

Windows + WSL Codex/PDF Merger 완전 설치 매뉴얼 (한글)

이 문서는 github.com/ffkaka/pdf-merger 프로젝트를 **Windows 환경에서 WSL2 기반으로 안정적으로 실행**하기 위한 전체 절차를 설명한다.
기준 날짜: **2026-02-22**

핵심 원칙:
• Codex CLI의 Windows 네이티브 지원은 실험적일 수 있으므로, 실사용은 **WSL2(Ubuntu)** 기준으로 구성한다.
• 프로젝트의 agent/skill 및 pmerge 실행도 모두 WSL에서 표준화한다.

1. 사전 준비 (Windows)

관리자 권한 PowerShell을 열고 아래를 순서대로 실행한다.

```
winget install --id Git.Git -e  
winget install --id OpenJS.NodeJS.LTS -e  
winget install --id Python.Python.3.12 -e  
wsl --install -d Ubuntu
```

설치 후 Windows를 재부팅한다.

확인:

```
git --version  
node --version  
npm --version  
python --version  
wsl --status
```

2. WSL(Ubuntu) 초기 설정

Ubuntu를 실행하고 Linux 사용자 계정을 만든다. 이후:

```
sudo apt update  
sudo apt upgrade -y  
sudo apt install -y build-essential python3 python3-pip python3-venv ghostscript poppler-utils
```

권장 설정:

```
git config --global user.name "YOUR_NAME"  
git config --global user.email "YOUR_EMAIL"
```

3. Codex CLI 설치 (WSL 내부)

공식 기본 설치(2026-02-22 기준):

```
npm install -g @openai/codex
```

인증(권장):

```
codex --login
```

또는 API 키 방식:

```
echo 'export OPENAI_API_KEY="YOUR_OPENAI_API_KEY"' >> ~/.bashrc  
source ~/.bashrc
```

동작 확인:

```
codex --version  
codex
```

4. 프로젝트 설치 (WSL 홈 기준)

요구사항대로 WSL 홈에 설치:

```
cd ~  
git clone https://github.com/ffkaka/pdf-merger.git  
cd ~/pdf-merger
```

디렉터리 확인:

```
pwd  
ls -la
```

기대 경로:

- 프로젝트 루트: ~/pdf-merger
- 입력 PDF 폴더: ~/pdf-merger/japan
- 결과 폴더: ~/pdf-merger/output/pdf
- 임시 폴더: ~/pdf-merger/tmp/pdfs

5. Skill/Agent 실행 준비

이 프로젝트는 AGENTS.md 지침에 따라 동작한다.

PDF 관련 작업 시 Codex에 pdf skill을 사용하게 하거나, 필요 시 아래를 실행:

```
codex skill install pdf
```

주의:

- PDF 검색은 텍스트 추출 기반으로 수행한다.
- 최종 병합 산출물은 output/pdf/에 생성된다.
- 중간 산출물/리포트는 tmp/pdfs/를 사용한다.

6. pmerge 사용법 (WSL 내부)

처음 사용하는 경우 아래 3단계만 따라하면 된다.

1. 프로젝트 폴더로 이동:

```
cd ~/pdf-merger
```

2. 키워드 파일 준비 (tmp/pdfs/keywords.txt):

```
IC-211  
IC-212  
IC-220
```

3. 병합 실행:

```
./pmerge japan --keywords-file tmp/pdfs/keywords.txt
```

결과:

- 병합 PDF: output/pdf/merged_keywords.pdf
- 리포트: tmp/pdfs/keyword_merge_report.json

자동 처리:

- 기본 검색 기준은 파일명이다.
- 결과가 2MB를 넘으면 자동으로 분할 파일을 만든다.
- 예: merged_keywords_01.pdf, merged_keywords_02.pdf

자주 쓰는 추가 예시(선택):

- 내용(본문) 기준으로 검색하고 싶을 때:

```
PERGE_EXTRA_ARGS="--match-mode content" ./pmerge japan --keywords-file tmp/pdfs/keywords.txt
```

• 먼저 매칭 결과만 확인할 때:

```
PERGE_EXTRA_ARGS="--dry-run" ./pmerge japan --keywords-file tmp/pdfs/keywords.txt
```

7. Codex 프롬프트 예시 (마지 + 검증)

아래처럼 **입력폴더 + 키워드 파일**만 알려주면 된다.

수동 실행 명령이 필요하면 6번을 참고한다.

7-1) 왜 경로를 붙이니

@경로를 붙이면 Codex가 어떤 파일/폴더를 기준으로 처리할지 정확히 이해한다.

자주 쓰는 형태:

- @AGENTS.md
- @japan
- @tmp/pdfs/keywords.txt

7-2) 가장 쉬운 요청 예시

@AGENTS.md 규칙대로 처리해줘.

입력폴더는 @japan 폴더와 @tmp/pdfs/keywords.txt 앤 필터링/병합하고 결과 파일과 검증 결과를 알려줘.

7-3) "사용방법 알려줘" / "사용방법" 요청 예시

@AGENTS.md 먼저 확인하고 이 프로젝트 사용방법 알려줘.

입력은 @japan 폴더와 @tmp/pdfs/keywords.txt 파일을 기준으로 설명해줘.

수동 실행이 필요하면 6번 항목을 보라고 함께 안내해줘.

8. pmerge를 전역 명령으로 등록 (WSL)

WSL 어디서든 pmerge 실행하려면:

```
mkdir -p ~/.local/bin  
ln -sf ~/pdf-merger/pmerge ~/.local/bin/pmerge  
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc  
source ~/.bashrc
```

검증:

```
which pmerge  
pmerge japan IC-211
```

9. 자주 발생하는 문제

1. codex: command not found

- 원인: 전역 npm bin 경로 미반영

• 해결:

```
echo 'export PATH="$(npm bin -g):$PATH"' >> ~/.bashrc  
source ~/.bashrc
```

2. Permission denied: ./pmerge

- 해결:

```
chmod +x ~/pdf-merger/pmerge
```

3. PDF 라이브러리 미설치 오류

- 현재 프로젝트 스크립트는 pypdf/pdfplumber가 없으면 ghostscript(gs) fallback을 사용한다.

• gs 확인:

```
which gs
```

10. 운영 권장사항

• 대량 병합 작업은 WSL 내부에서 실행한다.

- 결과물은 output/pdf/만 최종 산출물로 취급한다.

• tmp/pdfs/는 중간 산출물/리포트 용도로 유지한다.

• 키워드가 많으면 파일(--keywords-file) 기반으로 관리한다.

11. 공식 문서 링크

• Codex 소개: <https://openai.com/codex/>

• Codex CLI 시작 문서: <https://developers.openai.com/codex/cli>