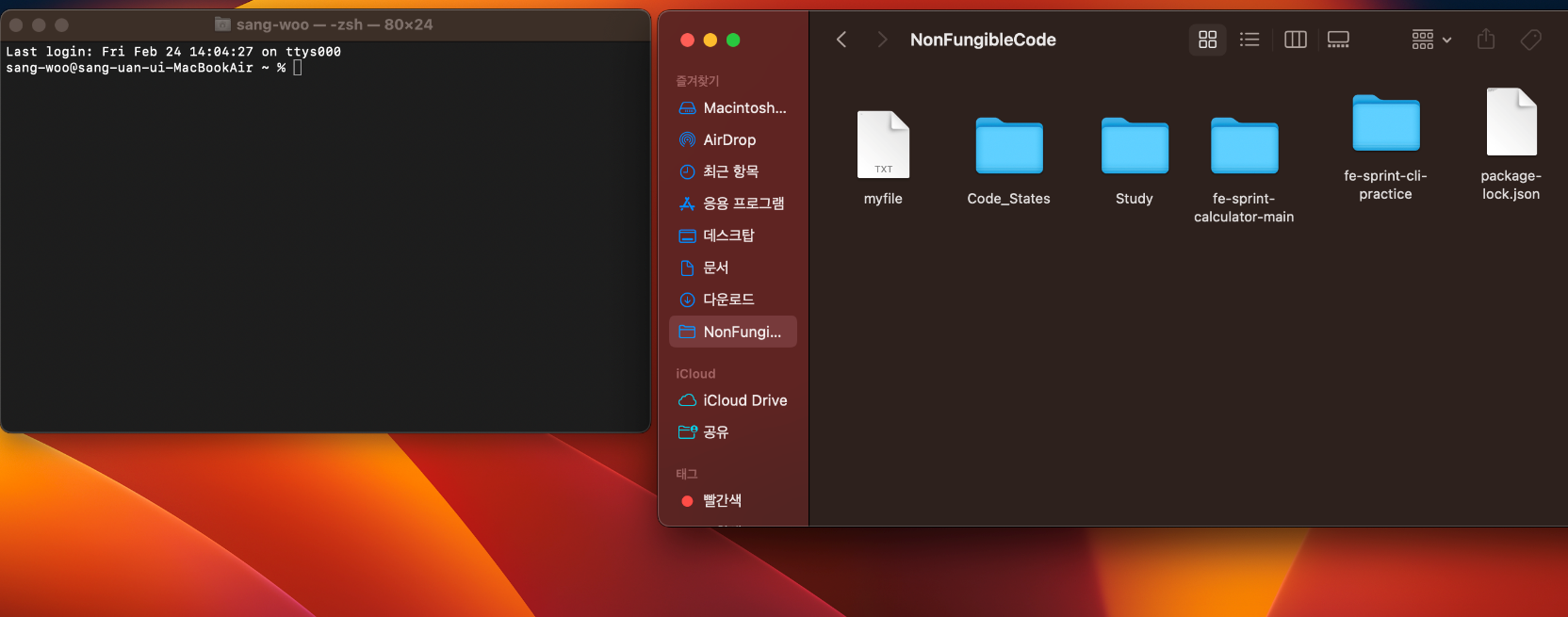
[2023.02.24.] -Linux/Git

기초(CLI 명령어 및 패키지매니저)

<맥북의 터미널과 Finder, CLI와 GUI의 차이를 한 눈에 보여준다>

CLI(Command Line Interface)는 터미널을 통한 텍스트 명령어로 컴퓨터와 상호 작용하는 방식을 가리키고, GUI(Graphical User Interface)는 사용자가 편리하게 사용할 수 있도록 입력이나 출력과 같은 기능을 아이콘 등을 활용한 시각적 요소를 사용해 컴퓨터의 기능을 간편하게 이용할 수 있도록 하는 방식이다.

컴퓨터를 조작하기 위해서는 반드시 입력(input)이 필요하다. 키보드와 마우스처럼 컴퓨터에 입력을 할 수 있도록 만들어주는 도구를 입력 소스(input source)라고 한다. 일반적인 사용자가 PC를 사용함에 있어서는 이러한 입력 소스를 사용하는 GUI가 CLI에 비해 압도적으로 편리한 것이 사실이지만, 개발자의 경우 원격으로 서버 컴퓨터에 접속하거나 소통해야 하기 때문에 이런 경우에는 CLI가 GUI 보다 훨씬 더 효율적이라고 할 수 있다.



오늘은 homebrew, wget, neofetch, nvm, node.js 등을 설치하고 Mac OS에서 리눅스 터미널 기반의 개발 환경을 구축하는 법과, 이러한 터미널을 이용한 간단한 CLI 명령어를 사용하는 방법 등을 배웠다. 설치 과정이 그렇게 복잡하지 않은 프로그램도 있었지만 예상치 못 한 에러가 발생하는 경우도 있었기에, 이럴 때는 구글링을 통해 비슷한 문제 상황과 해결 방법을 찾아보고 적용하느라 바빴다.



오늘 배운 중요한 개념들 중 하나가 바로 패키지와 패지키 매니저이다. **리눅스**는**apt,** **Mac OS**는**brew**, **node.js**는 **npm**이라는 패키지 매니저를 가지고 있다. 각각의 os나 프로그램이**패키지**이고 필요한 모듈, 소프트웨어를 다운로드할 수 있는 일종의 앱스토어와 같은 것이 바로 **패키지 매니저**이다. node.js 환경에서 외부 라이브러리를 다운로드하는 많은 방법 중에서 대표적인 것이 바로 npm이다. node.js에서 프로젝트에 필요한 모듈의 정보를 모아놓은 곳이 바로 package.json이라는 파일이다. 이곳에 프로젝트의 정보 및 CLI 상에서 사용 가능한 명령, 개발과 관련된 dependeny 등이 있는데, **이러한 수많은 모듈을 직접 전달하지 않고 필요에 따라 다운로드 해서 사용할 수 있도록 해주는 것이 바로 npm인 것이다.**