16. GitHub Copilot 및 AI 도구

GitHub Copilot 개요 및 설치

GitHub Copilot - 개요

- GitHub + OpenAI가 만든 **AI 기반 코드 자동 완성 도구**
- **작성 중인 코드 문맥을 이해** → 다음 코드 자동 예측
- 다양한 언어 지원 (JS/TS, Python, Java, Go, C/C++, HTML/CSS 등)
- 다양한 IDE 지원 (VS Code, JetBrains, Neovim 등)

주요 기능

✅ 함수/블록 자동 완성
✅ 반복 코드 자동 생성
✅ 테스트 코드 자동 생성
✅ 주석 기반 전체 코드 자동 작성

GitHub Copilot - 설치

1️⃣ 준비

- GitHub 계정 필요
- Copilot **구독 필요** → GitHub Copilot 설정 페이지 에서 활성화

2️⃣ 설치 (VS Code 기준)

1. VS Code → Extensions → `GitHub Copilot` 검색 → 설치
2. GitHub 계정 로그인 → Copilot 구독 확인 후 연동
3. 사용 방법:
    - 코드 작성 중 자동 제안 표시 (회색 텍스트)
    - `Tab` → 적용 / `Esc` → 무시 / `Alt + [` `]` → 다른 제안 보기

3️⃣ 설치 (JetBrains IDE 기준)

- Preferences → Plugins → Marketplace → `GitHub Copilot` 검색 → 설치
- GitHub 계정 연동 후 사용

정리

항목	내용
목적	AI 기반 코드 자동 완성
사용 효과	생산성 향상, 반복 작업 자동화

항목	내용
설치 위치	VS Code, JetBrains, Neovim 등
사전 필요	GitHub 계정 + Copilot 구독

코드 자동 완성

개요

- AI 기반 코드 자동 완성 도구 (GitHub + OpenAI)
- 현재 코드 **문맥(Context)** 을 분석해 다음에 올 코드를 예측
- 다양한 언어와 IDE 지원 → **VS Code** 에서 가장 많이 사용

사용법

- **설치**: GitHub Copilot 확장 설치 후 GitHub 계정 연동
- **작성**: 코드 작성 중 → 자동으로 **회색 ghost text**로 제안 표시
  - `Tab` → 적용
  - `Esc` → 무시
  - `Alt + [` / `]` → 다른 제안 보기

자동 완성 잘 되는 경우

- 주석 기반 함수 자동 생성
- 반복문, 조건문 블록 자동 완성
- API 호출 예제 자동 완성
- 테스트 코드 자동 생성
- 표준 라이브러리 사용 시 높은 정확도

예시

```
# Calculate factorial
def factorial(n):
# → Copilot이 함수 본문 자동 완성 제안
```

```
// Fetch user data from API
function fetchUserData(userId) {
```

장점

- 코드 작성 속도 크게 향상
- 반복 코드 자동 처리
- 테스트 코드 작성 시간 단축
- 표준 코드 스타일 학습 기반 → 코드 일관성 유지

주의사항

- 제안 코드가 항상 정확한 것은 아님 → 반드시 리뷰 후 사용
- 성능, 보안, 라이센스 문제 확인 필요
- 복잡한 비즈니스 로직은 직접 설계 후 보조용으로 활용

정리

장점	주의
생산성 증가	검토 필수
반복 작업 자동화	품질 확인 필요
테스트 코드 빠른 생성	라이센스 고려 필요

Pull Request 자동 요약

개요

- GitHub Copilot이 **Pull Request(= PR)** 에 작성된 변경사항(diff), 커밋 메시지, 코드 변경 context 등을 AI로 분석해서 → **자동으로 PR 설명(summary)** 를 생성해주는 기능

주요 목적

✅ PR 작성자 → 빠르게 PR 설명 초안 작성
✅ 리뷰어 → PR에서 어떤 변경이 있었는지 빠르게 파악 가능
✅ 팀 전체 → 더 깔끔한 PR 리뷰 culture 구축

사용 흐름

1️⃣ PR 작성 시

- GitHub Web UI → **Pull Request 작성 화면으로 이동**
- `Write` 탭 하단에 **"Copilot suggest summary"** 버튼 표시됨 (Copilot PR summary 기능 활성화 시)

2️⃣ 클릭하면

- AI가 PR diff, 커밋 메시지, 코멘트 등을 분석 → **PR Summary 초안 자동 생성**
- 사용자는 필요 시 수정 후 PR 제출

지원 조건

- GitHub Copilot 구독 필요 (개인, 조직용 모두 지원)
- GitHub.com → Public Repo 또는 Enterprise Repo (Enterprise Cloud 기준)
- GitHub Actions 통해 Copilot for PR Summary 사용도 가능

주요 장점

✅ PR 설명 작성 속도 향상 → 초안 자동 생성
✅ 일관된 PR Summary 스타일 유지 가능
✅ 대형 diff PR에서도 AI가 **핵심 변경 요약** 가능
✅ 리뷰어가 **변경 내용의 주요 포인트** 빠르게 이해 가능

한계

❗ PR 설명은 AI 제안 기반 → **최종 검토 필수**
❗ 복잡한 코드 변경의 "의도" 까지 정확하게 반영되지는 않음
❗ 단순 파일 구조 변경 등에서는 요약 품질이 낮을 수 있음

예시

자동 생성된 PR Summary 예시:

```
### Summary

- Refactored user authentication logic for better error handling
- Improved validation for signup form inputs
- Removed deprecated login API endpoint
- Updated test cases for new authentication flow
```

→ 리뷰어가 한눈에 "이번 PR에서 무엇이 변경되었는지" 빠르게 파악 가능

정리

항목	내용
기능명	Copilot for PR summaries
동작 시점	PR 작성 화면에서 수동 호출 or Actions 연계 자동 호출

항목	내용
주요 효과	PR Summary 자동 작성 → 리뷰/작업 속도 향상
지원 여부	GitHub Copilot 구독 필요 + GitHub.com 사용 시 지원
주의사항	AI가 생성한 summary → 반드시 검토 후 제출

결론

- GitHub Copilot의 PR 자동 요약 기능은 **팀 전체 PR culture 개선에 매우 유용**
- 코드 리뷰 품질과 속도를 동시에 높이는 효과
- 단, 최종 제출 전 반드시 **수정/검토 과정**은 필요

GitHub CLI 활용법

GitHub CLI (gh) 개요

- **gh** 명령어 → GitHub 공식 CLI 도구
- GitHub의 주요 기능을 **터미널에서 직접 사용** 가능
- Git과는 별도지만 **Git Workflow와 자연스럽게 통합됨**

주요 기능

✅ Pull Request 생성, 리뷰, 병합
✅ Issue 생성, 조회, 관리
✅ Repository 생성/관리
✅ Actions 확인 및 재실행
✅ Notifications 확인
✅ Gist 관리
✅ Copilot Chat (베타) 사용 가능

설치

macOS

```
brew install gh
```

Ubuntu

```
sudo apt install gh
```

Windows

- https://cli.github.com 에서 설치 파일 다운로드 or `winget install --id GitHub.cli`

인증

```
gh auth login
```

→ GitHub 계정으로 OAuth 인증 진행 → 터미널에서 gh 명령 사용 가능해짐

기본 사용법 예시

Repository 관리

현재 repo clone

```
gh repo clone owner/repo
```

새 repo 생성

```
gh repo create my-repo-name --public
```

Pull Request 관리

PR 목록 조회

```
gh pr list
```

PR 상세 보기

```
gh pr view 123
```

PR 생성

```
gh pr create --title "Feature XYZ" --body "Adds new feature XYZ"
```

PR 병합

```
gh pr merge 123 --squash --delete-branch
```

Issue 관리

Issue 목록 조회

```
gh issue list
```

Issue 상세 보기

```
gh issue view 42
```

Issue 생성

```
gh issue create --title "Bug in login flow" --body "Steps to reproduce..."
```

Actions 확인

Workflow 실행 상태 보기

```
gh run list
```

특정 run 확인

```
gh run view 1234567890
```

Run 재실행

```
gh run rerun 1234567890
```

Notifications 확인

```
gh notifications
```

→ GitHub Web으로 가지 않고 터미널에서 바로 확인 가능

유용한 추가 기능

✅ alias 기능으로 자주 쓰는 명령 단축 가능

```
gh alias set pr-open 'pr create --web'
```

✅ Copilot Chat (gh extensions) 으로 CLI에서 Copilot 활용 가능

```
gh extension install github/gh-copilot
gh copilot chat
```

정리

기능	명령 예시
PR 생성/리뷰/병합	`gh pr create`, `gh pr list`, `gh pr merge`
Issue 관리	`gh issue create`, `gh issue list`, `gh issue view`
Repo 관리	`gh repo clone`, `gh repo create`
Actions 확인	`gh run list`, `gh run rerun`
Notifications 확인	`gh notifications`

결론

- GitHub CLI는 **GitHub Web UI에서 하는 작업 대부분을 터미널에서 빠르게 처리 가능**
- Git Workflow + GitHub Workflow 자연스럽게 통합 가능
- PR/Issue/Actions 작업 흐름을 **끊김없이 터미널에서 자동화/가속화**할 수 있음 → 실무에서 아주 유용