

# VS Code와 Git 연동 가이드

## 1. VS Code 설치 및 설정

### VS Code 설치

1. [VS Code 공식 웹사이트](#) 접속
2. 운영체제에 맞는 버전 다운로드
3. 설치 프로그램 실행
4. 설치 옵션 선택:
  - "Add to PATH" 옵션 체크
  - "Register Code as an editor for supported file types" 체크
  - "Add to context menu" 체크

## VS Code 기본 설정

### 1. 필수 확장 프로그램 설치

- GitLens
- Git Graph
- Git History
- Git Blame
- Git Extension Pack

### 2. Git 연동 설정

- VS Code 실행
- 설정 메뉴 열기 (Ctrl+,)
- Git 관련 설정 확인:
  - `git.path` : Git 실행 파일 경로
  - `git.enabled` : Git 기능 활성화
  - `git.autofetch` : 자동 패치 활성화

## 테마 및 기본 설정

- File > Preferences > Settings
- 원하는 테마, 폰트 크기, 탭 크기 등 설정
- Git 관련 설정:

```
{  
  "git.enableSmartCommit": true,  
  "git.confirmSync": false,  
  "git.autofetch": true  
}
```

## 2. Git과 VS Code 연동

### 1. Git 저장소 초기화

1. VS Code에서 새 폴더 열기
2. 터미널 열기 (Ctrl+`)
3. Git 초기화:

```
git init
```

## 2. 소스 제어 기능 사용

1. VS Code 좌측 사이드바에서 소스 제어 아이콘 클릭
2. 변경사항 확인 및 스테이징
3. 커밋 메시지 입력 및 커밋
4. 원격 저장소 연동 및 푸시

### 3. Git 명령어 팔레트 사용

1. Ctrl+Shift+P 입력
2. Git 관련 명령어 검색 및 실행:
  - Git: Clone
  - Git: Pull
  - Git: Push
  - Git: Commit
  - Git: Checkout

## 3. VS Code Git 기능 활용

### 1. 변경사항 추적

- 소스 제어 뷰에서 변경된 파일 확인
- 변경사항 비교 (Diff) 뷰어 사용
- 변경사항 스테이징 및 언스테이징



## 2. 브랜치 관리

- 브랜치 생성 및 전환
- 브랜치 병합
- 충돌 해결

### 3. 히스토리 확인

- Git Graph 확장 프로그램 사용
- 커밋 히스토리 시각화
- 특정 커밋으로 이동

## 4. 문제 해결

### 1. Git 명령어 실행 오류

- Git 설치 경로 확인
- VS Code 재시작
- Git Bash에서 명령어 실행

## 2. VS Code Git 연동 문제

- Git 확장 프로그램 재설치
- VS Code 재시작
- Git 설정 확인

### 3. 일반적인 문제

- Git 자격 증명 관리자 확인
- 네트워크 연결 확인
- 저장소 권한 확인

## 5. VS Code Git 단축키

### 기본 단축키

- `Ctrl+Shift+G` : 소스 제어 뷰 열기
- `Ctrl+Enter` : 커밋
- `Ctrl+Shift+P` : 명령어 팔레트 열기

## Git 명령어

- Git: Clone
- Git: Pull
- Git: Push
- Git: Commit
- Git: Checkout

## 6. 추천 확장 프로그램

### Git 관련

- GitLens
- Git Graph
- Git History
- Git Blame
- Git Extension Pack



## 개발 도구

- ESLint
- Prettier
- Debugger for Chrome
- Python
- C/C++

## 7. 참고 자료

- VS Code 공식 문서
- Git 기본 가이드
- VS Code Git 확장 프로그램