

Flox는 특정 프로그래밍 언어에 구애받지 않는 패키지 및 환경 관리 도구로, macOS, Linux(x86, Arm) 등 다양한 플랫폼에서 재현 가능하고 일관된 개발 환경을 구축할 수 있게 돕습니다. 단일 매니페스트 파일을 사용하여 패키지, 도구, 환경 변수, 서비스 등 프로젝트의 모든 의존성을 통합하여 정의합니다.

핵심 개념: 환경 (Environment)

Flox의 가장 중요한 개념은 바로 **환경**입니다. 환경은 프로젝트 구동에 필요한 패키지와 특정 설정을 설치할 수 있는 격리된 공간을 의미합니다. 이 격리 방식은 무거운 컨테이너가 아니라 미리 구성된 하위 셸(Sub-shell)을 활용하기 때문에, 기존에 사용하던 도구나 셸 환경과 매우 매끄럽게 연동됩니다. 환경을 활성화하면 Flox는 해당 환경의 전용 패키지들을 우선적으로 바라보도록 **PATH** 설정을 동적으로 변경합니다.

Flox와 Nix Flakes

Nix Flake는 **flake.nix** 파일을 포함하는 독립적인 소프트웨어 패키징 단위입니다. 이 파일에는 의존성 구성과 소프트웨어 빌드 및 실행 방법이 정의되어 있습니다. 특히 **flake.lock** 파일을 통해 의존성 버전을 특정 시점에 고정함으로써 환경의 재현성을 극대화합니다. Flox는 이러한 Nix의 강력한 기능을 기반으로 구축되었지만, Nix 전문가가 아닐더라도 쉽게 접근할 수 있도록 설계되었습니다. 사용자는 **manifest.toml** 파일에서 환경을 정의하고, 내부적으로는 Nix Flake의 강력한 패키지 관리 기능을 활용하게 됩니다.

매니페스트를 통한 설치 및 관리

manifest.toml 파일을 편집하여 Flox 환경에 설치할 패키지(Flake)를 지정할 수 있습니다. **flox edit** 명령을 실행하여 현재 환경의 매니페스트를 바로 편집할 수 있습니다.

```
[install]
cowsay = "github:flox-examples/cowsay"
편집 후 환경을 활성화하면 설정이 적용됩니다.
```

flox activate

Flox가 자동으로 해당 패키지를 내려받고 빌드하여 현재 환경에 즉시 추가합니다.

Flox 프로필 (Profiles)

Flox 프로필 기능을 이용하면 **manifest.toml** 파일 내에서 셸 별칭(Alias), 함수, 환경 변수 등을 선언적으로 정의할 수 있습니다. 이러한 설정은 Flox 환경이 활성화될 때 자동으로 로드되므로, 프로젝트마다 최적화된 셸 환경을 구성할 수 있습니다.

```
[profile]
alias l="ls -alh"
```

기본 워크플로우

환경 초기화

프로젝트 디렉토리를 생성하고 해당 위치에서 **flox init** 명령을 실행하여 새로운 Flox 환경 설정을 시작합니다.

```
mkdir my-project
cd my-project
flox init
```

패키지 검색

flox search 명령을 사용하여 필요한 패키지를 찾습니다.

```
flox search cowsay
```

패키지 설치

flox install 명령으로 환경에 패키지를 추가합니다.

```
flox install cowsay
```

환경 활성화

설정된 환경을 활성화하려면 아래 명령을 사용합니다.

```
flox activate
```

설치된 패키지 확인

flox list를 사용하면 현재 활성화된 환경에 설치된 모든 패키지 목록을 확인할 수 있습니다.

```
flox list
```

환경 종료 (비활성화)

활성화된 환경을 종료하고 원래의 시스템 셸로 돌아가려면 **exit**를 입력하거나 **Ctrl+D**를 누릅니다.

```
exit
```

선언적 환경 관리

Flox는 **.flox** 디렉토리 내부의 **manifest.toml** 파일을 통해 모든 환경 설정을 관리합니다. **flox edit** 명령으로 이 파일을 열어 패키지 목록이나 셸 설정을 언제든지 수정할 수 있습니다.

환경 공유 및 협업

FloxHub를 이용하면 구축한 개발 환경을 다른 팀원과 손쉽게 공유할 수 있습니다. **flox push** 명령으로 환경 설정을 FloxHub에 업로드하고, 다른 사용자는 **flox pull**을 통해 동일한 구성을 그대로 복제하여 사용할 수 있습니다.

추가 학습 자원

- 공식 문서: <https://flox.dev/docs/>
- GitHub 저장소: <https://github.com/flox/flox>