AX 4주차 구현 교육 교안 초안: ② 결정론적 자동화(코딩)편

협찬 블로그 제작: 작업·사진 식별, 버전·승인 상태 기록, 누락·미승인 진행 확인 · 비개발자 입문 · 3차시 중 1·3차시 · 초안(미검증)

🟢 1차시: 작업·사진 식별과 버전·승인 상태 기록 (60분)

1. 학습목표

- 블로그 초안 파일에서 사진, GIF, 지도, 글자 수 등 핵심 자산 요소를 정규표현식 코드로 정밀 식별할 수 있어요 [1].
- 원고 수정 전후와 재열기 상태의 서식/컴포넌트 구조 차이를 JSON 데이터로 추출하고 대조할 수 있어요 [2].
- 원고의 버전 변경 이력과 승인 상태를 인계 묶음(Handoff Bundle) 및 대화 관리 도구로 안전하게 기록·보존할 수 있어요 [3,4].

2. 왜 필요한가?

협찬 블로그를 작성할 때 눈으로 사진 수나 지도 삽입 여부를 일일이 세다 보면 실수로 필수 요소를 빼먹거나 본문 글자 수 계산이 엉키기 쉬워요 [5]. 또한, 수정을 거치는 과정에서 어느 버전이 최종 승인된 원고인지 헷갈리면 미완성 초안이 포스팅되는 사고가 발생하므로 이를 코드로 자동 식별하고 기록해야 해요 [5,4].

3. 핵심 개념

- **정규표현식(Regex) 자산 식별**: 마크다운 원고 안에서 `[사진 01]`, `[GIF]`, `[지도]`, `[링크카드]`, `[스티커]` 같은 마커를 일반 본문 텍스트와 분리해 내어 각각의 개수를 정밀 측정하는 기술이에요 [1].
- **스마트에디터 DOM 캡처 파싱**: 네이버 스마트에디터 임시저장본을 새 탭에서 열어 캡처한 JSON 객체(`before.json`, `after.json`)의 `components` 배열에서 사진 URL, 정렬 방식, 형광 강조(#fff8b2) 위치를 추출해 비교하는 방식이에요 [2].
- **인계 묶음(Handoff Bundle) 및 Agit 버전 기록**: 작업 단계 간 이동 시 공개 URL, 원본 경로, 적용 조건, 기준 버전, 승인 상태를 하나의 구조화된 기록으로 보존하는 개념이에요 [3,4]. *(Agit은 대화 및 세션 이력을 브랜치 단위로 관리해 주는 오픈소스 도구예요 [3]).*

4. 따라하기 실습

Step 1: 마크다운 파일에서 사진과 글자 수 자동 식별하기

1. 터미널을 열고 실습용 초안 파일(`draft.md`)을 준비해요 [1].
2. Node.js 기반 검수 스크립트(`check-draft-metrics.mjs`)를 구동해 텍스트와 자산 마커를 식별해요 [1].

```
# 초안의 자산 요소 및 하드 규격 식별 구동
node check-draft-metrics.mjs draft.md --preset cafe-town --venue "성복동 카페"
```

1. 스크립트가 `PHOTO_RE` 및 `GIF_RE` 정규식 패턴을 이용해 `[사진 01 — 01-외관.jpg]`이나 `[GIF]` 라인을 감지하여 사진/GIF 수량을 따로 세고, 본문 글자 수 계산에서 제외하는 결과를 확인해요 [1].

Step 2: 임시저장 원고의 서식 및 정렬 상태 대조하기

1. 스마트에디터에서 원고 수정 전(`before.json`)과 저장 후 새 탭 재열기(`after.json`), 서식 계획(`plan.json`) JSON 파일을 준비해요 [2].

2. 서식 검증 스크립트(`verify-draft-format.mjs`)를 실행해요 [2].

```
node verify-draft-format.mjs --before before.json --after after.json --plan plan.json
```

1. 본문 문단 순서, 가운데 정렬(`align: center`), 핵심어 형광 강조(#fff8b2) 개수가 계획과 정확히 일치하는지 출력값을 확인해요 [2].

Step 3: 버전 및 승인 상태 기록하기 (Agit / JSON 저장)

1. Agit 도구를 활용해 현재 대화 세션과 원고 상태를 브랜치에 커밋하고 푸시해요 [3].

```
# 세션 및 작업 이력 기록
agit push
```

1. 이동 관문(Handoff Gate) 수칙에 따라 Drive 원본 경로, 기준 버전, 승인 상태(`Approved`)가 담긴 인계 묶음 JSON을 기록해요 [4].

5. 자주 막히는 곳

- **사진 마커와 글자 수 엉킴**: 사진 설명이나 파일명이 본문 글자 수로 함께 계산되는 경우 정규식 스크립트가 `PHOTO_RE` 패턴을 정확히 걸러내는지 확인해야 해요 [1].
- **사진 지연 로딩 미완료**: 스마트에디터 캡처 시 스크롤을 내리지 않아 이미지 URL이 `https://`로 로딩되지 않으면 `verify-draft-format.mjs` 스크립트가 로딩 미완료 오류를 반환해요 (스크롤을 끝까지 내린 후 재캡처해야 해요) [2].
- **Git/Agit 명령 미인식**: CLI 환경에서 Git이나 Node.js가 설치되어 있지 않아 명령어가 실행되지 않을 수 있어요 (AI에게 "Git과 Node.js 설치를 진행해 줘"라고 요청하여 환경을 정비해요) [6].

6. 정리·퀴즈

- **정리**: 정규표현식으로 원고 자산을 분리 식별하고, DOM 캡처 JSON과 버저닝 도구로 원고 상태를 기록하면 수작업 검수의 누락을 방지할 수 있어요 [1,4,2].
- **퀴즈 1**: 마크다운 원고 검수 시 사진 수량 계산과 본문 글자 수 계산을 분리하기 위해 사용하는 주요 기술은 무엇일까요?
  - *정답*: 정규표현식(Regex) 기반 마커 식별 스크립트 (`check-draft-metrics.mjs`) [1].
- **퀴즈 2**: Agit 도구에서 AI 세션과 대화 이력을 저장하고 복원할 때 활용하는 저장 단위는 무엇일까요?
  - *정답*: Commit(커밋) 및 Branch(브랜치) [3].

(※ 소스 근거: `check-draft-metrics.mjs.txt` [1], `verify-draft-format.mjs.txt` [2], `This New Open Source Tool Just Fixed Your Claude Code Workflow` [3], `gpters-case-2.txt` [4])

🔴 3차시: 누락·미승인 진행 차단을 위한 결정론적 코딩 검증 (60분)

1. 학습목표

- Claude Code 훅(Hook)을 설정하여 하드 규칙 위반 원고의 진행 및 생성을 결정론적으로 차단할 수 있어요 [7,8].

- `exit 2` 종료 코드를 활용해 위험 명령어 및 미승인 커밋/발행 시도를 시스템적으로 차단할 수 있어요 [8,7].
- 실제 공개 발행된 네이버 블로그 HTML DOM을 파싱하여 초안 대비 누락 요소와 별표(**`bold`**) 유출을 일대일 대조 검증할 수 있어요 [9].

2. 왜 필요한가?

텍스트 지침(CLAUDE.md)만으로는 AI가 대화가 길어지면 소제목 구분선이나 지도를 누락하거나, 마크다운 별표(**`bold`**)가 글자 그대로 블로그에 발행되는 실수를 범해요 [9,8,10]. 셸 스크립트와 훅을 이용해 조건 미충족 시 `exit 2`나 `exit 1`로 실행을 물리적으로 막아야 완전한 자동화 검수가 완성돼요 [8,9,11].

3. 핵심 개념

- **Claude Code PreToolUse / PostToolUse 훅**: 도구가 실행되기 직전(`PreToolUse`)에 입력을 검사해 차단하거나, 도구 실행 직후(`PostToolUse`) 포맷터 및 검수 스크립트를 자동 구동하는 통제 장치예요 [7,8].
- **종료 코드(Exit Code) 차단 원리**:
  - `exit 0`: 정상 허용 및 성공 [8,12].
  - `exit 2`: **사전 차단(Block)**. `stderr` 메시지가 AI에게 피드백되어 작업을 원상 복구하도록 강제해요 [8,7].
  - `exit 1`: 검수 실패 판정. 작업 완결 보고를 차단해요 [9,11,1].
- **발행본 HTML DOM 대조 검증 (`verify-published.mjs`)**: 네이버 모바일 PostView HTML을 직접 스크레이핑하여 고유 컴포넌트 ID(`SE-...`)를 추출해 초안의 사진, 소제목, 구분선, 지도, 표 개수를 일대일 대조 검사하는 기술이에요 [9].

4. 따라하기 실습

Step 1: 하드 금지 규칙 검수 스크립트 연결하기

1. `check-draft-metrics.mjs` 스크립트에 하드 금지 규칙을 확인해요 [1].
   - 안 먹은 메뉴 가격 나열(5줄 초과) 감지 [1].
   - 협찬 건 '내돈내산' 오표기 감지 [1].
   - 마크다운 별표(**`bold`**) 문명 유출 감지 [1].
   - 독자 궁금증 6대 요소(가격, 양, 맛, 주차, 가족, 혼잡) 중 5개 이상 충족 검사 [1].
2. 조건을 만족하지 못하면 `FAIL` 문구와 함께 `exit 1`로 종료되어 다음 단계로 진행할 수 없음을 확인해요 [11,1].

Step 2: Claude Code 훅 설정으로 미승인 진행 자동 차단하기

1. 프로젝트의 `.claude/settings.json` 파일에 훅 매처를 등록해요 [7,8].

```
{
  "hooks": {
"PostToolUse": [
      {
"matcher": "Edit|Write",
        "hooks": [
          {
```

```
"type": "command",
"command": "node \"$CLAUDE_PROJECT_DIR/check-draft-metrics.mjs\" draft.md --preset cafe-town --venu
e \"성복동 카페\""
          }
        ]
      }
    ],
    "PreToolUse": [
      {
"matcher": "Bash",
        "hooks": [
          {
"type": "command",
"command": "\"$CLAUDE_PROJECT_DIR/.claude/hooks/protect-files.sh\""
          }
        ]
      }
    ]
  }
}
```

1. 파일 변경 직후 `PostToolUse` 훅이 자동으로 작동하여 하드 규격을 검수함을 확인해요 [7,8].
2. `PreToolUse` 스크립트에서 파괴적 명령어 감지 시 `exit 2`를 반환하도록 작성하여 도구 실행이 차단되는지 시험해요 [7,8].

Step 3: 발행본 HTML DOM과 초안 일대일 대조하기

1. 실제 네이버 블로그에 발행된 포스팅의 `blogId`와 `logNo`를 수집해요 [9].
2. 발행본 대조 스크립트(`verify-published.mjs`)를 구동해요 [9].

```
node verify-published.mjs draft.md --blog myblogid --logno 123456789
```

1. 발행본 HTML의 `se-image`, `se-quotation`, `se-horizontalLine`, `se-placesMap` 개수가 초안과 일치하지 않거나, 별표(`**bold**`)가 글자로 노출되었을 경우 `exit 1`이 반환되어 완료 보고가 차단됨을 확인해요 [9].

5. 자주 막히는 곳

- **종료 코드 오류로 인한 차단 실패**: 차단 훅 스크립트 작성 시 `exit 1`을 쓰면 차단되지 않고 로그만 남은 채 계속 진행돼요 (반드시 `exit 2`를 반환해야 물리적으로 실행이 차단돼요) [8,13].
- **훅 수정 후 미반영**: `settings.json` 스크립트를 수정한 뒤 세션을 그대로 유지하면 스냅샷 때문에 적용되지 않아요 (세션을 재시작하거나 `/hooks` 메뉴에서 수동 갱신해야 해요) [12].
- **네이버 서버 연결 거부(ECONNRESET)**: 발행본 HTML 수집 시 헤더에 모바일 브라우저 User-Agent를 설정하지 않으면 네이버가 요청을 끊어버려요 (스크립트의 User-Agent 헤더 설정을 확인해야 해요) [9].

6. 정리·퀴즈

- **정리**: Claude Code 훅과 `exit 2` 종료 코드, 그리고 발행본 DOM 대조 스크립트를 연결하면 누락이나 미승인 진행을 결함 없이 통제할 수 있어요 [9,8,7].
- **퀴즈 1**: Claude Code PreToolUse 훅에서 위험한 명령어나 미승인 변경을 즉시 사전 차단하기 위해 스크립트가 반환해야 하는 종료 코드는 몇 번일까요?
  - *정답*: `exit 2` [8,7].
- **퀴즈 2**: 초안 검사를 통과했더라도 실제 네이버 블로그에 발행된 후 지도나 소제목이 빠졌는지 확인하기 위해 사용하는 검수 방식은 무엇일까요?
  - *정답*: 모바일 PostView HTML DOM 파싱 및 대조 검증 스크립트 (`verify-published.mjs`) [9].

(※ 소스 근거: `Automate actions with hooks` [7], `Claude Code hooks: a practical guide` [8], `check-draft-metrics.mjs.txt` [1], `verify-published.mjs.txt` [9], `클로드 코드 훅 실전` [13])

📋 [전체 소스 충실도 및 소스 미흡 요약 표]

| 차시 | 주요 다루는 주제 | 소스 충실도 | 소스 미흡 / 외부 커리큘럼 보완 필요 항목 |

| :--- | :--- | :--- | :--- |

| **1차시** | 마크다운 자산 식별, DOM 캡처 파싱, 버전/승인 상태 기록 | **매우 높음** (`check-draft-metrics.mjs` , `verify-draft-format.mjs` , `Agit` 소스 완비) [1,2,3] | 실제 바이너리 이미지 파일 자체의 EXIF 메타데이터 추출 및 OCR 인식 코딩 *(소스 미흡)* |

| **2차시** | CAC 4대 관문, 4대 레인 역할 분담, 예외 승인 기준표 | **높음** (`gpters-case-1` , `gpters-case-2` 소스 완비) [4,5] | 외부 체험단/체험 플랫폼과의 API 실시간 자동 접수/수집 코딩 *(소스 미흡)* |

| **3차시** | 훅(Hook) 기반 진행 차단, exit 2 차단, 발행본 DOM 일대일 대조 | **매우 높음** (`Claude Code Hooks Docs` , `verify-published.mjs` 소스 완비) [9,7] | 네이버 자동 포스팅을 위한 Headless 브라우저(Playwright/Selenium) 세션 컨트롤 및 OAuth/CAPCHA 우회 코딩 *(소스 미흡)* |

부록: 이 교안의 구현 목적, 준비물, 적용·테스트, 성과 측정

이 부록은 4주차 과제가 교안에 넣으라고 한 항목(기대 성과, 필요한 기술과 준비 사항, 실제 업무 적용 및 테스트 방법, 성과 측정 방법)을 모은 것이다.

(1) 구현 목적과 기대 성과

- **구현 목적**:
  - **조건 누락 및 품질 결함 방지**: 작성 가이드라인이 존재하더라도 네이버 지도, 동영상, 필수 사진, 소제목 구분선 등이 최종 발행 단계에서 누락되거나 별표(**`bold`**) 문자가 유출 발행되는 실수를 코드로 자동 검수하여 차단하기 위함입니다 [5,9,11].
  - **하드 규격 자동 통제**: 글자 수, 줄 수, 줄당 글자 수(10~20자, 한 줄 24자 이하), 미디어 수(사진 28~40장, GIF 2~3개), 금지 규칙(안 먹은 메뉴 가격 나열, 대가성 협찬 건 '내돈내산' 오표기, 정보 블록 이모지 누락 등)을 사람이 눈으로 일일이 세지 않고 결정론적 코드로 정밀 통제하기 위함입니다 [1,11].

    - **버전 및 승인 상태 고정**: 원고 수정 과정에서 어느 기준 버전을 읽고 작업했는지 혼동하지 않도록 공개 URL, 원본 경로, 기준 버전, 승인 상태를 인계 묶음(Handoff Bundle)으로 명확히 남기고 미승인 진행을 차단하기 위함입니다 [4].
- **기대 성과**:
    - **첫 발행본 조건 충족률(First-Pass Compliance Rate) 향상**: 최초 발행본에서 필수 요소(지도, 동영상, 서식) 누락을 0건으로 통제합니다 [5].
    - **수정 왕복 횟수 및 검토 시간 절감**: 초안 검토 시 불필요하게 반복되던 피드백 왕복 횟수를 기존 평균 5~6회(추정)에서 2~3회 이하로 줄이고 수작업 검토 시간을 대폭 절감합니다 [4,5].
    - **마감 기한 준수 및 위험 사전 차단**: 예약 날짜 및 리뷰 마감 기한 준수율을 높이고, 미승인 상태에서의 무단 발행/커밋 행동을 물리적으로 차단합니다 [4,5,8,13].

(2) 필요한 기술과 수강자가 미리 준비할 것 (준비물 체크리스트)

- **필요한 기술**:
    - **정규표현식(Regex) 기반 자산 식별 기술 (`check-draft-metrics.mjs`)**: 마크다운 파일 내 `[사진]`, `[GIF]`, `[지도]`, `[링크카드]` 등의 마커를 인식하여 본문 글자 수/줄 수 계산과 분리 집계하는 기술입니다 [1,9].
    - **SmartEditor JSON 및 모바일 HTML DOM 파싱/대조 기술 (`verify-draft-format.mjs`, `verify-published.mjs`)**: 네이버 스마트에디터 임시저장 캡처 JSON에서 문단/미디어의 가운데 정렬, 형광 강조(#fff8b2) 위치, 지연 로딩 완료 상태를 대조하고, 발행본 모바일 HTML의 고유 ID(`SE-...`) 및 컴포넌트를 파싱하여 초안과 일대일 비교하는 기술입니다 [2,9].
    - **Claude Code 훅(Hooks) 자동화 및 종료 코드 통제**: `.claude/settings.json` 에 `PreToolUse`, `PostToolUse` 이벤트를 등록하고 스크립트의 `exit 2` (사전 차단) 및 `exit 1` (검수 실패) 종료 코드를 활용해 실행을 강제 차단하는 기술입니다 [8,13,12,10].
    - **버전 및 세션 관리 (Git / Agit / Handoff Bundle)**: 커밋·브랜치 단위 세션 이력 관리(Agit) 및 기준 버전, 적용 조건, 승인 상태를 구조화하여 남기는 인계 관리 기술입니다 [4,3].
- **수강자 준비물 체크리스트**:
    - [ ] **기초 터미널 환경**: Node.js, npm, jq, Bash 실행 환경 [1,13].
    - [ ] **에이전트 앱**: Claude Code CLI 및 설정 파일(`.claude/settings.json`) [12,13].
    - [ ] **버전 관리 도구**: Git 및 Agit (선택/권장) [3,13].
    - [ ] **실습용 파일 데이터**:
        - 마크다운 초안 원고 파일 (`draft.md`) [9,1].
        - 스마트에디터 임시저장 구조 캡처 및 서식 계획 파일 (`before.json`, `after.json`, `plan.json`) [2].
        - 대조 검증용 네이버 블로그 PostView 정보 (`blogId`, `logNo`) [9].
- ⚠️ **소스에 없는 항목**: 수강자가 미리 준비해야 할 **특정 PC 하드웨어 사양(GPU/RAM 스펙)**, **외부 체험단 플랫폼 API 정식 인증키**, **컴퓨터 비전/OCR 전용 라이브러리** 등은 소스에 수록되어 있지 않습니다.

(3) 실제 업무에 적용하고 테스트하는 방법

1. **초안 하드 규격 자동 검수 테스트**:

- CLI 터미널에서 `node check-draft-metrics.mjs draft.md --preset cafe-town --venue "가게명"` 을 구동합니다 [1,11].
- 업종별 프리셋 기준(글자수, 줄수, 사진수), 정보 블록 이모지(📍, ⏰, 📞, 🚗) 부착 여부, 안 먹은 메뉴 가격 나열(5줄 초과), 대가성 협찬 건 '내돈내산' 오표기, 요 종결 비중(40% 이하)을 정규식으로 검사하여 조건 미달 시 `exit 1` 로 차단되는지 확인합니다 [1].

2. **SmartEditor 임시저장 원고 서식/구조 대조 테스트**:
   - 임시저장 원고를 새 탭에서 연 뒤 캡처한 파일을 대상으로 `node verify-draft-format.mjs --before before.json --after after.json --plan plan.json` 을 실행합니다 [2].
   - 본문·미디어·소제목의 가운데 정렬, 사진/GIF 지연 로딩 로딩 완료 여부, 핵심어 형광 강조(#fff8b2) 위치 및 개수가 계획과 일치하는지 검증합니다 [2].
3. **Claude Code 훅(Hook) 연동 및 종료 코드 차단 테스트**:
   - `.claude/settings.json` 에 `PostToolUse` (`Edit|Write`) 매처를 연결하여 파일 수정 직후 메트릭 스크립트가 자동 실행되도록 설정합니다 [10,12].
   - `PreToolUse` (`Bash`) 매처에 `pre-commit-check.sh` 를 연결하고, 스크립트에서 실패 조건 발생 시 `exit 2` 와 `stderr` 차단 안내문이 반환되는지 테스트하여 명령 실행 및 커밋이 물리적으로 차단되는지 시험합니다 (`echo $?` 로 종료코드 2 확인) [13,10].
   - `/hooks` 명령어나 `claude --debug` 모드를 실행해 등록된 훅 스냅샷과 실행 기록을 검증합니다 [13,12].
4. **공개 발행본 HTML DOM 일대일 대조 테스트**:
   - `node verify-published.mjs draft.md --blog <blogId> --logno <logNo>` 를 구동합니다 [9].
   - 실제 네이버 모바일 PostView HTML을 파싱해 고유 컴포넌트 ID(`SE-...`)를 추출하고 초안 대비 사진, 소제목, 구분선, 지도, 표 수량 불일치, 문단 누락, 마크다운 별표(**`bold`**) 문명 유출을 대조합니다 [9].
   - 차이가 감지되면 `FAIL` 및 `exit 1` 이 반환되어 완료 보고를 하지 못하도록 차단합니다 [9].

(4) 성과를 측정하는 방법

- **수정 왕복 횟수 (Revision Cycles)**: 사람이 초안을 검토하고 AI와 수정 피드백을 주고받는 왕복 횟수가 단축되었는지 측정합니다 (기존 기록 없는 추정 5~6회에서 목표 2~3회로 단축) [4,5].
- **첫 발행본의 조건 충족률 (First-Pass Compliance Rate)**: 최초 생성/발행된 원고가 지도, 동영상, 필수 사진, 서식 등 캠페인 가이드라인을 100% 충족하는 비율을 측정합니다 [5].
- **검토 및 대조 소요 시간 (Verification Time)**: 수작업으로 글자 수, 사진 수, 지도를 대조하던 검토 시간이 자동 스크립트 실행(수 초 내 완료)으로 단축된 수치를 측정합니다 [4,5].
- **기한 준수율 (On-time Completion Rate)**: 예약 일자 및 리뷰 접수 마감 기한을 준수했는지 측정합니다 [4,5].
- **위반 차단 및 통제 성공률 (Guardrail Success Rate)**: 금지 규칙 위반, 별표 유출 발행, 미승인 커밋 시도가 훅 및 스크립트(`exit 2` , `exit 1`)에 의해 사전/사후에 100% 차단되었는지 측정합니다 [13,1,9].
- ⚠️ **소스에 없는 항목**: **ROI 금액 환산 공식, A/B 테스트에 따른 블로그 상위 노출 랭킹 점수 측정 알고리즘, 자동 발행에 따른 매출 증가율** 등은 소스에 명시되어 있지 않습니다.

참고 소스

참고 소스

1. check-draft-metrics.mjs.txt
2. verify-draft-format.mjs.txt
3. This New Open Source Tool Just Fixed Your Claude Code Workflow — https://www.youtube.com/watch?v=CHHEjuNBxoQ (게시 2026-09-22)
4. gpters-case-2.txt
5. gpters-case-1.txt
6. 클로드 코드 왕초보 2026년 완벽 가이드 (60분 정리) | Claude 입문 1단계부터 7단계까지 (팁 40개 및 스킬 사용법) — https://www.youtube.com/watch?v=2R-kUFv7GZg (게시 2026-10-15)
7. Automate actions with hooks - Claude Code Docs — https://code.claude.com/docs/en/hooks-guide
8. Claude Code hooks: a practical guide (with real config) - Scalably — https://scalably.io/blog/claude-code-hooks-guide
9. verify-published.mjs.txt
10. Hooks로 품질 게이트 자동화: 포맷·린트·테스트·위험 차단 - TILNOTE — https://tilnote.io/pages/69bfafd7edaa74850764739a
11. blog-post-format.md
12. Claude Code 공식문서 리뷰-참고자료(Reference)[5] : Hooks 참조(Hooks reference) - 갓대희의 작은공간 — https://goddaehee.tistory.com/498
13. 클로드 코드 훅 실전: 저장 뒤 검사와 커밋 전 차단 - 준이아빠블로그 — https://www.digitalmarketer.co.kr/class/claude-code-in-practice/claude-code-hooks-in-practice