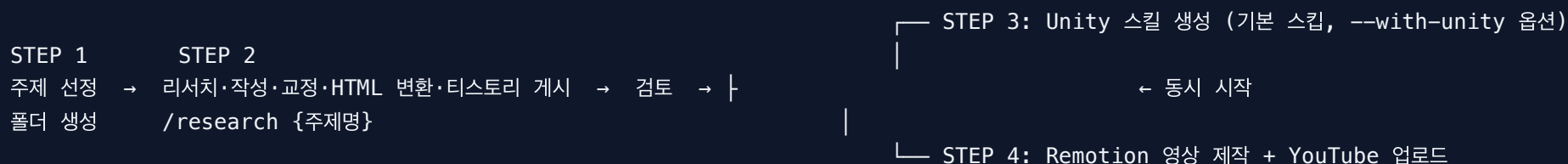


# WORKFLOW

## WORKFLOW.md — 전체 작업 흐름

모든 값은 `config/settings.yaml` 에서 결정됩니다.

### 전체 흐름



- STEP 1 ~ 2: 순차 실행, STEP 2 종료 시 사용자 검토 1회
- STEP 3, 4: `/publish-parallel` 로 동시 시작

### STEP 1 — 주제 선정 및 폴더 생성

커리큘럼에서 미완료 주제를 선택합니다.

- 주제 목차: `docs/curriculum.md`

완료된 주제는 [v] 또는 ✅ 표시가 들어갑니다. 새 주제는 적절한 PART에 추가 후 사용자 확인을 거칩니다.

폴더는 /research 스킴이 STEP 0에서 자동 생성합니다. 수동 생성도 가능:

```
mkdir -p "content/topic/PART {N}/{주제명}"/{research,tistory,youtube_make,youtube_result}
```

산출물 기준 경로 = TOPIC\_FOLDER.

## STEP 2 — 리서치 + 작성 + 교정 + HTML 변환 + 티스토리 게시

```
/research {주제명}
```

/research 내부 단계:

| 단계 | 내용  |
|----|---|
| 0  | 입력값 수집 + 출력 폴더 · 카테고리 · 재생목록 결정   |
| 1  | 3개 소스 병렬 리서치 (NotebookLM + Gemini + Claude)   |
| 2  | <code>/c#-blog-write-guide</code> 규칙으로 게시글 작성 → <code>result.md</code>                |
| 3  | 교정 + 작성 가이드 검토 보고 ( <code>/c#-blog-reviewer</code> )                                  |
| 4  | <code>/tistory-format</code> 으로 <code>tistory.html</code> 생성                          |
| 5  | <code>/tistory-publish</code> 로 티스토리 게시 (카테고리는 <code>config.tistory.category</code> ) |
| 6  | 텔레그램 전송 + 사용자 최종 검토   |

산출물:

```
TOPIC_FOLDER/  
├─ research/  
│   ├── notebook1m.md  
│   └─ gemini.md  
├─ tistory/  
│   ├── tistory.html    ← + 티스토리 게시 완료  
│   ├── tistory_url.txt  
│   └─ diagram_*.svg/png  
└─ result.md
```

## STEP 3 · 4 — 병렬 배포

```
# 기본 (Unity 스킵 스킵)  
/publish-parallel  
  
# Unity 스킵 포함이 필요한 경우  
/publish-parallel --with-unity
```

두 에이전트가 동시 시작합니다.

```
/publish-parallel  
├─ [Agent 1] unity-skills-creator    (기본 스킵, --with-unity 시 실행)  
└─ [Agent 2] video-production        (Remotion + YouTube 업로드)
```

텔레그램으로 시작 → 각 에이전트 완료 알림 → 전체 완료 집계.

### STEP 3 — Unity 실전 가이드 ( `unity-skills-creator` )

`result.md` 에서 Unity 고유 함정·패턴을 추출해 DO/DON'T SKILL.md를 생성합니다. 기본은 스킵이며 `--with-unity` 명시 시에만 실행됩니다.

산출물:

```
content/topic/unity-skills/PART {N}/{주제명}/SKILL.md
content/topic/unity-skills/README.md ← 인덱스 업데이트
```

### STEP 4 — Remotion 영상 제작 + YouTube 업로드 ( `/video-production` )

| 단계    | 내용  | 담당                                    |
|-------|---|---------------------------------------|
| 0     | 환경 확인 (Node, Remotion, TTS API 키 등)                   | 메인                                    |
| 1     | HTML 파싱 → <code>youtube_make/scenes.json</code>       | 메인                                    |
| 2     | 대화 스크립트 → <code>youtube_make/script.json</code>       | 메인                                    |
| 2.5   | <code>/script-reviewer</code> — 기술 정확성·이해도 검증 + 자동 수정 | 메인                                    |
| 3~5.7 | 나레이션 검증 + TTS + Remotion 렌더링 + 썸네일 + 무결성 + SRT        | <code>/video-render</code>            |
| 6~7   | YouTube 업로드 + 텔레그램 알림                                 | <code>/youtube-upload-pipeline</code> |

재생목록: `config.youtube.playlist` ( `C# 기초` , 공개).

인증 실패 시 `/google-auth` 가 자동 선행됩니다.

산출물:

```
TOPIC_FOLDER/
├── youtube_make/
│   ├── scenes.json
│   ├── script.json
│   ├── props.json
│   └── audio/
└── youtube_result/
    ├── production.mp4
    ├── subtitles_ko.srt
    ├── subtitles_en.srt
    ├── thumbnail.jpg
    └── youtube_url.txt
```

## 배치 · 스케줄

여러 주제를 한 번에 진행할 때:

```
# 여러 주제 순차 리서치
/batch-research {주제1} {주제2} ...

# 리서치 + 배포 전체 자동화 (4개 × 배치, 5시간 간격)
/scheduled-pipeline {주제들...}
```

스케줄 상태 파일: `config.pipeline.state_file` (기본 `.claude/pipeline-state.json`)

## 최종 산출물 디렉토리 구조

```
content/topic/PART {N}/{주제명}/
├─ research/
├─ tistory/
├─ youtube_make/
├─ youtube_result/
└─ result.md
```

```
content/topic/unity-skills/PART {N}/{주제명}/SKILL.md # --with-unity 시에만
```

## 스킬 · 에이전트 맵

### 메인 워크플로우

| 스킬                               | STEP    | 설명                             |
|----------------------------------|---------|--------------------------------|
| <code>/research</code>           | 2       | 리서치 + 작성 + 교정 + HTML + 티스토리 게시 |
| <code>/publish-parallel</code>   | 3·4     | 병렬 배포 오케스트레이터                  |
| <code>/batch-research</code>     | 2 (x N) | 여러 주제 순차 리서치                   |
| <code>/scheduled-pipeline</code> | 2~4     | 4개 배치 × 5시간 간격 스케줄             |

## 에이전트

| 에이전트                                | STEP | 담당             |
|-------------------------------------|------|----------------|
| <code>blog-reviewer</code>          | 2    | 품질 검수          |
| <code>tistory-publisher</code>      | 2    | 티스토리 게시        |
| <code>unity-skills-creator</code>   | 3    | Unity SKILL.md |
| <code>video-overview-creator</code> | 별도   | NotebookLM 동영상 |

## 단독 실행

| 스킬                                      | 용도                   |
|---|----------------------|
| <code>/c#-blog-write-guide</code>       | 작성 규칙 참조             |
| <code>/c#-blog-reviewer</code>          | result.md 품질 검수      |
| <code>/il-analysis</code>               | C# → IL 분석           |
| <code>/tistory-format</code>            | 마크다운 → HTML          |
| <code>/tistory-publish</code>           | 티스토리 게시·수정·삭제        |
| <code>/tistory-login</code>             | 티스토리 세션 로그인          |
| <code>/unity-skills</code>              | Unity SKILL.md 생성    |
| <code>/notebooklm-studio video</code>   | NotebookLM 동영상       |
| <code>/notebooklm-studio podcast</code> | NotebookLM 팟캐스트 + 대본 |
| <code>/video-production</code>          | Remotion 영상 파이프라인    |
| <code>/video-render</code>              | TTS + 렌더링 + 자막       |
| <code>/script-reviewer</code>           | script.json 검증       |
| <code>/youtube-upload-pipeline</code>   | 업로드 + 알림             |
| <code>/youtube-upload</code>            | YouTube Data API 업로드 |
| <code>/youtube-upload-web</code>        | YouTube Studio 웹 업로드 |

| 스킬                          | 용도                |
|-----------------------------|-------------------|
| <code>/youtube-login</code> | YouTube Studio 세션 |
| <code>/google-auth</code>   | Google OAuth 재인증  |