

## 기본 개념

**mise** (이전의 **rtx**, **asdf-vm**의 영향을 받음)는 여러 프로그래밍 언어와 도구의 버전을 프로젝트별로 관리할 수 있게 해주는 CLI 도구입니다. **.tool-versions** 파일을 사용하여 특정 프로젝트에 필요한 도구와 버전을 선언적으로 관리합니다.

## 기본 명령어

- **mise install** 또는 **mise i: .tool-versions** 파일에 명시된 모든 도구의 버전을 설치합니다.
- **mise use <tool>@<version>**: 특정 도구의 버전을 전역 또는 로컬 **.tool-versions** 파일에 추가하고 활성화합니다.
- **mise current**: 현재 디렉토리에서 활성화된 모든 도구와 그 버전을 보여줍니다.
- **mise ls**: 설치된 모든 도구와 그 버전 목록을 보여줍니다.
- **mise exec -- <command>**: **.tool-versions**에 명시된 버전의 도구 환경에서 특정 명령어를 실행합니다. (예: **mise exec -- npm test**)
- **mise reshim**: shim을 재구성합니다. (플러그인 설치/제거 후 필요할 수 있음)

## 플러그인 관리

- **mise plugins ls**: 설치된 모든 플러그인 목록을 보여줍니다.
- **mise plugins add <name>**: 새 플러그인을 추가합니다. (예: **mise plugins add nodejs**)
- **mise plugins add <name> <git-url>**: Git URL로 플러그인을 추가합니다.
- **mise plugins rm <name>**: 플러그인을 제거합니다.
- **mise plugins update <name>**: 특정 플러그인을 업데이트합니다. (모두 업데이트하려면 **mise plugins update --all**)

## 버전 관리

- **mise ls-remote <tool>**: 설치 가능한 모든 버전의 목록을 보여줍니다.
- **mise install <tool>@<version>**: 특정 버전의 도구를 설치합니다.
- **mise uninstall <tool>@<version>**: 특정 버전의 도구를 제거합니다.
- **mise global <tool>@<version>**: 전역 (~/.tool-versions) 버전을 설정합니다.

- **mise local <tool>@<version>**: 로컬(현재 디렉토리) 버전을 설정합니다. **.tool-versions** 파일이 생성/수정됩니다.

## .tool-versions 파일 예시

이 파일을 프로젝트의 루트 디렉토리에 위치시키면, 해당 디렉토리에 들어갈 때 **mise**가 자동으로 명시된 버전들을 활성화합니다.

```
# .tool-versions
nodejs 20.11.0
python 3.11.7
rust 1.76.0
```

## 셸 통합

**mise**가 제대로 작동하려면 셸 설정 파일 (**.bashrc**, **.zshrc** 등)에 **mise**를 활성화하는 스크립트를 추가해야 합니다.

```
# .zshrc 예시
eval "$(mise activate zsh)"
```

이 설정을 통해 디렉토리를 변경할 때마다 **mise**가 **.tool-versions** 파일을 자동으로 감지하고 해당 버전의 shim을 **PATH**에 추가해줍니다.

## asdf-vm과의 관계

**mise**는 **asdf-vm**과 매우 유사한 목표와 기능을 가지고 있으며, **asdf-vm**의 플러그인 생태계와 호환됩니다. 하지만 **mise**는 Rust로 작성되어 **asdf-vm**보다 훨씬 빠른 성능을 제공하며, 추가적인 기능(예: 환경 변수 관리, 작업 실행 등)을 통해 사용자 경험을 개선했습니다.