

# GitHub Copilot 맞춤 설정 가이드

프로젝트별 AI 지침 파일 작성법

동의대학교 서진석(jsseo@deu.ac.kr)

최종 수정: 2025. 11. 27. 09:31

## 차례

---

1	개요	2
1.1	Copilot 지침 파일이란?	2
1.2	왜 필요한가? (대학 업무 관점)	2
1.3	주요 이점	2
2	주요 파일 유형	2
2.1	copilot-instructions.md (프로젝트 기본 지침)	3
2.2	Instruction Files (.instructions.md)	3
2.3	Prompt Files (.prompt.md)	4
2.4	AGENTS.md (에이전트 모드용)	4
3	지침 설정 방법	4
3.1	프로젝트 레벨 설정 (팀 공유용)	4
3.2	개인 설정 (모든 프로젝트 적용)	5
3.3	VS Code 설정 예시	5
4	구성원별 활용 예시	5
4.1	교수용	5
4.1.1	연구 제안서 작성 지침	5
4.1.2	논문 작성 지침	6
4.2	학생용	6
4.2.1	과제 및 리포트 작성 지침	6
4.3	연구원용	6
4.3.1	실험 보고서 작성 지침	6
4.4	행정직원용	7
4.4.1	공문서 작성 지침	7
5	applyTo 패턴 활용법	7
6	시작하기 (단계별 가이드)	7
6.1	Step 1: 첫 지침 파일 만들기	7
6.2	Step 2: 동작 확인	8
6.3	Step 3: 활용 확장	8

<b>7 자주 묻는 질문 (FAQ)</b>	<b>8</b>
7.1 Q1: 지침 파일이 적용되었는지 어떻게 확인하나요? . . . . .	8
7.2 Q2: 여러 지침 파일이 충돌하면 어떻게 되나요? . . . . .	8
7.3 Q3: 지침 파일을 팀원과 공유하려면? . . . . .	8
7.4 Q4: 지침이 너무 길면 문제가 있나요? . . . . .	9
7.5 Q5: 영문 응답을 한글로 바꾸고 싶어요. . . . .	9
<b>8 빠른 참조</b>	<b>9</b>

## 1 개요

### 1.1 Copilot 지침 파일이란?

GitHub Copilot 지침 파일은 AI 어시스턴트인 Copilot에게 “이 프로젝트에서는 이렇게 답변해 줘”라고 미리 알려주는 설정 파일입니다.

비유로 이해하기:

- 새로 채용한 조교에게 “우리 연구실에서는 보고서를 이런 형식으로 작성해”라고 알려주는 매뉴얼
- 학생에게 “이 과목 리포트는 APA 형식으로 작성하세요”라고 알려주는 가이드라인
- 공문 작성 시 참고하는 기관 내부 양식집

### 1.2 왜 필요한가? (대학 업무 관점)

상황	지침 파일 없이	지침 파일 있으면
연구 제안서 작성	매번 형식 설명 필요	자동으로 형식 준수
논문 작성	학술 용어 번역 불일관	일관된 전문용어 사용
강의자료 제작	학과별 스타일 무시	학과 맞춤형 콘텐츠
행정 문서	공문서 형식 오류	기관 양식 자동 적용

### 1.3 주요 이점

- **일관성**: 동일한 프로젝트에서 항상 같은 스타일로 응답
- **효율성**: 매번 같은 설명을 반복할 필요 없음
- **협업**: 연구실/학과 단위로 설정 공유 가능
- **맥락 이해**: 프로젝트별 특수 용어나 요구사항 자동 적용

## 2 주요 파일 유형

GitHub Copilot은 네 가지 유형의 지침 파일을 지원합니다.

## 2.1 copilot-instructions.md (프로젝트 기본 지침)

---

### 항목    내용

---

용도	프로젝트 전체에 적용되는 기본 규칙
위치	<code>.github/copilot-instructions.md</code>
특징	프로젝트 폴더를 열면 자동으로 적용, 팀원과 Git으로 공유 가능

---

#### 예시 (연구 프로젝트용):

```
# 연구 프로젝트 기본 지침
## 프로젝트 개요
이 프로젝트는 2026년도 NRF 기초연구사업 신청을 위한 연구 제안서 작성 프로젝트
입니다.
## 문서 작성 원칙
- 모든 문서는 한국어로 작성
- 학술 용어는 영문 병기 (예: 인공지능(Artificial Intelligence))
- 참고문헌은 APA 7th Edition 형식 사용
```

## 2.2 Instruction Files (.instructions.md)

---

### 항목    내용

---

용도	특정 파일 유형이나 폴더에만 적용되는 세부 규칙
위치	<code>.github/instructions/*.instructions.md</code>
특징	<code>applyTo</code> 패턴으로 적용 범위 지정

---

#### 예시 (논문 작성용):

```
---
applyTo: "**/paper/**/*.md"
---
# 논문 작성 지침
## 구조: 초록(200단어 이내) → 서론 → 방법 → 결과 → 논의 → 결론
## 인용 형식: 본문 내 인용 (저자, 연도), 참고문헌 APA 7th Edition
```

## 2.3 Prompt Files (.prompt.md)

항목	내용
----	----

용도	재사용 가능한 프롬프트 템플릿
----	------------------

위치	.github/prompts/*.prompt.md
----	-----------------------------

특징	채팅창에서 / 명령으로 호출, 변수 사용 가능 (\${selection}, \${input:제목})
----	---

### 예시 (논문 요약용):

```

---
description: "학술 논문을 구조화된 형식으로 요약"
mode: "ask"
---
다음 논문을 요약해주세요: ${selection}
형식: 연구 목적(1-2문장), 연구 방법(3-4문장), 주요 결과(목록), 인용 정보(APA)
    
```

## 2.4 AGENTS.md (에이전트 모드용)

항목	내용
----	----

용도	Copilot 에이전트 모드에서 자율적 작업 수행 시 참조
----	----------------------------------

위치	프로젝트 루트 또는 하위 폴더의 AGENTS.md
----	-----------------------------

특징	에이전트가 코드 수정, 명령 실행 시 참조
----	-------------------------

## 3 지침 설정 방법

### 3.1 프로젝트 레벨 설정 (팀 공유용)

장점: Git으로 공유, 팀원 모두 동일 설정 사용

1. 프로젝트 폴더에 .github 폴더 생성
2. copilot-instructions.md 파일 생성
3. 지침 내용 작성
4. Git commit하여 팀과 공유

권장 프로젝트 폴더 구조:

```

프로젝트폴더/
├── .github/
    
```

```

| |—— copilot-instructions.md          # 기본 지침
| |—— instructions/                    # 세부 지침
| |   |—— paper.instructions.md
| |   |—— proposal.instructions.md
| |—— prompts/                          # 프롬프트 템플릿
| |—— summarize.prompt.md
| |—— translate.prompt.md
|—— (프로젝트 파일들...)

```

## 3.2 개인 설정 (모든 프로젝트 적용)

**장점:** 한 번 설정으로 모든 프로젝트에서 사용

1. VS Code 설정 열기 (Cmd+, 또는 Ctrl+,)
2. “copilot chat prompts” 검색
3. “Chat: Prompts Folders” 설정에 폴더 경로 추가
4. 해당 폴더에 .instructions.md 또는 .prompt.md 파일 생성

## 3.3 VS Code 설정 예시

```

{
  "github.copilot.chat.codeGeneration.useInstructionFiles": true,
  "chat.promptFiles.locations": [
    { "path": "/Users/username/my-copilot-prompts" }
  ]
}

```

# 4 구성원별 활용 예시

## 4.1 교수용

### 4.1.1 연구 제안서 작성 지침

**파일:** .github/instructions/proposal.instructions.md

```

---
applyTo: "**/proposal/**"
---

```

# 연구 제안서 작성 지침

작성 원칙:

1. 간결하고 명확한 문장: 한 문장에 하나의 아이디어만
2. 객관적 근거 제시: 주장마다 참고문헌 인용

3. 정량적 표현 선호: “많은” 대신 “약 70%의”

필수 포함 요소: 연구 목적, 국내외 연구 동향, 연구 내용 및 방법, 기대 효과

#### 4.1.2 논문 작성 지침

파일: .github/instructions/paper.instructions.md

```
---
applyTo: "**/papers/**/*.*md"
---
```

# 학술 논문 작성 지침  
 문체: 수동태보다 능동태 선호, 1인칭 복수 사용 (We propose, Our approach)  
 인용 형식: IEEE 형식 [1], [2], [3-5] / 저자 언급 시: Kim et al. [1] proposed...

### 4.2 학생용

#### 4.2.1 과제 및 리포트 작성 지침

파일: .github/instructions/assignment.instructions.md

```
---
applyTo: "**/assignments/**"
---
```

# 과제 작성 지침  
 기본 형식: 맑은 고딕 11pt, 줄간격 1.5줄, 여백 상하좌우 2.5cm  
 AI 활용 시 주의사항:  
 - AI는 아이디어 브레인스토밍에 활용  
 - 생성된 내용 반드시 검증  
 - AI 활용 사실 명시 (교수님 지침 따름)

### 4.3 연구원용

#### 4.3.1 실험 보고서 작성 지침

파일: .github/instructions/experiment.instructions.md

```
---
applyTo: "**/experiments/**"
---
```

# 실험 보고서 작성 지침  
 보고서 구조: 실험 개요 → 실험 환경 → 실험 방법 → 실험 결과 → 결론  
 재현성 보장 체크리스트:  
☐ 랜덤 시드 기록 / ☐ 코드 버전 (commit hash) / ☐ 데이터셋 버전 / ☐ 환경 설정 파일 첨부

## 4.4 행정직원용

### 4.4.1 공문서 작성 지침

파일: `.github/instructions/official-document.instructions.md`

```

---
applyTo: "**/documents/**"
---
# 공문서 작성 지침
문서 형식: A4 세로, 휴먼명조 또는 바탕체, 본문 13pt, 줄간격 160%
문체 규칙: 경어체 사용 (~습니다, ~합니다), 간결하고 명확한 표현
날짜 표기: 2025. 11. 26. / 금액 표기: 금 1,000,000원(일백만 원)
    
```

## 5 applyTo 패턴 활용법

패턴	설명	적용 대상
<code>**/*.md</code>	모든 Markdown 파일	프로젝트 내 모든 .md 파일
<code>**/paper/**</code>	paper 폴더 내 모든 파일	paper 폴더 하위 전체
<code>src/**/*.py</code>	src 폴더의 Python 파일	src 하위 모든 .py 파일
<code>*.md,txt</code>	루트의 md 또는 txt	최상위 폴더만

주의사항:

- 패턴은 프로젝트 루트 기준
- 대소문자 구분 (OS에 따라 다름)
- 여러 instructions 파일이 같은 파일에 적용될 수 있음
- 충돌 시 더 구체적인 패턴 우선

## 6 시작하기 (단계별 가이드)

### 6.1 Step 1: 첫 지침 파일 만들기

1. 프로젝트 폴더 열기: File → Open Folder
2. `.github` 폴더 생성: 탐색기에서 우클릭 → New Folder
3. `copilot-instructions.md` 파일 생성
4. 기본 템플릿 작성:



```
# 프로젝트 지침
## 기본 정보
이 프로젝트는 [프로젝트 설명]을 위한 것입니다.
## 작성 규칙
- 한국어로 작성
- 존댓말 사용
- 간결한 문장 선호
## 용어집
- [용어1]: [설명]
```

## 6.2 Step 2: 동작 확인

1. Copilot Chat 열기 (Cmd+Shift+I 또는 Ctrl+Shift+I)
2. 테스트 질문: “이 프로젝트에 대해 알려줘”
3. 지침 파일 내용이 반영되었는지 확인

## 6.3 Step 3: 활용 확장

1. .github/instructions/ 폴더 생성하여 상황별 지침 추가
2. .github/prompts/ 폴더 생성하여 프롬프트 템플릿 추가
3. VS Code 설정에서 개인 프롬프트 폴더 경로 추가 (선택)

## 7 자주 묻는 질문 (FAQ)

---

### 7.1 Q1: 지침 파일이 적용되었는지 어떻게 확인하나요?

Copilot Chat에서 “현재 적용된 지침을 알려줘”라고 물어보면 됩니다. 또는 채팅창 상단의 컨텍스트 영역에서 첨부된 파일을 확인할 수 있습니다.

### 7.2 Q2: 여러 지침 파일이 충돌하면 어떻게 되나요?

모든 해당 지침이 합쳐져서 적용됩니다. 내용이 충돌하는 경우, Copilot이 가장 관련성 높은 규칙을 선택하거나 두 규칙을 조합합니다.

### 7.3 Q3: 지침 파일을 팀원과 공유하려면?

.github 폴더를 Git에 커밋하면 됩니다. 팀원이 저장소를 clone하면 자동으로 같은 지침이 적용됩니다.

## 7.4 Q4: 지침이 너무 길면 문제가 있나요?

Copilot은 컨텍스트 길이에 제한이 있으므로, 핵심 내용 위주로 간결하게 작성하는 것이 좋습니다. 긴 지침보다는 여러 개의 짧은 지침 파일로 분리하세요.

## 7.5 Q5: 영문 응답을 한글로 바꾸고 싶어요.

copilot-instructions.md에 다음을 추가하세요:

- ```
## 언어 설정
- 모든 응답은 한국어로 작성
- 전문용어는 영문 병기 가능
```

## 8 빠른 참조

| 파일 유형 | 확장자                     | 위치                    | 용도          |
|-------|-------------------------|-----------------------|-------------|
| 기본 지침 | copilot-instructions.md | .github/              | 프로젝트 전체 규칙  |
| 세부 지침 | *.instructions.md       | .github/instructions/ | 특정 파일/폴더 규칙 |
| 프롬프트  | *.prompt.md             | .github/prompts/      | 재사용 템플릿     |
| 에이전트  | AGENTS.md               | 루트 또는 하위 폴더           | 에이전트 모드 지침  |

| 동작              | Mac         | Windows      |
|-----------------|-------------|--------------|
| Copilot Chat 열기 | Cmd+Shift+I | Ctrl+Shift+I |
| 인라인 제안 수락       | Tab         | Tab          |
| 다음 제안           | Alt+]       | Alt+]        |
| 이전 제안           | Alt+[       | Alt+[        |
| 제안 거부           | Esc         | Esc          |