

파일 무시하기: Git의 .gitignore 사용법

Git을 사용할 때 자동으로 추가되거나 추적되지 않기를 원하는 파일들이 있습니다. 주로 로그 파일이나 빌드 시스템에서 생성된 파일들이 이에 해당합니다. 이런 경우 `.gitignore` 라는 파일을 만들어 해당 패턴과 일치하는 파일들을 무시하도록 설정할 수 있습니다.

기본 .gitignore 파일 예시

다음은 간단한 `.gitignore` 파일의 예시입니다:

```
*.[oa]
*~
```

첫 번째 줄은 ".o" 또는 ".a"로 끝나는 파일들을 무시하라는 의미입니다. 이는 코드 빌드 과정에서 생성되는 오브젝트 파일과 아카이브 파일입니다. 두 번째 줄은 틸드(~)로 끝나는 모든 파일을 무시하라는 의미로, Emacs와 같은 텍스트 에디터가 임시 파일을 표시하는 데 사용합니다.

새 저장소를 시작할 때 미리 `.gitignore` 파일을 설정해두면 실수로 원치 않는 파일을 커밋하는 것을 방지할 수 있습니다.

.gitignore 패턴 규칙

`.gitignore` 파일에 사용할 수 있는 패턴 규칙은 다음과 같습니다:

- 빈 줄이나 `#` 으로 시작하는 줄은 무시됩니다.
- 표준 글로브 패턴이 작동하며, 전체 작업 트리에 재귀적으로 적용됩니다.
- 패턴 앞에 슬래시(/)를 붙이면 재귀적 적용을 피할 수 있습니다.
- 패턴 뒤에 슬래시(/)를 붙이면 디렉토리를 지정합니다.
- 패턴 앞에 느낌표(!)를 붙이면 해당 패턴을 무시하지 않습니다(예외 지정).

글로브 패턴 이해하기

글로브 패턴은 셸에서 사용하는 단순화된 정규표현식과 유사합니다:

- 별표(*): 0개 이상의 문자와 일치
- 대괄호([abc]): 대괄호 안의 문자 중 하나와 일치(이 경우 a, b, c)
- 물음표(?): 단일 문자와 일치

- 하이픈으로 구분된 문자가 있는 대괄호(`[0-9]`): 범위 내 문자와 일치(이 경우 0부터 9까지)
- 이중 별표(`**`): 중첩 디렉토리와 일치(예: `a/**/z` 는 `a/z` , `a/b/z` , `a/b/c/z` 등과 일치)

고급 .gitignore 예시

다음은 더 복잡한 .gitignore 파일의 예시입니다:

```
# 모든 .a 파일 무시
*.a

# 위에서 .a 파일을 무시하도록 했지만, lib.a는 추적함
!lib.a

# 현재 디렉토리의 TODO 파일만 무시하고 하위 디렉토리의 TODO는 무시하지 않음
/TODO

# build라는 이름의 모든 디렉토리 내 파일 무시
build/

# doc/notes.txt는 무시하지만 doc/server/arch.txt는 무시하지 않음
doc/*.txt

# doc/ 디렉토리와 그 하위 디렉토리에 있는 모든 .pdf 파일 무시
doc/**/*.*pdf
```

추가 정보

GitHub는 수십 개의 프로젝트와 언어에 대한 좋은 .gitignore 파일 예시 목록을 <https://github.com/github/gitignore> 에서 제공하고 있어 프로젝트 시작 시 참고할 수 있습니다.

간단한 경우에는 저장소의 루트 디렉토리에 단일 .gitignore 파일을 두고 이를 전체 저장소에 재귀적으로 적용할 수 있습니다. 그러나 하위 디렉토리에 추가 .gitignore 파일을 두는 것도 가능합니다. 이러한 중첩된 .gitignore 파일의 규칙은 해당 파일이 위치한 디렉토리 아래의 파일에만 적용됩니다. 예를 들어, 리눅스 커널 소스 저장소에는 206개의 .gitignore 파일이 있습니다.

여러 .gitignore 파일 사용에 대한 자세한 내용은 `man gitignore` 명령어로 확인할 수 있습니다.