

# VS CODE 빠른 참조

내비게이션, 편집, 멀티커서, 터미널, 디버깅, 확장

## 내비게이션

파일 및 실행 내비게이션	
<b>Ctrl+P</b>	파일 이름으로 빠른 열기
<b>Ctrl+Shift+P</b>	명령 팔레트
<b>Ctrl+Shift+O</b>	파일 내 실행물 이동
<b>Ctrl+T</b>	워크스페이스 내 실행물 이동
<b>Ctrl+G</b>	선택된 줄로 이동
<b>F12</b>	다음 줄로 이동
<b>Alt+F12</b>	전행의 미리보기
<b>Shift+F12</b>	뒤로 참조 찾기
<b>Ctrl+Shift+\</b>	대응하는 괄호로 이동

## 탭 및 에디터 내비게이션

<b>Ctrl+Tab</b>	열린 탭 순환
<b>Ctrl+1 / 2 / 3</b>	에디터 그룹 1/2/3에 포커스
<b>Ctrl+\</b>	에디터 그룹 열기
<b>Ctrl+B</b>	사이드바 전환
<b>Ctrl+J</b>	하단 패널 전환

## 편집

줄 작업	
<b>Ctrl+Shift+K</b>	줄 전체 삭제
<b>Alt+Up / Down</b>	위 / 아래로 이동
<b>Shift+Alt+Up / Down</b>	위 / 아래로 복사
<b>Ctrl+Enter</b>	아래에 줄 삽입
<b>Ctrl+Shift+Enter</b>	위에 줄 삽입
<b>Ctrl+Shift+[</b>	왼쪽 줄 들여쓰기
<b>Ctrl+Shift+]</b>	오른쪽 줄 들여쓰기

## 텍스트 조작

<b>Ctrl+D</b>	다음 선택 (반복으로 다음 일치 추가)
<b>Ctrl+Shift+L</b>	모든 일치 선택
<b>Ctrl+L</b>	현재 선택
<b>Ctrl+ /</b>	줄식 전환
<b>Shift+Alt+A</b>	문맥 줄식 전환
<b>Tab / Shift+Tab</b>	여스기 / 내어쓰기

## 멀티커서

커서 추가	
<b>Alt+Click</b>	클릭 위치에 커서 추가
<b>Ctrl+Alt+Up / Down</b>	위 / 아래로 커서 추가
<b>Ctrl+D</b>	선택의 다음 발생 추가
<b>Ctrl+Shift+L</b>	모든 발생에 커서 추가
<b>Ctrl+U</b>	마지막 커서 추가 취소
<b>Esc</b>	멀티커서 모드 종료

## 멀티커서 워크플로

```
// Select a word, then Ctrl+D to add matches
// Type replacement - all cursors update
// Ctrl+Shift+L selects ALL matches at once
// Alt+Click for arbitrary cursor placement
```

## 터미널

터미널 단축키	
<b>Ctrl+`</b>	통합 터미널 전환
<b>Ctrl+Shift+`</b>	새 터미널 생성
<b>Ctrl+Shift+5</b>	터미널 분할
<b>Ctrl+PgUp / PgDn</b>	터미널 위 / 아래 스크롤
<b>Ctrl+Shift+C</b>	터미널에서 선택 복사
<b>Ctrl+Shift+V</b>	터미널에 붙여넣기

## 터미널 관리

<b>Ctrl+Shift+P → Terminal: Kill</b>	활성 터미널 종료
<b>Ctrl+Shift+P → Terminal: Rename</b>	터미널 탭 이름 변경
<b>Terminal dropdown</b>	터미널 전환
<b>terminal.integrated.shell.*</b>	기본 셸 변경 설정

## 디버깅

디버그 컨트롤	
<b>F5</b>	디버깅 시작 / 계속
<b>Shift+F5</b>	디버깅 중지
<b>Ctrl+Shift+F5</b>	디버깅 재시작
<b>F9</b>	중단점 전환
<b>F10</b>	스텝 오버
<b>F11</b>	스텝 인
<b>Shift+F11</b>	스텝 아웃

## launch.json 예제

```
{
  "type": "node",
  "request": "launch",
  "name": "Run App",
  "program": "${WorkspaceFolder}/app.js"
}
```

## 디버깅 기능

<b>Watch panel</b>	디버깅 중 표현식 모니터링
<b>Call Stack</b>	실행 스택 보기
<b>Debug Console</b>	중단점에서 표현식 평가
<b>Conditional breakpoint</b>	중단점 오른쪽 클릭 → 조건 편집
<b>Logpoint</b>	실행 중 단 없이 메시지 로그

## 확장

확장 관리	
<b>Ctrl+Shift+X</b>	확장 보기 열기
<b>@installed</b>	설치된 확장으로 필터
<b>@enabled / @disabled</b>	활성화 / 비활성화로 필터
<b>@recommended</b>	추천 확장 표시

## 필수 확장

<b>Prettier</b>	코드 포매터 (JS, TS, CSS, HTML)
-----------------	----------------------------

<b>ESLint</b>	JavaScript/TypeScript 린팅
<b>GitLens</b>	항상된 Git 통합 및 플러그인
<b>Remote - SSH</b>	SSH를 통해 원격 머신에서 개발
<b>Live Server</b>	라이트 리모드 로컬 개발 서버
<b>Thunder Client</b>	VS Code 내 REST API 클라이언트

## 설정

설정 접근	
<b>Ctrl+,</b>	설정 UI 열기
<b>Ctrl+Shift+P → settings.json</b>	설정 JSON 열기

## 일반 설정 (JSON)

```
{
  "editor.fontSize": 14,
  "editor.tabSize": 2,
  "editor.formatOnSave": true,
  "editor.wordWrap": "on",
  "files.autoSave": "afterDelay"
}
```

## 워크스페이스 vs 사용자

<b>User settings</b>	전역, 모든 프로젝트에 적용
<b>Workspace settings</b>	프로젝트별, vscode/settings.json에 저장
<b>Workspace overrides User</b>	워크스페이스 설정이 우선

## 키보드 단축키

커스터마이징	
<b>Ctrl+K Ctrl+S</b>	키보드 단축키 에디터 열기
<b>keybindings.json</b>	커스텀 바인딩용 JSON 파일

## 필수 단축키

<b>Ctrl+Shift+P</b>	명령 팔레트 (모든 것 접근)
<b>Ctrl+P</b>	파일 빠른 열기
<b>Ctrl+Shift+F</b>	모든 파일 검색
<b>Ctrl+H</b>	파일에서 찾기 및 바꾸기
<b>Ctrl+Shift+H</b>	파일 전체에서 찾기 및 바꾸기
<b>F2</b>	선택된 줄 변경
<b>Ctrl+.</b>	빠른 수정 (코드 액션)
<b>Ctrl+Space</b>	IntelliSense 트리거
<b>Ctrl+Shift+I</b>	문서 포맷

## Git 통합

소스 컨트롤 보기	
<b>Ctrl+Shift+G</b>	소스 컨트롤 패널 열기
<b>+</b>	변경 사항 스테이지 (SCM 보기에서)
<b>-</b>	변경 사항 언스테이지
<b>Ctrl+Enter</b>	스테이지된 변경 사항 커밋
<b>...</b>	더 많은 작업 (푸시, 풀 등)

## 인라인 Git 기능

<b>Gutter indicators</b>	추가/수정/삭제에 대한 녹색/파란색/빨간색 표시
<b>Ctrl+Shift+P → Git: Checkout</b>	현재 커밋으로 전환
<b>Ctrl+Shift+P → Git: Pull</b>	원격에서 풀
<b>Ctrl+Shift+P → Git: Push</b>	원격으로 푸시
<b>Click gutter change</b>	수정된 줄의 인라인 diff 보기

## Diff 및 머지

<b>SCM → Open Changes</b>	나란히 diff 보기
<b>Inline diff arrows</b>	개별 변경 사항 되돌리기
<b>3-way merge editor</b>	중첩을 시각적으로 해결 (1.69+)

## 자주 쓰는 패턴

워크스페이스 기능	
<b>Ctrl+Shift+E</b>	파일 탐색기에 포커스
<b>Ctrl+Shift+D</b>	디버그 보기에 포커스
<b>Ctrl+Shift+M</b>	문제 패널 표시
<b>Ctrl+K Ctrl+T</b>	색상 테마 변경
<b>Ctrl+K Z</b>	Zen 모드 진입

## 스니펫

```
// File - Preferences -> Configure Snippets
{
  "print": {
    "prefix": "log",
    "body": "console.log('$1');",
    "description": "Console log"
  }
}
```

## 태스크 (tasks.json)

```
{
  "label": "build",
  "type": "shell",
  "command": "npm run build",
  "group": "build"
}
```