

M01-02 · GIBALJA VISUAL MANUAL

터미널에서 프로젝트 다루기

한 줄 목표: Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.

1. 왜 지금 터미널에서 프로젝트 다루기인가

M01-02 · 01

배우는 이유: 터미널에서 프로젝트 다루기

- 1 인터넷 명령을 의미 없이 붙여넣음
- 2 현재 폴더를 확인하지 않음
- 3 space가 있는 경로를 quote하지 않음
- 4 stderr를 성공 메시지로 오해함
- 5 위험한 삭제 명령을 backup 없이 실행함

그림 1. 배우는 이유를 하나의 핵심 메시지로 정리합니다.

터미널은 빠르지만 현재 위치와 command 의미를 모르면 잘못된 파일을 변경하거나 위험한 명령을 그대로 실행할 수 있습니다.

지금 상태	문제	학습 뒤 변화
아이디어·도구 중심	인터넷 명령을 의미 없이 붙여넣음	터미널 명령 실행 기록
느낌으로 완료	재현 evidence 없음	다른 사람이 확인 가능한 기준
AI 결과 의존	사람 판단 경계 없음	원문·실행·승인 분리

2. 학습 전 확인과 완료 계약

M01-02 · 02

학습 계약: 터미널에서 프로젝트 다루기

- 1

오늘 목표 · 터미널에서 프로젝트 다루기
- 2

산출물 · 터미널 명령 실행 기록
- 3

완료 · 실습 90분 + 셀프 테스트 + 자기 설명

그림 2. 학습 계약을 하나의 핵심 메시지로 정리합니다.

항목	계약
대상	IT 비전공 성인 학습자
선수지식	M01-01 파일·경로·프로젝트 폴더 이해하기
예상시간	그림 학습 30분 + 실습 45분 + 복습 15분
산출물	터미널 명령 실행 기록
완료	실습 영수증 + 셀프 테스트 + 자기 설명

실습에는 실제 개인정보·비밀번호·API key를 넣지 않습니다.

3. 전체 구조 한눈에 보기

M01-02 · 03

전체 개념 지도: 터미널에서 프로젝트 다루기

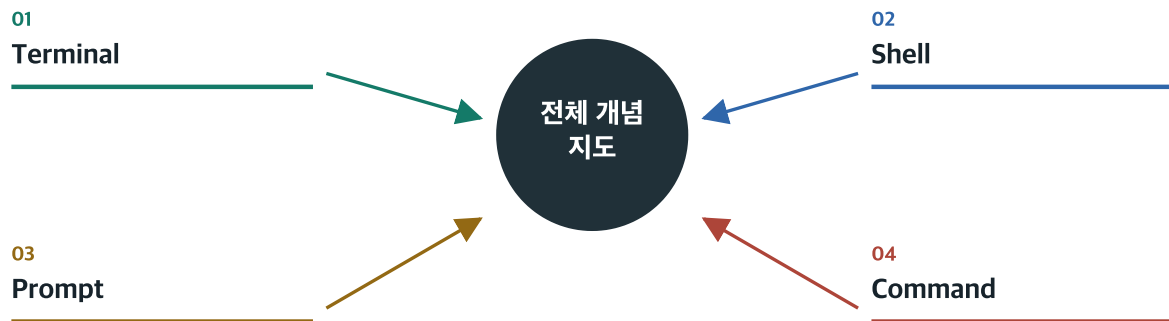


그림 3. 전체 개념 지도를 하나의 핵심 메시지로 정리합니다.

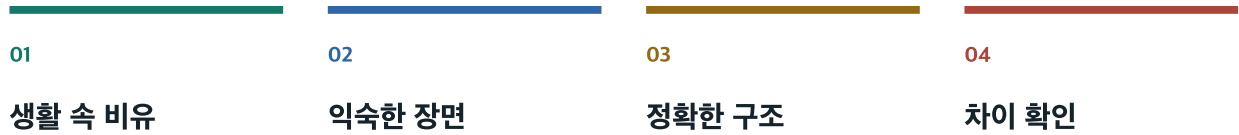
그림의 화살표를 먼저 따라가고, 각 단계에서 어떤 evidence가 남는지 찾습니다.

핵심 요소	학습 질문
Terminal	이 명령은 읽기인가 변경인가
Shell	어떤 경로가 대상인가
Prompt	관리자 권한이 필요한가
Command	실패 output은 어디에 있는가
Argument·Option	되돌릴 방법이 있는가
stdout·stderr·exit code	이 명령은 읽기인가 변경인가

4. 실생활 비유에서 정확한 구조로

M01-02 · 04

실생활 비유: 터미널에서 프로젝트 다루기



자기 말로 설명

그림 4. 실생활 비유를 하나의 핵심 메시지로 정리합니다.

터미널 명령은 택배 요청서와 같습니다. Command는 작업 종류, argument는 대상, option은 처리 조건, output과 exit code는 처리 영수증입니다. 비유는 시작점일 뿐입니다. 정확한 구조는 역할·입력·처리·상태·실패·책임으로 다시 분리합니다.

5. 핵심 요소 여섯 가지

M01-02 · 05

핵심 요소: 터미널에서 프로젝트 다루기

DO

① Terminal

② Shell



CHECK

③ Prompt

④ Command

⑤ Argument·Option

그림 5. 핵심 요소를 하나의 핵심 메시지로 정리합니다.

#	핵심 요소	확인 행동
1	Terminal	pwd·ls로 위치 확인
2	Shell	mkdir로 연습 폴더 생성
3	Prompt	printf로 파일 작성
4	Command	cp·mv로 복사·이름 변경
5	Argument·Option	실패 명령과 exit code 기록
6	stdout·stderr·exit code	pwd·ls로 위치 확인

6. 입력에서 산출물까지

M01-02 · 06

입력과 출력: 터미널에서 프로젝트 다루기

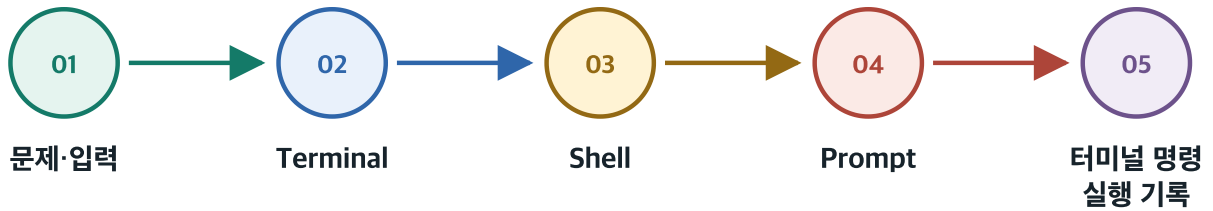


그림 6. 입력과 출력을 하나의 핵심 메시지로 정리합니다.

구간	입력	처리	출력
시작	사용자·문제·현재 상태	prompt와 현재 위치를 읽는다	명령 전 목적
중간	가정·범위·도구	읽기 전용 명령부터 실행한다	정상 output
완료	검증 결과·한계	변경 전 diff·backup을 확인한다	터미널 명령 실행 기록

30초 확인: 이 명령은 읽기인가 변경인가? 답을 말한 뒤 다음 장으로 이동하세요.

7. 누가 무엇을 책임하는가

M01-02 · 07

사람과 역할: 터미널에서 프로젝트 다루기

1 학습자

2 사용자

3 기획 책임

4 개발·AI

5 운영·검수

그림 7. 사람과 역할을 하나의 핵심 메시지로 정리합니다.

역할	책임	하면 안 되는 일
학습자	관찰·실행·기록	모르는 결과를 성공으로 표시
기발자	문제·범위·완료 기준	기술·법률 승인을 대신함
개발자·AI	구현·설명·검사 보조	최종 결정 자동 확정
Owner	승인·위험·운영 책임	evidence 없는 승인
사용자	실제 과업과 피드백	합성 persona로 대체

8. 헛갈리는 경계 분리하기

M01-02 · 08

경계와 책임: 터미널에서 프로젝트 다루기

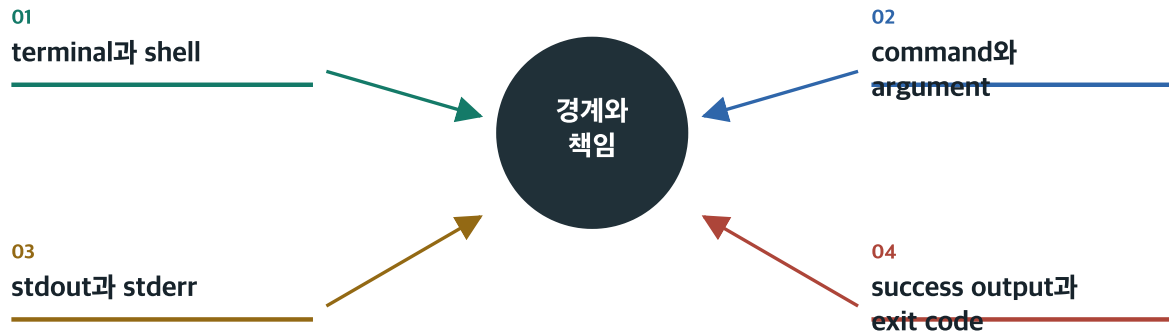


그림 8. 경계와 책임을 하나의 핵심 메시지로 정리합니다.

구분할 경계	왜 분리하나
terminal과 shell	같이 보이지만 책임·검증·실패 영향이 다르기 때문
command와 argument	같이 보이지만 책임·검증·실패 영향이 다르기 때문
stdout과 stderr	같이 보이지만 책임·검증·실패 영향이 다르기 때문
success output과 exit code	같이 보이지만 책임·검증·실패 영향이 다르기 때문
일반 권한과 관리자 권한	같이 보이지만 책임·검증·실패 영향이 다르기 때문

9. 정상 workflow 다섯 단계

M01-02 · 09

정상 workflow: 터미널에서 프로젝트 다루기

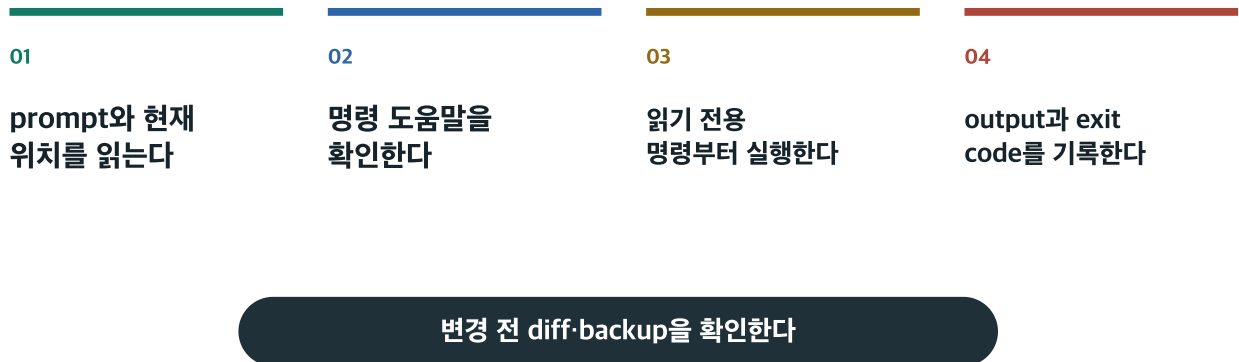


그림 9. 정상 workflow을 하나의 핵심 메시지로 정리합니다.

순서	행동	남길 evidence
1	prompt와 현재 위치를 읽는다	명령 전 목적
2	명령 도움말을 확인한다	정확한 command
3	읽기 전용 명령부터 실행한다	정상 output
4	output과 exit code를 기록한다	실패 output
5	변경 전 diff·backup을 확인한다	되돌리기 결과

10. 개발자가 결정할 다섯 질문

M01-02 · 10

판단 기준: 터미널에서 프로젝트 다루기

DO

- 1 이 명령은 읽기인가 변경인가
- 2 어떤 경로가 대상인가



CHECK

- 3 관리자 권한이 필요한가
- 4 실패 output은 어디에 있는가
- 5 되돌릴 방법이 있는가

그림 10. 판단 기준을 하나의 핵심 메시지로 정리합니다.

질문	선택 evidence
이 명령은 읽기인가 변경인가	명령 전 목적
어떤 경로가 대상인가	정확한 command
관리자 권한이 필요한가	정상 output
실패 output은 어디에 있는가	실패 output
되돌릴 방법이 있는가	되돌리기 결과

11. 좋은 예: 작은 evidence가 이어지는 경우

M01-02 · 11

좋은 예: 터미널에서 프로젝트 다루기

- 1 pwd·ls로 위치 확인
- 2 mkdir로 연습 폴더 생성
- 3 printf로 파일 작성
- 4 cp·mv로 복사·이름 변경
- 5 실패 명령과 exit code 기록

그림 11. 좋은 예를 하나의 핵심 메시지로 정리합니다.

좋은 예는 완벽한 문서가 아니라 질문·행동·결과·한계가 이어지는 작은 기록입니다.

행동	좋은 기록
pwd·ls로 위치 확인	명령 전 목적
mkdir로 연습 폴더 생성	정확한 command
printf로 파일 작성	정상 output
cp·mv로 복사·이름 변경	실패 output
실패 명령과 exit code 기록	되돌리기 결과

12. 나쁜 예: 그럴듯하지만 재현되지 않는 경우

M01-02 · 12

나쁜 예: 터미널에서 프로젝트 다루기

- 1 인터넷 명령을 의미 없이 붙여넣음
- 2 현재 폴더를 확인하지 않음
- 3 space가 있는 경로를 quote하지 않음
- 4 stderr를 성공 메시지로 오해함
- 5 위험한 삭제 명령을 backup 없이 실행함

그림 12. 나쁜 예들 하나의 핵심 메시지로 정리합니다.

나쁜 예	왜 위험한가	바꿀 행동
인터넷 명령을 의미 없이 붙여넣음	원인·범위·책임을 잃음	pwd·ls로 위치 확인
현재 폴더를 확인하지 않음	원인·범위·책임을 잃음	mkdir로 연습 폴더 생성
space가 있는 경로를 quote하지 않음	원인·범위·책임을 잃음	printf로 파일 작성
stderr를 성공 메시지로 오해함	원인·범위·책임을 잃음	cp·mv로 복사·이름 변경
위험한 삭제 명령을 backup 없이 실행함	원인·범위·책임을 잃음	실패 명령과 exit code 기록

30초 확인: 어떤 경로가 대상인가? 답을 말한 뒤 다음 장으로 이동하세요.

13. 오류·오해·실패 위치 지도

M01-02 · 13

실패 위치 지도: 터미널에서 프로젝트 다루기

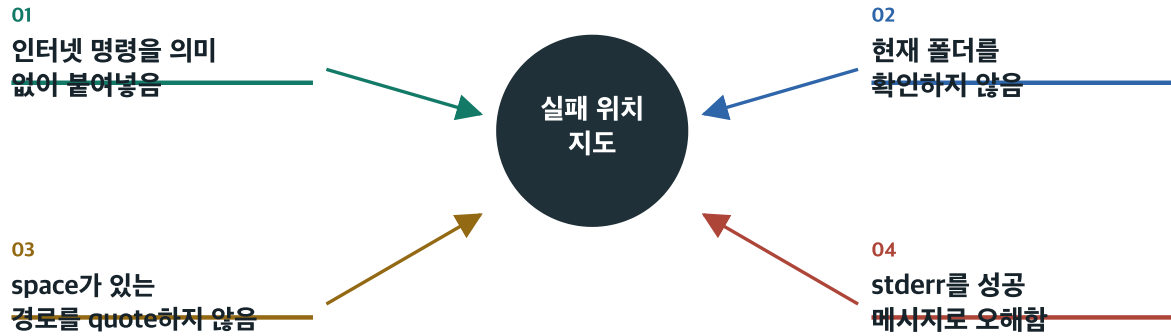


그림 13. 실패 위치 지도를 하나의 핵심 메시지로 정리합니다.

실패 신호	먼저 확인	복구
인터넷 명령을 의미 없이 붙여넣음	terminal과 shell	pwd·ls로 위치 확인
현재 폴더를 확인하지 않음	command와 argument	mkdir로 연습 폴더 생성
space가 있는 경로를 quote하지 않음	stdout과 stderr	printf로 파일 작성
stderr를 성공 메시지로 오해함	success output과 exit code	cp·mv로 복사·이름 변경
위험한 삭제 명령을 backup 없이 실행함	일반 권한과 관리자 권한	실패 명령과 exit code 기록

14. 보안·개인정보·변경 안전

M01-02 · 14

안전·개인정보: 터미널에서 프로젝트 다루기

01

최소 권한

02

실제 정보 금지

03

변경 전 확인

04

로그·
evidence

복구 가능성

그림 14. 안전·개인정보를 하나의 핵심 메시지로 정리합니다.

안전 질문	최소 통제
실제 정보인가	합성 값·redaction
변경 command인가	목적·대상·backup
권한이 필요한가	최소 권한·사람 승인
외부로 전송되는가	destination·보존·비용 확인
실패 뒤 돌아갈 수 있는가	diff·history·restore

15. 도구와 기록에서 무엇을 볼 것인가

M01-02 · 15

도구에서 찾기: 터미널에서 프로젝트 다루기

DO

- 1 pwd
- 2 ls -la



CHECK

- 3 mkdir -p practice
- 4 printf 'hello\n' > practice/note.txt
- 5 cp practice/note.txt practice/note-copy.txt

그림 15. 도구에서 찾기를 하나의 핵심 메시지로 정리합니다.

도구 화면에서는 큰 성공 문구보다 현재 위치·대상·version·output·exit 상태를 먼저 찾습니다.

```
pwd
ls -la
mkdir -p practice
printf 'hello\n' > practice/note.txt
cp practice/note.txt practice/note-copy.txt
```

16. 실습 1: 관찰하고 표시하기

M01-02 · 16

실습 1 · 관찰: 터미널에서 프로젝트 다루기

- 1 pwd·ls로 위치 확인
- 2 mkdir로 연습 폴더 생성
- 3 printf로 파일 작성
- 4 cp·mv로 복사·이름 변경
- 5 실패 명령과 exit code 기록

그림 16. 실습 1 · 관찰을 하나의 핵심 메시지로 정리합니다.

실습 순서	행동	완료 표시
1	pwd·ls로 위치 확인	☐ 관찰 · ☐ 기록 · ☐ 확인
2	mkdir로 연습 폴더 생성	☐ 관찰 · ☐ 기록 · ☐ 확인
3	printf로 파일 작성	☐ 관찰 · ☐ 기록 · ☐ 확인
4	cp·mv로 복사·이름 변경	☐ 관찰 · ☐ 기록 · ☐ 확인
5	실패 명령과 exit code 기록	☐ 관찰 · ☐ 기록 · ☐ 확인

17. 실습 2: 내 사례 작성하기

M01-02 · 17

실습 2 · 작성: 터미널에서 프로젝트 다루기

- 1 pwd·ls로 위치 확인
- 2 mkdir로 연습 폴더 생성
- 3 printf로 파일 작성
- 4 cp·mv로 복사·이름 변경
- 5 실패 명령과 exit code 기록

그림 17. 실습 2 · 작성을 하나의 핵심 메시지로 정리합니다.

[02_Labs/G01_Development_Environment/L01-02_terminal-command-rehearsal.html](#) 을 열고 02 수행 화면에서 내 사례를 작성합니다. 정답처럼 보이는 문장보다 선택 이유와 남은 질문을 적습니다.

18. Evidence package 만들기

M01-02 · 18

Evidence 남기기: 터미널에서 프로젝트 다루기



그림 18. Evidence 남기기를 하나의 핵심 메시지로 정리합니다.

Evidence	필수 내용	검수 질문
명령 전 목적	pwd·ls로 위치 확인	다른 사람이 같은 판단을 재현하는가
정확한 command	mkdir로 연습 폴더 생성	다른 사람이 같은 판단을 재현하는가
정상 output	printf로 파일 작성	다른 사람이 같은 판단을 재현하는가
실패 output	cp·mv로 복사·이름 변경	다른 사람이 같은 판단을 재현하는가
되돌리기 결과	실패 명령과 exit code 기록	다른 사람이 같은 판단을 재현하는가

30초 확인: 관리자 권한이 필요한가? 답을 말한 뒤 다음 장으로 이동하세요.

19. 개발자·외주사에게 확인할 질문

M01-02 · 19

개발자에게 물을 질문: 터미널에서 프로젝트 다루기

- 1 어느 shell에서 실행하는가
- 2 이 option은 무엇을 바꾸는가
- 3 명령이 성공했다는 기준은 무엇인가
- 4 동일 작업의 안전한 dry run이 있는가
- 5 취소·복구 방법은 무엇인가

그림 19. 개발자에게 물을 질문을 하나의 핵심 메시지로 정리합니다.

확인 질문	좋은 답의 증거
어느 shell에서 실행하는가	명령 전 목적
이 option은 무엇을 바꾸는가	정확한 command
명령이 성공했다는 기준은 무엇인가	정상 output
동일 작업의 안전한 dry run이 있는가	실패 output
취소·복구 방법은 무엇인가	되돌리기 결과

20. AI에 안전하게 작업 요청하기

M01-02 · 20

AI에 작업 요청하기: 터미널에서 프로젝트 다루기

DO

1 목적

2 맥락

CHECK

3 입력

4 완료 기준

5 금지·한계



그림 20. AI에 작업 요청하기를 하나의 핵심 메시지로 정리합니다.

AI 요청은 다음 구조로 씁니다.

목적: 터미널 명령 실행 기록 초안을 만든다.

맥락: Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.

입력: 아래 합성 사례와 확인된 사실만 사용한다.

완료 기준: 표의 모든 field, 정상·실패·한계, TBR를 포함한다.

금지: 실제 정보 추정, 확인하지 않은 성공 단정, 위험한 command 자동 실행.

출력 뒤: 누락·가정·검증 방법을 별도 목록으로 적는다.

21. AI 결과를 사람이 검수하기

M01-02 · 21

AI 결과 검수: 터미널에서 프로젝트 다루기



그림 21. AI 결과 검수를 하나의 핵심 메시지로 정리합니다.

검수	질문	실패 시
원문	입력에 실제 존재하는가	삭제·수정
범위	이번 manual의 경계 안인가	TBR로 이동
정상	expected를 재현했는가	실행 evidence
실패	거부·오류·복구가 있는가	case 추가
승인	사람 owner가 이유를 남겼는가	확정 금지

22. 실습 화면에서 전체 지도 찾기

YEONCORE · GIBALJA LAB

터미널 명령 리허설

M01-02 · v0.1.0

합성 사례 · 로컬 저장 · 외부 전송 없음

01 이해

02 수행

03 복습

전체 구조 이해

학습용 · SELF REVIEW

Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.

CORE 01

Terminal

terminal과 shell를 함께 확인합니다.

CORE 02

Shell

command와 argument를 함께 확인합니다.

CORE 03

Prompt

stdout과 stderr를 함께 확인합니다.

CORE 04

Command

success output과 exit code를 함께 확인합니다.

CORE 05

Argument·Option

일반 권한과 관리자 권한을 함께 확인합니다.

CORE 06

stdout·stderr·exit code

terminal과 shell를 함께 확인합니다.

경계: 이 화면의 안료는 학습 산출물 작성 준비이며 실제 서비스·보안·운영 승인이 아닙니다.

실습 화면 1. 핵심 요소와 경계를 desktop에서 찾습니다.

23. 실습 화면에서 단계별 수행하기

YEONCORE · GIBALJA LAB

터미널 명령 리허설

M01-02 · v0.1.0

합성 사례 · 로컬 저장 · 외부 전송 없음

01 이해

02 수행

03 복습

단계별 수행

학습용 · SELF REVIEW

Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.

1

pwd·ls로 위치 확인

2

mkdir로 연습 폴더 생성

3

printf로 파일 작성

4

cp·mv로 복사·이름 변경

5

실패 명령과 exit code 기록

내 사례와 evidence

관찰한 사실, 선택 이유, 남은 질문을 적으세요.

학습 영수증 만들기

초기화

실습 화면 2. 다섯 단계를 체크하고 내 사례 evidence를 작성합니다.

24. 모바일에서 복습 영수증 읽기

YEONCORE · GIBALJA LAB

터미널 명령 리허설

M01-02 · v0.1.0

합성 사례 · 로컬 저장 · 외부 전송 없음

01 이해02 수행03 복습

학습 영수증·복습

Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.

학습용 · SELF REVIEW

터미널 명령 실행 기록 · 학습 영수증

Manual

M01-02 · 터미널에서 프로젝트 다루기

완료 단계

5 / 5

남긴 기록

변경 전에 현재 경로와 명령 목적을 확인하고 stdout·stderr·exit code를 기록했다.

Evidence

명령 전 목적 · 정확한 command · 정상 output · 실패 output · 되돌리기 결과

다음

M01-03 · 개발 도구와 실행 환경 점검하기

셀프 질문: 다른 사람이 이 영수증과 터미널 명령 실행 기록만 보고 같은 판단을 재현할 수 있나요?

실습 화면 3. 390×844에서 완료 단계·기록·다음 매뉴얼을 복습합니다.

30초 확인: 실패 output은 어디에 있는가? 답을 말한 뒤 다음 장으로 이동하세요.

25. 90분 실습 워크북

시간	행동	산출물
0~10분	전체 그림·경계 읽기	한 문장 설명
10~25분	핵심 요소·정상 흐름	표시된 concept map
25~45분	실습 화면 01·02	5단계 수행
45~60분	실패·안전·질문	오류·TBR
60~75분	템플릿 완성	터미널 명령 실행 기록
75~90분	셀프 테스트·영수증	오답 복습

26. 터미널 명령 실행 기록 템플릿

재사용 템플릿: `03_Templates/T01-02_terminal-command-log.md` 빈칸을 한 번에 채우지 말고 관찰한 사실 → 선택 이유 → 미정 사항 순서로 작성합니다.

27. 기초 통합 용어집 300

통합 기초 용어집 `04_Glossary/GLOSSARY_foundation_orientation_environment.md` 의 810, 1314 묶음을 먼저 봅니다. 용어를 외우는 대신 내 실습 화면과 산출물에서 실제 예를 찾습니다.

28. 공식 자료와 적용 경계

아래 링크는 현재 원리와 도구 동작을 확인하는 1차 자료입니다. 링크 사용은 적합성·인증·운영 승인을 뜻하지 않습니다. 확인 기준일은 2026-07-17입니다.

공식 자료	확인할 원리	적용 경계
Apple Terminal Guide	macOS terminal·shell·command의 기본 역할	학습용 요약이며 실제 환경·사용자 검증 추가
Apple Execute commands	대화형 command와 script 실행 방식	학습용 요약이며 실제 환경·사용자 검증 추가
Bash Reference Manual	command·redirection·pipeline·exit status의 정확한 의미	학습용 요약이며 실제 환경·사용자 검증 추가
VS Code Terminal Basics	workspace root에서 integrated terminal을 사용하는 방식	학습용 요약이며 실제 환경·사용자 검증 추가

29. 셀프 테스트 30

1. 이 매뉴얼의 최종 산출물은 무엇인가요?

정답 보기

정답: 터미널 명령 실행 기록입니다.

2. 이 주제를 배우는 가장 직접적인 이유는 무엇인가요?

정답 보기

정답: 터미널은 빠르지만 현재 위치와 command 의미를 모르면 잘못된 파일을 변경하거나 위험한 명령을 그대로 실행할 수 있습니다.

3. 전체 지도에서 구분할 여섯 핵심 요소는 무엇인가요?

정답 보기

정답: Terminal · Shell · Prompt · Command · Argument·Option · stdout·stderr·exit code입니다.

4. 정상 workflow의 첫 행동은 무엇인가요?

정답 보기

정답: prompt와 현재 위치를 읽는다입니다.

5. 정상 workflow의 마지막 행동은 무엇인가요?

정답 보기

정답: 변경 전 diff·backup을 확인한다입니다.

6. 가장 먼저 답할 의사결정 질문은 무엇인가요?

정답 보기

정답: 이 명령은 읽기인가 변경인가입니다.

7. 완료 전에 확인할 마지막 결정은 무엇인가요?

정답 보기

정답: 되돌릴 방법이 있는가입니다.

8. 대표적인 실패 한 가지는 무엇인가요?

정답 보기

정답: 인터넷 명령을 의미 없이 붙여넣음입니다.

9. 또 다른 실패 신호는 무엇인가요?

정답 보기

정답: 현재 폴더를 확인하지 않음입니다.

10. 왜 정상 경로만 기록하면 부족한가요?

정답 보기

정답: space가 있는 경로를 quote하지 않음 같은 실패와 대안을 놓치기 때문입니다.

11. 첫 번째 책임 경계는 무엇인가요?

정답 보기

정답: terminal과 shell를 구분하는 것입니다.

12. 학습용 결과와 실제 승인을 왜 분리하나요?

정답 보기

정답: 학습 evidence가 실제 사용자·데이터·보안·운영 책임까지 자동으로 승인하지 않기 때문입니다.

13. 실습의 첫 단계는 무엇인가요?

정답 보기

정답: pwd·ls로 위치 확인입니다.

14. 실습의 마지막 단계는 무엇인가요?

정답 보기

정답: 실패 명령과 exit code 기록입니다.

15. 첫 번째 evidence는 무엇인가요?

정답 보기

정답: 명령 전 목적입니다.

16. 마지막 evidence는 무엇인가요?

정답 보기

정답: 되돌리기 결과입니다.

17. 개발자에게 가장 먼저 물을 질문은 무엇인가요?

정답 보기

정답: 어느 shell에서 실행하는가입니다.

18. AI 요청에 반드시 포함할 다섯 요소는 무엇인가요?

정답 보기

정답: 목적·맥락·입력·완료 기준·금지/한계입니다.

19. AI 결과를 바로 확정하면 안 되는 이유는 무엇인가요?

정답 보기

정답: AI는 누락·과잉 단정·잘못된 경로·위험한 command를 만들 수 있으므로 원문과 실행 evidence로 사람이 검수해야 합니다.

20. 좋은 검수는 정상 결과만 보나요?

정답 보기

정답: 아닙니다. 정상·실패·권한·경계·되돌리기를 함께 확인합니다.

21. 실습 기록에 command나 행동 전 목적을 쓰는 이유는 무엇인가요?

정답 보기

정답: 결과가 예상과 다를 때 무엇을 시험했는지 되짚고 위험한 행동을 줄이기 위해서입니다.

22. 색상만으로 PASS·FAIL을 구분하면 안 되는 이유는 무엇인가요?

정답 보기

정답: 인쇄·저대비·색각 차이에서도 상태를 알 수 있도록 문자·아이콘·선으로 함께 표시해야 하기 때문입니다.

23. 실제 개인정보나 secret을 실습에 넣어도 되나요?

정답 보기

정답: 아닙니다. 합성 값과 placeholder를 사용하고 secret은 환경변수 등 분리된 경로로 다룹니다.

24. 한 번에 여러 원인을 바꾸면 왜 안 되나요?

정답 보기

정답: 어떤 변경이 결과를 바꿨는지 알 수 없으므로 가설 하나씩 시험해야 합니다.

25. 완료 기준은 느낌으로 적어도 되나요?

정답 보기

정답: 아닙니다. 다른 사람이 관찰·실행·비교할 수 있는 evidence로 적어야 합니다.

26. 실패는 항상 학습 실패인가요?

정답 보기

정답: 아닙니다. 예상한 실패를 안전하게 재현하고 원인·대안·복구를 설명하면 중요한 학습 evidence입니다.

27. 공식 출처를 남기는 이유는 무엇인가요?

정답 보기
정답: 변동 가능한 도구·단계·명령의 현재 의미를 다시 확인하고 교육 설명의 적용 경계를 밝히기 위해서입니다.

28. 모바일 화면에서도 확인할 핵심은 무엇인가요?

정답 보기
정답: 제목·입력·행동·상태·완료 evidence가 가로 넘침 없이 같은 순서로 읽히는지 확인합니다.

29. 이 매뉴얼을 완료했다는 가장 좋은 증거는 무엇인가요?

정답 보기
정답: 터미널 명령 실행 기록과 실습 영수증을 다른 사람이 보고 같은 판단을 재현하는 것입니다.

30. 다음 매뉴얼로 넘어가기 전 한 문장으로 무엇을 설명해야 하나요?

정답 보기
정답: 터미널에서 프로젝트 다루기의 핵심 구조와 한계를 자기 사례로 설명해야 합니다.

30. 한 장 요약과 다음 매뉴얼 M01-03

기억할 것	한 문장
목적	Prompt·command·argument·option·표준 출력·exit code를 읽고 안전한 확인 중심 terminal workflow를 실행합니다.
핵심	prompt와 현재 위치를 읽는다 → 명령 도움말을 확인한다 → 읽기 전용 명령부터 실행한다 → output과 exit code를 기록한다 → 변경 전 diff·backup을 확인한다
산출물	터미널 명령 실행 기록
이전	M01-01 파일·경로·프로젝트 폴더 이해하기
다음	M01-03 · 개발 도구와 실행 환경 점검하기

완료 확인: 터미널 명령 실행 기록과 실습 영수증을 보지 않고 핵심 흐름·실패·경계를 설명한 뒤 다음 매뉴얼로 이동합니다.