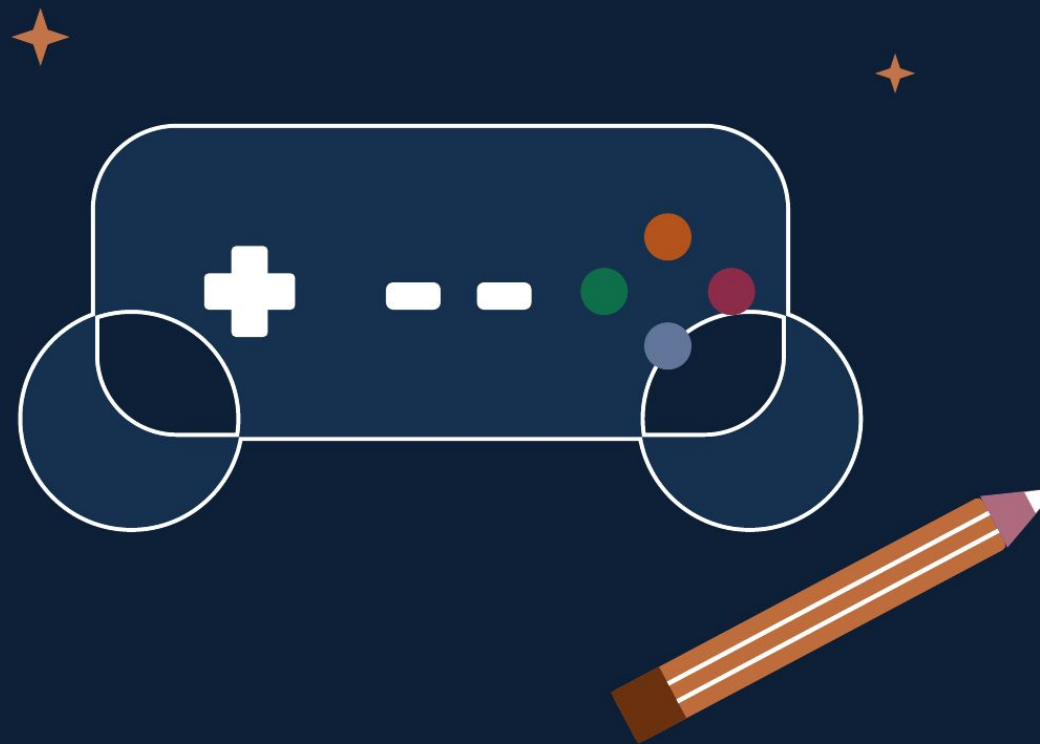


교사 게임리터러시 · 연수 자료

AI 에이전트와 함께하는

교실 맞춤형 게임 제작 마스터

게임을 읽는 눈에서
우리 반 게임을 만드는 손까지



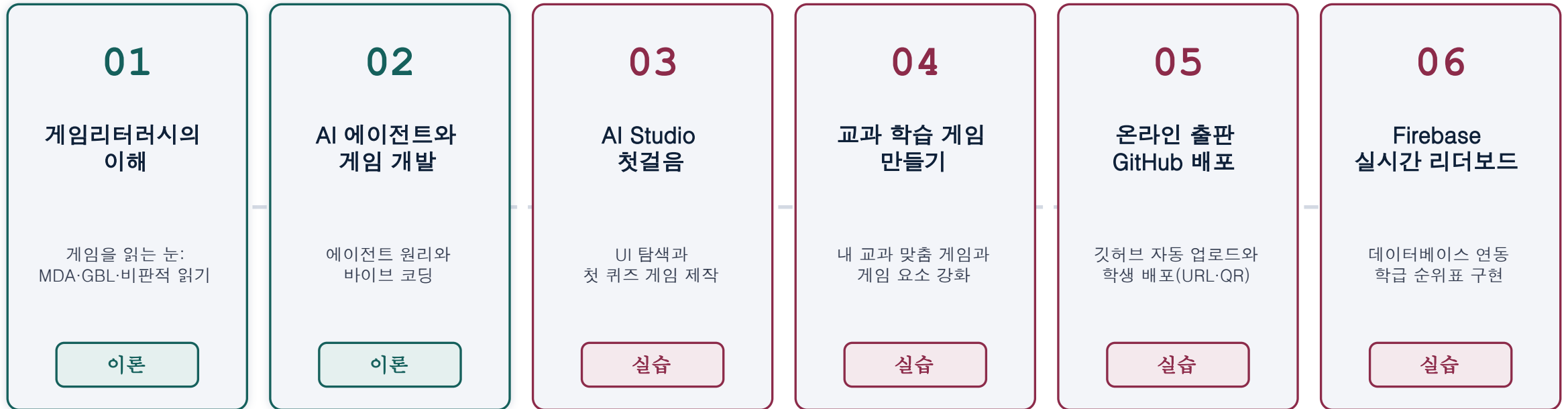
게임문화재단

게임문화교육원
Game Culture Education Institute

(사)한국창의정보문화학회

전체 여정 지도 — 6개의 월드

본 교재는 차시가 아닌 주제 중심의 6개 단원(월드)으로 구성된 모듈형 교재입니다. 각 단원은 독립적으로 완결되므로 연수 시간과 참여자 수준에 따라 순서와 배분을 유연하게 조정할 수 있습니다. 1~2단원은 이론, 3~6단원은 Google AI Studio 기반 실습입니다.



최종 산출물 연수를 마치면 교사 1인당 ① 교과 내용을 담은 학습 웹게임 ② 학생이 링크·QR로 접속하는 배포본 ③ 실시간 리더보드가 작동하는 완성작을 갖게 됩니다.

게이미터러시의 이해

게임을 읽고, 해석하고, 만드는 교사의 눈

게임은 학생들이 또래 관계와 성취와 이야기를 먼저 만나는 생활 공간이자, 규칙과 서사와 상호작용이 얹힌 하나의 복합 텍스트입니다. 이 단원에서는 그 텍스트를 읽는 언어를 익힙니다. 호이징가와 카이와의 놀이 이론에서 출발해 게임 분석의 표준 틀인 MDA를 익히고, 게임기반학습과 게이미피케이션을 구분하며, 보상 루프와 확률형 아이템 같은 상업적 설계를 해부합니다. 여기서 익힌 언어가 3단원부터 게임을 직접 만들 때의 기획 언어가 됩니다.

1

교재 1단원

1단원 학습 퀘스트

1단원을 마치면 다음 네 가지를 할 수 있어야 합니다. 각 퀘스트는 단원 말미의 정리 슬라이드에서 스스로 점검합니다.



QUEST 1

게임리터러시의 개념과 3층위(읽기·해석·창작)를 설명할 수 있다

리터러시 개념이 문자에서 게임으로 확장되어 온 맥락과 함께 설명할 수 있어야 합니다.



QUEST 2

MDA 프레임워크로 임의의 게임을 분석할 수 있다

규칙(M) → 상호작용(D) → 재미(A)의 연결 고리를 실제 게임에서 짚어낼 수 있어야 합니다.



QUEST 3

GBL과 게이미피케이션을 구분하고 내 수업의 적용점을 찾는다

두 접근의 정의·사례·한계를 대비하고 자신의 교과에 맞는 조합을 구상합니다.



QUEST 4

게임의 설득 구조(보상 루프·확률형 아이템 등)를 비판적으로 읽는다

금지가 아닌 해부의 관점에서 학생과 함께 분석하는 수업을 설계할 수 있어야 합니다.



CLEAR 보상

**게임을 보는
교사의 눈**

학생 문화의 절반을 차지하는
게임 세계를 전문가의 렌즈로
읽어내게 됩니다

왜 게임인가 — 학생들이 사는 세계

게임 교육의 출발점은 통계입니다. 한국콘텐츠진흥원이 전국 10~69세 1만 명을 조사한 결과, 전 국민 게임 이용률은 50.2%인 반면 10대는 80.9%에 달했습니다. 성인에게 게임이 여러 여가 중 하나라면, 학생에게 게임은 또래 관계·성취 경험·이야기 소비가 일어나는 생활 공간 그 자체입니다.

80.9%

10대 청소년 게임 이용률

전 국민 이용률 50.2%의 약 1.6배입니다. 전체 이용률이 3년 연속 하락하는 중에도 10대는 여전히 10명 중 8명이 게임을 즐기는, 압도적 이용층으로 남아 있습니다.

조사: 전국 10~69세 일반 국민 1만 명 대상



게임은 학생들의 제1언어

학생들은 친구·성취·서사·경쟁이라는 인간의 근본 경험을 게임 안에서 먼저 만납니다. 게임 용어와 문법은 이미 또래 문화의 공용어입니다.



언어를 모르면 세계를 못 읽는다

학생이 어떤 게임을 왜 좋아하는지 읽지 못하는 교사는 학생 문화의 절반을 놓칩니다. 게임 이해는 생활지도와 상담의 자원이기도 합니다.



읽기를 넘어 쓰기로

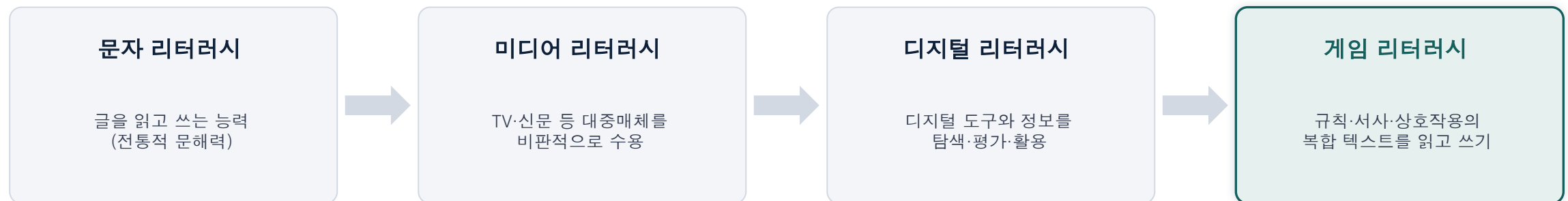
이 언어를 수업의 도구로 바꾸는 것이 게임리터러시 교육의 목표입니다. 본 교재는 그 마지막 단계, 즉 교사가 직접 만드는 단계까지 안내합니다.

게이미터러시란 무엇인가

리터러시(literacy)는 본래 글을 읽고 쓰는 능력을 가리켰지만, 미디어 환경이 변할 때마다 그 대상을 넓혀 왔습니다. 게임 연구자 James Paul Gee는 게임이 고유한 기호 체계와 학습 원리를 지닌 텍스트이며, 따라서 읽는 법과 쓰는 법을 배워야 할 대상이라고 주장했습니다.

게이미터러시(Game Literacy)란, 게임을 **비판적으로 이해** 하고, 그 의미와 구조를 **해석** 하며, 자신의 목적에 맞게 **만들어 표현** 할 수 있는 능력을 말합니다.

리터러시 개념의 확장



게임은 글·그림·소리·규칙·상호작용이 결합된 가장 복합적인 텍스트입니다. 문자 텍스트처럼 게임에도 문법이 있고, 그 문법을 아는 사람만이 읽기를 넘어 쓰기 (제작)에 이를 수 있습니다.

게이미터러시의 세 층위 — 레벨업 구조

게이미터러시는 세 층위가 차례로 쌓이는 능력입니다. 문자 문해력이 "읽기 → 독해 → 작문"으로 발달하듯, 게이미터러시도 "플레이 → 분석 → 제작"의 계단을 오릅니다. 본 교재의 목표 지점은 가장 높은 층위인 창작(LV.3)입니다.

LV.1



읽기 · 이해

규칙과 장르 관습, 조작 체계를 이해하며 게임을 온전히 플레이하는 단계입니다. 튜토리얼을 읽어내고, 목표를 파악하고, 실패에서 전략을 수정하는 것 자체가 하나의 문해 활동입니다.

LV.2



해석 · 비판

왜 재미있는지, 어떤 장치가 이용자를 붙잡는지 분석·평가하는 단계입니다. MDA 같은 분석 틀로 재미의 구조를 언어화하고, 확률형 아이템 등 상업적 설계를 비판적으로 검토합니다.

LV.3



쓰기 · 창작

자신의 목적에 맞는 게임을 직접 만드는 단계입니다. 교사에게 그 목적은 수업입니다. 학습 목표를 게임 규칙으로 번역하는 이 단계가 게이미터러시의 완성이며, 3~6단원 실습의 내용입니다.

본 교재의 종착점 = LV.3 창작. 과거에는 코딩이 이 계단을 가로막았지만, AI 에이전트가 그 장벽을 없앴습니다(2단원).

이론적 뿌리 — 놀이를 사유한 두 고전

게임 연구의 뿌리에는 두 권의 고전이 있습니다. 두 학자 모두 놀이를 시간 낭비가 아닌 인간 문명의 근본 활동으로 격상시켰고, 오늘날 게임 교육을 정당화하는 이론적 토대가 되었습니다.

요한 호이징가

『호모 루덴스』 (1938)

인간 = 놀이하는 존재(Homo Ludens)

놀이는 문화의 부산물이 아니라 문화를 만들어낸 원천입니다. 법정의 공방, 예술의 형식, 학문의 논쟁 모두 그 뿌리에 규칙 있는 겨루기, 곧 놀이가 있습니다.

또한 놀이는 일상과 구분되는 시공간, 이른바 "마법의 원(magic circle)" 안에서 고유한 규칙으로 진행됩니다. 게임 속 실패가 안전한 이유이며, 교실 게임이 "안전한 도전"을 제공할 수 있는 근거입니다.

수업 함의: 게임은 시간 낭비가 아니라 인간 학습의 원형이다.

로제 카이와

『놀이와 인간』 (1958) — 놀이의 4분류

아곤 (경쟁)

실력을 겨루어 승부를 가리는 놀이.
달리기 시합, 바둑, 대전 게임, 퀴즈 대결

알레아 (운)

결과를 우연에 맡기는 놀이.
주사위, 제비뽑기, 카드 뽑기

미미크리 (모방)

다른 존재가 되어 보는 놀이.
역할놀이, 연극, RPG, 아바타 꾸미기

일링크스 (현기증)

감각의 혼란 자체를 즐기는 놀이.
회전목마, 롤러코스터, 스피드 질주

좋은 학습 게임은 대개 두 가지 이상의 놀이 요소를 섞습니다. 예컨대 퀴즈 대결(아곤)에 뽑기 보상(알레아)을 더하면 실력 차가 있어도 모두가 끝까지 참여합니다.

게임 분석의 렌즈 — MDA 프레임워크

MDA는 게임 연구자 Hunicke, LeBlanc, Zubek이 2004년 제안한 게임 분석·설계의 표준 틀로, 게임을 메커닉스(규칙) – 다이내믹스(상호작용) – 미학(재미 경험)의 세 겹으로 나누어 봅니다. 복잡해 보이는 게임도 이 세 겹으로 분해하면 구조가 선명하게 드러납니다.

M 메커닉스 Mechanics

게임을 구성하는 규칙과 부품

점프, 점수, 타이머, 아이템, 승리 조건처럼 설계자가 직접 정하는 요소들입니다. 게임의 "법전"에 해당합니다.

D 다이내믹스 Dynamics

플레이 중 실제로 벌어지는 일

규칙이 플레이어와 만나 만들어내는 행동 패턴입니다. 눈치 싸움, 역전극, 협력과 배신은 규칙에 적혀 있지 않지만 규칙에서 태어납니다.

A 미학 Aesthetics

플레이어가 느끼는 재미 경험

성취감, 몰입, 짜릿함, 우정처럼 플레이어의 마음에 남는 정서적 반응입니다. 게임의 최종 목적지입니다.

두 개의 시점 디자이너는 M에서 출발해 규칙을 설계함으로써 재미(A)를 빚어내고, 플레이어는 A에서 출발해 재미를 먼저 느낀 뒤에야 규칙(M)을 의식합니다. 교사는 이 두 시점을 오가는 사람입니다 — 게임을 분석할 때는 플레이어처럼 재미에서 출발해 규칙을 역추적하고, 게임을 만들 때는 디자이너처럼 규칙을 조립해 재미를 겨냥합니다.

MDA로 게임 읽기 — 함께 해보는 예시

누구나 아는 달리기 점프 게임(플랫폼)을 MDA로 분해해 봅시다. 아래 표처럼 세 층위를 나누어 적기만 해도, "이 게임의 재미가 어느 규칙에서 나오는지"가 보이기 시작합니다.

CASE : 달리기 점프 게임

메커닉스

무엇으로 되어 있나

달리기 · 점프 · 장애물 · 코인 · 목숨 3개 · 제한 시간 · 스테이지 클리어 조건. 하나하나가 설계자가 정한 규칙이며, 이 목록을 바꾸면 전혀 다른 게임이 됩니다.

다이내믹스

무슨 일이 벌어지나

아슬아슬한 연속 점프, 코인 욕심을 내다 낭떠러지로 떨어지는 선택의 긴장, 시간이 얼마 안 남았을 때의 전력 질주. 규칙 어디에도 "긴장하라"고 적혀 있지 않지만, 규칙의 조합이 이 행동들을 만들어냅니다.

미학

무엇을 느끼나

도전(어렵지만 될 것 같아!), 성취(드디어 땀!), 감각(속도감과 리듬). 실패해도 다시 도전하게 만드는 이 감정이 게임이 최종적으로 판매하는 상품입니다.

핵심 질문 — "이 게임의 재미(A)는 어떤 규칙(M)에서 나오는가?" 이 연결 고리를 언어로 표현하는 것이 게임 분석의 전부입니다.

재미에도 종류가 있다 — 미학의 8분류

MDA의 저자들은 "재미"라는 뭉뚱그린 말 대신 여덟 가지 미학을 제시했습니다. 같은 "재미있는 게임"이라도 어떤 재미인지가 다르며, 학습 게임을 설계할 때는 이 중 무엇을 노릴지 의식적으로 선택해야 합니다.



감각 Sensation

보고 듣는 즐거움. 색·소리·타격감 등 감각적 만족



환상 Fantasy

다른 존재가 되는 상상. 역할과 세계관 몰입



서사 Narrative

이야기를 따라가는 맛. 전개와 결말의 궁금증



도전 Challenge

한계를 넘는 짜릿함. 학습 게임의 핵심 재미



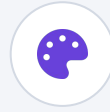
친교 Fellowship

함께 노는 즐거움. 협력·경쟁의 사회적 재미



발견 Discovery

숨겨진 것을 찾는 재미. 탐험과 새로움



표현 Expression

나를 드러내는 기쁨. 꾸미기·창작·선택



소일 Submission

가볍게 반복하는 편안함. 부담 없는 루틴

설계 질문 — 내가 만들 학습 게임은 8가지 중 어떤 재미를 줄 것인가? 잘 만든 학습 게임은 보통 도전을 축으로 2~3가지 재미를 조합합니다. (예: 도전+친교 = 모둠 퀴즈 대전, 도전+발견 = 탐험형 문제 풀이)

활동 — 내가 아는 게임을 해부하라

이론은 직접 적용해 보아야 몸에 붙습니다. 모듈별로 게임 하나를 골라 MDA 세 층위로 분해하고, 재미와 규칙의 연결 고리를 한 문장으로 정리해 발표합니다. 소요 시간은 약 15분입니다.

활동 방법

- 1 모듈별로 모두가 아는 게임 1개를 선정합니다. 학생들이 실제로 즐기는 게임일수록 좋습니다.
- 2 MDA 세 칸에 나눠 적습니다 — 규칙은 무엇인가? 플레이 중 무슨 일이 벌어지는가? 어떤 재미를 느끼는가?
- 3 "이 게임의 (A)는 (M)에서 나온다" 형식으로 연결 고리를 한 문장으로 정리합니다.
- 4 모듈별 1분 공유 — 같은 게임도 사람마다 다르게 읽힘을 확인하는 것이 이 활동의 숨은 목표입니다.

TIME : 15 MIN

워크시트 미리보기

게임 이름 : _____

M · 규칙과 부품 어떤 규칙과 요소로 이루어져 있나요?

.....

D · 벌어지는 일 플레이 중 어떤 행동·상황이 반복되나요?

.....

A · 느끼는 재미 8분류 중 어떤 재미인가요? (복수 선택)

.....

진행 팁: "그 게임이 왜 재미있어요?"라는 발문에 학생들이 쏟아내는 대답이 곧 미학(A)의 목록입니다.

헛갈리는 두 개념 — GBL vs 게이미피케이션

게임을 수업에 들이는 길은 크게 두 갈래입니다. Prensky가 정립한 게임기반학습(GBL)은 게임 그 자체를 학습 경험으로 삼는 접근이고, Deterding 등이 정의한 게이미피케이션은 게임이 아닌 활동에 게임의 요소만 이식하는 접근입니다. 목적과 설계 방법이 다르므로 구분해서 사용해야 합니다.

게임기반학습 (GBL) "게임으로 배우기"

- 게임을 플레이하는 것 자체가 학습 경험이 되도록 설계합니다.
- 게임 속 문제 해결이 곧 학습 목표 도달입니다. 내용이 게임 밖에 있지 않고 게임 안에 녹아 있습니다.
- 예: 샌드박스 월드에서 지형 만들며 지리 학습, 분수가 적힌 물고기를 낚으며 분수 개념 연습
- 강점: 깊은 몰입과 맥락적 이해 | 유의점: 설계 난이도가 높고 학습 목표와 게임 목표의 정렬이 관건

게이미피케이션 "게임처럼 배우기"

- 수업·과제 등 게임이 아닌 활동에 게임의 요소만 가져와 동기를 높입니다.
- 포인트 · 배지 · 레벨 · 리더보드 · 도전과제가 대표적 장치입니다.
- 예: 발표 포인트제, 과제 완료 배지, 모둠 순위판, 출석 스트리크
- 강점: 어떤 수업에도 즉시 적용 | 유의점: 보상이 목적이 되면 내적 동기를 해칠 수 있음(과잉정당화)

본 교재의 전략 = 둘의 결합. 교과 내용을 담은 게임(GBL)을 만들고(3~4단원), 실시간 리더보드(게이미피케이션 요소)를 얹어(6단원) 두 접근의 강점을 함께 취합니다.

여기에 하나 더 — 게임리터러시는 어디에 있나

GBL과 게이미피케이션을 구분해 놓고도, 거기에 게임리터러시까지 나란히 올려놓고 셋 중 하나를 고르려 하면 계속 헷갈립니다. 셋이 같은 층에 있지 않기 때문입니다. 앞의 둘은 수업을 어떻게 구성할 것인가에 대한 답, 곧 방법이고 게임리터러시는 그래서 무엇을 할 줄 알게 되는가에 대한 답, 곧 능력입니다.

방법

게임기반학습(GBL)
게임 자체를 학습 경험으로 삼는다

게이미피케이션
게임의 요소만 활동에 이식한다

수업을 어떻게 구성할 것인가

능력

게임리터러시
LV.1 읽기·이해 → LV.2 해석·비판 → LV.3 쓰기·창작

그래서 무엇을 할 줄 알게 되는가

같은 층이 아니다 — 앞의 둘은 방법, 게임리터러시는 그 방법으로 기르는 능력

자주 보는 네 가지 오해

“게임으로 수업했으니 게임리터러시를 한 것이다.”

분수 게임으로 분수를 가르쳤다면 학습 목표는 분수입니다. 게임 자체를 들여다보는 활동이 한 번은 들어가야 합니다.

“포인트와 배지를 줬으니 게임리터러시다.”

게이미피케이션은 게임의 부품만 빌려온 것입니다. 순위표를 운영하는 일과 순위표가 사람을 어떻게 움직이는지 아는 일은 다릅니다.

“게임리터러시는 과몰입 예방 교육이다.”

절제와 자기조절은 LV.2 해석·비판의 한 갈래일 뿐입니다. 읽기(LV.1)와 창작(LV.3)이 빠지면 반쪽입니다.

“게임리터러시는 게임 개발이나 코딩 교육이다.”

창작은 셋째 층위이지 전부가 아니고, 목적도 기술 습득이 아닙니다. 읽고 해석한 것을 자기 목적에 맞게 표현하는 행위입니다.

헷갈릴 때 던지는 질문 하나 — “이 수업이 끝나고 학생에게 남는 것이 교과 지식인가, 게임을 보는 눈인가?” 앞이면 GBL·게이미피케이션을 쓴 교과 수업, 뒤면 게임리터러시 수업입니다. 둘 다 남는다면 가장 좋은 설계입니다.

교실 속 게임 활용 — 세 갈래 길

현재 학교 현장에서 게임을 활용하는 방식은 크게 세 갈래입니다. 각각 뚜렷한 장점과 한계가 있으며, 본 교재는 지금까지 진입 장벽이 가장 높았던 세 번째 길을 엽니다.



기성 교육용 게임

샌드박스 · 교육용 플랫폼

교육용으로 개발된 완성형 게임을 그대로 수업에 활용하는 방식입니다. 세계 재현, 협력 건축, 코딩 학습 등 완성도 높은 콘텐츠를 즉시 쓸 수 있고 관련 수업 사례도 풍부합니다.

장점: 높은 품질과 검증된 사례 | 한계: 우리 반 진도·수준에 맞춘 세부 조정 불가



퀴즈 게임 플랫폼

실시간 퀴즈 · 골든벨형 도구

문항만 입력하면 즉시 경쟁형 복습 게임이 되는 플랫폼입니다. 준비 부담이 적어 전 과목에서 가장 널리 쓰이며, 실시간 순위 공개로 수업 열기를 끌어올립니다.

장점: 간편함과 즉시성 | 한계: 게임 형식이 퀴즈로 고정되어 확장이 어려움



교사 자작 웹게임

내가 만드는 우리 반 게임

내용·규칙·난이도·디자인을 모두 교사가 결정하는 맞춤형 게임입니다. 지금까지는 코딩 역량이 필요해 소수의 길이었지만, AI 에이전트의 등장으로 모든 교사에게 열렸습니다.

이 교재가 여는 길 → OUR PATH

비판적으로 읽기 — 게임이 나를 붙잡는 장치들

상업 게임은 이용자의 시간과 지갑을 붙잡도록 정교하게 설계됩니다. 게임리터러시의 "해석·비판" 층위는 바로 이 설계를 읽어내는 능력입니다. 대표적인 세 가지 장치를 해부해 봅시다.



보상 루프

언제 나올지 모르는 보상(가변 비율 보상)이 행동을 가장 강하게 붙잡습니다. 슬롯머신과 동일한 심리 기제로, "한 판만 더"를 만들어내는 핵심 장치입니다. 행동주의 심리학의 강화 계획 개념으로 설명할 수 있어 교과 연계도 가능합니다.



확률형 아이템

낮은 확률의 희귀 아이템을 반복 구매로 유도하는 수익 모델입니다. 표시 확률로 기댓값을 계산해 보면 구조가 선명히 보입니다 — 실제로 확률 계산은 초등 고학년~중등 수학 수업의 훌륭한 실생활 소재가 됩니다.



다크 패턴

출석 보상, 한정 이벤트, 랭킹 압박, 소셜 알림처럼 이탈을 막기 위해 설계된 장치의 총칭입니다. Zagal 등의 연구는 게임 속 다크 패턴을 시간·금전·사회관계 착취로 유형화했습니다. 장치의 이름을 아는 것만으로도 영향력이 줄어듭니다.

교육의 방향 — 금지가 아니라 해부. 구조를 아는 학생은 휘돌리지 않습니다. 학생과 함께 게임의 설계를 분석하는 수업은 그 자체로 실질적인 과몰입 예방교육이 됩니다.

출처: 가변 비율 강화 계획의 원 논고는 다음과 같습니다. Ferster, C. B., & Skinner, B. F. (1957). Schedules of reinforcement. · 확률형 아이템 이용과 문제적 도박 성향의 관련을 다룬 대규모 조사가 다음이 자주 인용됩니다. Zendle, D., & Cairns, P. (2018). Video game loot boxes are linked to problem gambling: Results of a large-scale survey. · Zagal, J. P., Björk, S., & Lewis, C. (2013). Dark patterns in the design of games.

소비자에서 생산자로 — 왜 교사가 만들어야 하는가

기성 게임을 고르는 것과 직접 만드는 것 사이에는 질적인 차이가 있습니다. 게임리터러시의 관점에서, 그리고 교실의 현실에서, 교사가 생산자가 되어야 할 이유는 분명합니다.



창작이 가장 깊은 리터러시

직접 만들어 본 사람만이 게임의 설계 의도를 꿰뚫어 봅니다. 글쓰기가 독해력을 완성하듯, 게임 제작 경험은 게임을 읽는 눈을 완성합니다. LV.3(창작)에 오른 교사는 LV.2(비판)의 깊이가 달라집니다.



우리 반 맞춤 게임은 시중에 없다

우리 반의 진도, 우리 학생들의 수준, 이번 주에 배운 어휘가 담긴 게임은 어느 스토어에도 없습니다. 교육과정과 학급 맥락을 아는 유일한 사람, 즉 교사만이 만들 수 있습니다.



만드는 교사가 학생도 만들게 한다

교사의 창작 경험은 곧 학생 게임 제작 수업의 설계 역량이 됩니다. 제작 과정의 막힘과 해법을 몸으로 아는 교사는 학생의 창작 프로젝트를 지도할 수 있는 코치가 됩니다.

그동안의 장벽

코딩



AI 에이전트가
대신 씁니다

아이디어를 한국어로 설명하면
앱이 되어 돌아오는 시대,
남은 질문은 "무엇을 만들 것인가"
뿐입니다.

→ WORLD 2에서 계속

함께 가는 길 — 게임리터러시 지원 생태계

게임리터러시 교육은 개별 교사의 열정만으로 굴러가지 않습니다. 국내에는 이미 공적 기관과 교사 공동체로 이루어진 지원 생태계가 갖추어져 있으며, 연수 이후의 성장은 이 생태계에 접속할 때 지속됩니다.



게임문화재단 · 한국콘텐츠진흥원 게임문화 교육의 공적 기반

전국 단위의 게임리터러시 교육 사업과 게임과몰입 상담 인프라를 운영하고, 게임이용자 실태조사 등 정책 연구의 기반 데이터를 매년 발간합니다. 교사가 수업 근거로 삼을 수 있는 공신력 있는 통계와 자료의 출처입니다.



게임문화교육연수원 (2026 개원) 교원 · 보호자 대상 공식 연수 기관

게임문화 이해와 게임리터러시 교육과정을 다루는 공식 원격 연수를 제공합니다. 2026년 개원 이후 교원 대상 신규 과정을 지속 개설하고 있어, 본 교재 이수 후의 심화 학습 경로로 적합합니다.



게임리터러시 연구교사 프로그램 실천 공동체

전국 초·중등 교사들이 게임 활용 수업을 연구하고 사례를 공유하는 실천 공동체입니다. 혼자 만들다 막히는 지점을 함께 넘고, 서로의 산출물을 검증해 주는 동료 네트워크가 지속 성장의 열쇠입니다.

연수 운영 팁: 소속 시도교육청의 게임리터러시 관련 연수·연구회 정보를 마지막 차시에 함께 안내하면 연수의 효과가 현장으로 이어집니다.



WORLD 1 CLEAR!

획득한 아이템 — 1단원 핵심 정리



게임리터러시 3층위

게임을 읽고(플레이) 해석하고(비판) 쓰는 (창작) 능력. 창작이 리터러시를 완성하며, 본 교재의 목표는 LV.3입니다.



MDA 분석 렌즈

규칙(M)이 상호작용(D)을 낳고 재미(A)로 완성되는 구조. 분석할 땐 A→M으로, 설계할 땐 M→A로 오갑니다.



GBL · 게이미피케이션

게임으로 배우기 vs 게임처럼 배우기. 본 교재는 GBL 게임 제작에 리더보드 (게이미피케이션)를 결합합니다.



비판적 읽기 방패

보상 루프·확률형 아이템·다크 패턴. 금지 대신 해부 — 구조를 아는 학생은 휘둘리지 않습니다.

NEXT ► WORLD 2 AI 에이전트와 게임 개발의 만남

AI 에이전트와 게임 개발

대화가 곧 개발이 되는 자리

1단원에서 좋은 학습 게임이 무엇인지 살펴보았다면, 이 단원은 그것을 실제로 만들 수 있게 해 주는 기술의 지형을 다룹니다. 챗봇과 구분되는 AI 에이전트의 작동 원리, 자연어 대화로 소프트웨어를 만드는 바이브 코딩, 그리고 이 책의 실습 도구인 Google AI Studio까지 차례로 정리합니다. 단원 끝의 준비 점검을 마치면, 3단원부터는 곧바로 첫 게임 제작에 들어갑니다.

2

교재 2단원

2단원 학습 퀘스트

2단원을 마치면 다음 네 가지를 할 수 있어야 합니다. 마지막 퀘스트는 실습 환경 점검으로, 3단원 시작 전에 반드시 완료되어야 합니다.



QUEST 1

챗봇과 AI 에이전트의 차이를 설명할 수 있다

"답을 주는 AI"와 "일을 끝내는 AI"의 차이를 작동 방식(계획·도구 사용·반복)으로 설명합니다.



QUEST 2

교육 현장의 AI 에이전트 활용 사례를 파악한다

수업 자료 제작, 업무 자동화, 학습 게임 제작 등 활용 영역별 사례와 가능성을 이해합니다.



QUEST 3

바이트 코딩의 원리와 교사의 역할 변화를 이해한다

코드 문법은 AI가, 교육 설계는 교사가 맡는 새로운 분업 구조를 설명할 수 있어야 합니다.



QUEST 4

Google AI Studio의 핵심 기능을 조망하고 실습 준비를 마친다

빌드 모드·주석 모드·GitHub 동기화·Firebase 연동의 용도를 알고, 계정·브라우저 점검을 완료합니다.



CLEAR 보상

실습 월드
입장 열쇠

WORLD 3부터는 이론을 손으로
확인합니다 — 직접 게임을
만들기 시작합니다

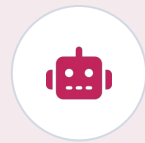
AI 에이전트란 — 챗봇과 무엇이 다른가

인공지능 분야에서 에이전트(agent)란 환경을 인식하고 목표 달성을 위해 스스로 행동하는 주체를 뜻합니다(Russell & Norvig, 2021). 최근의 AI 에이전트는 대규모 언어모델에 도구 사용·계획·기억이 결합되어, 질문에 답하는 수준을 넘어 여러 단계의 일을 끝까지 수행합니다.



챗봇 "답을 주는 AI"

- 한 번 묻고 한 번 답하는 문답 구조입니다.
- 작업의 모든 단계를 사람이 지시하고, 답변을 사람이 옮겨 실행해야 합니다.
- 산출물: 텍스트 답변. 활용 예: 아이디어 얻기, 문장 다듬기, 개념 질문.



AI 에이전트 "일을 끝내는 AI"

- 목표를 주면 스스로 계획을 세우고 단계를 나눕니다.
- 코드 실행, 파일 생성, 검색 같은 도구를 직접 사용하며 여러 단계를 연속 수행하고, 오류가 나면 스스로 수정합니다.
- 산출물: 완성된 결과물(문서·앱·게임). 활용 예: "퀴즈 게임 만들어줘" 한마디로 동작하는 게임 완성.

교사에게 갖는 의미 — 챗봇 시대에는 AI의 답을 수업 자료로 "옮기는" 일이 남았지만, 에이전트 시대에는 완성물이 바로 나옵니다. 게임 제작처럼 여러 단계가 필요한 일이 한 번의 대화로 가능해진 것입니다.

에이전트는 어떻게 일하는가 — 다섯 단계 루프

에이전트의 작동은 다섯 단계의 루프로 이해할 수 있습니다. 사람이 하는 일은 처음의 목표 제시(STEP 1)와 중간중간의 방향 조정뿐이며, 나머지는 에이전트가 만족스러운 결과가 나올 때까지 STEP 3~5를 스스로 반복합니다.



← 만족할 때까지 STEP 3~5를 스스로 반복합니다. 교사는 결과를 보고 대화로 방향만 잡아주면 됩니다.

핵심 통찰 — 에이전트 시대에 교사에게 필요한 역량은 코딩 실력이 아니라, ① 좋은 목표를 명확히 제시하는 능력(프롬프트)과 ② 결과물을 교육적으로 검토하는 안목입니다. 이 두 가지가 이 교재 전체를 관통하는 훈련 목표입니다.

교육 현장의 AI 에이전트 — 이미 시작된 변화

AI 에이전트는 이미 학교 현장 곳곳에서 쓰이고 있습니다. 시도교육청 단위의 교사 연수도 "AI 활용"을 넘어 "AI 에이전트 직접 제작"으로 진화하는 중입니다. 대표적인 네 가지 활용 영역을 살펴봅시다.



수업 자료 제작

학습지·평가 문항·프레젠테이션을 교육과정 성취기준에 맞춰 통째로 생성합니다. 초안 생성에 그치지 않고 수정·보완까지 대화로 반복할 수 있어, 자료 제작 시간이 크게 줄어듭니다.



학교 업무 자동화

가정통신문 초안 작성, 명렬표 정리, 일정 관리 등 반복적인 행정 업무를 에이전트에게 위임합니다. 절약된 시간은 수업 연구와 학생 상담이라는 교사 본연의 일로 돌아갑니다.



교사 연수의 확산

여러 시도교육청이 교사가 직접 수업용 AI 에이전트를 설계·제작하는 직무연수를 개설하고 있습니다. "쓰는 교사"에서 "만드는 교사"로의 전환이 정책 차원에서 진행 중입니다.



학습 게임 제작

교과 내용을 담은 웹게임을 교사가 직접 개발합니다. 본 교재의 핵심 활용처로, 3단원부터 Google AI Studio로 이 과정을 단계별로 실습합니다.

네 영역의 공통점 — "AI에게 물어보는 교사"에서 "AI에게 일을 맡기는 교사"로. 다음 슬라이드에서 그 방법론인 바이트 코딩을 다룹니다.

바이브 코딩 — 대화가 곧 개발이다

2025년 초 안드레 카파시가 이름 붙인 "바이브 코딩(vibe coding)"은 개발의 문법을 바꾸었습니다. 원하는 것을 자연어로 설명하면 AI가 코드를 쓰고, 사람은 결과를 보며 대화로 다듬는 방식입니다. 전문 개발자의 생산성 도구이자, 비개발자가 소프트웨어 제작에 진입하는 문이 되었습니다.

바이브 코딩 (Vibe Coding) 코드를 직접 작성하지 않고, 자연어 대화로 AI에게 의도를 설명하며 소프트웨어를 만들어 가는 개발 방식

BEFORE

전통적 개발

- 프로그래밍 언어의 문법부터 배워야 시작할 수 있었습니다.
- 한 줄 한 줄 코드를 직접 작성하고, 오류 해결에 몇 시간씩 쏟았습니다.
- 아이디어에서 완성까지 수개월 — 교사에게는 사실상 진입 불가의 영역이었습니다.



AFTER

바이브 코딩

- 하고 싶은 것을 한국어로 설명하면 AI가 코드 작성·실행·수정을 담당합니다.
- 결과물을 눈으로 보며 "여기를 이렇게 바꿔줘"라고 대화로 다듬습니다.
- 아이디어에서 동작하는 게임까지 수십 분 — 교육 전문성만 있으면 누구나 만듭니다.

유의: 바이브 코딩에서도 결과 검증의 책임은 사람에게 있습니다. 교육 자료라면 더욱 그렇습니다(→ STAGE 2-11 안전 수칙).

역할 분담 — 교사의 전문성이 곧 개발력

바이브 코딩은 "누가 무엇을 잘해야 하는가"를 다시 정의합니다. 기술 구현은 AI의 몫이 되었고, 남은 것은 교육의 몫 — 무엇을 가르칠지, 어떤 재미로 배우게 할지, 결과물이 교육적으로 타당한지의 판단입니다. 1단원의 이론이 여기서 기획 언어로 쓰입니다.



AI 에이전트가 맡는 것

- 프로그래밍 문법과 코드 작성 전체
- 화면 디자인과 상호작용 구현
- 오류(버그) 찾기와 수정
- 배포·저장 등 기술적 처리(GitHub, 클라우드)



교사가 맡는 것

- 학습 목표와 성취기준을 게임 내용으로 번역
- 학생 수준에 맞는 난이도와 문항 설계
- 게임 규칙과 재미의 기획 — 1단원의 MDA가 바로 이 언어입니다
- 결과물의 교육적 검토와 품질 최종 판단

그래서 1단원(게임리터러시·MDA)이 먼저였습니다. 기술 장벽이 사라진 시대에 좋은 학습 게임과 그렇지 않은 게임을 가르는 것은 코딩 실력이 아니라 교육 설계의 질입니다. 기술은 AI가, 교육은 교사가 — 이것이 새로운 분업입니다.

대표 프로그램 — 바이트 코딩 도구 3파전

바이트 코딩을 지원하는 대표 도구는 세 가지입니다. 셋 다 자연어로 앱을 만들 수 있지만, 게임 제작 연수의 관점에서는 생성 → 배포 → 백엔드(리더보드)까지 한곳에서 끝나는지가 핵심 비교 기준입니다.

★ PICK

Google AI Studio

Gemini 기반 · 본 교재의 선택

형태	웹 기반 빌드(Build) 모드
강점	생성→배포→백엔드 일원화
배포	공유 링크 · GitHub · Cloud Run
백엔드	Firebase 자동 연동(리더보드)
비용	무료 (구글 계정만 필요)

Claude

Artifacts · 아티팩트

형태	대화 중 미리보기 생성
강점	긴 맥락 처리 · 코드 품질
배포	아티팩트 링크 공유
백엔드	별도 서비스 연결 필요
비용	무료 + 유료 플랜

ChatGPT

Canvas · 캔버스

형태	캔버스에서 코드 편집
강점	대중성 · 범용성
배포	코드 복사 후 별도 호스팅
백엔드	별도 서비스 연결 필요
비용	무료 + 유료 플랜

셋 다 훌륭한 도구이며 수업 목적에 따라 선택이 달라질 수 있습니다. 본 교재는 무료로 제작부터 배포·리더보드까지 한곳에서 완결되는 Google AI Studio를 실습 도구로 채택합니다.

왜 Google AI Studio인가 — 네 가지 결정타

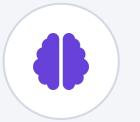
교사 연수 도구는 성능만으로 선택할 수 없습니다. 진입 장벽(비용·가입 절차), 결과물의 수업 적용 가능성(배포), 확장성(백엔드)까지 갖춰야 합니다. AI Studio가 네 조건을 모두 충족합니다.

01 완전 무료 시작



구글 계정만 있으면 카드 등록 없이 바로 시작합니다. 연수 참여자 전원이 5분 안에 같은 출발선에 설 수 있다는 것은 연수 운영에서 결정적인 장점입니다.

02 최신 Gemini 탑재



구글의 최신 AI 모델 Gemini가 게임을 생성합니다. 한국어 이해가 뛰어나 한국어 프롬프트만으로 충분하며, 생성 품질이 실습 만족도를 좌우합니다.

03 배포까지 한 번에



만든 게임을 공유 링크로 즉시 배포하고, GitHub 양방향 동기화로 원본을 보관하며, Cloud Run으로 정식 서비스화할 수 있습니다. "만들고 끝"이 아니라 "학생에게 도달"까지 갑니다(5만원).

04 Firebase 자동 연동

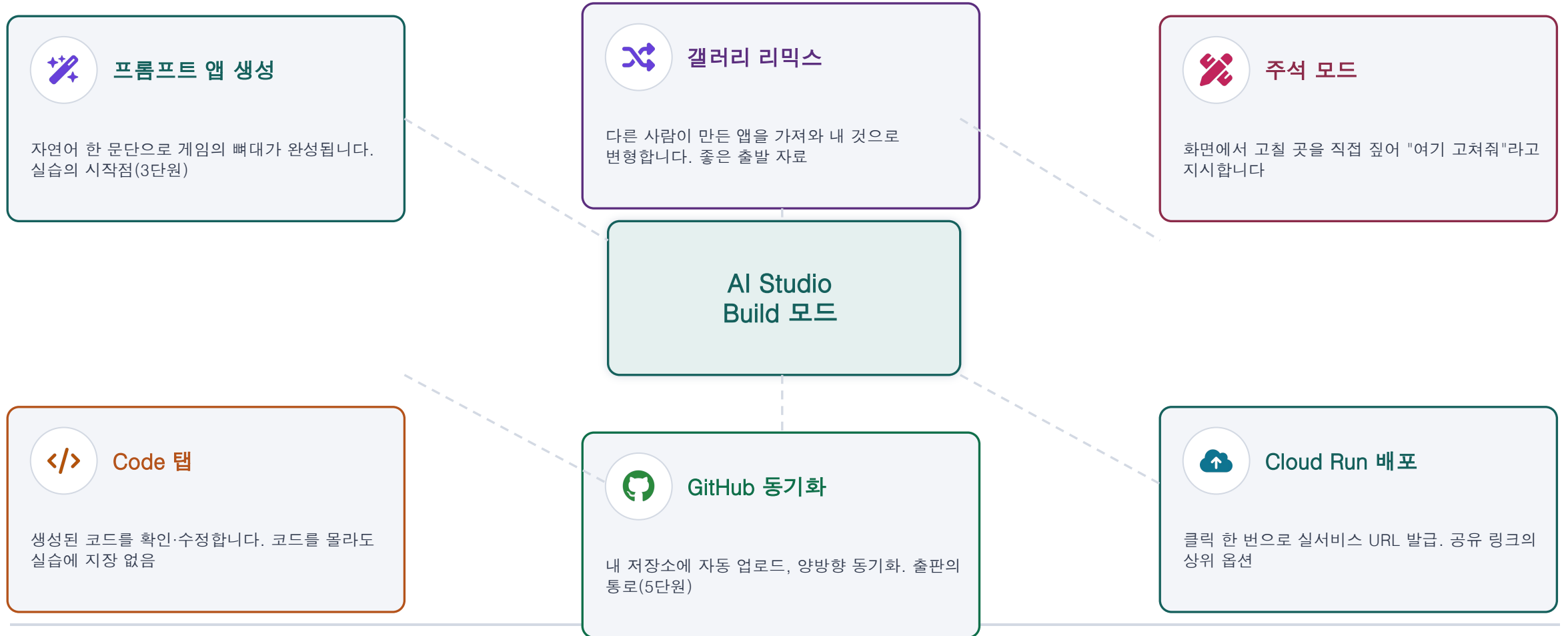


프롬프트만으로 Firestore 데이터베이스와 인증을 자동 구성합니다. 서버 지식 없이도 학급 전체의 점수가 모이는 실시간 리더보드를 만들 수 있습니다(6만원).

접속 주소: aistudio.google.com — WORLD 3에서 실제 화면과 함께 만납니다.

AI Studio 기능 지도 — 앞으로 만날 여섯 무기

실습에서 사용할 여섯 기능의 지도입니다. 모두 빌드(Build) 모드라는 하나의 화면에서 이루어지며, 3~6단원에서 실제 화면 캡처와 함께 하나씩 익힙니다.



프롬프트 한 줄이 게임이 되는 순간

이론의 마무리는 눈으로 확인하는 것입니다. 강사가 아래 프롬프트를 실제로 입력해 게임이 만들어지는 전 과정을 시연합니다. 참가자는 오른쪽 관전 포인트를 중심으로 에이전트의 작업을 관찰합니다.

> PROMPT

"초등 3학년이 분수 개념을 연습하는 낚시 게임을 만들어줘.
바다에 분수가 적힌 물고기들이 헤엄치고, 문제에 맞는 물고기를
클릭하면 낚아 올려. 정답이면 점수, 오답이면 물고기가 도망가.
밝고 귀여운 스타일로, 효과음도 넣어줘."

시연 중 함께 볼 포인트

- 프롬프트 속 요소가 어떻게 구현되는가 — "분수가 적힌 물고기"가 문제 출제 로직으로 번역되는 과정
- 요청하지 않은 부분을 AI가 어떻게 채우는가 — 배경·점수판·시작 화면 등 스스로 보완하는 부분
- 수정 요청 한 번에 무엇이 얼마나 바뀌는가 — 대화형 개선(iteration)의 감각

1

프롬프트 입력

왼쪽 문장을 빌드 모드에 그대로 붙여넣습니다.

2

생성 관찰 (1~2분)

에이전트가 계획→코드→실행을 진행하는 과정을 지켜봅니다.

3

즉시 플레이

미리보기 화면에서 직접 낚시하며 결과를 확인합니다.

4

수정 한 번

"물고기를 더 천천히 움직여줘" — 대화로 개선되는 과정을 봅니다.

실습 로드맵 — 우리가 만들 것들

3단원부터는 매 단원 손에 잡히는 산출물이 하나씩 쌓입니다. 각 단원의 산출물이 다음 단원의 재료가 되는 누적 설계이므로, 순서대로 진행하는 것을 권장합니다.



최종 결과물 우리 반 교과 내용이 담긴, 링크 하나로 접속하는, 실시간 순위표가 도는 나만의 학습 게임 — 그리고 그것을 만들 줄 아는 교사 자신입니다.

시작 전 안전 수칙 — 네 개의 방패

AI로 만드는 교육 자료에는 네 가지 점검 지점이 있습니다. 실습 단위마다 해당 장면에서 다시 짚겠지만(배포=개인정보, 리더보드=닉네임), 시작 전에 원칙으로 함께 확인해 둡시다.



저작권

기존 게임의 캐릭터·이미지·음악을 그대로 가져다 쓰지 않습니다. AI가 생성한 이미지·코드도 이용 조건을 확인하는 습관이 필요하며, 유명 IP를 닮은 생성물은 수업용이라도 피하는 것이 안전합니다.



개인정보

게임 화면과 리더보드에 학생 실명·사진을 넣지 않습니다. 닉네임 사용을 원칙으로 하고, 수집하는 정보(점수·닉네임)를 최소화합니다. 6단원 리더보드 설계에서 필수 점검 사항입니다.



학생 계정과 연령

구글 계정 연령 정책과 소속 학교(교육청) 계정의 기능 제한을 사전에 확인합니다. 학생은 계정 없이 "만든 게임을 플레이"만 하도록 설계할 수 있으며, 본 교재도 그 방식을 권장합니다.



AI 산출물 검토

AI가 만든 문항의 사실 오류·편향·수준 적합성 점검은 교사의 몫입니다. 교육 자료의 최종 책임자는 언제나 교사이며, 이 검토 안목이 에이전트 시대 교사 전문성의 핵심입니다.

참고: 소속 시도교육청의 생성형 AI 활용 지침이 있다면 그것이 우선입니다. 연수 운영 시 해당 지침을 함께 배부하는 것을 권장합니다.

실습 준비 체크리스트

3단원 실습을 시작하기 전, 아래 다섯 항목을 점검합니다. 연수 운영자라면 이 목록을 사전 안내문으로 발송해 두면 첫 실습 시간의 손실을 크게 줄일 수 있습니다.



구글 계정 개인 계정을 권장합니다. 학교(워크스페이스) 계정은 관리자 설정에 따라 AI Studio 접속이 막혀 있을 수 있으므로, 사전에 확인하거나 개인 계정을 준비합니다.



크롬 브라우저 최신 버전으로 업데이트합니다. AI Studio는 크롬에서 가장 안정적으로 동작하며, 실습 중 브라우저 문제로 흐름이 끊기는 일을 예방합니다.



GitHub 계정 가입 github.com에서 무료로 가입합니다. 5단원(온라인 출판)에서 사용하며, 가입 시 만든 사용자명이 게임 주소의 일부가 되므로 신중히 정합니다.



노트북 + 안정적 인터넷 실습은 개인 노트북 기준으로 진행합니다. 태블릿은 일부 기능이 제한되며, 게임 생성 과정에서 안정적인 네트워크가 필요합니다.



aistudio.google.com 접속 테스트 로그인 후 첫 화면이 열리는지만 미리 확인해 둡니다. 접속이 안 되는 경우 계정 유형(연령·기관 제한)을 점검합니다.



WORLD 2 CLEAR!

이론 장비 장착 완료 — 2단원 핵심 정리



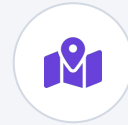
에이전트를 부리는 법

챗봇은 답을 주고 에이전트는 일을 끝냅니다. 목표 제시와 결과 검토가 사람의 몫입니다.



바이브 코딩 마인드셋

한국어로 설명하면 앱이 됩니다. 교사의 전문성(교육 설계)이 곧 개발력입니다.



AI Studio 기능 지도

빌드 모드·주석 모드·GitHub 동기화·Firebase 연동 — 실습에서 쓸 여섯 무기의 위치를 알았습니다.



안전 수칙 네 방패

저작권·개인정보·학생 계정·AI 산출물 검토. 최종 책임자는 언제나 교사입니다.

READY? NEXT ► WORLD 3 — AI Studio 첫걸음 (실습 시작!)

참고문헌

본 교재 이론편에서 인용한 문헌의 전체 목록입니다(APA 7판 양식). 각 슬라이드 하단의 출처 표기는 아래 목록의 축약형입니다.

Buckingham, D. (2003). Media education: Literacy, learning and contemporary culture. Polity Press.

Buckingham, D., & Burn, A. (2007). Game literacy in theory and practice. Journal of Educational Multimedia and Hypermedia, 16(3), 323–349.

Caillois, R. (2001). Man, play and games. University of Illinois Press. (원저 1958년 출판)

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. MindTrek 2011.

Ferster, C. B., & Skinner, B. F. (1957). Schedules of reinforcement. Appleton–Century–Crofts.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. Palgrave Macmillan.

Gilster, P. (1997). Digital literacy. Wiley.

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. AAAI Workshop.

Huizinga, J. (1955). Homo ludens. Beacon Press. (원저 1938년 출판)

Lepper, M. R., Greene, D., & Nisbett, R. E. (1973). Undermining children's intrinsic interest with extrinsic reward. Journal of Personality and Social Psychology, 28(1), 129–137.

New London Group. (1996). A pedagogy of multiliteracies: Designing social futures. Harvard Educational Review, 66(1), 60–92.

Prensky, M. (2001). Digital game-based learning. McGraw–Hill.

Salen, K., & Zimmerman, E. (2004). Rules of play: Game design fundamentals. MIT Press.

Zagal, J. P., Björk, S., & Lewis, C. (2013). Dark patterns in the design of games. FDG 2013.

Zendle, D., & Cairns, P. (2018). Video game loot boxes are linked to problem gambling: Results of a large-scale survey. PLOS ONE, 13(11), e0206767.

게임문화재단. (2026). 게임문화교육연수원. <https://www.game-edu.kr>

한국콘텐츠진흥원. (2025). 2025 게임이용자 실태조사.

Karpathy, A. (2023). “The hottest new programming language is English” [게시물]. X.

Karpathy, A. (2025). “Vibe coding” [게시물]. X.

Russell, S., & Norvig, P. (2021). Artificial intelligence: A modern approach (4th ed.). Pearson.

Google. (2026). Build apps in Google AI Studio. Gemini API 개발자 문서. <https://ai.google.dev/gemini-api/docs/aistudio-build-mode>

Google. (2026). Deploying from Google AI Studio. Gemini API 개발자 문서. <https://ai.google.dev/gemini-api/docs/aistudio-deploying>

교육부. (2024). 교육(지원)청·학교 개인정보 보호 가이드라인.

개인정보보호위원회. (2023). 개인정보 보호법 제22조의2(만 14세 미만 아동의 개인정보 보호).

본 자료는 교육 목적의 연수 교재입니다. 통계와 프로그램 기능 설명은 2026년 8월 기준이며, AI Studio의 화면·기능은 이후 업데이트로 달라질 수 있습니다.



WORLD 3

AI Studio 첫걸음

접속부터 나의 첫 게임 완성까지

이 단원은 처음부터 끝까지 실습입니다. 브라우저 주소창에 주소를 입력하는 순간부터 로그인, 화면 구성 파악, 프롬프트 입력, 생성 관찰, 완성된 게임 플레이, 대화와 주석 모드로 고치기까지 모든 단계를 실제 화면과 함께 하나씩 따라갑니다. 오늘의 목표는 단 하나, “내 손으로 만든 첫 게임”입니다. 화면이 교재와 조금 달라도 흐름은 같습니다.



3

교재 3단원

3단원 학습 퀘스트

3단원을 마치면 다음 네 가지를 할 수 있어야 합니다. 이번 단원의 퀘스트는 모두 "직접 해보기"로만 달성됩니다.



QUEST 1

AI Studio에 접속해 로그인하고 화면 구성을 설명할 수 있다

주소 입력 → 로그인 → 빌드 홈의 각 영역(사이드바·입력창·갤러리)의 역할을 파악합니다.



QUEST 2

프롬프트로 첫 게임을 생성할 수 있다

준비된 OX 퀴즈 프롬프트를 입력하고, 에이전트가 게임을 만드는 과정을 관찰합니다.



QUEST 3

대화와 주석 모드로 게임을 수정할 수 있다

문항 교체, 디자인 변경 등 수정 요청을 채팅과 화면 짚기(주석 모드)로 전달합니다.



QUEST 4

내 교과 내용으로 OX 퀴즈를 변형할 수 있다 (미션)

실습 마지막 미션 — 프롬프트 틀의 빈칸을 내 과목·단원으로 바꿔 나만의 게임을 만듭니다.



시작 전 점검

- ✓ 크롬 브라우저 최신 버전
- ✓ 구글 개인 계정 로그인 가능
- ✓ 노트북 + 안정적인 인터넷

(2단원 체크리스트 완료 기준)

오늘의 실습 여정 — 8개의 발판

아래 8단계를 순서대로 밟습니다. 각 단계마다 실제 화면 캡처가 함께 제공되므로, 자신의 화면과 비교하며 따라오면 됩니다. 소요 시간은 여유 있게 총 60~80분입니다.



중간에 길을 잃으면 → 마지막 "트러블슈팅 FAQ" 슬라이드를 펼쳐세요. 자주 막히는 여섯 지점의 해법이 정리되어 있습니다.

접속 — 주소를 직접 입력하세요

크롬을 열고 주소창에 아래 주소를 정확히 입력한 뒤 Enter를 누릅니다. 포털 검색으로 들어가면 광고나 비슷한 이름의 다른 사이트를 거칠 수 있으므로, 처음부터 주소 직접 입력을 권장합니다. 즐겨찾기(북마크)에 바로 추가해 두면 다음 실습부터 편합니다.

브라우저 주소창에 입력

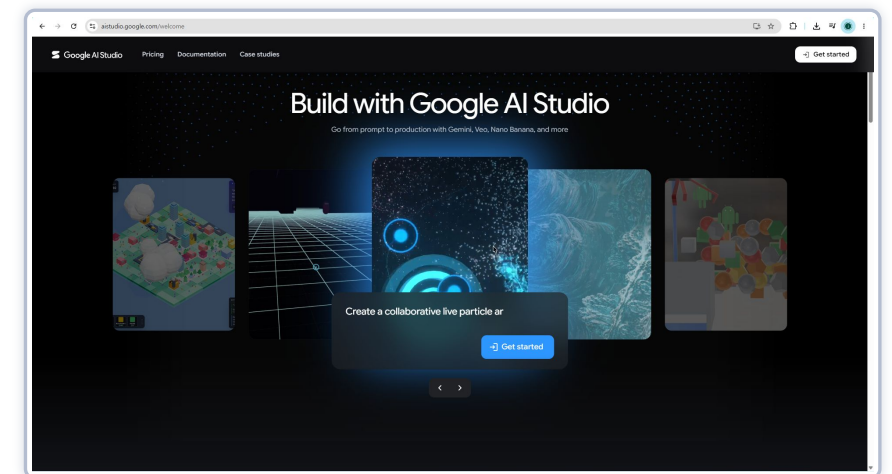
 **aistudio.google.com**

  aistudio.google.com/welcome

실제 크롬 주소창 (부분 캡처)

입력 팁

- https:// 는 생략해도 됩니다. 자동으로 붙습니다.
- "ai studio"로 검색하지 말고 주소를 입력하세요 — 이름이 비슷한 서비스가 여럿 있습니다.
- 주소창 오른쪽 별(☆) 아이콘을 눌러 북마크에 추가해 둡니다.



접속 직후 화면 — 오른쪽 위 [Get started]로 시작

왜 주소가 aistudio.google.com일까? AI Studio는 구글의 AI 개발 도구 사이트입니다. 구글 계정 하나로 로그인되며, 별도 설치 없이 브라우저에서 모든 실습이 이루어집니다. 접속 화면의 "Go from prompt to production with Gemini"라는 문구가 이 도구의 정체성을 요약합니다 — 프롬프트에서 완성품까지.

로그인 — 세 가지 경우의 수

접속 화면에서 오른쪽 위(또는 가운데)의 [Get started] 버튼을 누르면 로그인 절차가 시작됩니다. 자신의 상황이 아래 세 경우 중 어디에 해당하는지 확인하세요. 크롬에 이미 구글 계정으로 로그인되어 있는 경우가 가장 흔하며, 이때는 버튼 하나로 곧장 작업 화면에 도착합니다.

경우 ① 이미 구글에 로그인되어 있다 (가장 흔함)

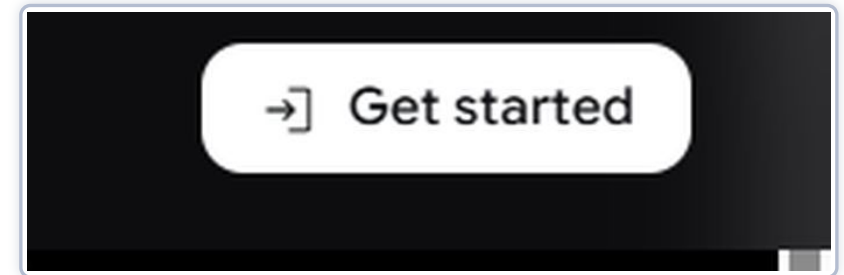
[Get started]를 누르면 계정 확인 없이 곧바로 빌드 홈으로 들어갑니다. 크롬 오른쪽 위에 프로필 사진이 보인다면 이 경우입니다.

경우 ② 로그인이 안 되어 있다

구글 계정 선택 화면이 나타납니다. 개인 계정을 선택하고 비밀번호를 입력하면 됩니다. 여러 계정이 있다면 실습용으로 정한 계정 하나만 계속 사용하는 것이 좋습니다.

경우 ③ 이 계정으로 AI Studio가 처음이다

이용 약관 동의 화면이 한 번 나타납니다. 내용을 확인하고 동의하면 이후에는 다시 나타나지 않습니다.



접속 화면 오른쪽 위 [Get started] (부분 확대)

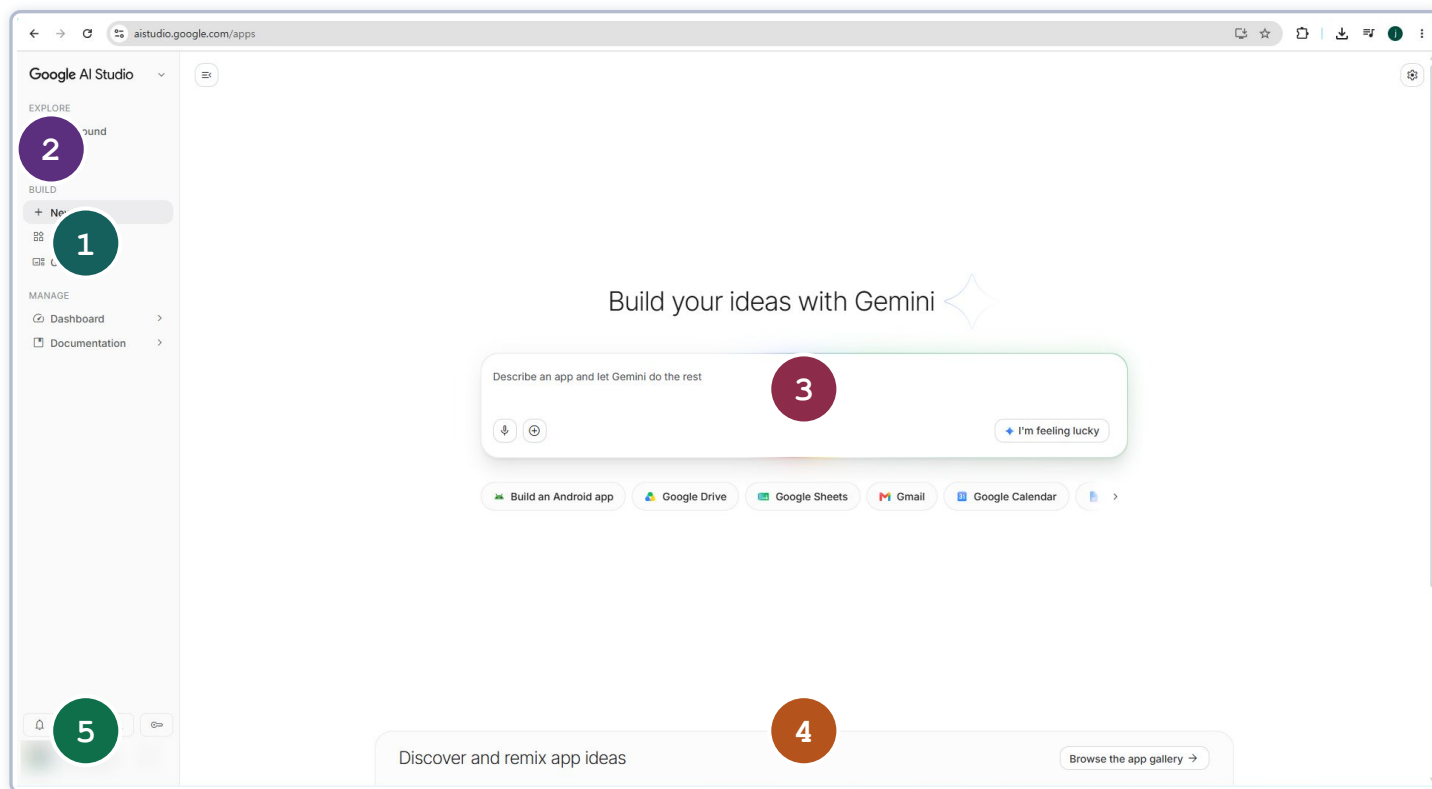
학교 계정이 안 될 때

학교(워크스페이스) 계정은 관리자 설정에 따라 차단 안내가 뜰 수 있습니다. 개인 계정으로 다시 로그인하세요. 계정 전환은 화면 오른쪽 위 프로필 아이콘에서 합니다.

로그인에 성공하면 주소가 aistudio.google.com/apps로 바뀌며 다음 슬라이드의 화면이 나타납니다.

빌드 홈 투어 — 다섯 구역만 알면 됩니다

로그인하면 별도의 이동 없이 곧바로 이 화면, 빌드 홈(aistudio.google.com/apps)에 도착합니다. 가운데 "Build your ideas with Gemini"라는 문구 아래 프롬프트 입력창이 보이면 제대로 온 것입니다. 아래 캡처의 번호와 오른쪽 표를 짚지어, 각자의 화면에서 다섯 구역을 직접 찾아보세요.



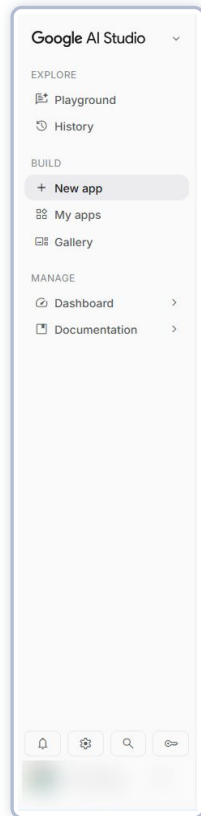
빌드 홈 — 로그인 직후 실제 화면

- 1 사이드바**
EXPLORE·BUILD·MANAGE 세 묶음의 기능 메뉴
- 2 History (기록)**
모든 작업이 자동 저장 — 실수로 닫아도 여기서 복구
- 3 프롬프트 입력창**
게임 설명을 입력하는 시작점. 오늘의 주 무대
- 4 갤러리 입구**
다른 사람들의 앱 구경 — Browse the app gallery
- 5 계정·알림·설정**
프로필 확인, 계정 전환, 환경 설정

※ 메뉴 이름·위치는 업데이트로 달라질 수 있습니다. 이름이 달라도 역할은 같습니다.

사이드바 확대 — 메뉴 하나하나 뜯어보기

왼쪽 사이드바를 부분 확대한 실제 캡처입니다. 메뉴는 EXPLORE(탐색)·BUILD(만들기)·MANAGE(관리) 세 묶음으로 구성됩니다. 본 교재 실습의 주 무대는 BUILD 묶음이며, 그중에서도 New app이 새 게임을 만드는 시작 버튼입니다.



사이드바 부분 캡처 (실제 화면)



Playground
수 있습니다.

EXPLORE · Gemini 모델과 자유롭게 대화하고 실험하는 공간입니다. 게임에 넣을 문항 아이디어를 얻는 용도로 활용할



History (기록)

EXPLORE · 지금까지의 작업이 자동 저장됩니다. 실수로 창을 닫아도 여기서 다시 열 수 있습니다.



New app ★

BUILD · 새 앱(게임) 만들기 — 이번 실습의 시작점입니다. 빌드 훔의 입력창과 같은 기능입니다.



My apps

BUILD · 내가 만든 앱들의 목록입니다. 지난 실습에서 만든 게임을 다시 열 때 사용합니다.



Gallery

BUILD · 다른 사람들이 만든 앱을 구경하고 리믹스(복제·수정)할 수 있는 전시장입니다.



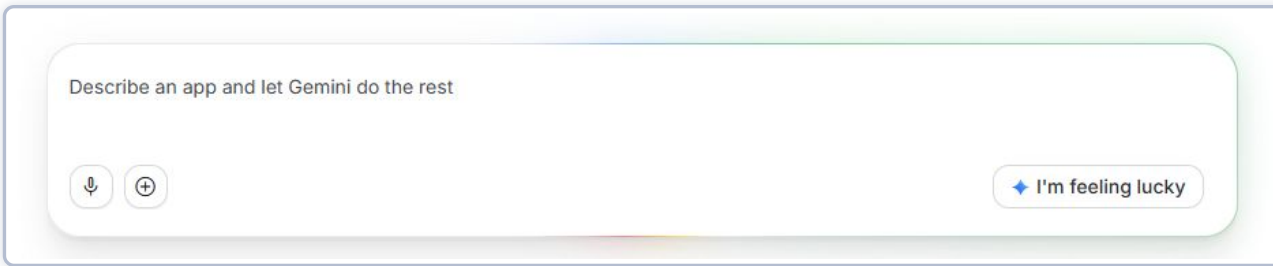
Dashboard · Documentation

MANAGE · API 키 등 개발자용 관리 메뉴 — 본 교재 실습에서는 사용하지 않습니다.

★ 표시가 오늘의 주인공입니다. 사이드바가 아이콘만 보인다면 위쪽의 접기/펼치기 버튼으로 펼칠 수 있습니다.

입력창 확인 — 게임을 시작하는 세 가지 방법

빌드 홈에서 게임 제작을 시작하는 방법은 세 가지입니다. 직접 프롬프트를 쓰는 것이 기본이고, 막막할 때는 나머지 두 방법이 좋은 출발점이 됩니다. 오늘은 ①번, 직접 입력으로 첫 게임을 만듭니다. 다른 화면으로 갔다면 사이드바의 New app을 누르면 언제나 입력창으로 돌아옵니다.



프롬프트 입력창 (부분 확대) — 오른쪽의 I'm feeling lucky도 보입니다

① 프롬프트 직접 입력 (오늘의 방법)

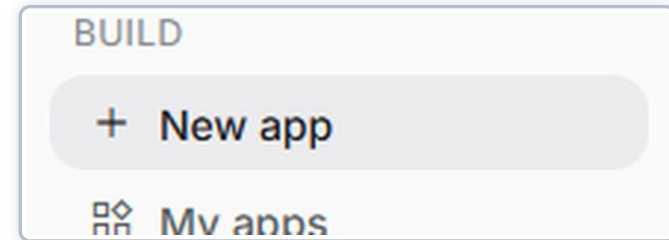
만들고 싶은 게임을 한국어로 설명하고 Build 버튼을 누릅니다. 다음 슬라이드에 준비된 프롬프트가 있습니다.

② I'm feeling lucky

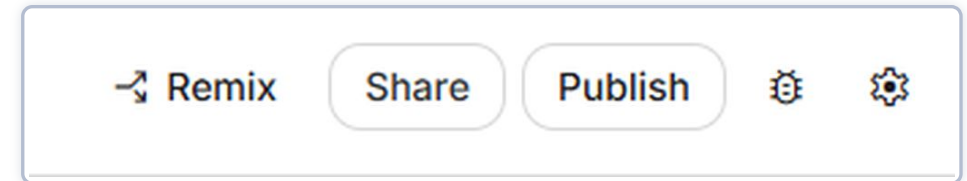
버튼을 누르면 AI가 무작위 아이디어를 채워 줍니다. 어떤 것이 가능한지 감을 잡을 때 좋습니다.

③ 갤러리 리믹스 (Remix)

사이드바 Gallery 또는 홈 하단의 Browse the app gallery에서 마음에 드는 앱을 골라 Remix하면 내 것으로 가져와 수정할 수 있습니다.



사이드바의 New app — 입력창으로 돌아오는 문



열린 앱 오른쪽 위의 Remix · Share · Publish

입력창 꿀팁 — 회색 글씨의 정체

프롬프트를 쓰다 보면 뒤에 회색 글씨로 이어질 내용이 제안됩니다. Gemini의 자동 완성 제안으로, Tab 키를 누르면 채택되고 무시하면 반영되지 않습니다. 좋은 제안이면 받아들여 프롬프트를 풍성하게 만들 수 있습니다.

Discover and remix app ideas

Browse the app gallery →

실습 프롬프트 — 오늘의 주문서

아래 프롬프트를 그대로 입력(또는 배부된 파일에서 복사)합니다. 그대로 써도 게임이 완성되지만, 오른쪽 해설을 읽으면 "왜 이렇게 쓰는지"가 보입니다. 좋은 프롬프트는 대상·형식·규칙·스타일 네 요소를 담습니다.

> 실습 1 프롬프트 (그대로 입력)

"초등학생용 OX 퀴즈 게임을 만들어줘.
주제는 우리 몸과 건강이야. 문제가 10개 나오고,
학생이 O 또는 X 버튼을 누르면 정답 여부를 알려줘.
정답이면 100점씩 올라가고, 다 풀면 총점과 함께
칭찬 메시지를 보여줘. 초등학생이 좋아할
밝고 알록달록한 디자인으로, 글자는 크게 해줘."

대상

"초등학생용", "글자는 크게" — 눈높이를 정합니다

형식

"OX 퀴즈", "문제 10개" — 게임의 뼈대입니다

규칙

"정답이면 100점", "총점 표시" — 1단원의 메커닉스(M)!

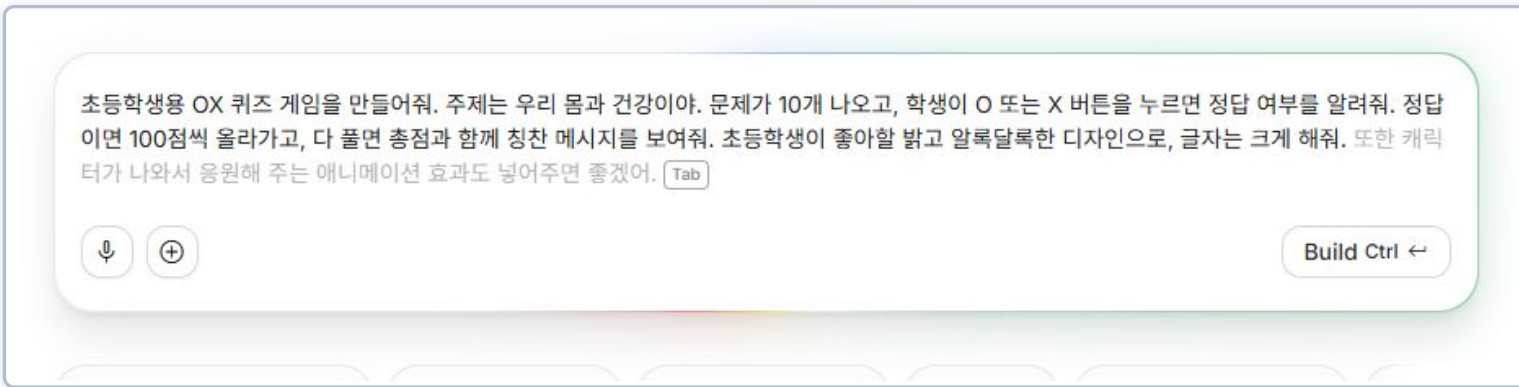
스타일

"밝고 알록달록", "칭찬 메시지" — 미학(A)을 지정합니다

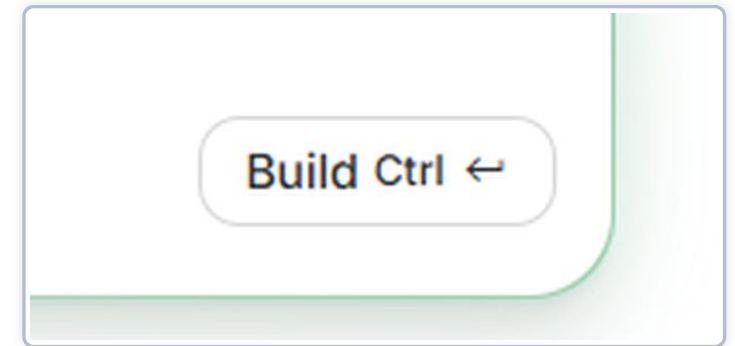
프롬프트에 정답을 다 적지 않아도 됩니다 — 문항 자체는 AI가 만들어 줍니다. 대신 생성된 문항에 사실 오류가 없는지 확인하는 것은 교사의 몫입니다(2단원 안전 수칙 ④). 다음 단계에서 실제로 확인합니다.

입력하고 실행 — Build 버튼 하나로 개발 시작

실습 프롬프트를 입력창에 붙여넣은 실제 화면입니다. 입력창 오른쪽 아래의 [Build] 버튼(단축키 Ctrl+Enter)을 누르는 순간 에이전트가 일을 시작합니다. 캡처를 자세히 보면 프롬프트 끝에 회색 글씨의 자동 완성 제안("캐릭터가 나와서 응원해 주는 애니메이션...")이 붙어 있는 것도 보입니다 — Tab을 누르면 이 제안까지 채택됩니다.



프롬프트가 입력된 입력창 — 끝의 회색 글씨가 Gemini의 자동 제안



[Build Ctrl+↵] 실행 버튼 (부분 확대)

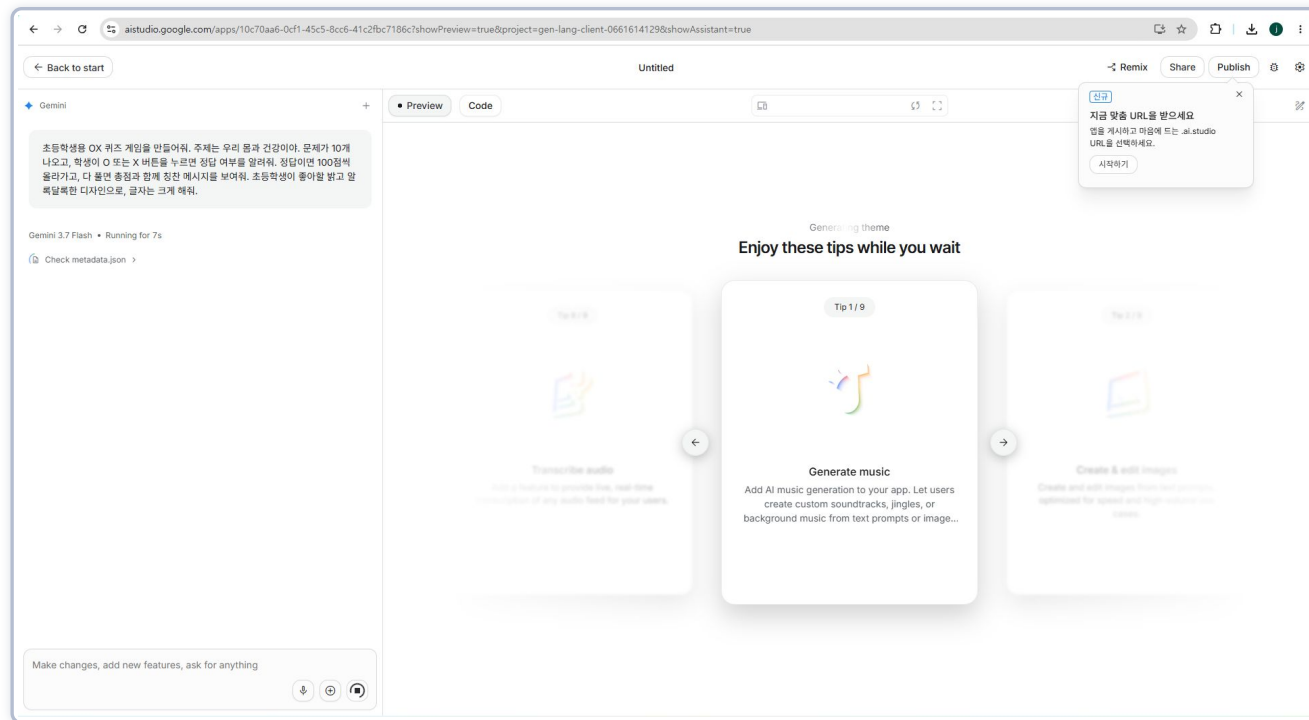
단축키 Ctrl+Enter로도 실행됩니다. 입력창 안에서 줄바꿈은 Enter.

실행 전 마지막 점검

프롬프트가 중간에 잘리지 않고 전부 들어갔는지 확인하세요. 복사-붙여넣기 과정에서 마지막 문장이 빠지는 경우가 흔합니다. 일부만 입력되어도 게임은 만들어지지만, 빠진 요구사항(예: 점수 규칙)은 반영되지 않습니다. 알림 권한 팝업이 뜨면 [허용]과 [차단] 어느 쪽을 눌러도 실습에는 지장이 없습니다.

생성 관찰 — 에이전트가 일하는 2~3분

실행하면 화면이 작업 모드로 바뀌고 에이전트의 진행 상황이 실시간으로 표시됩니다. 실제 캡처에서 보듯 대기 중에는 활용 팁 카드("Enjoy these tips while you wait")가 넘어가고, 왼쪽에 사용 모델(Gemini)과 경과 시간이 표시됩니다. 이 2~3분이 2단원에서 배운 "에이전트 루프"가 눈앞에서 돌아가는 순간입니다.



생성 중 화면 — 왼쪽에 모델명과 경과 시간(Running for ...s)

왼쪽: 진행 상황

사용 중인 모델과 경과 시간이 표시되고, 파일이 만들어질 때마다 목록이 늘어납니다 — STEP 2·3(계획·도구 사용)

오른쪽: 팁 카드

기다리는 동안 AI Studio의 기능 소개 카드가 넘어갑니다. 가볍게 읽어두면 4~6단원에서 도움이 됩니다

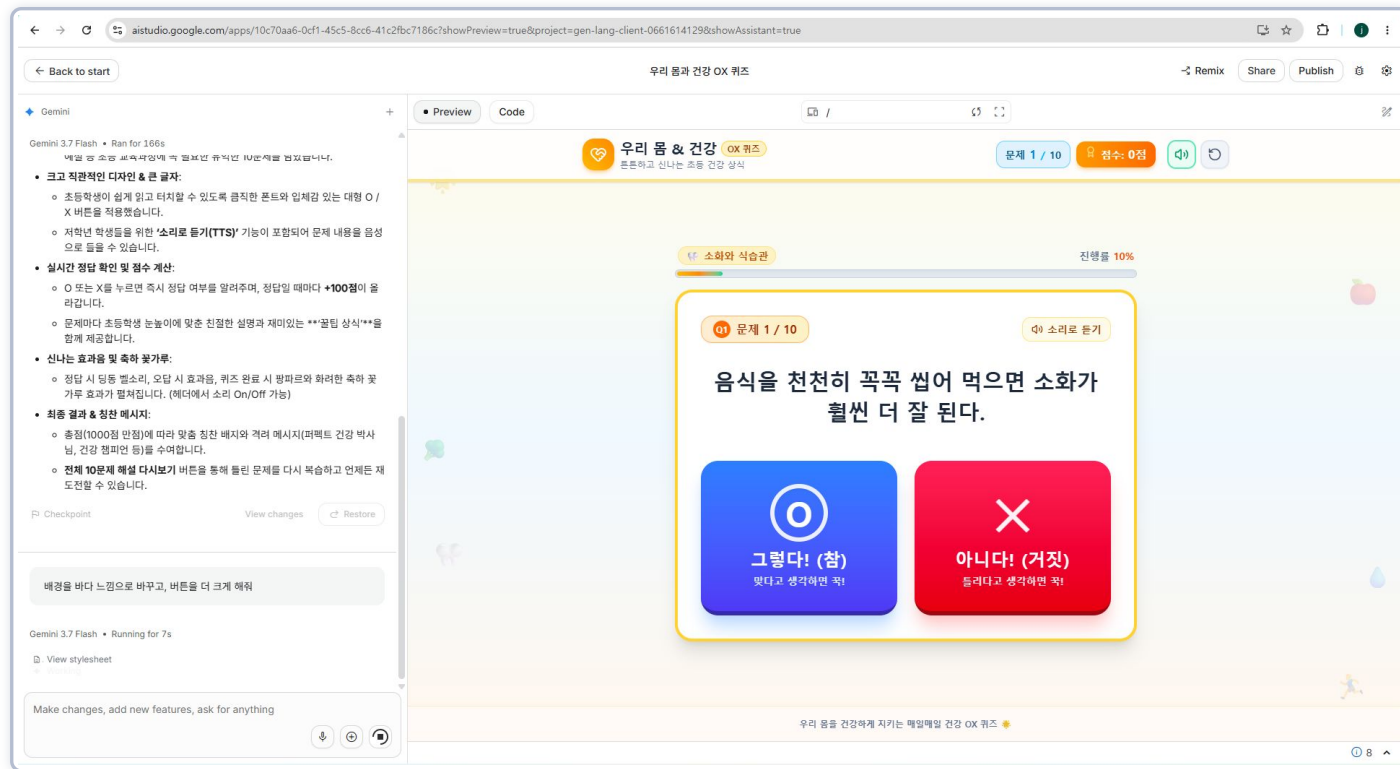
완료되면

오른쪽에 게임 미리보기가 자동으로 열립니다 — STEP 4(실행·생성). 보통 2~3분 안에 완료됩니다

생성이 5분 이상 멈춰 있으면? → 트러블슈팅 FAQ ③번을 보세요. 대부분 새로그침 없이 해결됩니다.

완성 — 첫 게임을 플레이하고 점검하기

축하합니다, 첫 게임이 완성되었습니다! 실제 생성 결과를 보면 요청한 10문제·100점 규칙은 물론, 요청하지 않은 소리로 듣기(TTS) 기능과 진행률 표시까지 에이전트가 스스로 더해 놓았습니다. 미리보기에서 직접 플레이하며 교사의 눈으로 아래 다섯 항목을 점검하세요 — 이 점검이 다음 단계(수정)의 재료가 됩니다.



완성된 게임 플레이 — 문제 1/10, O·X 버튼, 점수·진행률 표시

플레이하며 점검할 5가지

- ✓ 프롬프트의 요구(10문제·100점·총점·칭찬)가 다 반영되었나요?
- ✓ 문항 내용에 사실 오류나 어색한 표현은 없나요?
- ✓ 우리 반 학생 수준에 맞는 난이도인가요?
- ✓ O/X 버튼이 잘 눌리고 반응이 명확한가요?
- ✓ 글자 크기·색이 교실 화면(TV·빔)에서도 잘 보일까요?

불만족스러운 점을 두세 개 메모해 두세요 — 바로 다음 단계에서 고칩니다.

대화로 수정 — 개발자와 회의하듯 요청하기

수정은 새 프롬프트가 아니라 이어지는 대화로 합니다. 왼쪽 채팅창에 원하는 변경을 말하면 에이전트가 기존 게임을 고쳐 줍니다. 실제 캡처는 "배경을 바다 느낌으로 바꾸고, 버튼을 더 크게 해줘"라는 요청과 그 결과입니다 — 배경만 바뀐 것이 아니라 제목·캐릭터·아이콘까지 바다 테마로 일관되게 다시 디자인된 점에 주목하세요.

문항 교체

"3번 문제를 빼와 근육에 관한 문제로 바꿔줘"

난이도 조절

"문제를 조금 더 쉽게, 5학년 수준으로 바꿔줘"

디자인 변경

"배경을 바다 느낌으로, 버튼을 더 크게 해줘"

규칙 변경

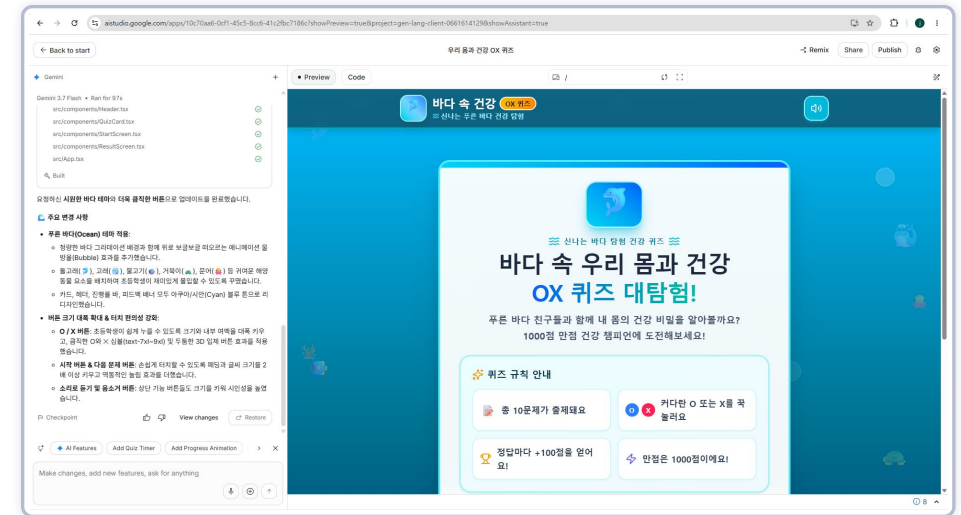
"제한 시간 10초 타이머를 문제마다 추가해줘"

효과 추가

"정답일 때 박수 소리와 폭죽 효과를 넣어줘"

한국어 다듬기

"문장을 초등학생에게 더 친근한 말투로 바꿔줘"



수정 결과 — "바다 속 우리 몸과 건강 OX 퀴즈 대탐험"으로 재탄생

배경을 바다 느낌으로 바꾸고, 버튼을 더 크게 해줘



요령: 한 번에 하나씩, 구체적으로. 되돌리려면 "방금 수정 취소하고 원래대로".

수정 요청을 입력한 채팅창 (부분)

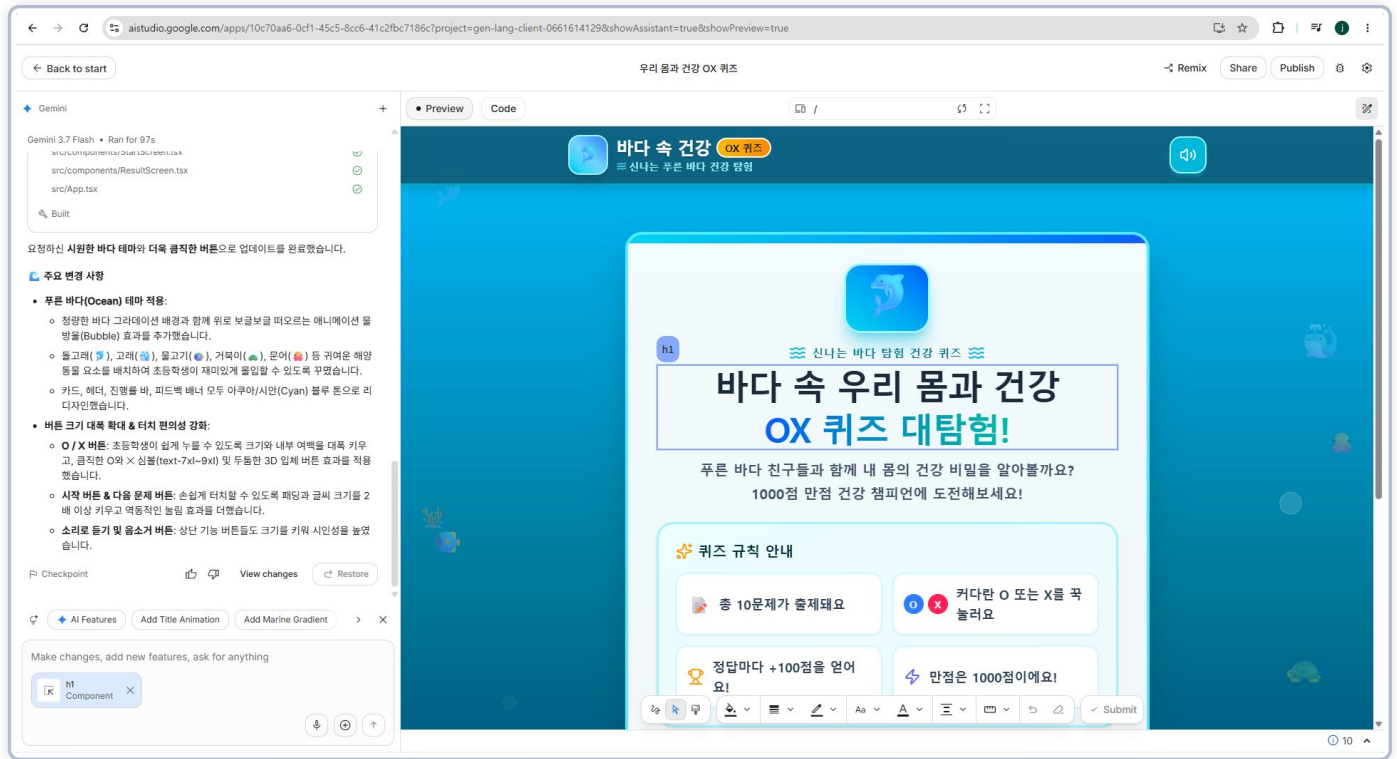
주석 모드 — 말 대신 화면을 짚어서 수정

"오른쪽 위의 그거"를 말로 설명하기 어려울 때가 있습니다. 미리보기 오른쪽 위의 연필 모양 아이콘을 누르면 주석 모드가 켜집니다. 실제 캡처에서 보듯 화면 요소를 클릭하면 파란 테두리로 선택 표시가 되고, 선택한 요소의 이름표(h1 Component)가 채팅창에 붙으며, 하단에 선택·형광펜·색상 등 주석 도구 막대가 나타납니다. 디자인 세부 수정에서 특히 강력합니다.



① 미리보기 오른쪽 위의 연필 아이콘 (부분)

- 1 연필 아이콘을 눌러 주석 모드 켜기
- 2 고칠 요소를 클릭 — 파란 테두리로 선택
- 3 요청 입력 — 요소 이름표가 함께 전달됨
- 4 Submit — 해당 부분만 정확히 수정됨

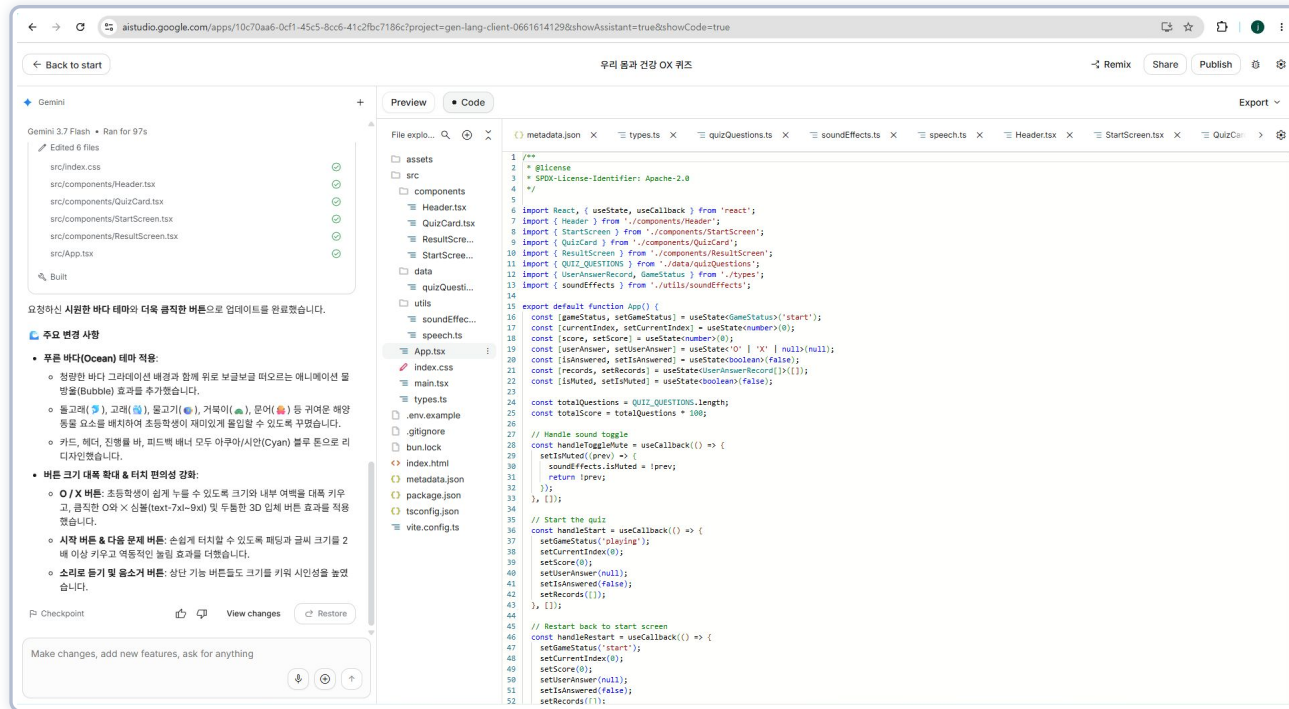


② 제목(h1) 요소를 선택한 모습 — 요소 이름표가 채팅창에 붙고, 하단에 주석 도구 막대가 열림

활용 예: "이 버튼만 노란색으로", "이 글자를 두 줄로 나눠줘"

Code 탭 구경 — 그리고 내 게임의 이름표

미리보기 옆의 Code 탭을 누르면 에이전트가 작성한 실제 코드가 보입니다. 실제 캡처에서 보듯 화면 구성(components), 문항 데이터(quizQuestions), 효과음(soundEffects) 등 파일이 역할별로 정리되어 있습니다. 읽지 못해도 전혀 상관없습니다 — "내 한국어 문장이 이만큼의 코드로 번역되었구나"를 확인하는 것으로 충분합니다.



Code 탭 — 파일 목록과 코드. 우리가 쓴 프롬프트의 번역본

우리 몸과 건강 OX 퀴즈

화면 위 가운데의 프로젝트 이름

이름 확인과 변경

AI가 알아서 "우리 몸과 건강 OX 퀴즈"라는 이름을 붙여 주었습니다. 화면 위 가운데의 제목을 클릭하면 직접 바꿀 수 있습니다.

이름 규칙 제안: 「과목_주제_게임형식」 예: 과학_우리몸_OX퀴즈. 작업물이 쌓일수록 이름 규칙의 힘이 커집니다.

작업은 자동 저장되며, 사이드바의 My apps와 History에서 언제든지 다시 열 수 있습니다.

미션 — 내 교과서의 OX 퀴즈로 변형하라

이제 배운 것을 내 것으로 만들 시간입니다. 아래 프롬프트 틀의 빈칸을 자신의 학년·과목·단원으로 채워 새 게임을 만들어 보세요. 완성하면 옆 사람과 서로의 게임을 플레이하고 한 가지씩 칭찬과 제안을 나눕니다.

> 미션 프롬프트 틀 (빈칸 채우기)

"[]학년용 OX 퀴즈 게임을 만들어줘.
주제는 [과목의 단원]이야.
문제가 []개 나오고, 정답이면 []점씩 올라가.
[우리 반 아이들이 좋아할 스타일 설명]으로 꾸며줘.
글자는 교실 TV에서도 잘 보이게 크게 해줘."

1. 빈칸 채워 새 게임 생성

사이드바 New app으로 새로 시작 — 기존 게임 대화에 이어 쓰지 않기

2. 생성된 문항 검토·수정

사실 오류 점검(교사의 몫!) 후 대화로 문항 다듬기

3. 짝과 교차 플레이

서로의 게임을 해보고 칭찬 1 + 제안 1 나누기

4. 이름 붙여 저장 확인

「과목_주제_OX퀴즈」 규칙으로 이름 바꾸기

시간이 남는다면 (심화) "문제를 쉬움 5개 + 어려움 5개로 나눠줘", "마지막에 틀린 문제만 다시 풀게 해줘" 같은 요청으로 게임을 발전시켜 보세요. 4단원에서 이 게임을 더 크게 키웁니다.

트러블슈팅 — 막히는 여섯 지점과 해법

실습 중 자주 발생하는 상황과 해결 순서입니다. 대부분 1분 안에 해결됩니다. 강사·연수 운영자는 이 슬라이드를 책상에 펼쳐 두면 좋습니다.

① 접속했는데 로그인 화면이 안 보여요

이미 다른 구글 계정으로 로그인된 상태일 수 있습니다. 오른쪽 위 프로필 아이콘 → 계정 전환에서 원하는 계정을 선택하세요.

② 학교 계정이라 차단된대요

개인 계정으로 전환하세요(2단원 체크리스트). 크롬 프로필을 개인용으로 새로 만들면 계정이 섞이지 않습니다.

③ 생성이 5분 넘게 멈춰 있어요

채팅창에 "계속해줘"라고 입력해 보세요. 그래도 멈춰 있으면 새로고침(F5) — 작업은 자동 저장되어 사라지지 않습니다.

④ 게임에 오류가 나요 / 버튼이 안 눌려요

오류 화면을 그대로 두고 채팅에 "오류가 났어, 고쳐줘"라고 입력하세요. 에이전트가 스스로 원인을 찾아 수정합니다.

⑤ 만들던 게임이 사라졌어요

사이드바의 My apps 또는 History를 여세요. 모든 작업이 자동 저장되어 있습니다. 프로젝트 이름을 붙여 두면 찾기 쉽습니다.

⑥ 문항 내용이 이상해요

정상입니다 — AI는 사실 오류를 낼 수 있습니다. "3번 문제가 틀렸어. ~로 바꿔줘"라고 구체적으로 수정을 지시하세요.



WORLD 3 CLEAR!

첫 게임 완성 — 3단원 핵심 정리



접속과 화면의 지도

aistudio.google.com 직접 입력 —
로그인하면 바로 빌드 홈. 사이드바 다섯
구역의 위치를 익혔습니다.



첫 게임 생성

대상·형식·규칙·스타일을 담은 프롬프트
하나로 OX 퀴즈 게임을 만들고 2~3분의
생성 과정을 관찰했습니다.



두 가지 수정 기술

대화로 고치기(한 번에 하나씩·구체적으로)
와 주석 모드(화면 짚기)를 손에 익혔습니다.



나의 교과 게임

미션으로 내 과목·단원의 OX 퀴즈를 만들고
이름을 붙여 저장했습니다.

NEXT ► WORLD 4 교과 맞춤 학습 게임 만들기 — 게임을 키웁니다



WORLD 4

교과 맞춤 학습 게임

퀴즈를 넘어 내 수업을 담은 게임으로

3단원에서 우리는 “게임을 만들 수 있다”는 것을 확인했습니다. 이번 단원의 질문은 한 단계 위입니다 — “좋은 학습 게임을 만들 수 있는가.” 1단원의 MDA를 기획 언어로 꺼내어 학습 목표를 게임 규칙으로 번역하고, 내 교과에 맞는 장르를 골라 본격 게임을 만듭니다. 단원의 끝에서 선생님의 게임은 “퀴즈”에서 “우리 반 수업 도구”가 됩니다.



교재 4단원

4단원 학습 퀘스트

4단원을 마치면 다음 네 가지를 할 수 있어야 합니다. 이번 단원의 핵심은 "기술"이 아니라 "설계"입니다 — 같은 도구로도 기획이 다르면 전혀 다른 수업 도구가 나옵니다.

QUEST 1



학습 목표를 게임 규칙으로 번역하는 미니 기획서를 쓸 수 있다

성취기준 → 핵심 행동 → 게임 메커닉(M) → 노리는 재미(A)의 연결을 한 장으로 정리합니다.

QUEST 2



내 교과에 맞는 게임 장르를 고르고 본격 게임을 생성할 수 있다

장르 카탈로그 6종에서 교과 특성에 맞는 형식을 골라 프롬프트로 구현합니다.

QUEST 3



게임 요소(타이머·콤보·레벨 등)로 재미를 의도적으로 강화할 수 있다

요소 메뉴판에서 2~3개를 골라 추가하고, 각 요소가 어떤 재미(미학)를 만드는지 설명합니다.

QUEST 4



난이도 곡선·배려 설계·AI 실시간 기능으로 게임을 완성한다

쉬움→어려움 배치, 다시 풀기, 저학년 배려를 더하고 Gemini 실시간 기능을 실험합니다.



이 단원의 재료

**3단원에서 만든
나의 OX 퀴즈**

업그레이드 재료로 쓰거나,
새 장르로 갈아탈 수도 있습니다.
어느 쪽이든 기획서가 먼저입니다

오늘의 업그레이드 여정 — 6단계

이번 단원은 "설계 → 제작 → 강화 → 완성"의 흐름입니다. 3단원과 달리 화면 조작보다 판단할 것이 많습니다 — 각 단계에서 여러분이 내리는 교육적 결정이 게임의 품질을 좌우합니다. 소요 시간은 여유 있게 90~120분입니다.

1



미니 기획서

학습 목표를 게임 규칙으로
번역 (MDA 실전)

2



장르 선택

교과 특성에 맞는
게임 형식 고르기

3



본격 게임 생성

기획서를 프롬프트로 바꿔
새 게임 만들기

4



게임 요소 강화

타이머·콤보·레벨 등
재미 장치 2~3개 추가

5



난이도·배려 설계

쉬움→어려움 곡선,
다시 풀기, 저학년 배려

6



플레이테스트

교사·학생 시점 점검과
동료 피드백 루틴

진행 팁: STEP 1~2(설계)는 종이에서, STEP 3~6(제작)은 AI Studio에서 이루어집니다. 설계에 시간의 3분의 1을 쓰는 것이 결과적으로 가장 빠른 길입니다.

미니 기획서 — 학습 목표를 게임 규칙으로 번역

프롬프트를 치기 전에 종이 한 장을 먼저 씁니다. 1단원 MDA를 실전에 투입하는 순간입니다 — 왼쪽 틀의 네 칸을 채우면 그것이 곧 프롬프트의 뼈대가 됩니다. 오른쪽은 실제 작성 예시입니다.

① 학습 목표 · 성취기준

이 게임으로 무엇을 배우게 할 것인가? 교육과정의 성취기준 한 줄로 시작합니다.

② 핵심 행동

학생이 게임 속에서 반복할 행동은? (고르기·맞히기·모으기·막기·배치하기...) 이 행동이 학습 행동과 겹칠수록 좋은 GBL입니다.

③ 게임 메커닉 (M)

핵심 행동을 담을 규칙 — 무엇을 하면 점수가 오르고, 무엇을 하면 실패하는가?

④ 노리는 재미 (A)

8분류 중 어떤 재미를 노리는가? 도전을 축으로 2~3개를 고릅니다.

핵심: ②와 학습 행동이 겹치는가 — "분수를 비교하는 게임"인가, "분수 문제 사이에 끼어든 오락"인가.

작성 예시 — 수학 3~4학년 "분수"

- ① 목표: 단위분수·진분수의 크기를 비교할 수 있다
- ② 핵심 행동: 두 분수 중 큰 쪽을 빠르게 고르기
- ③ 메커닉: 바다에 분수가 적힌 물고기 두 마리가 헤엄침 → 큰 분수를 낚으면 +100점, 작은 쪽을 낚으면 물고기가 도망감. 20마리 중 15마리 이상 잡으면 클리어
- ④ 재미: 도전(비교 판단) + 수집(어항이 채워짐) + 감각(낚는 손맛)

→ 이 네 줄이 그대로 다음 슬라이드의 프롬프트가 됩니다

장르 카탈로그 — 교과에 맞는 게임 형식 고르기

AI 에이전트는 어떤 장르든 만들 수 있으므로, 선택 기준은 기술이 아니라 교과 특성입니다. 반복 연습이 필요한 내용인가, 개념 관계를 다루는가, 탐구 과정인가에 따라 어울리는 형식이 다릅니다. 각 카드의 프롬프트 씨앗을 기획서 내용과 합치면 됩니다.



퀴즈 대전

전 교과 복습 · 가장 안전한 출발

"○○ 단위 4지선다 퀴즈 대전, 연속 정답 콤보"



수집형 (낚시·농장)

수학 연산 · 어휘 — 반복을 수집의 재미로

"정답을 맞히면 물고기를 낚아 어항에 모으는 게임"



타워 디펜스

구구단·낱말 등 자동화 훈련

"문제를 맞혀야 포탑이 발사되는 낱말 방어 게임"



카드 배틀

사회·역사·과학 개념 관계

"역사 인물 카드로 시대 순서 대결하는 게임"



방탈출

통합·프로젝트 · 추론 과정

"단서 3개를 풀어야 문이 열리는 과학 방탈출"



러너 · 점프

저학년 · 신체 리듬 + 간단 문제

"장애물 앞 문제를 맞혀야 점프하는 달리기 게임"

선택 기준 한 줄 — 반복 연습이면 수집형·디펜스, 개념 관계면 카드 배틀, 탐구 과정이면 방탈출, 처음이면 퀴즈 대전.

본격 게임 생성 — 기획서를 프롬프트로

기획서의 네 칸을 문장으로 잇기만 하면 프롬프트가 됩니다. 아래는 앞의 분수 기획서를 그대로 옮긴 실습 프롬프트입니다. 사이드바의 New app으로 새 프로젝트에서 시작하세요 — 3단원 게임의 대화에 이어 쓰면 두 게임이 섞입니다.

> 실습 2 프롬프트 (기획서 → 문장)

"초등 3~4학년 분수 비교 낚시 게임을 만들어줘.
바다에 분수가 적힌 물고기 두 마리가 헤엄치고,
더 큰 분수 물고기를 클릭하면 낚아서 +100점,
작은 쪽을 클릭하면 물고기가 도망가.
낚은 물고기는 화면 아래 어항에 모이고,
20마리 중 15마리를 잡으면 스테이지 클리어야.
바다 느낌의 시원한 디자인, 글자는 크게 해줘."

생성은 3단원과 동일합니다(입력 → Build → 2~3분 관찰 → 플레이). 첫 결과가 기획과 다르면 당황하지 말고 "어항이 안 보여. 화면 아래에 추가해줘"처럼 기획서와 다른 점부터 하나씩 맞춰 가세요.

기획서 ①·②

→ 첫 문장: 대상·과목·핵심 행동(비교해 고르기)

기획서 ③ (M)

→ 가운데: 점수 규칙·실패 조건·클리어 조건

기획서 ④ (A)

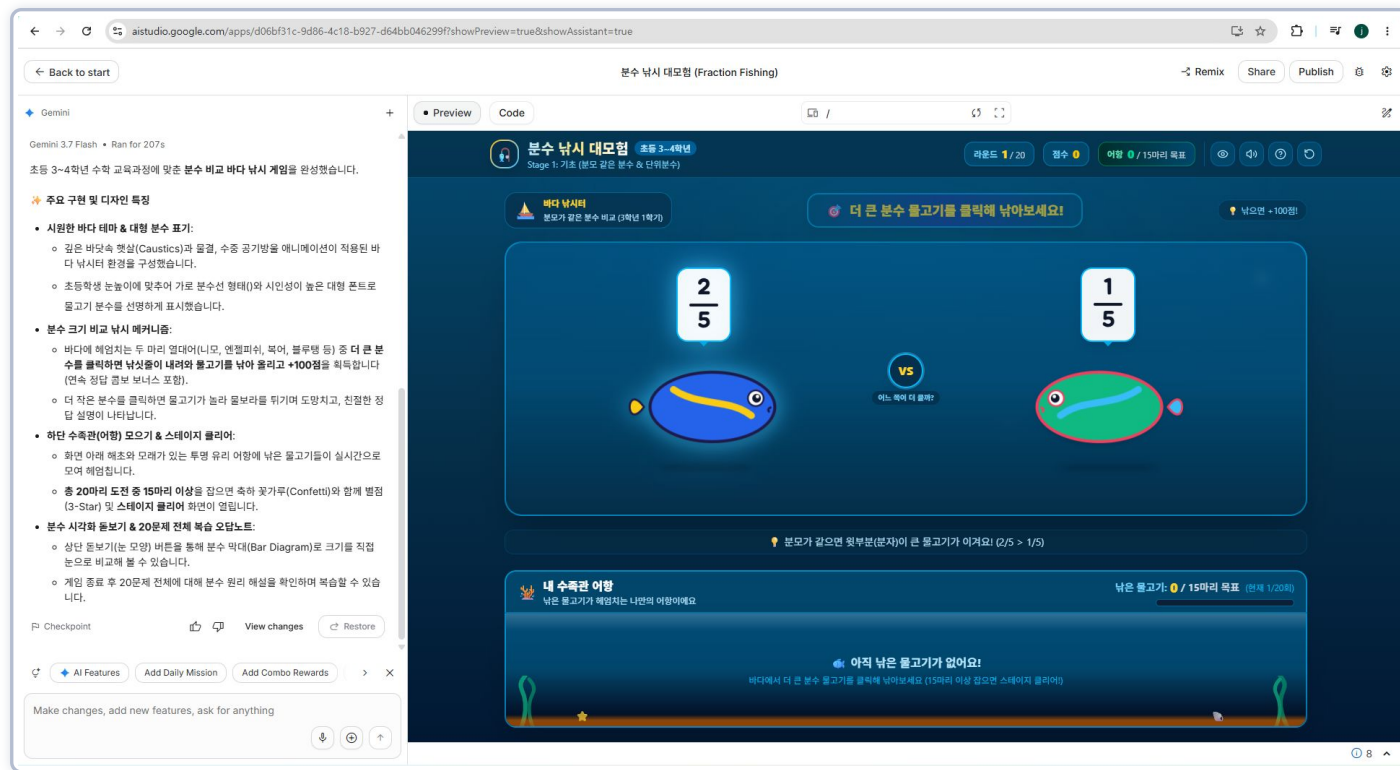
→ 마지막: 수집(어항)·감각(바다 디자인) 지정

문항 데이터를 직접 주고 싶다면

프롬프트 뒤에 "문제는 다음 목록을 사용해줘:"라고 쓰고 문항을 붙여넣으면 AI가 만든 문항 대신 내 문항이 들어갑니다. 우리 반 진도에 딱 맞추는 가장 확실한 방법입니다.

생성 결과 — 기획서와 대조하며 점검

완성된 게임을 기획서 옆에 놓고 대조합니다. 점검 질문은 3단원의 다섯 가지에 하나가 더해집니다 — "핵심 행동이 살아 있는가?" 학생이 이 게임에서 반복하는 행동이 정말 분수 비교인지, 아니면 클릭 반사신경인지 확인하는 것이 GBL 설계의 핵심 점검입니다.



생성된 분수 낚시 게임 (예시)

기획서 대조 점검

- ✓ ① 학습 목표 — 이 게임을 20분 하면 분수 비교가 늘까?
- ✓ ② 핵심 행동 — 학생이 반복하는 행동이 곧 학습 행동인가?
- ✓ ③ 메커닉 — 점수·실패·클리어 규칙이 기획대로 들어갔나?
- ✓ ④ 재미 — 노린 재미(도전·수집·감각)가 실제로 느껴지나?
- ✓ ⑤ 문항 — 분수 표기와 크기 비교에 오류는 없나? (교사 검수!)

기획서와 다른 점을 메모하고 하나씩 대화로 맞춰 가세요 — 이것이 STEP 4의 준비물입니다.

게임 요소 강화 — 재미 장치 메뉴판

이제 1단원에서 배운 보상 설계를 직접 사용합니다. 아래 여덟 요소에서 2~3개만 골라 한 번에 하나씩 추가하세요. 요소마다 만들어내는 재미(미학)가 다르므로, 기획서 ④에서 노린 재미와 맞는 것을 고르는 것이 요령입니다. 많이 넣을수록 좋은 게 아닙니다 — 초점이 흐려지면 학습도 재미도 잃습니다.

타이머 긴장감 · 도전

"문제마다 15초 제한 시간을 추가해줘"

콤보 몰입 · 성취

"연속 정답이면 콤보 보너스 점수를 넣어줘"

생명 신중함 · 긴장

"목숨 3개 시스템으로 바꿔줘"

레벨 성장 · 발견

"5문제마다 레벨이 오르고 배경이 바뀌게 해줘"

효과음·BGM 감각

"정답/오답 효과음과 배경음악을 넣어줘"

배지·보상 수집 · 성취

"만점이면 황금 트로피 배지를 주게 해줘"

랜덤 출제 반복 플레이

"문제 순서를 매번 섞어줘"

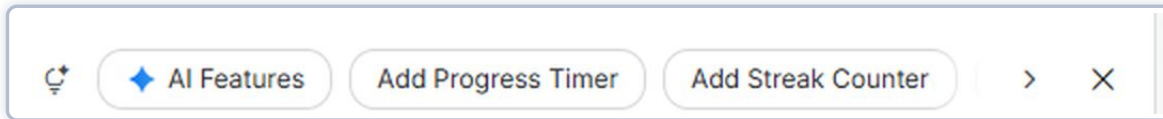
다시 풀기 학습 완성

"틀린 문제만 마지막에 다시 풀게 해줘"

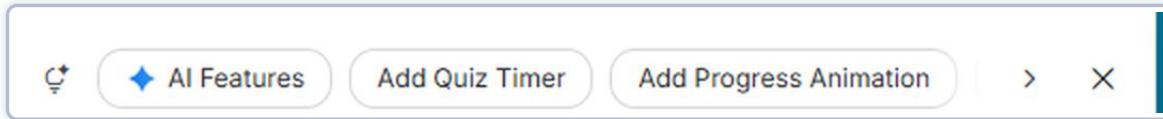
주의 — 보상 설계의 윤리 1단원에서 해부한 보상 루프를 이제 우리가 설계합니다. 학습을 돕는 보상(다시 풀기·성취 배지)과 시간만 붙잡는 보상(무한 랜덤 상자)의 차이를 의식하세요. 우리의 목표는 몰입이지 중독이 아닙니다.

강화 실전 — AI의 제안 칩과 되돌리기 안전장치

요소를 직접 요청하는 것 외에, AI Studio가 먼저 제안하는 길도 있습니다. 채팅창 위에는 지금 게임에 어울리는 기능 제안 칩이 자동으로 나타납니다 — 실제 캡처에서 보듯 OX 퀴즈에는 진행 타이머·연속 정답 카운터를, 바다 테마에는 퀴즈 타이머·진행 애니메이션을 제안했습니다. 칩을 누르면 해당 기능이 바로 추가됩니다.



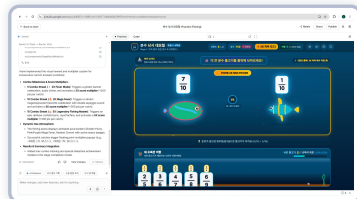
기능 제안 칩 ① — Add Progress Timer · Add Streak Counter



기능 제안 칩 ② — 게임이 바뀌면 제안도 바뀝니다



응답마다 생기는 Checkpoint — Restore로 그 시점 복원



강화 결과 — 콤보 5연속, 점수 배수 피버 모드(2X)가 켜진 실제 화면

제안 칩 활용법

칩은 편리하지만 "제안"일 뿐입니다. 누르기 전에 기획서 ④(노리는 재미)와 맞는지 한 번 판단하세요. 맞으면 클릭 한 번으로 강화 완료, 아니면 무시하면 됩니다. 아이디어가 고갈됐을 때 훑어보는 메뉴판으로도 좋습니다.

체크포인트 = 실험을 겁내지 않을 자유

에이전트의 응답마다 Checkpoint가 자동 저장됩니다. 새 요소를 넣었다가 게임이 이상해지면 해당 응답의 Restore 버튼으로 그 시점 상태를 복원하면 됩니다. View changes로 무엇이 바뀌었는지 확인할 수도 있습니다. 과감하게 실험하세요 — 언제든지 되돌아갈 수 있습니다.

난이도 곡선과 배려 설계 — 모두가 끝까지 하는 게임

좋은 학습 게임은 잘하는 학생만 살아남는 게임이 아니라, 모든 학생이 끝까지 도전하는 게임입니다. 난이도 곡선과 배려 장치는 게임 완성도의 마지막 조각이자, 교사만이 설계할 수 있는 부분입니다.



난이도 곡선

"문제를 쉬움 5개 → 보통 3개 → 어려움 2개 순서로 배치해줘" — 초반 성공 경험이 끝까지 가는 힘이 됩니다. 카이와의 아곤(경쟁)은 이길 가능성이 있을 때만 재미있습니다.



난이도 선택권

"시작할 때 쉬움/보통/어려움 모드를 고르게 해줘" — 같은 게임을 수준별 과제로 만드는 가장 간단한 방법입니다.



틀린 문제 재도전

"틀린 문제만 마지막에 다시 풀게 하고, 두 번째에 맞으면 절반 점수를 줘" — 실패를 학습 기회로 되돌리는 장치입니다.



저학년·읽기 배려

"문제를 소리로 읽어주는 버튼을 넣어줘"(TTS), "글자를 더 크게, 문장을 짧게" — 읽기 속도가 게임 실력이 되지 않도록 합니다.



색·대비 배려

"빨강-초록 대신 파랑-주황으로 구분하고, 모양 차이도 함께 줘" — 색각 이상 학생도 구분할 수 있는 신호 설계입니다.

이 다섯 가지는 특수교육 대상 학생만을 위한 것이 아닙니다 — 배려 설계는 언제나 모든 학생의 경험을 좋게 만듭니다(보편적 학습 설계).

AI 실시간 기능 — 문제가 무한히 샘솟는 게임

지금까지의 게임은 만들 때 정해진 문항을 보여주는 "정적" 게임이었습니다. AI Studio로 만든 앱에는 Gemini를 호출하는 기능을 넣을 수 있어서, 게임이 플레이 중에 실시간으로 문제를 만들거나 학생의 답을 채점하게 할 수 있습니다. 정적 문항과는 차원이 다른 가능성이 열립니다.

무한 문제 은행

"매 판마다 Gemini가 새로운 분수 비교 문제를 만들어 내게 해줘" — 문제가 소진되지 않아 반복 플레이가 가능해집니다.

서술형 채점 코치

"학생이 문장으로 답을 쓰면 Gemini가 채점하고 한 줄 조언을 주게 해줘" — 객관식을 넘어 서술형 게임이 됩니다.

스무고개 AI 솔래

"Gemini가 과학 개념 하나를 마음속에 정하고, 학생의 예/아니오 질문에 답하는 스무고개 게임을 만들어줘"

끝말잇기 심판

"학생과 Gemini가 번갈아 낱말을 잇고, Gemini가 규칙 위반을 판정하게 해줘" — 국어 어휘 수업의 단골 손님

실습 확인 포인트 — 기능이 켜졌는지 이렇게 확인

요청 후 2~3판을 연속으로 플레이해 문제가 매번 달라지는지 확인하세요. 문제가 나올 때 잠깐의 로딩이 생기는 것은 Gemini를 호출하는 정상 동작입니다. 수준을 벗어난 문제가 나오면 "3~4학년 교육과정 범위 안에서만"이라고 범위를 좁혀 다시 요청합니다.

유의점

AI가 실시간 생성하는 문제는 교사가 미리 검수할 수 없습니다. 학년 수준과 소재 범위를 프롬프트에 명확히 제한하고("3학년 수준, 교과서 어휘만"), 수업 전 반드시 여러 판 시험 플레이하세요. 미리보기에서는 바로 동작하며, 외부 배포 시의 설정은 5단원에서 다룹니다.

플레이테스트 — 두 개의 시점으로 검증하기

게임 개발의 마지막 단계는 언제나 플레이테스트입니다. 교사 시점과 학생 시점, 두 렌즈로 점검한 뒤 동료와 교차 테스트합니다. MDA로 말하면 — 설계자가 만든 규칙(M)이 의도한 재미(A)로 도착했는지를 플레이어의 눈으로 확인하는 과정입니다.

교사 시점 점검 (학습)

- ✓ 문항의 사실 오류·표기 오류가 없는가 (전 문항 직접 확인)
- ✓ 핵심 행동이 학습 행동과 일치하는가
- ✓ 한 판의 길이가 수업 흐름에 맞는가 (5~10분 권장)
- ✓ 가장 느린 학생도 끝까지 갈 수 있는가 (배려 설계 확인)

학생 시점 점검 (재미)

- ✓ 첫 30초 안에 무엇을 해야 하는지 알 수 있는가
- ✓ 실패했을 때 "다시 하고 싶은가", 아니면 "그만두고 싶은가"
- ✓ 점수·보상이 오를 때 기분이 좋은가 (효과가 체감되는가)
- ✓ 노린 재미 2~3개가 실제로 느껴지는가

동료 교차 테스트 프로토콜 (연수 현장용, 10분)

① 옆 사람과 게임을 바꿔 3분씩 플레이합니다 (만든 사람은 설명 금지 — 첫 30초 테스트를 겹칩니다). ② "칭찬 2개 + 제안 1개"를 서로 교환합니다. ③ 받은 제안 중 하나를 골라 즉시 대화로 수정합니다. 수정 전 상태는 Checkpoint가 지켜주고 있으니 과감하게 반영해 보세요. 이 루틴은 그대로 학생 게임 제작 수업의 상호 평가 활동으로 옮겨 쓸 수 있습니다.

미션 — 내 수업의 본격 게임을 완성하라

이번 미션은 3단원보다 깁니다. 기획서부터 플레이테스트까지 전 과정을 자신의 교과로 한 바퀴 돌립니다. 결과물은 6단원까지 계속 발전시킬 "나의 대표 게임"이 됩니다.

1

미니 기획서 작성 (10분)

성취기준 한 줄 → 핵심 행동 → 메커닉 → 노리는 재미 2~3개. 종이에 먼저 씁니다.

2

장르 선택 + 생성 (15분)

카탈로그에서 장르를 고르고, 기획서를 프롬프트로 옮겨 New app에서 생성합니다.

3

게임 요소 강화 (15분)

메뉴판·제안 칠판에서 2~3개를 골라 하나씩 추가합니다. 이상해지면 Restore.

4

난이도·배려 설계 (10분)

난이도 곡선 또는 모드 선택 + 배려 장치 1개 이상을 반영합니다.

5

플레이테스트 (10분)

두 시점 점검 후 동료 교차 테스트 — 칭찬 2, 제안 1, 즉시 수정 1.

완성 기준

- ✓ 내 교과 내용이 담겨 있다
- ✓ 게임 요소 2개 이상이 작동한다
- ✓ 배려 장치가 1개 이상 있다
- ✓ 전 문항을 교사가 검수했다
- ✓ 동료 1명이 플레이해 보았다
- ✓ 「과목_주제_장르」로 이름 저장

이 게임은 5단원에서 학생들에게 배포되고, 6단원에서 실시간 리더보드가 붙습니다 — 오늘의 완성도가 남은 여정의 출발선입니다.

교실 적용 — 언제, 어떻게 쓰는 게임인가

게임이 완성되어도 "수업 어디에 넣을 것인가"는 별도의 설계입니다. 같은 게임도 배치에 따라 역할이 달라집니다. 아래 세 가지 기본 배치와 두 가지 운영 방식을 조합하면 대부분의 수업 상황을 커버할 수 있습니다.



도입 5분 — 동기 유발

전 차시 복습 겸 오늘 주제 맛보기. 쉬움 모드로 전원 성공 경험을 주고 시작합니다.



정리 10분 — 형성평가

오늘 배운 내용으로 한 판. 결과 화면의 점수·오답이 곧 형성평가 데이터가 됩니다.



단원 마무리 — 복습 대회

단원 전체 문항으로 확장해 모둠 대항전. 6단원의 리더보드가 여기서 빛납니다.

운영 ① 전체 화면 (교사 1대)

TV·빔에 게임을 띄우고 모둠별로 번갈아 답합니다. 기기가 없어도 되고 전체 분위기가 뜨겁습니다. 골든벨식 진행에 적합 — 글자 크기를 미리 크게 해둔 이유입니다.

운영 ② 개별 기기 (태블릿·노트북)

학생마다 자기 속도로 플레이합니다. 개별 연습·수준별 과제에 적합하며, 링크 배포가 필요합니다 — 바로 다음 5단원에서 URL·QR 배포를 실습합니다.

지도안 한 줄 틀: "[단원]의 [성취기준] 도달을 위해 [게임명]을 [배치]에 [운영 방식]으로 활용하여 [핵심 행동]을 반복 연습한다."

트러블슈팅 — 강화 단계에서 자주 막히는 지점

요소를 추가하는 단계에서는 3단원과 다른 유형의 문제가 나타납니다. 공통 해법은 두 가지 — 하나씩 요청하기, 그리고 Checkpoint의 Restore입니다.

① 기능을 추가했더니 게임이 망가졌어요

해당 요청 이전 응답의 Restore 버튼을 누르면 그 시점으로 복원됩니다. 그다음 요청을 더 작게 쪼개서 다시 시도하세요.

② 여러 기능을 한꺼번에 요청했더니 이상해요

"타이머와 콤보와 레벨을 넣어줘"보다 세 번에 나눠 요청하세요. 하나 넣고 플레이 확인, 또 하나 — 가 가장 빠릅니다.

③ 그림·캐릭터가 어색해요

"캐릭터를 이모지로 바꿔줘" 또는 "도형과 색으로만 단순하게 표현해줘"가 안정적입니다. 저작권 걱정도 함께 사라집니다.

④ 게임이 무겁고 느려져요

"애니메이션 효과를 줄여서 가볍게 만들어줘"라고 요청하세요. 교실의 저사양 기기일수록 단순한 게임이 좋습니다.

⑤ 수정을 반복했더니 대화가 너무 길어요

완성 시점마다 이름을 붙여 두고, 큰 방향 전환은 New app에서 새로 시작하는 편이 결과가 깨끗합니다.

⑥ 아이디어가 안 떠올라요

채팅창 위 제안 칩을 훑어보거나, Gallery에서 다른 앱을 리믹스해 구조만 빌려오세요. 기획서의 ④(재미)로 돌아가는 것도 좋습니다.



WORLD 4 CLEAR!

나의 대표 게임 완성 — 4단원 핵심 정리



설계가 먼저다

성취기준→핵심 행동→메커닉→재미의 미니 기획서. 프롭프트는 기획서의 번역일 뿐입니다.



장르는 교과가 고른다

반복 연습=수집형·디펜스, 개념 관계=카드 배틀, 탐구=방탈출 — 기술이 아니라 교과 특성이 기준입니다.



강화는 2~3개, 하나씩

요소 메뉴판과 제안 칩, 그리고 Checkpoint Restore. 물입을 설계하되 중독은 설계하지 않습니다.



모두가 끝까지

난이도 곡선·다시 풀기·TTS·색 배려 — 배려 설계는 모든 학생의 경험을 좋게 만듭니다.

NEXT ► WORLD 5 온라인 출판 — 내 게임을 학생들에게 배포합니다



WORLD 5

온라인 출판

내 게임에 주소를 달아주기

4단원까지 우리는 “내 컴퓨터 화면 안에서만 도는 게임”을 만들었습니다. 이 단원의 목표는 단 하나입니다. 그 게임에 인터넷 주소를 붙여, 학생이 교실 태블릿이든 집 스마트폰이든 링크 하나로 열 수 있게 만드는 것. 설치도, 파일 전달도, 로그인도 필요 없습니다. 주소만 있으면 됩니다.



5

교재 5단원

5단원 퀘스트 — 무엇을 얻어 가나요?

"선생님, 그 게임 집에서도 해요?" — 4단원까지 만든 게임을 학생이 실제로 쓰려면 반드시 넘어야 하는 문턱이 출판(publishing)입니다. 이 단원은 그 문턱을 네 가지 방법으로 넘어보고, 각 방법이 교실의 어떤 상황에 맞는지 판단하는 눈을 기릅니다.

1 출판 방법을 비교한다

Share 링크 · Publish(Cloud Run) · GitHub Pages · ZIP 배포 네 가지의 비용·수명·AI 기능 지원 여부를 구분한다.

2 내 게임의 갈림길을 판단한다

AI 기능(Gemini 호출)이 들어간 게임인지 아닌지에 따라 갈 길이 달라진다는 원리를 이해하고 스스로 선택한다.

3 GitHub를 혼자 연다

계정·저장소·커밋·공개 범위까지, 교사에게 필요한 만큼만 실제 화면을 눌러 가며 익힌다.

4 코드를 저장소에 올린다

AI Studio의 GitHub 동기화로 Push하고, github.com에서 파일이 실제로 올라갔는지 눈으로 확인한다.

5 Pages로 URL을 만든다

Settings → Pages에서 게시 방식을 고르고, 무료·영구 주소를 발급받아 접속을 확인한다.

6 안전하게 배포하고 회수한다

QR·학급 게시판으로 학생에게 달게 하고, 문제가 생기면 즉시 닫는 절차까지 갖춘다.

8단계 출판 로드맵

총 100분 · 4단원까지 만든 게임을 그대로 재료로 씁니다

아래 8단계는 순서대로 밟되, 학교 사정에 따라 갈라집니다. GitHub 계정 개설이 어려운 환경이라면 2~3단계(공유 링크·Cloud Run 게시)만으로도 학생 배포는 충분히 가능합니다. 4단계부터는 "영구적이고 내 소유인 주소"를 원할 때 밟는 길입니다.

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6	STEP 7	STEP 8
출판 4종 비교	공유 링크	Cloud Run 게시	GitHub 익히기	코드 올리기	Pages로 URL	학생 배포	관리와 회수
무엇이 있는지 지도 보기	Share — 동료와 즉시 공유	Publish — AI 기능 유지	계정·저장소·공 개 범위	AI Studio에서 Push	게시 방식 고르고 켜기	QR · 게시판 · 수업 배치	고쳐 올리기 · 즉시 닫기
10분	8분	10분	15분	15분	18분	14분	10분



분기점 안내 — STEP 3(Cloud Run)과 STEP 6(GitHub Pages)은 대체 관계가 아니라 용도가 다릅니다. AI가 실시간으로 말을 걸거나 문제를 만들어주는 게임은 STEP 3, 코드만으로 완결되는 게임은 STEP 6이 정답입니다. 5쪽에서 그 이유를 다룹니다. 또 GitHub 계정 개설은 사전 과제로 돌리면 연수 시간을 15분 아낄 수 있습니다.

왜 "파일"이 아니라 "주소"여야 할까

많은 선생님이 자료를 만들면 파일로 나눠줍니다. 그러나 학습 게임은 파일로 배포하는 순간 절반이 죽습니다. 기기마다 열리는 앱이 다르고, 수정하면 다시 전부 나눠줘야 하며, 누가 어디까지 받았는지 추적할 수 없습니다. 주소는 이 세 문제를 한 번에 해결합니다.

배포 방식	파일로 나눠주기 (.html, .zip)	주소로 열게 하기 (URL)
학생 기기	기기·브라우저마다 다르게 열림. 모바일에서 zip 해제 불가	주소창에 붙여넣으면 어디서나 동일하게 열림
수정 반영	고칠 때마다 전원에게 다시 배포	내가 한 번 고치면 모두가 즉시 최신 버전
가정 연계	가정 기기로 옮기기 번거로움	문자·클래스룸에 링크 한 줄, QR 한 장
회수·중단	이미 받은 파일은 회수 불가	게시를 끄거나 공개 범위를 되돌리면 즉시 닫힘
교사 부담	기기 문제 문의가 교사에게 집중	"주소 다시 알려주세요" 수준으로 감소

+ 교실 관찰 — 링크 배포로 바꾼 뒤 가장 크게 줄어드는 것은 "안 열려요"라는 말입니다. 수업 시간의 도입 5분이 통째로 돌아옵니다.

출판 방법 4종 비교 — 지도부터 펴기

AI Studio로 만든 앱을 세상에 내보내는 길은 크게 네 가지입니다. 네 가지는 우열이 아니라 용도가 다릅니다. 아래 표에서 "AI 기능" 열과 "수명" 열을 특히 눈여겨보세요. 이 두 칸이 교실에서의 선택을 결정합니다.

방법	한 줄 정의	AI 기능	비용	수명·소유	이럴 때
① Share 링크	AI Studio 안에서 앱을 다른 사용자에게 공유. 상대는 코드를 보고 복제(fork)할 수 있음	살아있음	무료 (내 사용량 차감)	AI Studio 계정에 종속	동료 교사와 빠른 공유, 연수·수업 나눔
② Publish (Cloud Run)	Publish 버튼 3클릭으로 Cloud Run에 게시. API 키는 서버 쪽에 안전 보관	살아있음	Starter Tier 앱 2개까지 무료	Google Cloud 위, 커스텀 주소 가능	AI가 실시간으로 작동하는 게임을 학생에게 배포
③ GitHub + Pages	코드를 내 저장소에 올리고 정적 사이트로 공개. 무료·영구 URL	작동 안 함 (정적 호스팅)	무료	내 GitHub 계정 소유, 영구	코드만으로 완결되는 게임, 오래 쓸 학급 자료
④ ZIP 내려받기 + 외부 호스팅	코드를 내려받아 학교 홈페이지·다른 호스팅에 올림	설정에 따라 다름	무료~유료	올린 곳에 종속	학교 서버 정책상 외부 서비스가 막힌 경우

내 게임은 어느 길로? — 딱 하나만 물어보세요

"이 게임은 실행 중에 AI에게 말을 거는가?" 이 질문 하나로 갈립니다. AI에게 말을 건다는 것은 코드가 Gemini API를 호출한다는 뜻이고, 그 호출에는 열쇠(API 키)가 필요합니다. 정적 호스팅인 GitHub Pages에는 이 열쇠를 안전하게 숨길 곳이 없습니다.

내 게임은 실행 중 AI를 호출하는가?

YES — AI가 실시간으로 작동

예) 학생 답을 AI가 채점·피드백, AI가 매번 새 문제 출제, AI 캐릭터와 대화, TTS 음성 읽어주기

→ STEP 3 Publish (Cloud Run)

API 키가 서버 쪽에 보관되어 학생에게 노출되지 않습니다. Starter Tier는 Google Cloud 프로젝트·결제 설정 없이 앱 2개까지 게시할 수 있습니다.

NO — 코드만으로 완결

예) 4단원 분수 낚시 대모험, OX 퀴즈, 타워 디펜스 — 문항·규칙이 코드 안에 이미 들어 있는 게임

→ STEP 6 GitHub Pages

호출할 AI가 없으니 숨길 열쇠도 없습니다. 무료로, 기한 없이, 내 계정 소유의 주소를 갖게 됩니다. 우리 교재의 실습은 이 길로 갑니다.

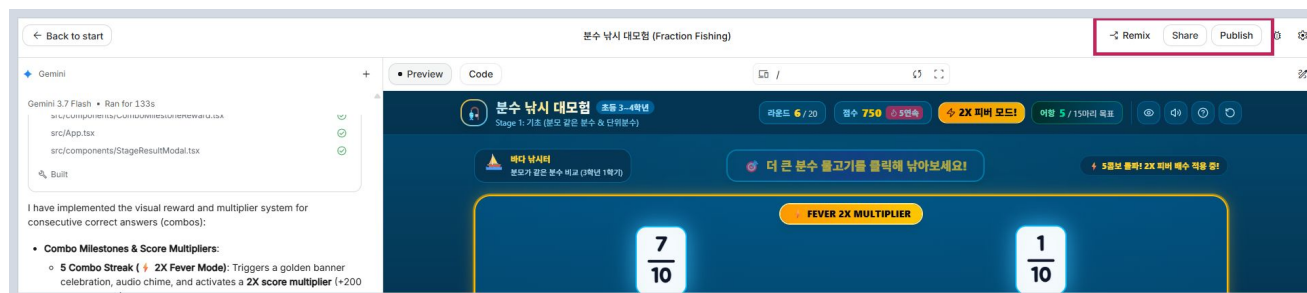


API 키 경고 — AI 호출 코드를 그대로 GitHub 공개 저장소에 올리면 키가 코드 안에서 그대로 읽힙니다. 남이 내 키로 요금을 쓰게 됩니다. AI 기능이 있는 앱은 반드시 Publish(Cloud Run) 쪽으로 보내세요.

앱 상단 툴바 — 세 버튼이 각각 무엇인가

Remix · Share · Publish

앱을 만들면 화면 위쪽 오른쪽에 버튼 세 개가 나타납니다(분홍 표시). 이름이 비슷해 헷갈리기 쉬운데, 하는 일은 완전히 다릅니다. 세 버튼의 차이를 먼저 정리하고 시작합니다.



앱 상단 툴바 — 오른쪽 끝에 Remix · Share · Publish (분홍 표시)

Remix

이 앱을 복제해 내 것으로 만들기. 내 앱을 원본은 두고 변형할 때 씁니다. 출판과는 무관합니다.

Share

AI Studio 안에서 다른 사용자에게 앱을 열어줍니다. 상대는 실행 화면과 코드를 함께 봅니다.

Publish

실제 인터넷 주소로 게시합니다. Cloud Run에 올라가고 AI 기능이 살아 있는 배포가 됩니다.

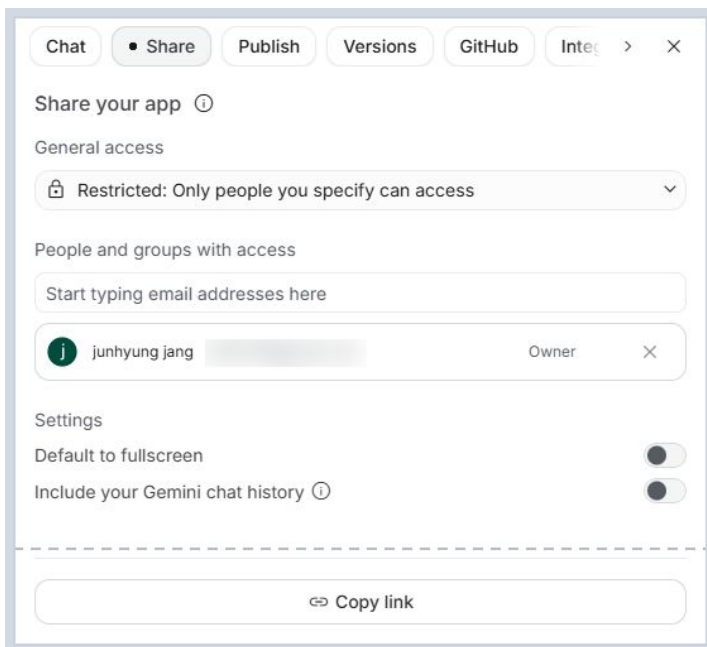
버튼이 안 보이면 — 3단원에서 배운 Preview / Code 탭 중 Preview 쪽인지 확인하세요.

i

오른쪽 위 톱니바퀴를 누르면 같은 기능이 Chat / Share / Publish / Versions / GitHub / Integrations / Secrets 탭으로도 열립니다. 5단원에서는 이 중 Share · Publish · GitHub 세 탭만 씁니다.

공유 링크 만들기 — 가장 빠른 한 걸음

앱은 기본이 비공개(private)입니다. Share는 그중 가장 가벼운 방법으로, 지정한 사람에게만 앱을 열어줍니다. 다만 공유받은 사람은 코드를 볼 수 있고 자기 것으로 복제할 수도 있습니다 — 이 점을 알고 쓰면 동료 나눔에는 최적입니다.



Share 대화상자 (이메일 흐름 처리)

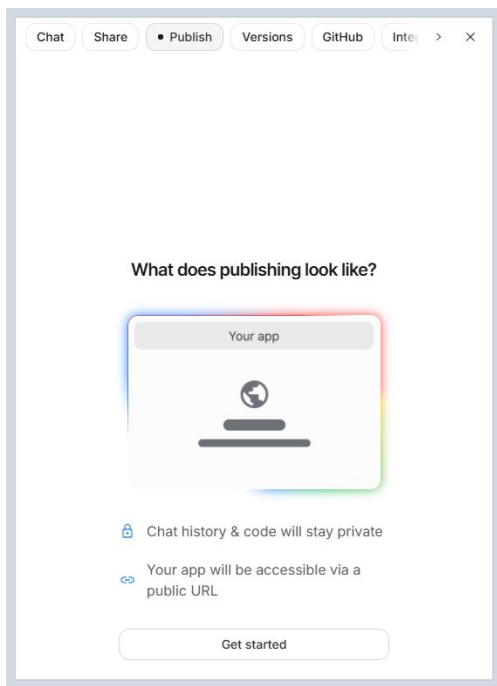
- 1 **상단 Share 클릭**
대화상자가 열립니다.
- 2 **General access 확인**
기본은 "Restricted: Only people you specify can access" — 지정한 사람만 접근입니다.
- 3 **참여자 추가**
이메일을 입력해 사람을 더합니다. 소유자(Owner)는 나입니다.
- 4 **Settings 확인**
Default to fullscreen(전체 화면으로 열기), Include your Gemini chat history(내 대화 기록 포함) 토글이 있습니다. 두 번째는 꺼 두는 편이 안전합니다.
- 5 **맨 아래 Copy link**
링크를 복사해 전달합니다.

! 학생 배포에는 권하지 않습니다 — ① 상대가 코드를 볼 수 있고 ② 공유한 앱의 API 호출이 내 사용량으로 차감되며 ③ 30명에게 각각 권한을 주는 일이 번거롭습니다. 동료 교사·연수 참가자와 나누는 용도로 쓰세요.

Publish — 진짜 인터넷 주소로 게시하기

Cloud Run 배포 · Starter Tier는 결제 설정 없이 앱 2개까지

Publish는 앱을 Google Cloud Run에 올려 실제 서비스처럼 돌립니다. AI 호출에 쓰는 API 키가 서버 쪽에 보관되므로 학생에게 노출되지 않습니다. Starter Tier 덕분에 Google Cloud 프로젝트를 따로 만들거나 결제 계정을 등록하지 않아도 앱 두 개까지 게시할 수 있습니다.



Publish 패널

① 3클릭이면 끝

Publish → Get started → Publish App. 배포가 끝나면 접속용 Cloud Run URL이 발급됩니다.

② 패널의 두 줄

"Chat history & code will stay private"와 "Your app will be accessible via a public URL" — 대화 기록과 코드는 비공개, 앱만 공개 주소로 열린다는 뜻입니다.

③ 커스텀 주소

ai.studio 하위 도메인을 붙일 수 있습니다. 전 세계에서 유일해야 하며 선착순 배정입니다.

④ Starter Tier 한도

앱 2개까지, 단일 리전 배포. 이미 Google Cloud 결제 계정을 쓰는 사용자는 대상이 아닙니다.

⑤ 학교에서 주의

학교·교육청 망에서 Cloud Run 도메인이 막히지 않는지 수업 전날 교실 기기 한 대로 접속 테스트를 하세요.



AI를 실시간 호출하는 기능(음성 읽어주기, AI 문제 생성 등)을 넣었다면 이 STEP 3이 최종 배포 경로입니다. 그런 기능이 없다면 GitHub Pages가 더 오래갑니다.

Cloud Run 주소를 받은 다음 — 챙길 것 다섯

Publish가 끝나면 주소가 나옵니다. 여기서 끝내지 말고 아래 다섯 가지를 확인해야 수업 당일 사고가 없습니다. 특히 ①과 ⑤는 실제 연수에서 가장 자주 문제가 된 항목입니다.

① 교실 기기로 접속 테스트

내 노트북에서 되는 것과 교실 태블릿에서 되는 것은 다릅니다. 학교 망 필터링에 Cloud Run 도메인이 걸리는 사례가 있습니다. 수업 전날 반드시 한 대로 확인하세요.

② 첫 로딩 시간 확인

Cloud Run은 접속이 없으면 잠들었다가 첫 요청에서 깨어납니다. 첫 학생이 몇 초 더 기다릴 수 있습니다. 수업 시작 전 교사가 먼저 한 번 열어 두면 해결됩니다.

③ 앱 2개 한도 관리

Starter Tier는 게시 앱 2개까지입니다. 세 번째를 올리려면 기존 게시를 내려야 합니다. 학기 중 여러 게임을 돌린다면 미리 계획하세요.

④ 주소를 기록해 둘 것

주소는 길고 기억하기 어렵습니다. 학급 자료 폴더나 수업 지도안에 바로 적어 두세요. 다음 학년도에 다시 찾을 때 고생합니다.

⑤ AI 기능이 실제로 도는지

배포본에서 AI 기능(음성·자동 채점 등)을 직접 눌러 보세요. 개발 화면에서는 되고 배포본에서는 안 되는 경우가 있습니다.

= AI 기능이 없는 게임이라면 여기서 멈추지 말고 다음 쪽(GitHub)으로 가세요 — 더 오래 가는 무료 주소가 기다립니다.

GitHub 기초 — 교사를 위한 최소한의 개념

용어 4개만 알면 충분합니다

GitHub는 프로그래머의 세계처럼 보이지만, 교사에게 필요한 개념은 네 개뿐입니다. 아래 비유는 교실에서 학생에게 설명할 때도 그대로 쓸 수 있습니다 — 6학년 학생도 "학급 문집 보관함"이라고 하면 이해합니다.

저장소 (Repository)

프로젝트 하나가 통째로 들어가는
폴더

비유 ▶ 학급 문집을 보관하는 서랍 한 칸. 게임 하나에 서랍 하나를 씁니다.
(이번 실습의 서랍 이름: adventure_math)

커밋 (Commit)

"여기까지 저장" 하고 도장 찍는
행위

비유 ▶ 문집을 고칠 때마다 "3월 12일, 표지 색 바꿈"이라고 적어둔 기록장.
언제든 그 시점으로 되돌아갈 수 있습니다.

푸시 / 풀 (Push / Pull)

내 작업을 올리기 / 남의 작업
내려받기

비유 ▶ Push = 내 서랍의 내용을 인터넷 서랍에 올림. Pull = 인터넷 서랍의
최신본을 내 쪽으로 가져옴.

공개 / 비공개 (Public / Private)

누구나 볼 수 있는지 여부

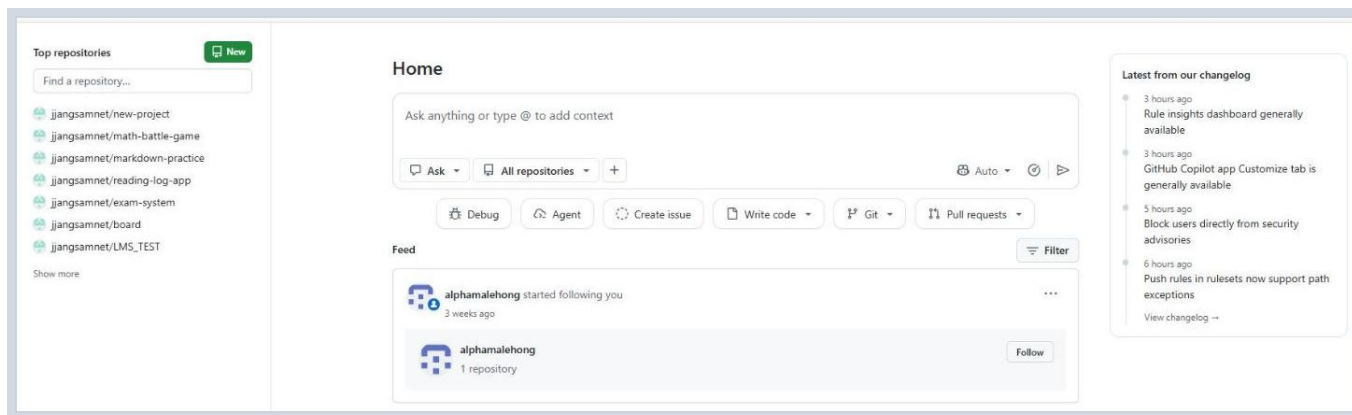
비유 ▶ 무료 계정에서 Pages로 주소를 만들려면 저장소가 공개(Public)
여야 합니다. 그래서 학생 개인정보가 코드에 절대 들어가면 안 됩니다.

1

계정 만들기 — github.com → Sign up → 이메일·비밀번호·사용자명 입력 → 이메일 인증. 사용자명은 곧 내 주소의 일부(<https://사용자명.github.io/...>)가
되니 교사용 아이디를 권합니다. 가입 직후 2단계 인증(2FA) 화면이 뜨면 안내대로 설정하면 되고, 무료 계정으로 이 단원의 모든 실습이 가능합니다.

로그인 후 첫 화면 — 여기서 길을 잃지 않기

github.com에 로그인하면 이 화면이 나옵니다. 처음 보면 정보가 많아 당황스럽지만, 교사가 쓰는 곳은 두 군데뿐입니다 — 왼쪽 위의 초록색 New 버튼(새 저장소 만들기)과 그 아래 내 저장소 목록.



GitHub 로그인 후 첫 화면 — 왼쪽 위 New 버튼과 Top repositories

왼쪽 위 New

새 저장소를 만드는 초록 버튼. 13쪽에서 이 버튼이 여는 화면을 자세히 봅니다.

Top repositories

내가 최근에 만진 저장소 목록. "Show more"로 전체를 볼 수 있습니다.

가운데 Home / Feed

내가 팔로우하는 사람과 인기 저장소 소식. 교사 실습에서는 볼 일이 없습니다.

오른쪽 위 프로필

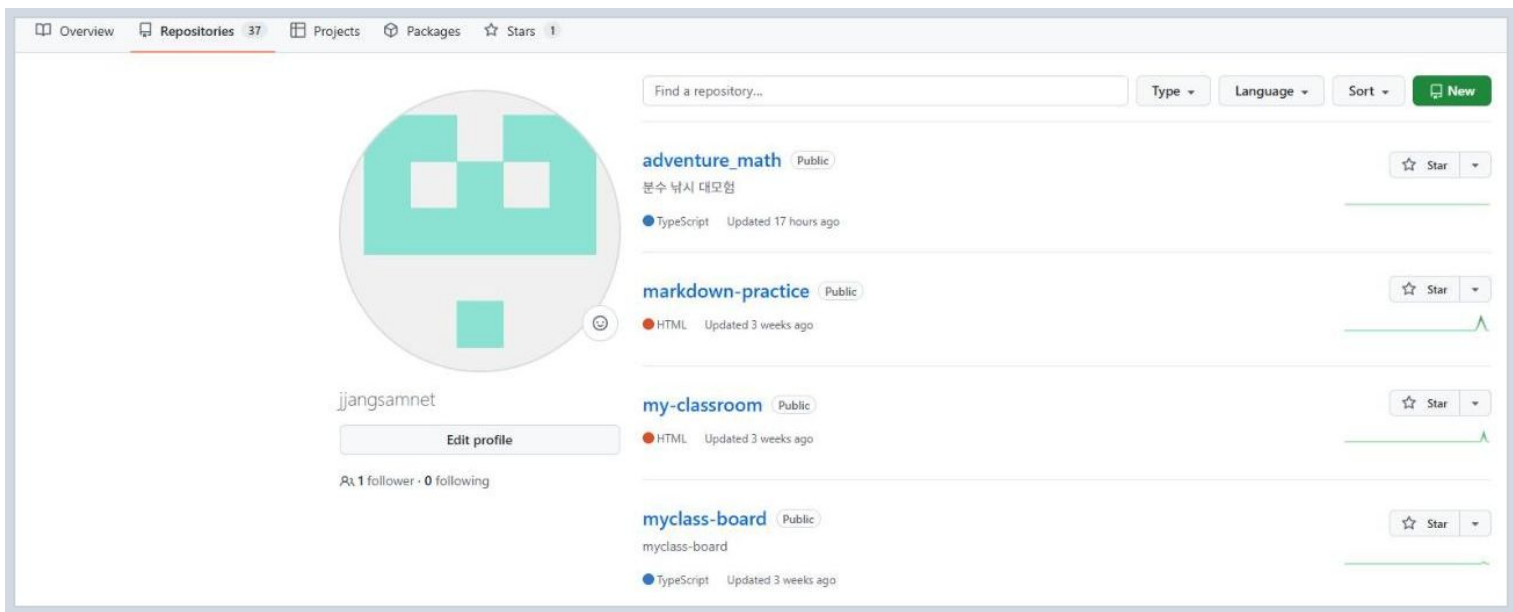
내 저장소 목록·설정으로 가는 입구. Your repositories를 누르면 다음 쪽 화면입니다.

i

GitHub는 원래 프로그래머들이 서로 코드를 보여주고 함께 고치는 곳입니다. 그래서 첫 화면이 소셜 미디어처럼 생겼습니다. 교사는 그 기능을 쓸 일이 없으니 가운데 영역은 무시하고, 왼쪽 New와 저장소 목록만 보면 됩니다.

내 저장소 목록 — 서랍장 전체 보기

프로필 → Repositories 탭입니다. 내가 가진 서랍이 전부 여기 모입니다. 각 줄에 저장소 이름, 공개 범위 배지(Public/Private), 설명, 사용 언어, 마지막 수정 시각이 표시됩니다. 오래 쓸 학급 자료를 여기서 한눈에 관리하게 됩니다.



프로필 → Repositories — 저장소 이름 · 공개 배지 · 설명 · 언어 · 수정 시각

1

공개 배지를 먼저 봅니다

Public이면 전 세계가 코드를 볼 수 있습니다. 무료 계정으로 Pages 주소를 만들려면 Public이어야 하므로, 학생 개인정보가 없는지가 늘 전제입니다.

2

설명을 꼭 적어 두세요

"분수 낚시 대모험"처럼 한글 설명을 넣으면 1년 뒤에도 무엇인지 알아봅니다. 저장소 이름은 영문이라 헛갈립니다.

3

수정 시각으로 최신본 찾기

같은 게임을 여러 번 만들었다면 "Updated ~"를 보고 최신 것을 고릅니다.

4

오른쪽 위 New

이 화면에서도 새 저장소를 만들 수 있습니다.

저장소 만들기 — 각 칸이 무슨 뜻인가

github.com/new · 이 화면을 읽을 줄 알면 GitHub의 절반은 끝납니다

New 버튼을 누르면 이 화면입니다. 칸이 많아 보이지만 교사가 정할 것은 두 개뿐입니다 — 이름과 공개 범위. 나머지는 기본값 그대로 두면 됩니다. 실습에서는 AI Studio가 이 과정을 대신 해 주지만(16~17쪽), 무엇이 만들어지는지 알고 말기는 것과 모르고 말기는 것은 다릅니다.

Create a new repository (github.com/new)

Owner / Repository name	★ 소유자(내 계정)와 저장소 이름. 영문 소문자·하이픈·밑줄을 씁니다. 이 이름이 주소의 뒷부분이 됩니다.
Description	한글 설명을 적습니다. 선택 사항이지만 꼭 적으세요.
Choose visibility	★ Public(누구나 열람) / Private(나만). 무료 계정으로 Pages를 켜려면 Public이어야 합니다.
Start with a template	기존 틀에서 시작하기. 우리는 쓰지 않습니다(No template).
Add README	설명 문서를 함께 만들지 여부. AI Studio가 코드를 올릴 것이므로 꺼 두어도 됩니다.
Add .gitignore	올리지 않을 파일 목록. AI Studio가 알아서 넣어 줍니다. No .gitignore로 둡니다.
Add license	저작권 이용 조건. 교실 자료라면 지금은 비워 두어도 무방합니다.
Create repository	초록 버튼을 누르면 빈 저장소가 만들어집니다.

두 갈래 — 내가 만들까, AI Studio가 만들게 할까

저장소는 두 가지 방법으로 만들 수 있습니다. 결과물은 같지만 실습 흐름이 다릅니다. 연수에서는 B안(AI Studio가 만들기)을 권합니다 — 화면을 오가지 않아 막히는 사람이 적습니다. 앞 쪽에서 A안 화면을 본 이유는, B안이 무엇을 대신 해 주는지 알아야 문제가 생겼을 때 손댈 수 있기 때문입니다.

A안 · 내가 직접 만들기

이해에 좋음

github.com/new에서 저장소를 먼저 만들고, AI Studio에서 그 저장소를 골라 연결합니다.

- 무엇이 만들어지는지 눈으로 봅니다
- 이름·설명·공개 범위를 내가 정합니다
- 화면을 두 번 오가야 해서 초보자는 헷갈릴 수 있습니다
- 이미 있는 저장소에 이어 올릴 때는 이 길뿐입니다

B안 · AI Studio가 만들기

실습 권장

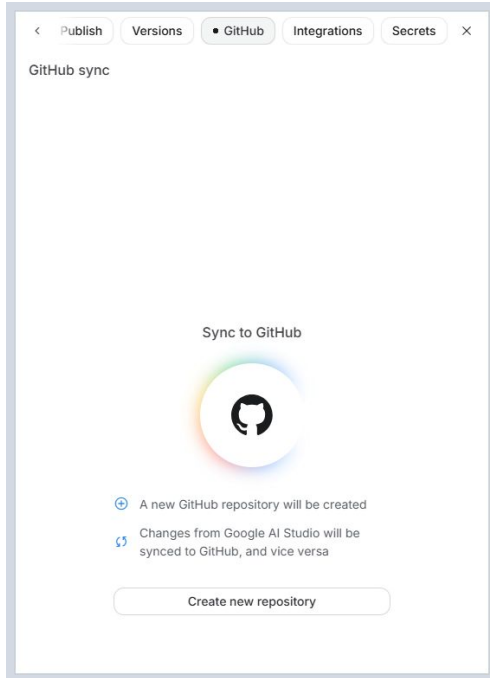
AI Studio의 GitHub 탭에서 이름과 공개 범위를 정하면, 저장소 생성과 연결이 한 번에 끝납니다.

- 화면을 옮기지 않아 실습이 매끄럽습니다
- 이름·설명·Public 선택을 그 자리에서 합니다
- 연수 시간이 짧을 때 유리합니다
- 다음 쪽부터 이 길로 진행합니다

! 공개 범위는 어느 길로 가든 Public — 무료 계정에서 GitHub Pages로 주소를 만들려면 저장소가 공개여야 합니다(비공개 저장소의 Pages는 유료 요금제 기능입니다). "공개"란 전 세계 누구나 내 코드를 읽을 수 있다는 뜻입니다. 그래서 이 단원의 안전 수칙(32쪽)이 중요합니다 — 코드에 학생 이름·사진·API 키가 하나도 없어야 마음 놓고 공개할 수 있습니다.

AI Studio에서 GitHub 탭 열기

무대가 GitHub에서 AI Studio로 옮겨 갑니다. 4단원에서 만든 앱을 열고 오른쪽 위 톱니바퀴를 눌러 패널을 연 뒤, GitHub 탭을 고릅니다. 아직 연결하지 않았다면 "Sync to GitHub" 안내와 함께 새 저장소를 만드는 버튼이 보입니다.



GitHub 탭 — Sync to GitHub

1 4단원에서 만든 앱을 엽니다

aistudio.google.com/apps → My apps에서 고릅니다.

2 오른쪽 위 톱니바퀴 클릭

패널이 열리고 Chat / Share / Publish / Versions / GitHub / Integrations / Secrets 탭이 보입니다.

3 GitHub 탭 선택

탭이 많아 화살표로 넘겨야 보일 수 있습니다.

4 "Create new repository"

새 저장소를 만들어 연결합니다. 이미 만든 저장소가 있다면 그것을 고르는 항목도 있습니다.

5 GitHub 로그인·권한 승인

처음 한 번만 합니다. AI Studio가 내 저장소에 코드를 올릴 수 있도록 허용하는 절차입니다.

i

권한 승인 화면이 낯설게 느껴질 수 있습니다. "AI Studio가 내 GitHub 저장소를 읽고 쓸 수 있게 한다"는 뜻이고, GitHub 설정에서 언제든지 취소할 수 있습니다.

저장소 정보 입력 — Public을 꼭 확인하세요

14쪽에서 본 github.com/new의 항목들이 여기에 요약되어 나타납니다. 이름·설명·공개 범위 세 가지만 정하면 됩니다. Visibility에서 Public을 고르지 않으면 나중에 Pages를 켤 수 없어 되돌아와야 하니, 이 칸을 반드시 확인하세요.

New repository name

영문 소문자로. 이번 실습은 adventure_math. 이 이름이 주소 뒤에 그대로 붙습니다.

New repository description

한글로 적어 둡니다. 예: 분수 낚시 대모험

Visibility — Private

나만 볼 수 있음. 무료 계정에서는 Pages를 켤 수 없습니다.

Visibility — Public ★

누구나 볼 수 있음. "This repo will be discoverable by everyone on GitHub.com" — 이쪽을 고릅니다.

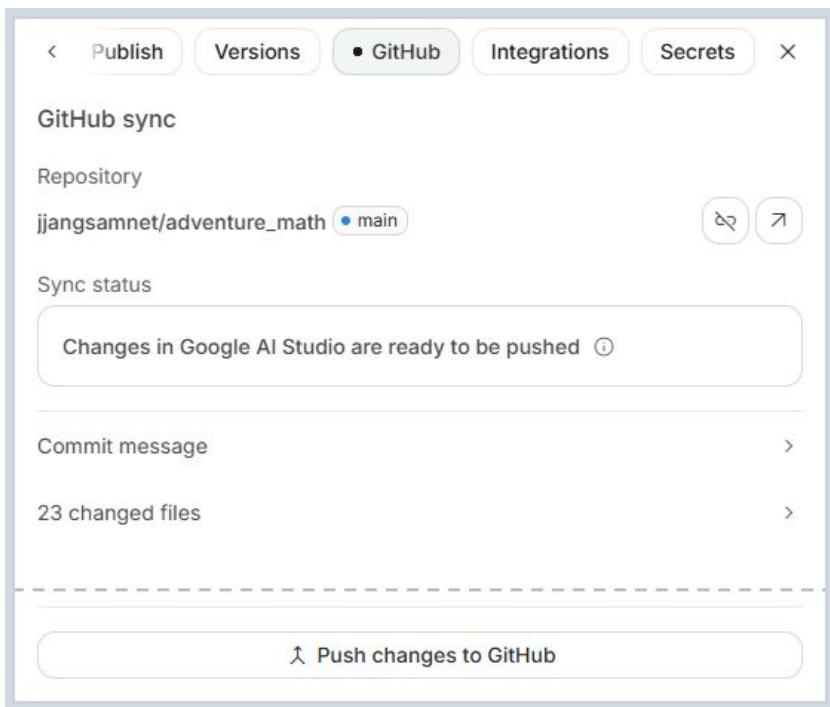
Create GitHub repository

누르면 "We're creating your repository" 안내가 뜨고, 잠시 뒤 동기화 화면으로 바뀝니다.

GitHub 탭 → 새 저장소: 이름 · 설명 · Public 선택 후 Create

Push 하기 — 무엇이 올라가는지 보고 누르기

저장소가 만들어지면 패널이 동기화 화면으로 바뀝니다. 여기서 저장소 이름과 브랜치(main), 그리고 "올릴 준비가 된 변경"이 몇 개인지 확인한 뒤 Push changes to GitHub를 누릅니다. 커밋 메시지는 AI가 대신 써 주므로 그대로 두어도 됩니다.



동기화 패널 — 저장소 · 상태 · 변경 파일 수 · Push 버튼

① 저장소와 브랜치가 맞나요?

jjangsamnet/adventure_math 처럼 "내 아이디 / 저장소 이름"이 표시되고 옆에 main 배지가 붙습니다. 이 두 값이 곧 나중에 만들 주소의 재료입니다.

② 올릴 준비가 되었나요?

"Changes in Google AI Studio are ready to be pushed" 문구와 변경 파일 수(예: 23 changed files)를 확인합니다. 0이면 이미 올라간 상태입니다.

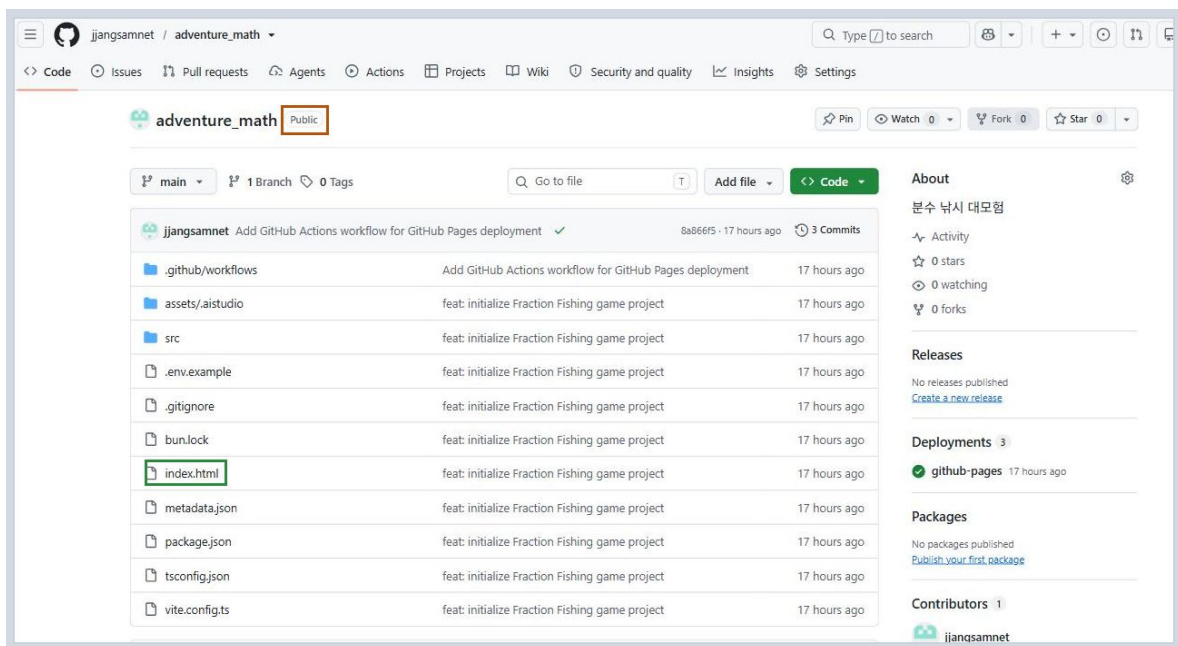
③ 커밋 메시지는 그대로

AI가 "feat: initialize Fraction Fishing game project" 같은 문장을 자동으로 씁니다. 직접 고쳐도 되지만 그대로 두어도 무방합니다.

올라갔는지 눈으로 확인 — github.com 저장소 화면

Push를 눌렀다고 끝이 아닙니다. 반드시 열어서 보세요

브라우저에서 github.com의 내 저장소를 직접 엽니다(주소: github.com/내아이디/저장소이름). 파일 목록이 보이고 그 안에 index.html이 있으면 성공입니다(초록 표시). 이 파일이 최상위에 없으면 나중에 주소를 열었을 때 404가 뜹니다 — 지금 확인하는 것이 가장 쌉니다.



저장소 화면 — Public 배지(주황) · 파일 목록 · index.html(초록) · 3 Commits

저장소 이름 옆 배지

Public인지 확인. Private이면 Pages를 켤 수 없습니다.

index.html

★ 최상위에 있어야 합니다. Pages는 이 파일을 첫 화면으로 씁니다.

main 배지 · 1 Branch

기본 가지가 main 하나. 이 이름을 Pages 설정에서 고르게 됩니다.

3 Commits

저장 지점이 3개 쌓였다는 뜻. 누르면 기록을 볼 수 있습니다 (21쪽).

오른쪽 About·Deployments

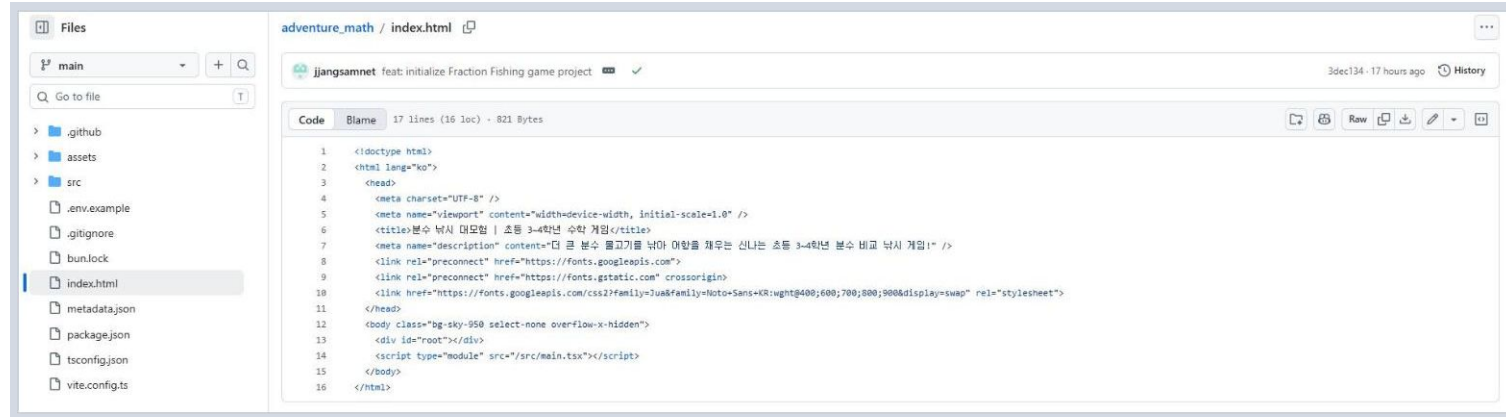
설명과 배포 상태가 표시됩니다. Pages를 켜면 여기에 github-pages가 나타납니다.

Add file / Code 버튼

파일을 직접 올리거나 코드를 내려받는 버튼. AI Studio가 올려 주므로 쓸 일이 없습니다.

index.html을 열어보면 — 무엇이 들어 있나

파일 이름을 클릭하면 내용이 보입니다. 코드를 다 이해할 필요는 없습니다. 다만 "게임이 어떤 조각들로 되어 있는지" 한 번 보고 나면, 문제가 생겼을 때 AI에게 무엇을 고쳐 달라고 말해야 할지 감이 잡힙니다.



index.html — 왼쪽에 파일 트리, 오른쪽에 코드 16줄

`<title>분수 낚시 대모험...</title>`

브라우저 탭에 뜨는 제목. 여기가 한글이면 학생 화면 탭에도 한글로 보입니다.

`<meta name="description">`

검색·공유 시 보이는 한 줄 설명. AI가 게임 설명을 자동으로 넣어 두었습니다.

`<div id="root"></div>`

게임이 그려지는 빈 자리. 실제 화면은 아래 script가 여기에 채웁니다.

`<script src="/src/main.tsx">`

게임 본체를 불러오는 줄. 게임 코드는 src 폴더 안에 있습니다.

커밋 기록 — 되돌릴 수 있다는 안심

저장소 화면의 "N Commits"를 누르면 이 목록이 나옵니다. 게임을 고칠 때마다 한 줄씩 쌓입니다. "AI에게 고쳐 달라고 했다가 더 나빠졌다"는 상황에서 이 기록이 안전망이 됩니다 — 언제든지 이전 시점의 코드를 볼 수 있습니다.



커밋 기록 — 메시지 · 작성자 · 시각 · Verified 배지 · 커밋 해시

커밋 메시지

무엇을 바꿨는지 적은 한 줄. AI가 자동으로 씁니다. "Initial commit"이 첫 저장, 그 위로 시간순으로 쌓입니다.

오른쪽 해시

8a866f5 같은 고유 번호. 그 시점을 가리킵니다. 누르면 그때의 코드 전체를 볼 수 있습니다.

Verified 배지

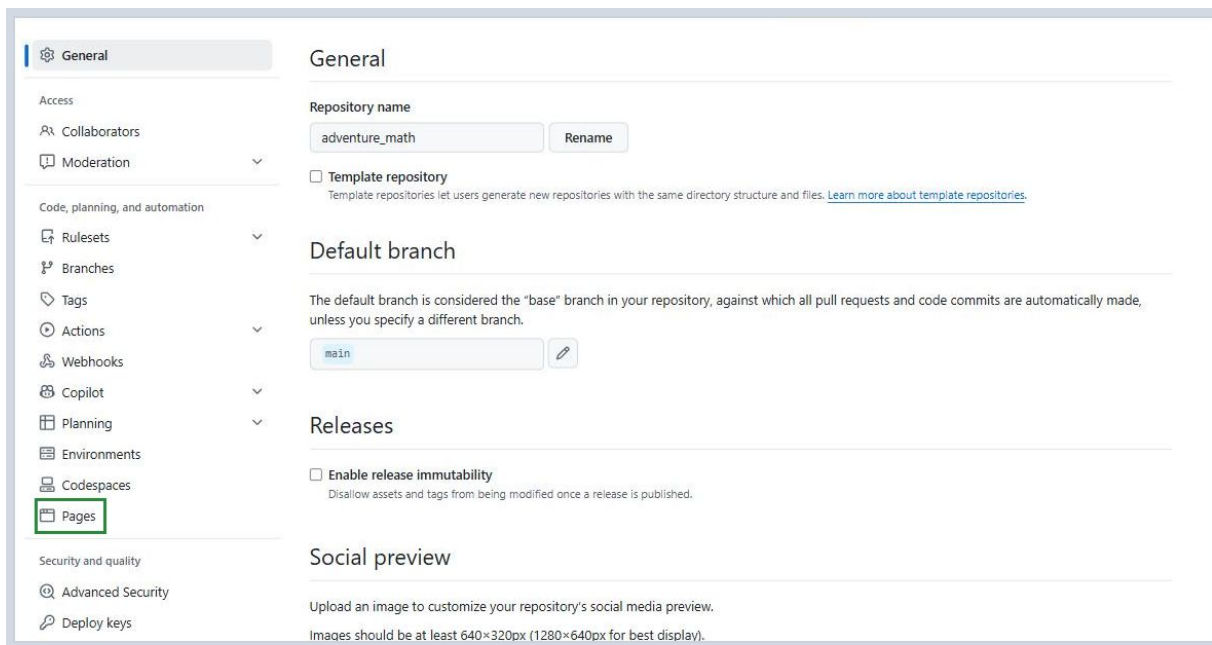
서명이 확인된 커밋이라는 표시. 교사 실습에서는 신경 쓰지 않아도 됩니다.

왜 자주 Push하나

커밋 하나가 곧 되돌아갈 수 있는 지점 하나입니다. 크게 고치기 전에 한 번 올려 두면 마음이 편합니다.

Settings 열기 — Pages 메뉴는 어디 있나

코드가 올라갔다고 해서 자동으로 주소가 생기지는 않습니다. Pages 기능을 한 번 켜 줘야 합니다. 저장소 화면 맨 위 탭 줄의 오른쪽 끝 Settings를 누르고, 왼쪽 사이드바를 아래로 내리면 Pages가 있습니다(초록 표시).



Settings → 왼쪽 사이드바 "Code, planning, and automation" 아래에 Pages

1

저장소 상단 Settings

탭 줄 맨 오른쪽, 톱니바퀴 아이콘입니다. 이 탭은 저장소 주인에게만 보입니다.

2

왼쪽 사이드바를 훑습니다

General → Access → Code, planning, and automation → Security and quality → Integrations 순서로 묶여 있습니다.

3

Pages를 클릭

"Code, planning, and automation" 묶음의 아래쪽, Environments·Codespaces 다음에 있습니다.

4

찾기 어렵다면

주소창에 저장소 주소 뒤 /settings/pages 를 직접 붙여도 같은 곳으로 갑니다.

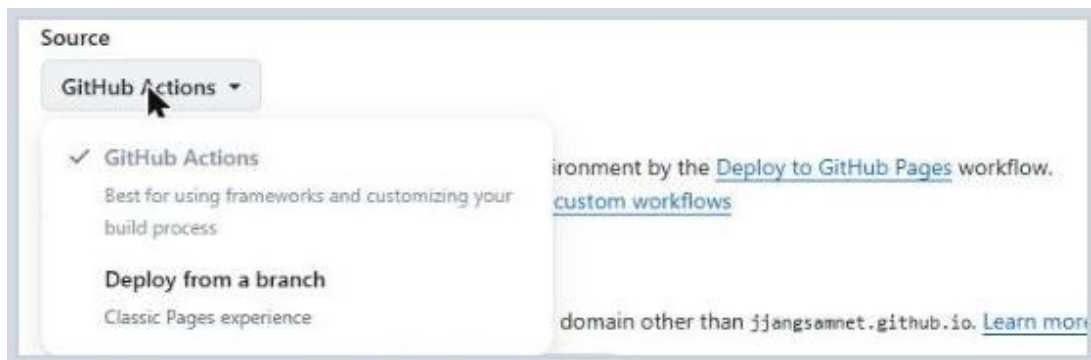
i

이 화면 맨 아래에는 Danger Zone이 있습니다 — 공개 범위 전환과 삭제. 31쪽에서 회수 절차로 다시 옵니다.

Source — 게시 방식 두 갈래

이 선택 하나로 다음 화면이 완전히 달라집니다

Pages 화면의 Build and deployment → Source에서 게시 방식을 고릅니다. 드롭다운을 열면 두 가지가 나옵니다. 어느 쪽을 골라도 최종 주소는 같습니다. 다만 실습 중에 보게 되는 화면과 확인 방법이 달라지니, 지금 차이를 알고 가세요.



Source 드롭다운 — GitHub Actions / Deploy from a branch

어느 쪽이 이미 선택되어 있나요? — AI Studio로 Push했다면 GitHub Actions가 켜져 있을 가능성이 큼니다. 저장소에 .github/workflows 폴더가 보였다면(18쪽 캡처) 배포 절차서가 함께 올라간 것입니다.

이미 잘 배포되고 있다면 굳이 바꾸지 마세요. 다음 쪽은 "Branch가 None이고 아무것도 배포되지 않은" 경우를 위한 안내입니다.

Deploy from a branch → 23쪽

"Classic Pages experience"

가지(branch)와 폴더를 지정하면 그 안의 파일을 그대로 올려 줍니다. 완성된 index.html이 이미 있는 경우에 가장 단순합니다.

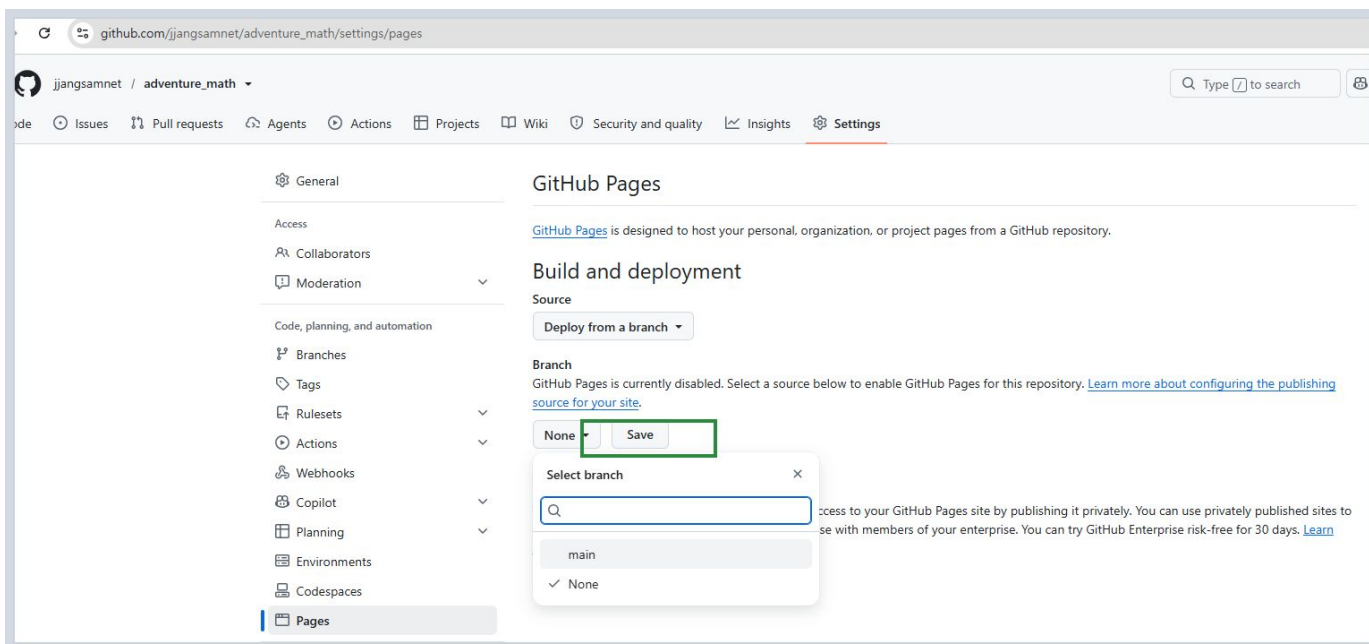
GitHub Actions → 24쪽

"Best for using frameworks and customizing your build process"

저장소에 들어 있는 배포 절차서(workflow)를 실행해 만들어진 결과물을 올립니다. AI Studio가 이 절차서를 함께 올려 두면 자동으로 이쪽이 선택되어 있습니다.

길 ① Deploy from a branch — 가지를 지정해 켜기

처음 들어가면 Branch가 None으로 되어 있고 "GitHub Pages is currently disabled"라고 적혀 있습니다. 드롭다운을 열어 main을 고르고 폴더는 / (root)로 둔 뒤 Save를 누르면 끝입니다. 1~2분 뒤 주소가 발급됩니다.



1

Source: Deploy from a branch

드롭다운에서 이쪽을 고릅니다.

2

Branch: None → main

None을 눌러 목록을 열고 main을 선택합니다. 저장소가 비어 있으면 main이 안 보입니다 — 먼저 Push하세요.

3

폴더: / (root)

index.html이 최상위에 있으므로 root 그대로 둡니다. 하위 폴더에 있다면 그 폴더를 고릅니다.

4

Save 클릭

저장하면 배포가 시작됩니다.

5

1~2분 뒤 새로고침

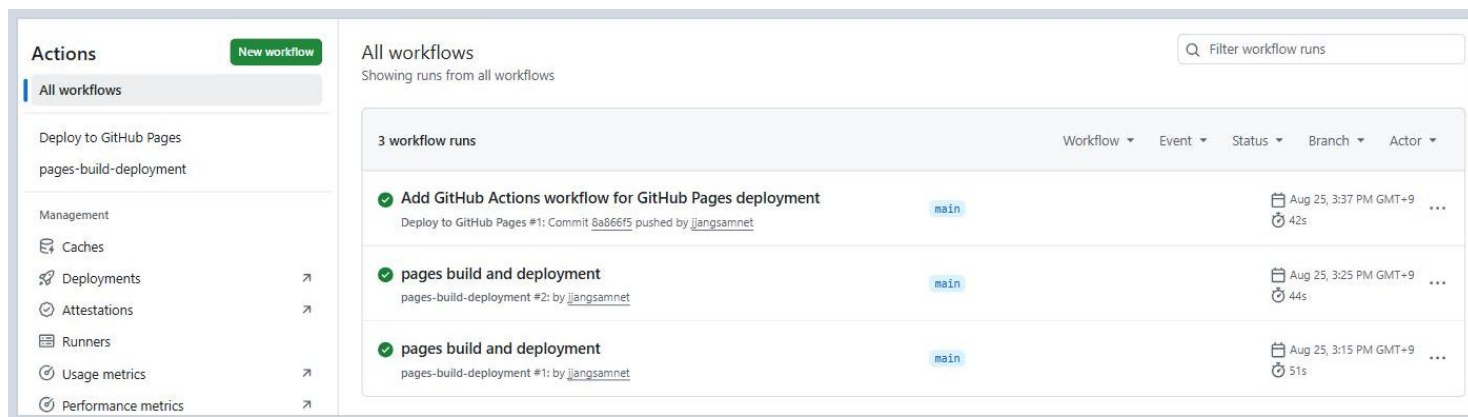
화면 위쪽에 초록 안내와 주소가 나타납니다(25쪽).

발급 주소 형식 ▶ `https://<내 GitHub 사용자명>.github.io/<저장소 이름>/`

예) `https://jjangsamnet.github.io/adventure_math/`

길 ② GitHub Actions — 배포가 되는 것을 지켜보기

Source가 GitHub Actions로 되어 있으면 Push할 때마다 배포 절차서가 자동으로 돕니다. 따로 Save를 누를 것이 없고, 대신 저장소 상단의 Actions 탭에서 배포가 잘 되었는지 확인합니다. 초록 체크(✓)면 성공, 빨간 X면 실패입니다.



Actions 탭 — 워크플로 실행 목록. 초록 체크가 배포 성공

왼쪽 워크플로 목록

Deploy to GitHub Pages / pages build and deployment. 배포 절차서의 이름입니다.

초록 체크 (✓)

그 실행이 성공했다는 뜻. 옆의 시각과 소요 시간(42s 등)도 함께 보입니다.

빨간 X가 뜬다면

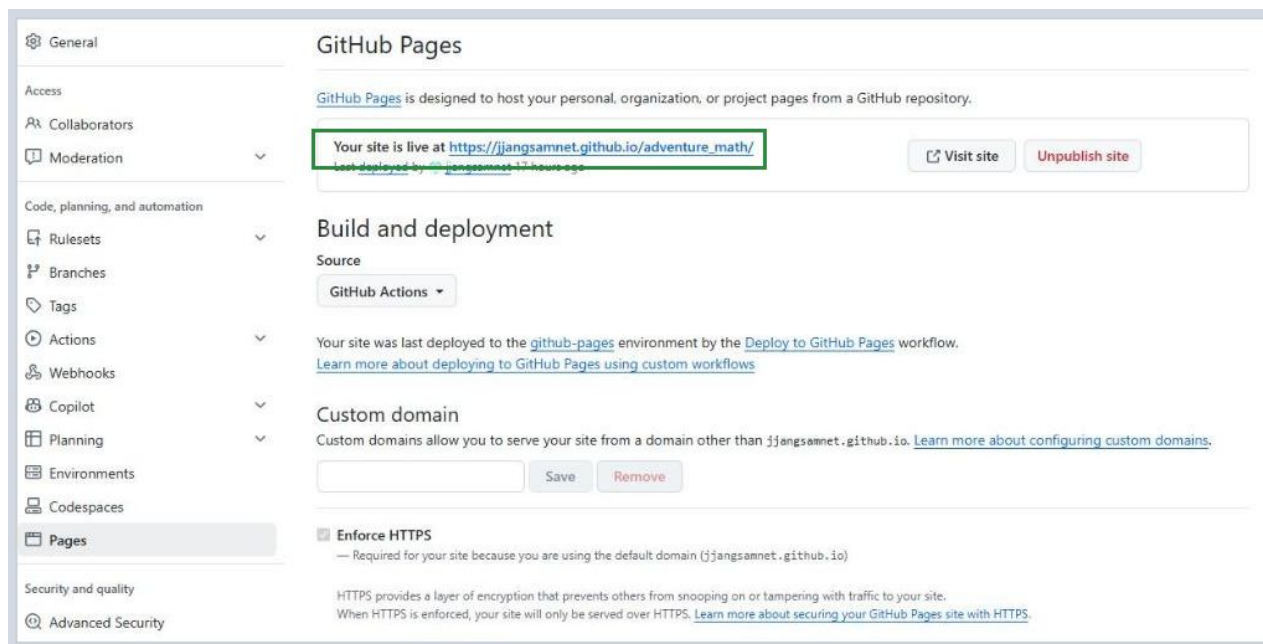
실행을 눌러 들어가면 어느 단계에서 멈췄는지 로그가 나옵니다. 그 문장을 그대로 AI에게 붙여넣고 물어보세요.

노란 원 (진행 중)

아직 배포가 돌고 있습니다. 보통 1~2분이면 끝납니다. 기다렸다 새로고침하세요.

"Your site is live at ..." — 주소가 태어난 순간

Pages 화면을 새로고침하면 위쪽에 초록 안내와 함께 주소가 나타납니다(초록 표시). 이 화면에는 주소 말고도 앞으로 쓸 기능이 몇 개 더 있으니 함께 봐 두세요 — 특히 오른쪽의 Unpublish site는 회수 버튼입니다.



Pages 배포 완료 — 주소 · Visit site · Unpublish site · Custom domain · Enforce HTTPS

Your site is live at ...

★ 발급된 주소. 학생에게 나눠 줄
한 줄입니다.

Visit site

새 탭에서 바로 열어 봅니다.

Unpublish site

★ 게시를 끕니다. 문제가 생겼을 때
가장 빠른 회수 방법(31쪽).

Last deployed by ...

마지막 배포 시각. 고쳤는데 반영이
안 될 때 여기부터 봅니다.

Custom domain

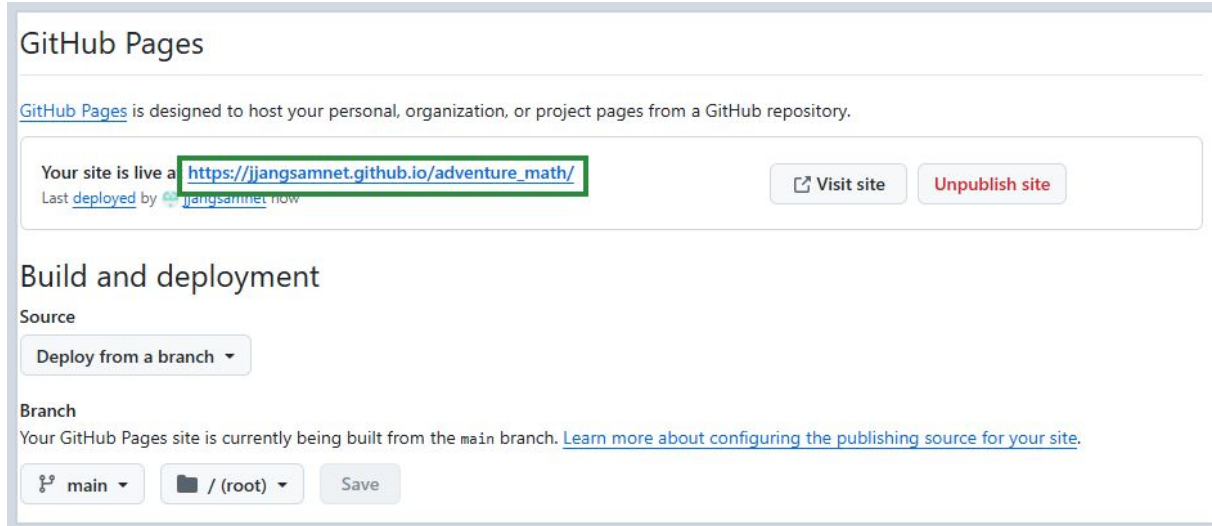
학교 도메인이 있다면 붙일 수
있습니다. 없으면 비워 둡니다.

Enforce HTTPS

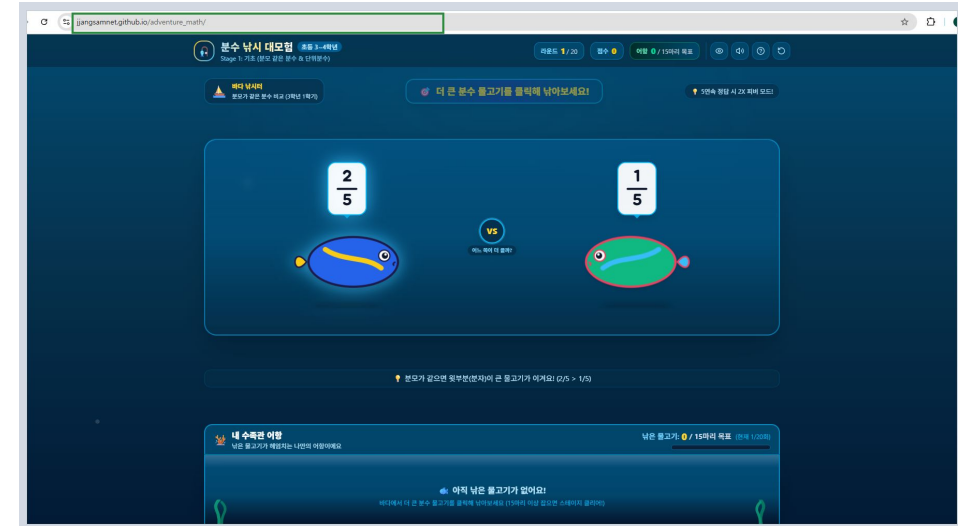
기본 도메인에서는 자동으로 켜지고
 끌 수 없습니다. 안전한 상태라는
 뜻입니다.

접속해서 확인하기 — 5단원의 목표 지점

주소를 새 탭에 붙여넣어 게임이 실제로 열리는지 확인합니다. 이 순간이 5단원의 도착점입니다 — 이제 이 한 줄만 있으면 누구나, 어떤 기기에서든 내 게임을 열 수 있습니다. 스마트폰으로도 꼭 한 번 열어 보세요.



"Your site is live at ..." + Branch가 main / (root)로 바뀐 상태



발급된 주소로 접속한 게임 — 주소창 확인



안 될 때 3가지 ▶ ① 404가 뜬다 — index.html이 저장소 최상위에 있는지 확인(18쪽). 하위 폴더면 파일을 최상위로 옮기고 다시 Push. ② 옛날 화면이 뜬다 — 브라우저 캐시입니다. Ctrl+F5 또는 시크릿 창으로 확인(배포 반영에 1~2분). ③ 화면이 깨진다 — 이미지·소리 경로가 내 PC 기준일 수 있습니다. AI Studio에 "모든 자원을 상대 경로로 바꿔줘"라고 요청하세요.

학생에게 닿게 하기 — 배포 경로 4가지

주소가 있어도 학생 손에 닿지 않으면 소용없습니다

주소는 길고, 초등학생이 타이핑하기에는 오타가 납니다. 실제 교실에서는 "타이핑하지 않고 도달하는 경로"를 반드시 마련해야 합니다. 아래 네 가지 중 학교 환경에 맞는 것을 두 개 이상 겹쳐 두면 수업 중 사고가 거의 사라집니다.

QR 코드

QR 생성 사이트에 주소를 넣어 이미지를 만들고, TV 화면에 띄우거나 학습지 귀퉁이에 인쇄합니다. 태블릿 카메라로 바로 접속.

가장 빠름 · 인쇄 가능

학급 플랫폼 링크

클래스룸·e학습터·학급 밴드 게시글에 링크를 겁니다. 가정 연계 과제로 낼 때 필수 경로입니다.

가정 연계에 강함

북마크 사전 설치

교실 공용 태블릿이라면 수업 전에 브라우저 즐겨찾기에 미리 등록해 둡니다. 저학년일수록 효과가 큼니다.

저학년 · 공용기기

짧은 주소

주소가 길면 단축 서비스로 줄여 칠판에 적습니다. 다만 학교 망에서 단축 도메인이 막히는 경우가 있어 사전 확인이 필요합니다.

칠판 판서용

i

겹쳐 두라는 뜻 — 예를 들어 QR(교실용)과 클래스룸 링크(가정용)를 함께 준비합니다. 태블릿 카메라가 안 되는 기기가 한둘 나오고, 집에서는 QR을 찍을 화면이 없습니다.

수업 전 날 체크리스트: ① 교실 기기 한 대에서 주소 열림 확인 ② QR 이미지 만들어 학습지에 넣음 ③ 클래스룸 게시글 예약 발행 ④ 예비용으로 짧은 주소도 준비.

QR 코드 만들기 — 5분이면 됩니다

QR 코드는 별도 프로그램 없이 만들 수 있습니다. 만드는 것보다 중요한 것은 "어디에 얼마나 크게 넣느냐"입니다. 교실 뒤에서도 찍히려면 생각보다 커야 하고, 학습지에 넣을 때는 인쇄 해상도가 중요합니다.

1

QR 생성 사이트를 엽니다

"QR 코드 생성"으로 검색하면 무료 사이트가 많이 나옵니다. 로그인도 필요 없고 광고가 적은 곳을 고르세요.

2

주소를 붙여넣습니다

<https://> 부터 마지막 / 까지 전부 복사해 넣습니다. 한 글자라도 빠지면 다른 곳으로 갑니다.

3

PNG로 내려받습니다

가능하면 크게(1000px 이상) 받아 두세요. 학습지에 인쇄할 때 작게 받으면 뭉개집니다.

4

교실 TV용 — 화면 가득

수업 시작 시 TV에 QR만 크게 띄웁니다. 학생들이 자리에서 찍을 수 있는 크기여야 합니다.

5

학습지용 — 3cm 이상

학습지 귀퉁이에 넣을 때 한 번이 3cm보다 작으면 인식률이 떨어집니다.

6

찍히는지 직접 확인

만든 QR을 내 스마트폰으로 한 번 찍어 봅니다. 이 한 번이 수업 중 사고를 막습니다.

학교 망에서 막힌다면

QR 자체는 그냥 주소를 담은 그림이라 막힐 일이 없습니다. 문제가 된다면 주소 쪽입니다 — github.io 도메인이 교육청 필터링에 걸리는 경우가 드물게 있습니다. 수업 전날 교실 기기로 확인하고, 막히면 정보부장·교육청에 도메인 허용을 요청하세요.

주소를 짧게 만들고 싶다면

저장소 이름을 짧게 지으면 주소도 짧아집니다. adventure_math 대신 math4 처럼요. 다만 이름은 나중에 바꿀 수 있지만(Settings → Rename), 바꾸면 예전 주소가 끊깁니다. 이미 학생에게 나눠 준 뒤라면 바꾸지 마세요. 단축 서비스는 학교 망에서 막히는 사례가 있으니 보조 수단으로만 쓰세요.

수업 배치 — 이 게임을 언제 꺼내는가

주소가 생겼다고 게임이 저절로 수업이 되지는 않습니다. 언제 꺼내느냐에 따라 같은 게임이 진단 도구가 되기도 하고 단순한 시간 때우기가 되기도 합니다. 아래 세 가지 배치를 각각 한 번씩 해 보고 우리 반에 맞는 것을 고르세요.

도입 5분 · 진단

단원 시작에 붙여 사전 지식을 확인합니다.

- 점수는 기록하지 않습니다 — "오늘 뭘 배울지 맛보기"라고 말해 둡니다
- 교사는 순회하며 어디서 많이 막히는지만 봅니다
- 5분을 넘기지 않습니다. 길어지면 본 수업이 밀립니다

정리 10분 · 형성평가

오늘 배운 개념을 게임으로 되짚습니다.

- 가장 효과가 큰 배치입니다. 방금 배운 것을 바로 써 보게 합니다
- 교사는 순회하며 개별 지도할 학생을 파악합니다
- 다 못 푼 학생에게 "집에서 더 해도 된다"고 링크를 안내합니다

가정 · 자율 반복

링크를 클래스룸에 걸어 원하는 만큼 반복하게 합니다.

- 강제 과제로 만들지 않는 편이 참여율이 높습니다
- "해 온 사람 손 들어"를 하지 않습니다 — 안 한 학생이 드러납니다
- 기기가 없는 가정을 반드시 고려합니다. 대체 활동을 함께 준비하세요



공통 주의 — 어느 배치든 "게임 점수를 성적·상점·자리 배치에 연결하지 않기"가 원칙입니다. 연결하는 순간 학생은 학습이 아니라 점수를 최적화합니다. 반복하면 오르는 구조라는 점도 미리 말해 두세요. (6단원 리더보드에서 이 이야기를 본격적으로 다룹니다.)

고치고 다시 올리기 — 한 번 연결하면 버튼 하나

수업을 해 보면 반드시 고칠 곳이 나옵니다. "문제가 너무 어려워요", "글씨가 작아요". 한 번 연결해 두었으면 그다음부터는 간단합니다 — AI Studio에서 고치고, Push 한 번. 학생이 보는 주소는 그대로입니다.

1

AI Studio에서 고칩니다

4단원에서 하던 대로 채팅으로 요청합니다. 예: "3단계 문제를 조금 쉽게 바꿔줘"

2

GitHub 탭에서 Push

변경 파일 수가 다시 표시됩니다. Push changes to GitHub를 누릅니다.

3

Actions에서 배포 확인

초록 체크가 뜰 때까지 1~2분. Deploy from a branch를 쓰고 있다면 자동으로 다시 배포됩니다.

4

주소를 강력 새로고침

Ctrl+F5. 학생 기기에서도 캐시가 남아 있을 수 있으니, 수업 전에 미리 갱신해 두세요.

주소는 바뀌지 않습니다

이것이 GitHub Pages의 가장 큰 장점입니다. 몇 번을 고쳐 올려도 학생에게 나눠 준 주소는 그대로입니다. 이미 인쇄한 QR 학습지도 계속 유효합니다.

잘못 고쳤다면

커밋 기록(21쪽)에서 이전 시점을 찾을 수 있습니다. AI Studio 패널의 Versions 탭에서도 이전 버전으로 돌아갈 수 있습니다. 크게 고치기 전에 한 번 Push해 두는 습관이 안전망입니다.

i

수업 중에는 고치지 마세요. 배포에 1~2분이 걸리고 그동안 화면이 흔들립니다.

회수하기 — 문제가 생겼을 때 즉시 닫는 법

미리 한 번 익혀 두면 급할 때 당황하지 않습니다

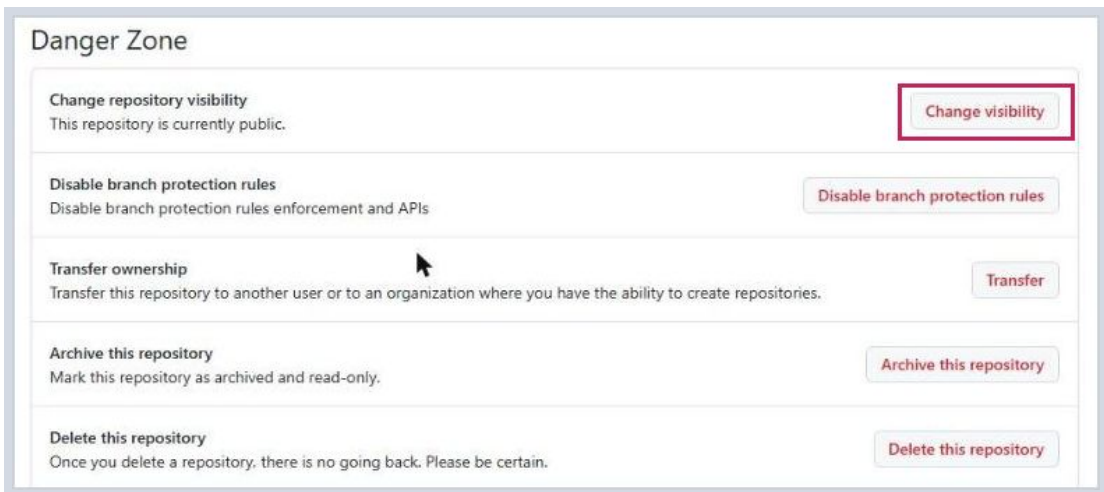
공개한 자료에 문제가 생기는 일은 실제로 일어납니다. 실수로 학생 이름이 들어갔거나, 저작권 있는 이미지를 썼거나. 중요한 것은 속도입니다. 아래 두 가지 방법을 미리 알아 두세요 — 둘 다 1분이면 됩니다.

방법 ① Unpublish site (권장 · 가장 빠름)

Settings → Pages 화면 오른쪽 위의 Unpublish site를 누릅니다(25쪽 캡처). 주소가 즉시 닫히고 접속하면 404가 뜹니다. 저장소와 코드는 그대로 남아 있으므로, 고친 뒤 다시 켜면 같은 주소가 살아납니다.

방법 ② 저장소를 Private으로 전환

Settings → General 맨 아래 Danger Zone → Change visibility. 주소가 닫히는 동시에 코드도 남에게 보이지 않게 됩니다. 코드 자체에 문제가 있을 때 (개인정보·API 키) 이쪽을 씁니다.



Settings → General 맨 아래 Danger Zone (분홍 표시가 공개 범위 전환)



Danger Zone에는 Transfer(소유권 이전) · Archive(읽기 전용) · Delete(영구 삭제)도 있습니다. Delete는 되돌릴 수 없으니 절대 급하게 누르지 마세요.

공개하기 전 3분 점검 — 안전 수칙 6칸

공개 저장소는 전 세계가 볼 수 있습니다

GitHub Pages로 주소를 만들었다는 것은 그 코드가 인터넷에 공개되었다는 뜻입니다. 게시 버튼을 누르기 전, 아래 여섯 칸을 소리 내어 읽으며 점검하는 습관을 들이세요. 한 번의 실수가 되돌리기 어려운 결과로 이어질 수 있습니다.

□ 학생 이름·학번

문항이나 예시에 실제 학생 이름, 학번, 반 번호가 들어가 있지 않은지. 이름이 필요하다면 "친구 1, 친구 2"로 바꿉니다.

□ 사진·음성 파일

학생 얼굴 사진이나 목소리가 게임 자원에 포함되어 있지 않은지. 초상권·개인정보 문제가 동시에 걸립니다.

□ API 키·비밀번호

코드 안에 API 키가 문자열로 박혀 있지 않은지. 공개 저장소의 키는 자동 수집 대상이 됩니다.

□ 저작권 자원

인터넷에서 가져온 이미지·음악·캐릭터가 들어 있지 않은지. 수업용이라도 공개 게시는 사적 이용 범위를 벗어납니다.

□ 저장소 설명

설명에 학교명·학급명을 그대로 적지 않았는지. "3학년 수학 게임" 정도면 충분합니다.

□ 회수 절차 숙지

31쪽의 Unpublish site와 Private 전환을 미리 한 번 눌러 보세요. 급할 때 찾느라 헤매지 않습니다.

! 법적 기준 — 만 14세 미만 아동의 개인정보를 처리하려면 법정대리인의 동의가 필요합니다(개인정보 보호법 제22조의2). 초등학교 수업에서 만든 게임은 애초에 개인정보를 수집하지 않는 설계가 원칙입니다. 점수·기록이 필요하다면 학생이 직접 정한 닉네임만 쓰게 하세요(6단원).

100분 미션 — 내 게임에 주소 달기

4단원에서 만든 게임을 그대로 재료로 씁니다. 캡처를 옆에 두고 순서대로 따라오세요. 마지막 산출물은 "동료에게 보낼 수 있는 링크 한 줄"과 "학습지에 붙일 QR 한 장"입니다.

0~10분	내 게임 진단	AI 기능(실시간 AI 호출)이 있는지 확인 → Publish 길인지 Pages 길인지 결정 (5쪽)
10~18분	공유 링크 만들기	Share로 옆자리 선생님에게 링크 전달 → 상대 화면에서 열리는지 확인 (7쪽)
18~28분	Cloud Run 게시	Publish → Get started → Publish App → 발급 주소로 접속 (8~9쪽)
28~43분	GitHub 익히기	계정 준비 → 첫 화면·저장소 목록·새 저장소 폼 눈으로 확인 (10~14쪽)
43~58분	코드 올리기	AI Studio GitHub 탭 → Public 저장소 생성 → Push → github.com에서 index.html 확인 (15~18쪽)
58~76분	Pages 켜기	Settings → Pages → Source 확인 → main/(root) 또는 Actions → 주소 발급 → 접속 성공 (21~26쪽)
76~90분	학생 배포 준비	QR 생성 → 학습지 하단에 삽입 → 스마트폰으로 실제 접속 테스트 (27~28쪽)
90~100분	회수 연습 + 안전 점검	Unpublish site를 눌러 닫아 보고 다시 켜기 → 6칸 점검표 소리 내어 읽기 (31~32쪽)

제출 양식 ▶ 내 게임 이름: _____ | 선택한 경로: ☐ Publish(Cloud Run) ☐ GitHub Pages | 저장소 이름: _____
 발급된 주소: https://_____ | 접속 확인: ☐ PC ☐ 스마트폰 ☐ 교실 태블릿 | QR 제작: ☐ 회수 연습: ☐

자주 막히는 지점 8가지

연수 현장에서 실제로 가장 많이 나온 질문들입니다. 대부분은 "설정을 하나 빠뜨렸다"거나 "기다림이 부족했다"에 해당합니다.

Q. Branch에 main이 안 보여요

A. 아직 Push가 되지 않아 저장소가 비어 있는 경우입니다. AI Studio에서 Push changes to GitHub를 먼저 누르고, Pages 화면을 새로고침하세요(17~18쪽).

Q. Pages 메뉴가 안 보여요

A. 저장소가 Private일 때 무료 계정에서는 제한됩니다. Settings → Danger Zone에서 Public으로 바꾸면 나타납니다(31쪽).

Q. 고쳤는데 주소에 반영이 안 돼요

A. ① Push를 눌렀는지 ② Actions에 초록 체크가 났는지 ③ Ctrl+F5로 강력 새로고침했는지 순서대로 확인합니다. 배포에 1~2분 걸립니다(24·30쪽).

Q. Publish에서 결제하라고 나와요

A. 이미 Google Cloud 결제 계정을 쓰고 있는 계정은 Starter Tier 대상이 아닙니다. Cloud 프로젝트를 연결하거나, AI 기능이 없다면 Pages로 우회하세요.

Q. 주소를 열었더니 404

A. 저장소 최상위에 index.html이 있어야 합니다(18쪽). 하위 폴더에 있다면 AI Studio에 "index.html을 최상위로 옮겨줘"라고 요청한 뒤 다시 Push하세요.

Q. 게임은 뜨는데 AI 기능이 안 돼요

A. 정상입니다. GitHub Pages는 정적 호스팅이라 서버가 없어 AI 호출이 작동하지 않습니다. AI 기능이 필요하다면 Publish(Cloud Run) 경로를 쓰세요(5쪽).

Q. Source가 이미 GitHub Actions예요

A. AI Studio가 배포 절차를 함께 올린 것입니다. 잘 배포되고 있다면 바꾸지 마세요. Actions 탭에서 결과만 확인하면 됩니다(22·24쪽).

Q. 학교 망에서 안 열려요

A. 교육청 망 필터링일 가능성이 큼니다. 수업 전 날 반드시 교실 기기 한 대로 접속 테스트를 하고, 막히면 정보부장·교육청에 도메인 허용을 요청하세요.

5단원 클리어 — 내 게임에 주소가 생겼습니다

이제 선생님의 게임은 선생님 컴퓨터를 떠났습니다. 링크 한 줄이면 교실 태블릿에서도, 학생의 집 스마트폰에서도 열립니다. 다음 6단원에서는 이 게임에 "우리 반 순위표"를 붙여, 학생들의 도전이 서로에게 보이도록 만듭니다.

STEP 1~2

출판 지도와 공유 링크

4종 비교 · 갈림길 판단 · Share

STEP 3

Cloud Run 게시

Publish 3클릭 · 배포 후 점검 5

STEP 4~5

GitHub와 코드 올리기

계정·저장소·공개 범위 · Push · 파일 확인

STEP 6

Pages로 URL 발급

Source 두 갈래 · 배포 확인 · 접속

STEP 7~8

배포와 회수

QR·수업 배치 · 고쳐 올리기 · 즉시 닫기

남은 숙제

만든 주소를 다음 수업 학습지 하단에 QR로 넣어 보고, 실제 학생 반응을 한 문장으로 기록해 두세요 — 6단원 리더보드 설계의 재료가 됩니다.

다음 단원 예고

WORLD 6 · 실시간 리더보드 — Firebase Firestore를 붙여 TOP 10과 내 순위를 보여주고, 교사용 회차 초기화까지 만듭니다.



WORLD 5 CLEAR

▶ NEXT: WORLD 6

주소를 가진 게임은 더 이상 내 자료가 아니라 우리 반의 자료입니다. 다음 단원에서 만나요.



WORLD 6

실시간 리더보드

우리 반의 기록이 서로에게 보이도록

5단원까지 선생님의 게임에는 주소가 생겼습니다. 그런데 학생 서른 명이 각자 자기 화면에서 점수를 얻어도, 그 기록은 서로에게 보이지 않습니다. 이 단원은 그 벽을 허웁니다. **Firebase Firestore**를 붙여 우리 반의 기록을 한곳에 모으고, 실명 없이 회차별로 보이는 순위표를 만듭니다. 그리고 그 데이터베이스 앞에 문지기 — 보안 규칙 — 를 세우는 일까지가 이 단원의 몫입니다.



6

교재 6단원

6단원 퀘스트 — 무엇을 얻어 가나요?

리더보드는 기술적으로는 30분이면 붙습니다. 어려운 것은 그 다음입니다 — 누구의 이름을 띄울 것인가, 몇 등까지 보여줄 것인가, 언제 지울 것인가. 그리고 "내가 만든 데이터베이스는 지금 안전한가". 이 단원의 절반은 그 판단에 씁니다.

1 순위표를 교육적으로 설계한다

순위표가 동기를 낮추기도 한다는 연구를 바탕으로, 교실에서 지킬 설계 원칙 5가지를 먼저 정한다.

2 Firebase 콘솔을 혼자 연다

프로젝트 만들기부터 Firestore, 인증, 보안 규칙까지 실제 화면을 눌러 가며 익힌다.

3 데이터를 눈으로 확인한다

컬렉션·문서·필드가 콘솔에서 어떻게 보이는지, 개인정보가 섞여 있지 않은지 직접 점검한다.

4 로그인을 걷어낸다

기본으로 켜지는 Google 로그인을 익명 로그인 + 닉네임으로 바꿔, 개인정보를 아예 수집하지 않는 구조로 만든다.

5 보안 규칙을 읽고 고친다

"allow read, write: if true" 가 무슨 뜻인지 알아보고, 점수 범위·수정 금지를 직접 넣는다.

6 교사용 회차 초기화를 만든다

기록을 지우지 않고 새 회차를 여는 방식으로, 매 수업 깨끗한 순위표에서 시작한다.

8단계 리더보드 로드맵

총 100분 · 5단원까지 만든 게임을 그대로 재료로 씁니다

앞의 두 단계(설계 판단·개념 이해)를 건너뛰고 바로 만들면, 만들고 나서 지우게 됩니다. 실제 연수에서 가장 많이 나온 후회가 "실명으로 만들었다가 다시 뜯어고쳤다"입니다. 순서대로 가세요.

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6	STEP 7	STEP 8
Firestore 이해	프로젝트 만들기	Firestore 열기	익명 인증 켜기	보안 규칙	앱에 붙이기	리더보드 구현	회차 초기화
클라우드 칠판 비유	콘솔 3단계 마법사	컬렉션·문서·필드	로그인 건너내기	내 규칙 검사 → 수정	Enable Firebase	TOP 10 + 내 순위	교사용 패널
8분	12분	15분	12분	18분	10분	15분	10분



먼저 읽고 가세요 — STEP 4(익명 인증)를 뒤로 미루면 안 됩니다. Firebase 인증은 기본적으로 Google 로그인이고, 초등학생 계정으로 로그인시키는 순간 만 14세 미만 개인정보 문제가 그대로 걸립니다. 그리고 STEP 5(보안 규칙)는 절대 건너뛸 수 없는 단계입니다 — 22쪽에서 그 이유를 실제 화면으로 보여드립니다.

순위표는 약인가 독인가 — 만들기 전에 정할 것

게이미피케이션 연구는 리더보드에 대해 한 방향으로 말하지 않습니다. Hanus와 Fox(2015)는 배지와 리더보드를 넣은 수업에서 한 학기 동안 내재적 동기·만족도·시험 점수가 오히려 낮아지는 것을 관찰했습니다. 도구가 나쁜 것이 아니라, 순위표가 "남과의 비교"를 학습의 중심에 놓을 때 그렇게 됩니다. 아래 다섯 가지는 그 위험을 줄이는 설계입니다.

① 실명 금지 — 닉네임만

개인정보 문제이자 낙인 문제입니다. "3반 김OO 28위"가 화면에 뜨는 순간, 그 학생에게 이 게임은 벌이 됩니다.

② 회차마다 초기화

누적 순위는 만들지 않습니다. 순위표의 수명은 한 차시입니다. 어제 진 학생이 오늘 다시 0에서 시작할 수 있어야 합니다.

③ 내 기록을 나란히

"우리 반 1위 820점" 옆에 "내 최고 기록 340 → 410점"을 함께 띄웁니다. 비교 대상을 남에서 어제의 나로 옮기는 장치입니다.

④ TOP 10까지만 공개

전체 순위는 공개하지 않습니다. 상위권은 보이게 하되 하위권은 남에게 보이지 않게 — 자기 순위는 자기 화면에만.

⑤ 점수는 성취가 아니다

반복하면 오르는 구조임을 학생에게 먼저 말해 둡니다. 순위를 평가·상점·자리 배치에 연결하는 순간 설계가 무너집니다.



이론적 배경 — 숙달목표(내가 어제보다 나아졌는가)와 수행목표(내가 남보다 나은가)는 학습에 다른 결과를 냅니다(Ames, 1992). ③번은 숙달목표를 화면에 남겨두기 위한 장치입니다.

길이 바뀝니다 — 이제 서버가 필요합니다

5단원의 갈림길을 다시 꺼내 봅니다

5단원에서 우리는 "이 게임은 실행 중에 AI에게 말을 거는가?"로 길을 갈랐습니다. 리더보드는 AI를 부르지는 않지만, 여러 사람의 기록을 한곳에 모아 두고 서로 읽어야 합니다. 즉 저장소가 필요합니다. 질문을 한 칸 넓혀 봅시다.

내 게임은 "내 화면 밖"의 무언가가 필요한가? (AI 호출 · 공용 데이터)

NO — 코드만으로 완결

5단원의 분수 낚시 대모험. 문항·규칙이 코드 안에 있고 점수는 내 화면에만 남습니다.

→ **GitHub Pages**

무료 · 영구 · 내 계정 소유. 그대로 두면 됩니다.

YES — 서버가 필요함

리더보드가 붙은 순간 여기로 넘어옵니다. 남의 점수를 읽고 내 점수를 남기려면 공용 저장소가 있어야 합니다.

→ **Firestore + Publish (Cloud Run)**

Starter Tier로 결제 없이 앱 2개까지. Firestore·Auth·Cloud Run이 한 세트로 붙습니다.



5단원에서 만든 GitHub Pages 주소는 그대로 살아 있습니다 — 다만 리더보드가 붙은 버전은 그 주소에서 순위표만 비어 보입니다. 정적 호스팅에는 데이터를 담을 곳이 없기 때문입니다. 리더보드 버전의 최종 주소는 Publish로 발급되는 Cloud Run 주소이고, 학생에게는 그쪽을 나눠 줍니다.

Firestore — 교실 뒤에 걸린 클라우드 칠판

용어 3개면 충분합니다

Firestore는 구글이 운영하는 데이터베이스입니다. 교사에게 필요한 이미지는 하나입니다 — 전국 어디서든 보이는 칠판이 하나 걸려 있고, 누가 거기에 무엇을 적으면 그 칠판을 보고 있던 모든 사람의 화면이 즉시 바뀝니다. "새로고침을 누르지 않아도 바뀐다"가 실시간(real-time)의 뜻입니다.

컬렉션 (Collection)

같은 종류의 기록을 모아 두는 상자

비유 ▶ 칠판에 붙인 "점수" 코너. 우리는 scores 라는 상자 하나만 씁니다.

문서 (Document)

상자 안에 들어가는 쪽지 한 장

비유 ▶ "바다탐험가 · 740점 · 3-2반 · 10시 12분" 이 한 줄이 문서 하나입니다.

필드 (Field)

쪽지에 적는 항목

비유 ▶ nickname · score · classCode · createdAt. 여기에 이름·학번·이메일은 넣지 않습니다.

실제로 저장되는 모습 — scores 컬렉션 안의 문서 하나

```
{
  nickname: "바다탐험가",
  score: 740,
  classCode: "3-2-r1",
  createdAt: 2026-08-25 10:12:07
}
```

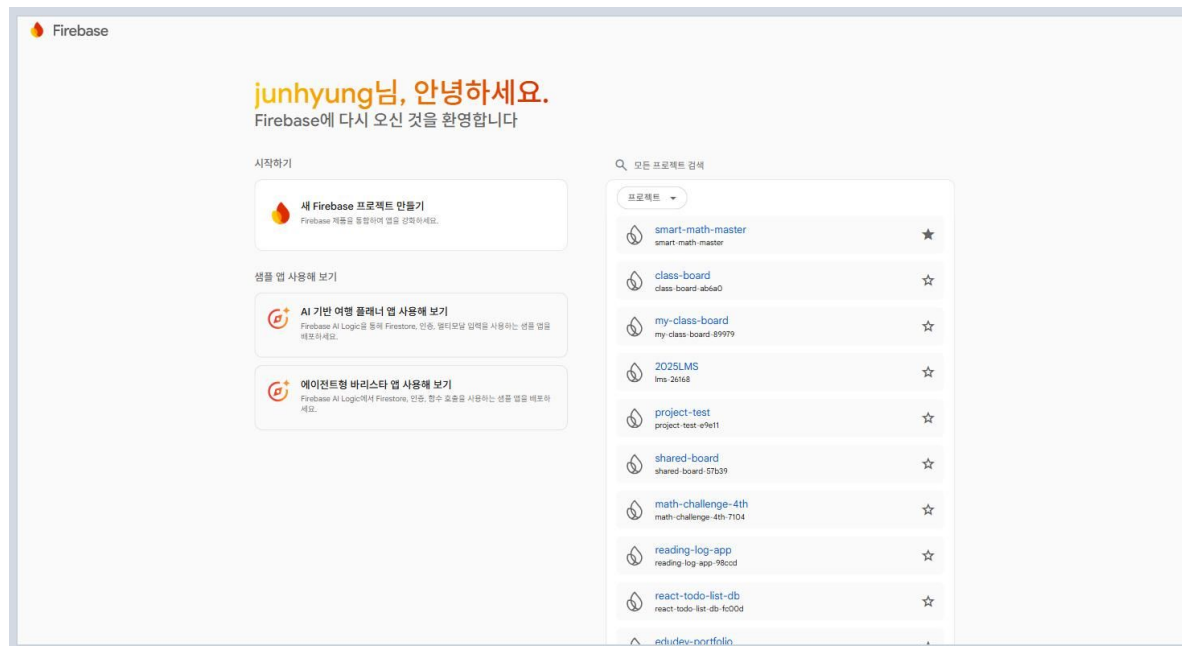
"실시간"이 공짜가 아닌 이유

학생 화면 하나하나가 칠판을 계속 지켜보는 눈(리스너)을 하나씩 갖습니다. 누군가 점수를 올리면 지켜보던 모든 눈에게 바뀐 쪽지가 전달되고, 그 전달 하나하나가 "읽기"로 계산됩니다. 그래서 리더보드는 쓰기보다 읽기를 훨씬 많이 씁니다. 31~32쪽에서 실제 사용량 화면과 함께 다시 다룹니다.

Firebase 콘솔 첫 화면 — 지도부터 보기

console.firebase.google.com · 교사 업무용 구글 계정으로 로그인

주소창에 console.firebase.google.com 을 칩니다. 구글 계정으로 로그인하면 아래 화면이 나옵니다. 왼쪽은 새로 만들기, 오른쪽은 이미 가진 프로젝트 목록입니다. 처음이라면 오른쪽이 비어 있습니다.



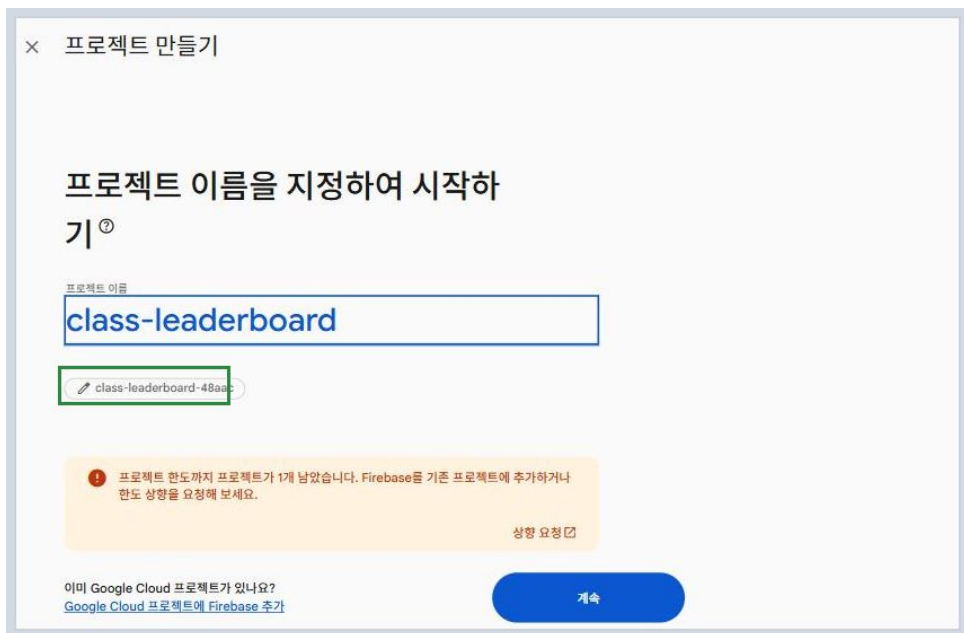
Firebase 콘솔 홈 — 왼쪽 "새 Firebase 프로젝트 만들기", 오른쪽 프로젝트 목록

- 주소창에 console.firebase.google.com**
검색하지 말고 주소를 직접 치는 편이 빠릅니다. 즐겨찾기에 넣어 두세요.
- 교사 업무용 구글 계정으로 로그인**
이 프로젝트는 만든 계정에 묶입니다. 개인 계정보다 업무용 계정을 권합니다.
- 왼쪽 "새 Firebase 프로젝트 만들기"**
처음이라면 여기서 시작합니다. 다음 쪽으로.
- 오른쪽은 이미 만든 프로젝트**
한 계정에 만들 수 있는 프로젝트 수에는 한도가 있습니다 (다음 쪽 참고).

i 무료로 씁니다. Firebase의 Spark 요금제는 결제 카드 등록 없이 쓸 수 있고, 이 단원의 모든 실습이 그 안에서 끝납니다.

프로젝트 만들기 ① — 이름 정하기

프로젝트는 "우리 반 게임 데이터가 사는 집" 한 채입니다. 이름을 치면 그 아래에 자동으로 전 세계에서 유일한 프로젝트 ID가 붙습니다(초록 표시). 이 ID는 나중에 주소와 설정 곳곳에 나타나므로 알아보기 쉬운 영문 이름을 쓰는 편이 좋습니다.



프로젝트 이름 입력 → 아래에 자동 생성된 프로젝트 ID (class-leaderboard-48aac)

1

영문 이름으로 짓습니다

class-leaderboard, math-4th-board 처럼. 한글도 되지만 ID가 복잡해집니다.

2

프로젝트 ID는 자동으로 붙습니다

이름 뒤에 임의의 다섯 글자가 붙어 전 세계에서 유일해집니다. 연필 아이콘으로 고칠 수 있지만 그대로 두면 됩니다.

3

"계속"을 누릅니다

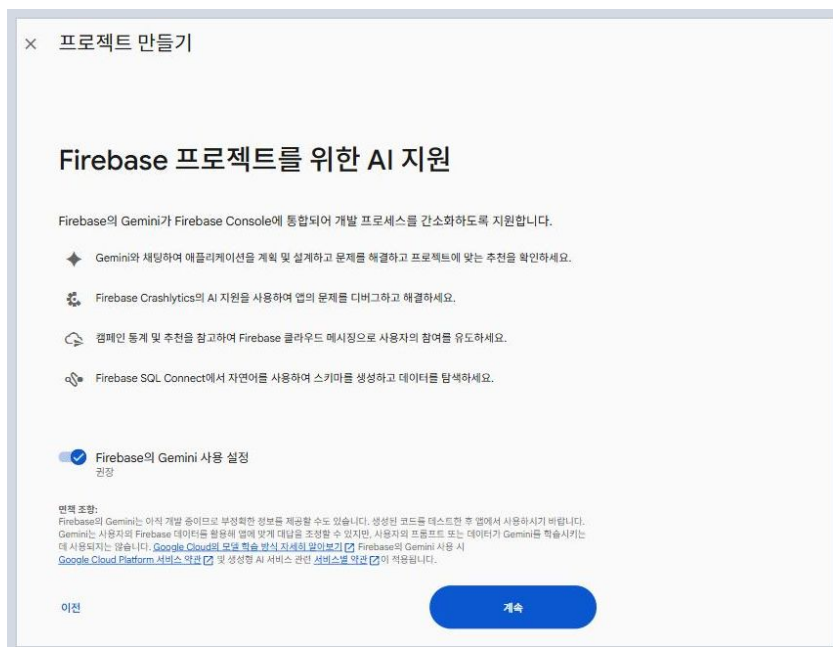
아직 아무것도 만들어지지 않았습니다. 3단계를 다 지나야 실제로 생성됩니다.

!

프로젝트 한도 — 한 구글 계정이 만들 수 있는 프로젝트 수에는 상한이 있습니다. 남은 개수가 적으면 캡처의 주황 박스 같은 안내가 뜹니다. 한도에 걸렸다면 ① 안 쓰는 프로젝트를 정리하거나 ② 기존 프로젝트에 Firestore만 추가해 쓰거나 ③ 안내의 "상향 요청"을 누릅니다. 학년 전체가 각자 만든다면 미리 확인해 두세요.

프로젝트 만들기 ② — Firebase의 Gemini

두 번째 화면은 Firebase 콘솔 안에서 Gemini의 도움을 받을지 묻습니다. 콘솔에서 막힐 때 채팅으로 물어볼 수 있게 해 주는 기능입니다. 켜 두어도 되고 꺼도 됩니다 — 우리 실습의 결과는 달라지지 않습니다.



"Firebase의 Gemini 사용 설정" 토글 — 기본은 켜짐

켜 두면 좋은 점

콘솔 화면에서 "이 규칙이 무슨 뜻이야?" 같은 질문을 바로 할 수 있습니다. Firestore 화면 위쪽에 뜨는 "Firestore를 사용하기 위한 핵심 개념에 관해 Gemini에게 물어보기" 버튼이 그것입니다.

학교에서 생각할 점

면책 조항에 적혀 있듯 생성된 답이 부정확할 수 있습니다. 또 교육청 지침상 AI 기능 사용에 제한이 있는 경우가 있으니, 판단이 서지 않으면 꺼 두고 진행해도 이 교재의 모든 실습이 그대로 됩니다.

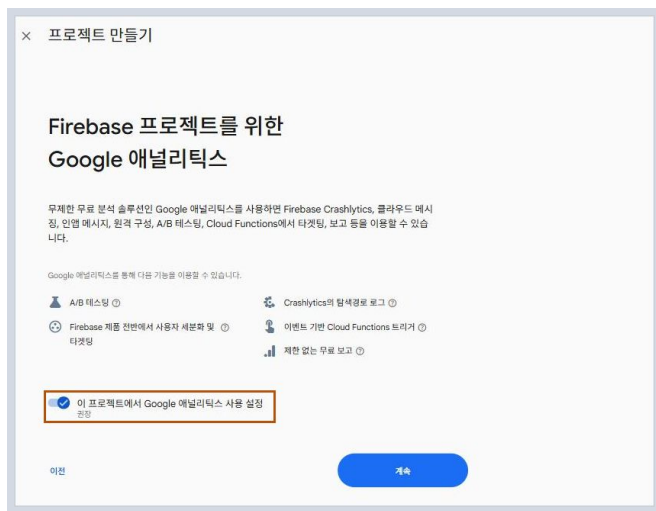
i

결정했으면 "계속"을 누릅니다. 아직도 프로젝트는 만들어지지 않았습니 — 다음 쪽이 마지막 단계이고, 거기서 처음으로 실제 생성이 일어납니다.

프로젝트 만들기 ③ — Google 애널리틱스는 끄세요

이 단원에서 기본값을 바꾸는 유일한 곳입니다

마지막 화면은 Google 애널리틱스를 켜지 묻고, 기본값은 "켜짐"입니다. 애널리틱스는 누가 언제 어떤 화면을 봤는지 수집·분석하는 도구입니다. 우리가 만드는 것은 초등학생이 쓰는 학습 게임이고, 이 단원의 원칙은 "개인정보를 아예 수집하지 않는다"입니다. 그래서 이 토글은 끕니다.



기본값: 켜짐 (주황 표시)



토글을 끄면 기능 목록에 취소선이 붙습니다

1

토글을 클릭해 끕니다

오른쪽 캡처처럼 취소선이 그어지면 꺼진 것입니다.

2

"프로젝트 만들기" 클릭

버튼 이름이 "계속"에서 바뀌어 있습니다. 여기서 처음으로 실제 생성이 시작됩니다.

3

30초~1분 기다립니다

"준비 완료" 안내가 뜨면 계속 진행합니다.

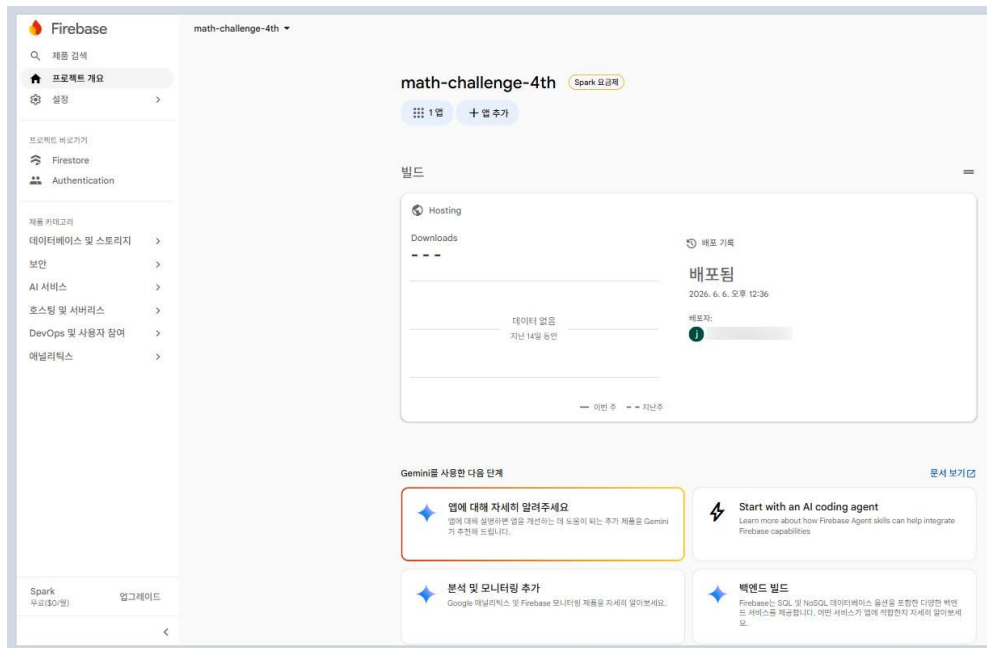
!

왜 끄는가 — 애널리틱스는 그 자체로 불법이 아닙니다. 다만 만 14세 미만 아동의 정보를 다루려면 법정대리인 동의가 필요하고(개인정보 보호법 제22조의2), 학생 이용 기록을 구글 계정과 연결해 쌓아 둘 이유가 우리 수업에는 없습니다. 켜지 않으면 동의 문제 자체가 생기지 않습니다. 나중에 필요해지면 콘솔에서 언제든지 켤 수 있으니, 지금은 꺼 두는 쪽이 안전합니다.

프로젝트 개요 화면 읽는 법

앞으로 이 화면을 계속 오가게 됩니다

프로젝트가 만들어지면 이 화면으로 들어옵니다. 메뉴가 많아 보이지만 이 단원에서 쓰는 곳은 왼쪽 사이드바의 두 군데뿐입니다 — Firestore(데이터)와 Authentication(인증). 나머지는 지금 몰라도 됩니다.



프로젝트 개요 — 왼쪽 사이드바에 Firestore · Authentication 바로가기

프로젝트 이름 옆 배지

"Spark 요금제" = 무료 요금제라는 뜻. 왼쪽 아래에도 "Spark 무료(\$0/월)"이 늘 표시됩니다.

프로젝트 바로가기

Firestore와 Authentication. 이 단원에서 실제로 누르는 곳은 여기 둘뿐입니다.

제품 카테고리

데이터베이스 및 스토리지 / 보안 / AI 서비스 / 호스팅 등. 바로가기에 없을 때 여기서 찾습니다.

앱 추가

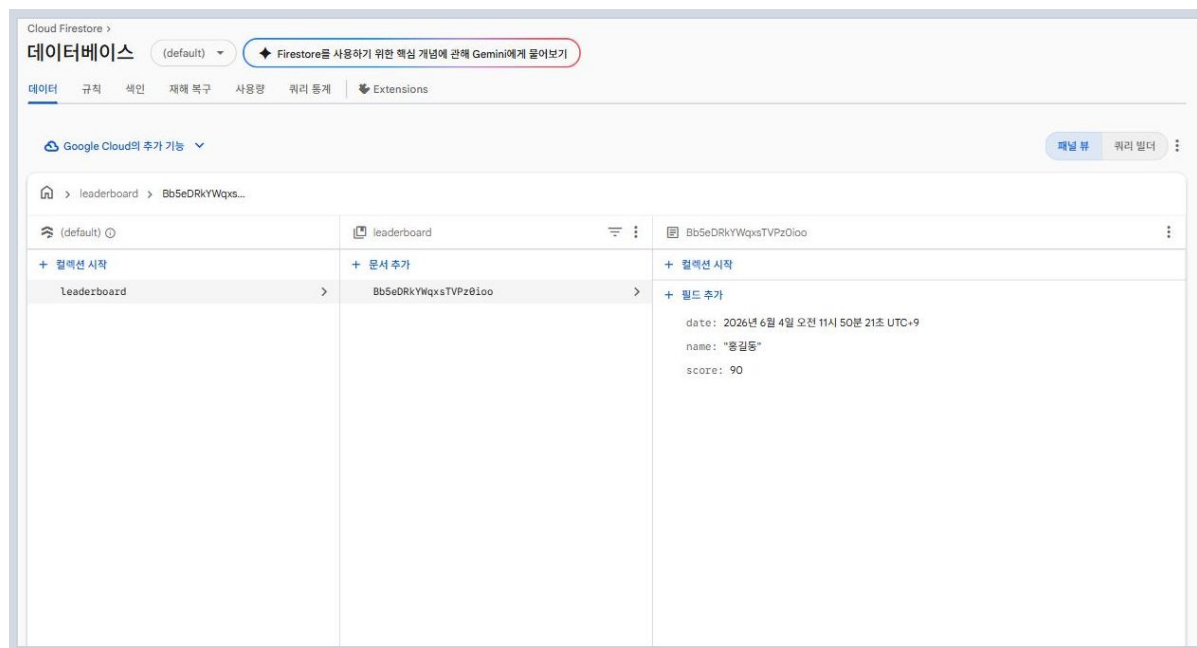
웹·안드로이드·iOS 앱을 이 프로젝트에 연결합니다. AI Studio를 쓰면 자동으로 붙습니다.

왼쪽 아래 "업그레이드"

유료(Blaze) 전환 버튼입니다. 이 단원에서는 누르지 않습니다.

Firestore 열기 — 3단 패널 구조 익히기

사이드바의 Firestore를 누릅니다(처음이라면 "데이터베이스 만들기"부터 나오고, 위치와 보안 규칙 모드를 고르면 이 화면이 됩니다). 화면은 왼쪽부터 컬렉션 → 문서 → 필드 순서로 세 칸이 이어집니다. 위쪽 탭 여섯 개도 함께 봐 두세요.



Firestore 데이터 화면 — 왼쪽부터 컬렉션 / 문서 / 필드

왼쪽 칸

컬렉션 목록. 캡처에서는 leaderboard 하나가 있습니다.

가운데 칸

선택한 컬렉션 안의 문서 목록. 문서 ID는 자동 생성된 무작위 문자열입니다.

오른쪽 칸

선택한 문서의 필드. date · name · score 세 개가 보입니다.

위쪽 탭

데이터 / 규칙 / 색인 / 재해 복구 / 사용량 / 쿼리 통계. 우리는 앞의 둘과 사용량만 씁니다.

맨 아래

"데이터베이스 위치: nam5"처럼 리전이 표시됩니다. 한 번 정하면 바꿀 수 없습니다.

컬렉션 만들기 ① — 상자에 이름 붙이기

왼쪽 칸 위의 "+ 컬렉션 시작"을 누르면 두 단계짜리 대화상자가 열립니다. 1단계는 상자 이름(컬렉션 ID)만 정하면 됩니다. 우리 리더보드는 점수를 담은 상자 하나면 충분하므로 scores 라고 짓겠습니다.

컬렉션 시작 1단계 — 상위 경로 "/" 아래에 컬렉션 ID 입력

1

왼쪽 칸의 "+ 컬렉션 시작"

데이터 탭에서 첫 번째 칸 맨 위에 있습니다.

2

상위 경로가 "/" 인지 확인

최상위에 만든다는 뜻입니다. 문서 안에 또 컬렉션을 넣을 수도 있지만 우리는 쓰지 않습니다.

3

컬렉션 ID에 scores 입력

영문 소문자 권장. 나중에 프롬프트와 보안 규칙에서 이 이름을 그대로 씁니다.

4

"다음"을 누릅니다

컬렉션은 문서가 하나라도 있어야 만들어집니다. 그래서 2단계로 이어집니다.

i

이름 짓기 요령 — 컬렉션은 여러 쪽지가 들어가는 상자이므로 복수형(scores, students)이 자연스럽습니다. 나중에 AI에게 지시할 때도 "scores 컬렉션에 저장해줘"처럼 그대로 부르게 됩니다.

컬렉션 만들기 ② — 첫 쪽지 한 장 써 보기

2단계는 상자에 넣을 첫 쪽지(문서)를 만듭니다. 여기서 문서 ID, 필드 이름, 값의 유형, 실제 값을 정합니다. 실제 수업에서는 학생이 게임을 하면 코드가 알아서 쪽지를 넣지만, 구조를 눈으로 익히려면 한 번은 직접 써 보는 편이 좋습니다.

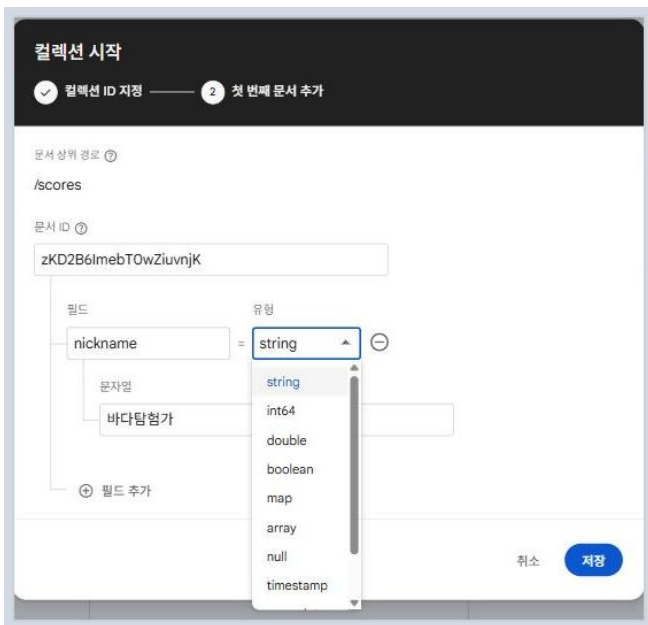
컬렉션 시작 2단계 — 문서 ID · 필드 · 유형 · 값

- 1 **문서 ID → "자동 ID" 클릭**
오른쪽의 "자동 ID"를 누르면 무작위 문자열이 채워집니다. 직접 짓지 마세요 — 겹치면 서로 덮어씁니다.
- 2 **필드 이름에 nickname**
왼쪽 "필드" 칸입니다. 영문 소문자로 짓고, 코드·프롬프트·규칙에서 같은 철자를 씁니다.
- 3 **유형은 string 그대로**
닉네임은 글자이므로 문자열(string)입니다. 유형 목록은 다음 쪽에서 자세히 봅니다.
- 4 **값에 예시 닉네임 입력**
캡처에서는 "바다탐험가". 실명이 아니라 사물·동물·색깔 조합을 예시로 쓰세요.
- 5 **"필드 추가"로 나머지 3개**
score(int64) · classCode(string) · createdAt(timestamp)를 같은 방식으로 더합니다.
- 6 **"저장"을 누릅니다**
컬렉션과 첫 문서가 함께 만들어집니다.

필드 유형 — 무엇을 골라야 하나

유형을 잘못 고르면 순위가 엉뚱하게 정렬됩니다

유형 칸을 누르면 목록이 펼쳐집니다. 이름은 낮설지만 교사가 실제로 쓰는 것은 셋뿐입니다. 특히 점수를 문자열(string)로 저장하면 "100"이 "90"보다 앞에 오는 사고가 납니다 — 글자 순서로 정렬되기 때문입니다.



유형 목록 — string · int64 · double · boolean ·
map · array · null · timestamp ...

string

글자. 닉네임, 반 코드처럼 글로 된 값. ← 우리 nickname, classCode

int64

정수. 점수, 문제 수처럼 소수점이 없는 숫자. ← 우리 score

double

소수. 평균 점수처럼 소수점이 있는 숫자. 리더보드에는 잘 안 씁니다.

boolean

참/거짓. "클리어했는가" 같은 예-아니오 값.

timestamp

날짜와 시각. 언제 기록했는지. ← 우리 createdAt

map / array

값 안에 값이 또 들어가는 형태. 초등 실습에서는 쓰지 않아도 됩니다.

null

값이 비어 있음을 뜻합니다. 일부러 쓸 일은 거의 없습니다.



score는 반드시 int64. string으로 두면 "1000점"이 "999점"보다 아래로 내려갑니다.

우리 리더보드의 데이터 설계 — 넣을 것과 넣지 않을 것

기능을 정하기 전에 "무엇을 저장할 것인가"를 먼저 정합니다. 여기서 정한 네 칸이 곧 프롬프트가 되고, 보안 규칙이 되고, 콘솔에서 눈으로 점검할 목록이 됩니다. 오른쪽 "넣지 않는 것"이 이 단원에서 더 중요합니다.

넣을 것 — scores 컬렉션의 문서 한 장

nickname string · 학생이 직접 정한 별명 (2~8자)

score int64 · 그 판에서 얻은 점수

classCode string · 반과 회차 (예: 3-2-r1)

createdAt timestamp · 기록한 시각

넣지 않는 것 — 하나라도 들어가면 설계가 무너집니다

이름 · 학번 실명은 물론 "3반민준이" 같은 별명도 실명입니다.

이메일 · 사진 익명 로그인을 쓰면 애초에 받아오지 않습니다.

학교 이름 반 코드만으로 충분합니다. 학교명이 붙으면 특정이 쉬워집니다.

기기 정보 · 위치 수집할 이유가 없습니다. 요청하지도 마세요.

이렇게 정해 두면 나중에 이 한 줄이 프롬프트가 됩니다

scores 컬렉션에 { nickname(문자), score(정수), classCode(문자), createdAt(시각) } 네 필드만 저장해줘.
이름·학번·이메일·사진·학교명은 절대 저장하지 마.

로그인을 건너냅니다 — 초등 교실의 핵심 갈림길

기본값은 Google 로그인입니다. 그대로 두면 안 됩니다

Firebase 인증을 켜면 Google 로그인이 기본으로 따라옵니다. 편리하지만 초등 교실에서는 그대로 쓸 수 없습니다. 학생 계정으로 로그인하는 순간 이메일·이름·프로필 사진이 앱에 넘어오고, 만 14세 미만 아동의 개인정보 처리에는 법정대리인 동의가 필요해집니다. 순위표에 필요한 것은 신원이 아니라 "누가 누구인지 구분만 되는 표식"입니다.

A안 · 익명 로그인 + 닉네임

권장

개인정보를 하나도 받지 않으면서 사용자마다 임시 식별자(uid)를 발급합니다. 학생은 닉네임만 입력합니다.

- 보안 규칙에서 "로그인한 사람만 쓰기 가능"을 걸 수 있어 장난 데이터를 막기 쉽습니다
- 콘솔에서 익명 제공업체를 한 번 켜 줘야 합니다 (18~21쪽에서 화면으로 안내)
- 익명 계정은 30일 자동 정리 대상이 될 수 있으나, 회차마다 초기화하는 우리 설계에는 문제가 없습니다

B안 · 인증 없이 공개 쓰기

차선

로그인 자체를 없애고 누구나 점수를 남길 수 있게 규칙을 엮습니다. 콘솔 설정이 필요 없어 가장 간단합니다.

- 주소만 알면 누구나 점수를 쓸 수 있습니다 — 장난 점수(999999)를 막을 장치가 규칙 검증뿐입니다
- 학교 밖에 주소가 퍼지면 손쓰기 어렵습니다
- 한 차시짜리 공개 수업처럼 짧고 통제된 상황에서만 쓰세요

실습 프롬프트 ① — 로그인 건너내고 닉네임으로 (A안)

이 앱은 초등학생이 쓰는 앱이라 개인정보를 모으면 안 돼. Google 로그인 화면을 없애고, 대신 Firebase 익명 로그인 (Anonymous Authentication) 을 써줘. 게임 시작 전에 닉네임 입력창만 보여주고, 닉네임은 한글·영문 2~8자로 제한해줘. 이메일·이름·프로필 사진 같은 개인정보는 받지도 저장하지도 마.

Authentication 열기 — 지금 누가 등록되어 있나

사이드바에서 Authentication을 누릅니다. 처음이라면 "시작하기" 버튼이 보이고, 누르면 아래 화면이 됩니다. 첫 탭인 "사용자"는 이 앱에 등록된 계정 목록입니다. 캡처에서는 구글 계정 하나가 등록되어 있습니다 — 우리가 바꾸려는 것이 바로 이 상태입니다.



사용자 탭 (이메일과 UID는 흐림 처리)



익명 로그인으로 바꾸면 이 목록은 어떻게 달라질까요? 식별자 칸이 비고 제공업체 아이콘이 사라진 줄이 학생 수만큼 쌓입니다. 이름도 이메일도 없는 임의의 UID만 남습니다 — 그것이 우리가 원하는 상태입니다.

식별자

누구인지 알아보는 값. 구글 로그인이면 이메일 주소가 그대로 들어옵니다. 익명 로그인이면 이 칸이 비어 있습니다.

제공업체

어떤 방식으로 로그인했는지. 캡처의 아이콘은 Google 로그인입니다.

사용자 UID

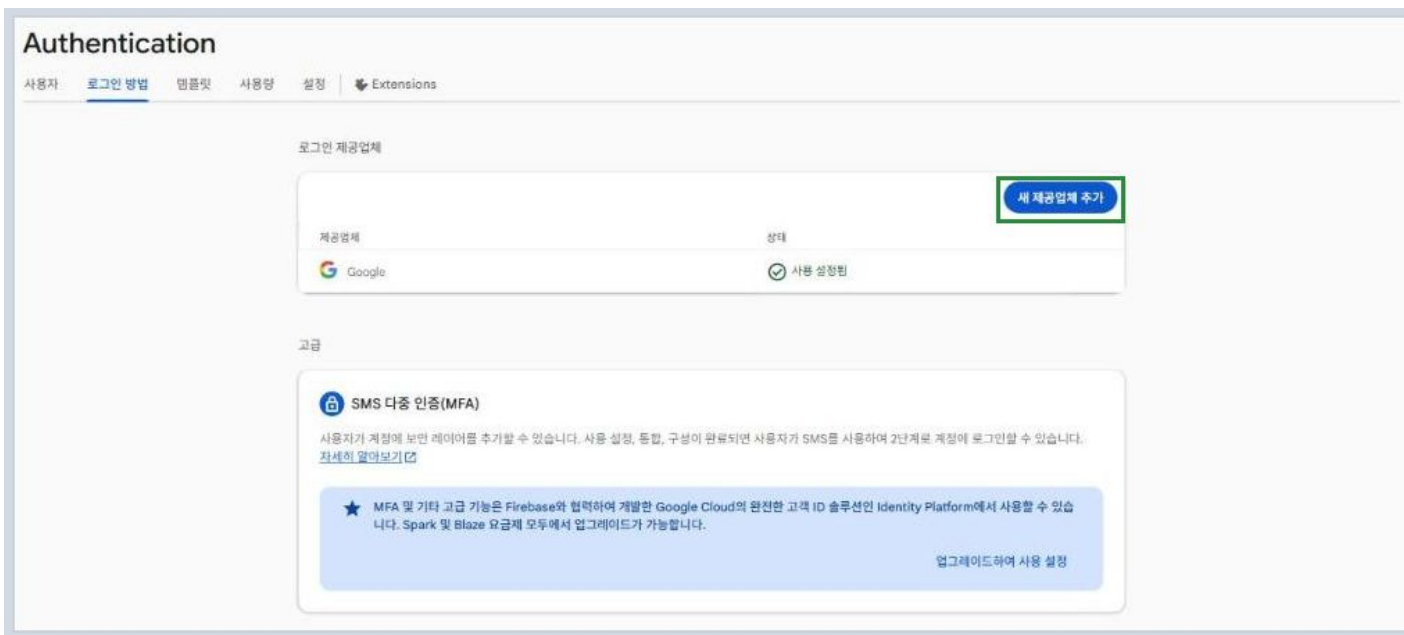
Firebase가 내부적으로 쓰는 임의의 식별자. 익명 로그인은 이것만 생깁니다 — 개인정보가 아닙니다.

다섯 탭

사용자 / 로그인 방법 / 템플릿 / 사용량 / 설정. 우리는 앞의 두 개만 씁니다.

"로그인 방법" 탭 — 제공업체 추가하기

두 번째 탭 "로그인 방법"으로 갑니다. 여기에 현재 켜져 있는 로그인 수단이 나열됩니다. 캡처에서는 Google 하나가 "사용 설정됨" 상태입니다. 익명 로그인을 더하려면 오른쪽 위 "새 제공업체 추가"(초록 표시)를 누릅니다.



로그인 방법 탭 — 로그인 제공업체 목록과 "새 제공업체 추가" 버튼

1 Authentication → "로그인 방법" 탭

제목 바로 아래 두 번째 탭입니다.

2 현재 목록을 확인

Google이 "사용 설정됨"이면 그대로 두어도 됩니다. 익명을 더해도 서로 부딪히지 않습니다.

3 "새 제공업체 추가" 클릭

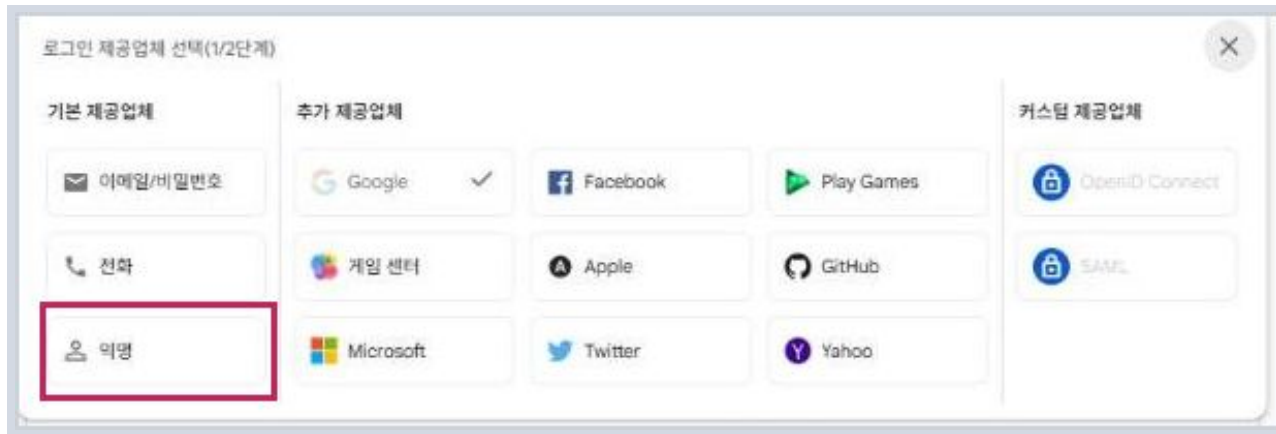
오른쪽 위 파란 버튼(캡처의 초록 표시).



Google 로그인을 지울까요? — 지우지 않아도 됩니다. 앱 코드에서 익명 로그인만 부르면 구글 로그인 화면은 나타나지 않습니다. 다만 학생용으로 배포하기 전에 실제로 구글 로그인 화면이 뜨지 않는지 반드시 한 번 확인하세요.

제공업체 목록에서 "익명" 고르기

대화상자가 열리면 로그인 수단이 세 묶음으로 나뉘어 보입니다. 우리가 찾는 것은 왼쪽 "기본 제공업체" 맨 아래의 익명입니다(분홍 표시). 이름 그대로, 아무 정보도 받지 않고 사용자를 구분만 해 주는 방식입니다.



로그인 제공업체 선택(1/2단계) — 기본 제공업체 맨 아래가 익명



"이메일/비밀번호"는 왜 안 되나 — 초등학생에게 계정과 비밀번호를 만들게 하는 순간 관리 책임이 교사에게 넘어옵니다. 잊어버린 비밀번호를 재설정해 주는 일이 매 수업 생깁니다. 우리 목적(순위 구분)에는 익명이면 충분합니다.

기본 제공업체

이메일/비밀번호, 전화, 익명. 우리가 쓰는 것은 익명 하나입니다.

추가 제공업체

Google, Facebook, Apple, GitHub 등 외부 계정 로그인. 초등 수업에서는 쓰지 않습니다.

커스텀 제공업체

학교·기관의 자체 인증 시스템을 붙일 때 씁니다. 유료 요금제가 필요합니다.



익명을 클릭하면 바로 2/2단계로 넘어갑니다. 다음 쪽이 이 실습의 마지막 관문입니다.

익명 사용 설정 — 이 스위치 하나가 실습을 좌우합니다

익명을 고르면 2/2단계가 나옵니다. 할 일은 스위치를 켜고 저장하는 것뿐입니다. 이 스위치를 켜지 않으면 앱에서 점수 저장이 조용히 실패합니다 — 화면에 오류도 안 뜹니다. 실습에서 가장 자주 막히는 지점이니 꼭 확인하세요.



제공업체 구성(2/2단계) — "사용 설정" 토글(분홍 표시)과 "자동 정리" 체크박스

1

"사용 설정" 토글을 켵니다

오른쪽 위 스위치. 캡처는 꺼진 상태입니다 — 켜면 파란색이 됩니다.

2

"자동 정리"도 켜길 권합니다

30일이 지난 익명 계정을 자동 삭제합니다. 회차마다 초기화하는 우리 설계에서는 잃을 것이 없고, 사용량·결제 할당량에도 집계되지 않습니다.

3

"저장"을 누릅니다

목록에 "익명 · 사용 설정됨"이 추가되면 완료입니다.

!

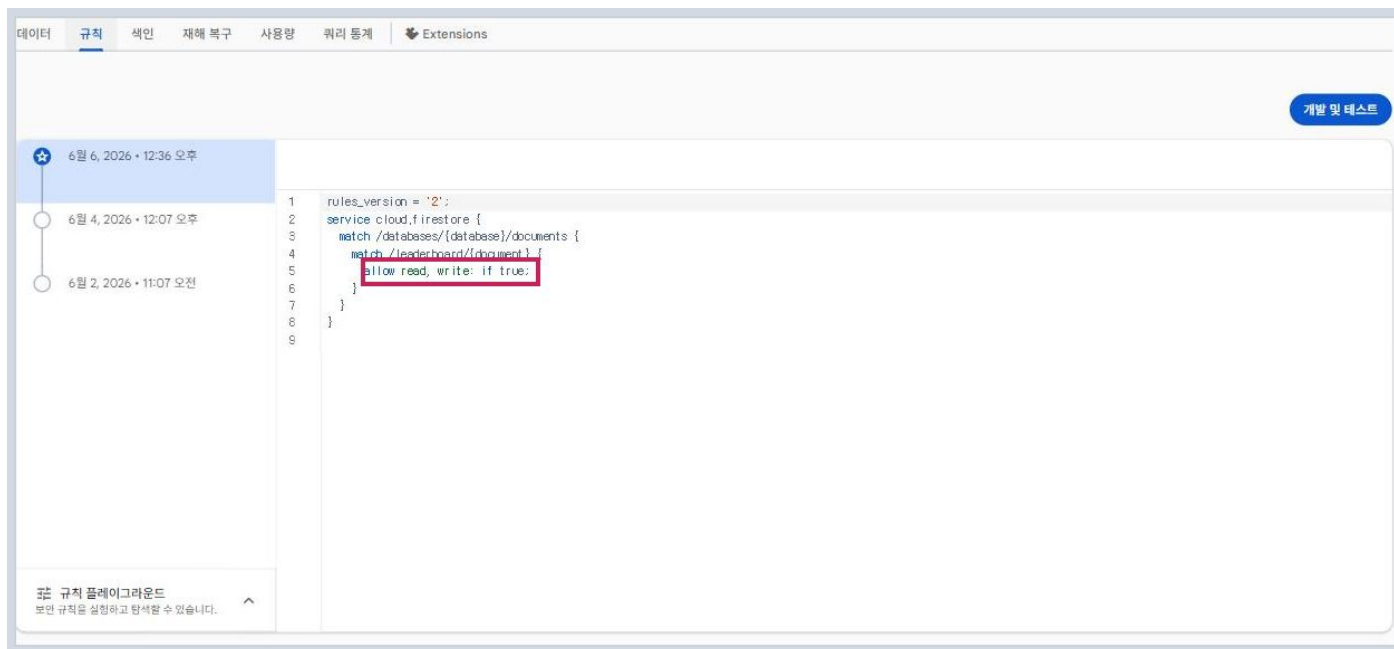
켜지 않으면 어떻게 되나 — 앱은 정상으로 보입니다. 게임도 잘 돌아가고 오류 메시지도 뜨지 않습니다. 그런데 점수만 저장되지 않고, 순위표는 계속 비어 있습니다. 원인을 찾느라 프롭트를 몇 번이고 다시 쓰게 되는 자리입니다.

실습 중 "순위표가 비어요"라는 말이 나오면 먼저 이 스위치부터 확인하세요 — 절반 이상이 여기서 해결됩니다.

보안 규칙 — 지금 내 데이터베이스는 안전한가?

이 단원에서 가장 중요한 화면입니다

Firestore의 "규칙" 탭을 엽니다. 여기에 쓰인 몇 줄이 데이터베이스 앞을 지키는 문지기 전부입니다. 아래는 실제 교사 프로젝트에서 촬영한 화면입니다. 5번째 줄을 소리 내어 읽어 보세요.



규칙 탭 — 왼쪽은 버전 기록, 오른쪽이 현재 적용 중인 규칙 (분홍 표시가 문제의 줄)

지금 적용 중인 규칙

```
match /leaderboard/{document} {
  allow read, write: if true;
}
```

이것이 뜻하는 것

"조건 없이(if true) 읽기와 쓰기를 모두 허용한다."

즉 주소만 알면 누구나 점수를 넣고, 남의 점수를 고치고, 전부 지울 수 있습니다.

드문 일이 아닙니다 — 테스트 모드로 시작하거나 AI가 초안을 쓰면 흔히 이 상태로 남습니다. 학생에게 열기 전에 반드시 고쳐야 합니다.

규칙 문장 해부 — 여섯 줄만 읽으면 됩니다

보안 규칙은 프로그래밍 언어처럼 보이지만 구조는 아주 단순합니다. "어느 경로에(match) 어떤 동작을(allow) 어떤 조건일 때(if) 허용한다" 한 문장의 반복입니다. 아래 표만 이해하면 내 규칙이 안전한지 스스로 판단할 수 있습니다.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /leaderboard/{document} {
      allow read, write: if true;
    }
  }
}
```

write는 셋을 합친 말입니다 — create(새로 만들기) · update(고치기) · delete(지우기). 그래서 "allow write"라고 쓰면 남의 점수를 고치고 지우는 것까지 함께 열립니다. 우리는 다음 쪽에서 이 셋을 갈라, create만 허용하고 update와 delete는 막을 것입니다.

i

반면 read를 열어 두는 것은 괜찮습니다 — 순위표는 모두가 봐야 하니까요. 다만 그것은 저장하는 필드에 개인정보가 없기 때문에 괜찮은 것입니다(16쪽 설계).

rules_version	규칙 문법의 버전. '2'로 두면 됩니다.
service cloud.firestore	이 규칙이 Firestore용이라는 선언. 고정입니다.
match /.../documents	이 데이터베이스의 모든 문서를 뜻하는 바깥 울타리. 고정입니다.
match /leaderboard/{...}	★ 어느 상자에 적용할지. leaderboard 컬렉션의 모든 문서.
allow read, write	★ 무엇을 허용할지. read=읽기, write=만들기·고치기·지우기 전부.
if true	★ 언제 허용할지. true = 언제나, 즉 조건이 없음. 가장 위험한 값.

고쳐 쓰기 — 우리 교실에 맞는 문지기

규칙도 프롬프트로 고칩니다. 직접 타이핑할 필요 없습니다

"점수는 몇 점까지가 정상인가" 같은 것은 우리 교실 사정이라 AI가 알 수 없습니다. 아래 프롬프트로 요청한 뒤, 결과를 규칙 탭에 붙여넣고 "게시"를 누릅니다. 오른쪽은 그렇게 나온 규칙의 예시입니다.

실습 프롬프트 ② — 문지기 손보기

firestore.rules 를 이렇게 고쳐줘.

- scores 컬렉션은 로그인 (익명 포함) 한 사용자만 create 가능
- score는 정수이고 0 이상 99999 이하일 때만 허용
- nickname은 문자열이고 8자 이하일 때만 허용
- 이미 저장된 문서는 update·delete 모두 거부
- 읽기 (read) 는 누구나 가능

왜 점수 상한을 두나

개발자 도구로 값을 바꿔 999999점을 올리는 학생이 실제로 나옵니다. 우리 게임의 최고점보다 조금 넉넉한 값을 상한으로.

왜 수정·삭제를 막나

한 번 올린 점수를 누가 고치거나 지우면 순위표를 믿을 수 없습니다. 지우는 일은 교사가 콘솔에서만 하도록 남깁니다.

이런 규칙이 나옵니다 (예시)

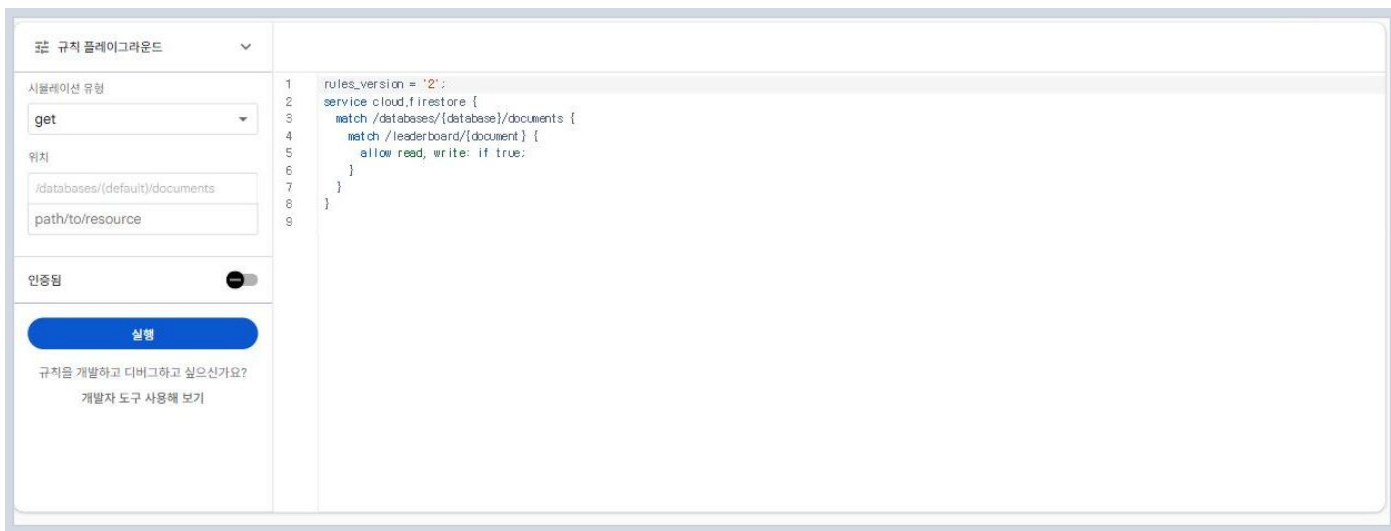
```
match /scores/{doc} {
  allow read: if true;
  allow create: if request.auth != null
    && request.resource.data.score is int
    && request.resource.data.score >= 0
    && request.resource.data.score <= 99999
    && request.resource.data.nickname is string
    && request.resource.data.nickname.size() <= 8;
  allow update, delete: if false;
}
```



붙여넣은 뒤 반드시 "게시"를 누르세요. 편집만 하고 게시하지 않으면 예전 규칙이 계속 살아 있습니다. 왼쪽 버전 기록에 새 날짜가 생기면 반영된 것입니다.

규칙 플레이그라운드 — 게시하기 전에 시험해 보기

규칙은 조용히 실패합니다. 화면에 오류가 안 뜬다고 통과한 것이 아닙니다. 그래서 콘솔에는 규칙을 실제로 시험해 보는 도구가 있습니다. 규칙 탭 왼쪽 아래의 "규칙 플레이그라운드"를 펼치면 됩니다.



1

시뮬레이션 유형

get(읽기) / create(만들기) / update(고치기) / delete(지우기) 중 시험할 동작을 고릅니다.

2

위치

경로를 적습니다. 예: scores/test1

3

"인증됨" 토글

켜면 로그인한 사용자로, 끄면 로그인하지 않은 사람으로 시험합니다.

4

"실행"을 누릅니다

허용/거부가 색으로 표시되고, 규칙의 어느 줄에서 걸렸는지도 알려 줍니다.

규칙 플레이그라운드 — 시뮬레이션 유형 · 위치 · 인증됨 토글 · 실행

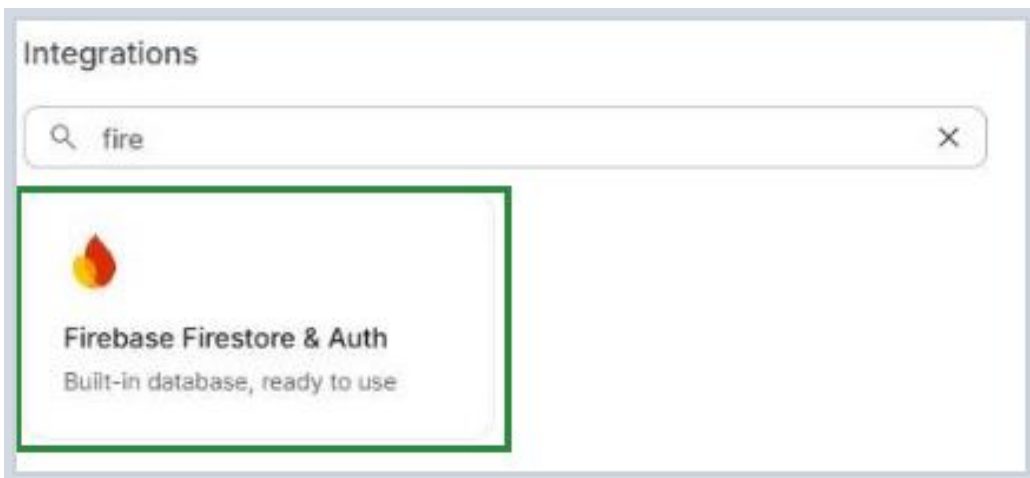
+

꼭 해 볼 네 가지 — ① 인증됨 켜고 create → 허용되어야 함 ② 인증됨 끄고 create → 거부되어야 함 ③ 인증됨 켜고 update → 거부되어야 함 ④ 인증됨 끄고 get → 허용되어야 함(순위표는 모두가 봄).

네 가지가 모두 예상대로 나오면 규칙이 제대로 걸린 것입니다. 하나라도 어긋나면 게시하지 말고 규칙을 다시 고치세요.

앱에 붙이기 ① — AI Studio에서 Firebase 찾기

여기서 무대가 Firebase 콘솔에서 AI Studio로 옮겨 갑니다. 5단원에서 쓰던 앱을 열고 오른쪽 위 톱니바퀴를 눌러 패널을 엽니다. 탭 중 Integrations를 고르고 검색창에 "fire"를 치면 하나가 나옵니다 — Firebase Firestore & Auth.



Integrations 탭에서 "fire" 검색 → Firebase Firestore & Auth

1

5단원에서 만든 앱을 엽니다

aistudio.google.com/apps → My apps에서 고릅니다.

2

오른쪽 위 톱니바퀴 클릭

패널이 열리고 Chat / Share / Publish / Versions / GitHub / Integrations / Secrets 탭이 보입니다.

3

Integrations 탭 선택

구글 앱 연동 목록이 나옵니다(드라이브·시트·지메일 등).

4

검색창에 fire 입력

"Firebase Firestore & Auth — Built-in database, ready to use" 카드가 남습니다.

5

카드를 클릭

다음 쪽의 설정 패널로 넘어갑니다.

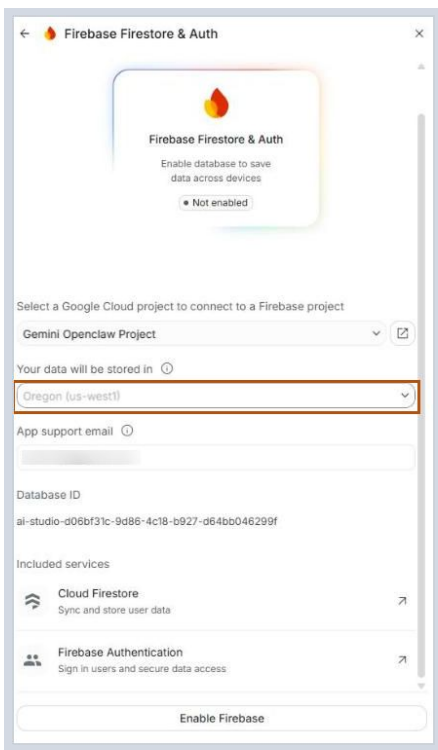
i

프롬프트에 "Firebase로 저장해줘"라고 쓰면 이 카드가 자동으로 뜨기도 합니다. 둘 중 어느 쪽이든 도착지는 같습니다.

앱에 붙이기 ② — Enable Firebase 패널 뜯어보기

누르기 전에 반드시 확인할 칸이 하나 있습니다

카드를 누르면 이 패널이 열립니다. 맨 아래 "Enable Firebase" 한 번이면 데이터베이스·인증·보안 규칙 초안이 자동으로 붙습니다. 다만 그 위의 "Your data will be stored in"(주황 표시)은 한 번 정하면 바꿀 수 없습니다. 누르기 전에 여기부터 보세요.



상태 표시

카드에 "Not enabled"라고 적혀 있으면 아직 붙지 않은 상태입니다.

Google Cloud 프로젝트

어느 프로젝트에 붙일지 고릅니다. 앞에서 만든 프로젝트를 선택하세요.

★ 데이터 저장 위치

캡처의 기본값은 Oregon(us-west1) — 미국입니다. 국내 asia-northeast3(서울)로 바꾸는 편이 좋습니다. 한 번 정하면 변경 불가.

App support email

문제가 생겼을 때 표시될 연락처. 교사 업무용 계정으로.

Database ID

자동으로 붙는 이름. 손대지 않습니다.

Included services

Cloud Firestore(데이터 저장·동기화)와 Firebase Authentication(로그인·접근 제어)이 함께 커집니다.

Enable Firebase

누르면 실제로 붙습니다. 잠시 뒤 코드에 firebase 설정 파일과 firestore.rules가 생깁니다.

firebaseConfig — 이 값은 비밀이 아닙니다

5단원의 API 키 경고와 헛갈리기 쉬운 지점

연동이 끝나면 코드에 firebaseConfig라는 덩어리가 생깁니다(콘솔의 프로젝트 설정에서도 볼 수 있습니다). apiKey라는 이름 때문에 5단원에서 배운 "키를 공개 저장소에 올리지 마라"와 충돌하는 것처럼 보이지만, 성격이 다릅니다.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDwoH",
  authDomain: "math-challenge-4th-7104.firebaseio.com",
  projectId: "math-challenge-4th-7104",
  storageBucket: "math-challenge-4th-7104.firebaseio.com",
  messagingSenderId: "135079135455",
  appId: "1:135079135455:web:1234567890"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

프로젝트 설정 → 일반 → 내 앱의 SDK 설정 (실제 값은 흐림 처리)

Gemini API 키 (5단원에서 경계한 것)

진짜 비밀입니다. 이 키를 가진 사람은 내 계정으로 AI를 호출해 요금을 발생시킬 수 있습니다. 그래서 공개 저장소에 올리면 안 되고, 서버 쪽에 숨겨야 합니다.

firebaseConfig의 apiKey (지금 보는 것)

비밀이 아닙니다. "우리 프로젝트가 어디에 있는지"를 가리키는 주소표에 가깝습니다. 누구나 브라우저 개발자 도구로 볼 수 있고, 그렇게 설계되어 있습니다. 실제 방어는 보안 규칙이 합니다.

=

한 줄 요약 — 문은 열려 있어도 됩니다. 문 안쪽에서 문지기(보안 규칙)가 지키니까요.

리더보드 구현 — 4쪽의 설계 원칙을 프롬프트로 옮기기

여기서 4쪽에서 정한 다섯 가지 원칙이 실제 문장이 됩니다. "순위표 만들어줘"라고만 하면 AI는 전체 순위를 실명 비슷하게 주르륵 나열하는 흔한 형태로 만듭니다. 원칙을 문장으로 적어 넣는 것이 이 단원의 실력입니다.

실습 프롬프트 ③ — 리더보드 구현

리더보드를 이렇게 만들어줘.

- scores 컬렉션에 { nickname, score, classCode, createdAt } 문서를 저장
- 화면 오른쪽에 TOP 10만 실시간으로 표시 (1~3위는 금·은·동 색)
- 전체 순위는 보여주지 마. 내가 10위 밖이면 목록 아래에 "나: 17위 / 340점" 한 줄만 내 화면에 따로 보여줘
- TOP 10 옆에 "내 최고 기록"을 항상 같이 띄워줘
- 같은 닉네임은 최고 점수 하나만 남겨줘
- classCode는 주소 뒤 ?class=3-2 로 받아 반별로 순위를 나눠줘

프롬프트의 각 줄이 어떤 원칙에서 나왔는가

① 실명 금지

nickname 필드만 저장 — 이름·학번 필드를 아예 만들지 않음

② 회차 초기화

classCode에 회차를 붙여 판을 나눔 (STEP 8에서 완성)

③ 내 기록 병기

"내 최고 기록"을 TOP 10 옆에 상설 배치

④ TOP 10까지만

전체 순위 금지, 내 순위는 내 화면에만

⑤ 점수 ≠ 성취

최고 점수 1개만 남겨 반복 누적을 막음

+

실행 화면에서 확인할 것 — ① TOP 10만 나오는가 ② 내 최고 기록이 옆에 있는가 ③ 10위 밖일 때 내 순위 한 줄이 뜨는가 ④ 옆자리 기기에서 점수를 올리면 새로고침 없이 바뀌는가.

그리고 Firebase 콘솔의 Firestore 데이터 화면을 열어, 문서에 필드가 정확히 네 개뿐인지 눈으로 확인하세요.

교사용 초기화 — 지우지 않고 새 판을 엽니다

4쪽 원칙 ②(회차마다 초기화)를 구현할 차례입니다. 그런데 방금 우리는 규칙에서 삭제를 막았습니다. 모순처럼 보이지만 아닙니다. 초기화를 "지우기"가 아니라 "새 회차 코드 발급"으로 설계하면, 기록은 안전하게 남고 화면은 깨끗하게 비워집니다.

실습 프롬프트 ④ — 교사용 회차 패널

교사용 패널을 만들어줘.

- 주소 뒤에 `?teacher=1` 을 붙였을 때만 보이는 작은 패널
- "새 회차 시작" 버튼: `classCode` 뒤에 회차 번호를 붙여 새 코드를 만들고 (예: 3-2-r2) 그 코드로 순위표를 새로 시작
- 이전 회차 기록은 지우지 말고 그대로 남겨둬
- 현재 회차의 학생용 링크와 QR을 패널에 보여줘

진짜로 지워야 할 때

실수로 부적절한 닉네임이 올라갔거나 학년 말에 정리할 때는 Firebase 콘솔에서 직접 지웁니다. 콘솔 → Firestore → 문서를 고른 뒤 오른쪽 위 점 세 개 메뉴에서 삭제하면 됩니다. 보안 규칙은 앱에서 오는 요청만 막을 뿐, 프로젝트 주인인 교사의 콘솔 작업은 막지 않습니다.

즉 삭제 권한을 교사 한 사람에게만 남겨 두는 구조 — 이것이 규칙에서 delete를 거부한 진짜 이유입니다.

회차 코드가 도는 방식

1교시

3-2-r1 · 1반 1회차. 학생 링크 `...?class=3-2-r1`

다음 차시

3-2-r2 · "새 회차 시작" → 화면이 비워지고 새 링크 발급

옆 반

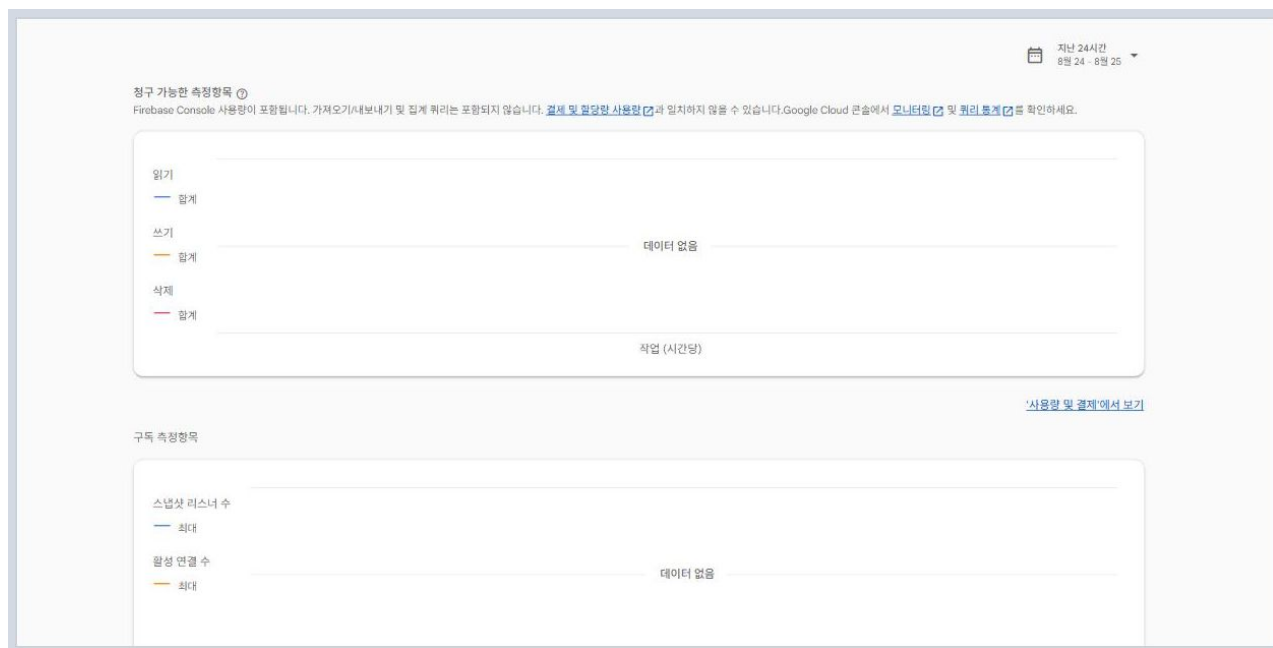
3-3-r1 · 반이 다르면 코드가 달라 순위가 섞이지 않음



`?teacher=1` 은 잠금장치가 아닙니다. 주소를 알아낸 학생은 똑같이 열 수 있습니다. 이것은 "평소에는 화면을 어지럽히지 않기 위한 정리 장치"이지 보안이 아닙니다. 진짜로 막아야 할 것 (점수 조작·삭제)은 22~24쪽의 보안 규칙이 막습니다.

사용량 화면 — 읽기가 먼저 닳는 것을 눈으로 확인

Firestore의 "사용량" 탭을 열면 이 데이터베이스가 얼마나 쓰였는지 그래프로 보입니다. 수업 한 차시를 돌린 다음 이 화면을 한 번 열어 보세요 — 6쪽에서 말한 "읽기가 쓰기보다 훨씬 많다"를 직접 확인할 수 있습니다.



사용량 탭 — 청구 가능한 측정항목(읽기·쓰기·삭제)과 구독 측정항목(스냅샷 리스너·활성 연결)

읽기

순위표가 갱신되어 화면에 전달될 때마다 1. 가장 먼저 닳습니다.

쓰기

학생이 점수를 저장할 때마다 1. 생각보다 적게 씁니다.

삭제

우리 규칙은 삭제를 막았으므로 거의 0이어야 정상입니다.

스냅샷 리스너 수

지금 순위표를 지켜보고 있는 화면의 개수 = 접속한 학생 수에 가깝습니다.

활성 연결 수

동시에 붙어 있는 연결. 수업 중 실시간 참여 인원을 가늠할 수 있습니다.

무료 한도 — 우리 학교는 몇 반까지 쓸 수 있나

AI Studio가 만들어 준 데이터베이스는 결제 계정 없이 쓰는 대신 하루 사용량 한도를 여러 앱이 나눠 씁니다. 한도를 넘으면 태평양 시간 자정 무렵 다음 날에 초기화됩니다. 아래 왼쪽이 공식 한도이고, 오른쪽은 그 한도를 교실 규모로 옮겨 본 추정입니다.

공식 공유 한도 (여러 앱이 함께 사용)

저장 용량	1 GiB	점수 한 건이 100바이트도 안 되니 사실상 걸릴 일 없음
쓰기 (write)	40,000 / 일	점수를 저장할 때마다 1
읽기 (read)	50,000 / 일	실시간 갱신이 여기를 씁니다 — 가장 먼저 닿는 칸
실시간 업데이트	50,000 / 일	지켜보던 화면에 변경이 전달될 때마다 1
네트워크 송신	10 GiB / 월	텍스트만 오가므로 여유 있음



사용량 및 결제 → 세부정보 및 설정에서 현재 요금제를 확인할 수 있습니다 (Spark = 무료).

한 반 40분 수업 추정 — 직접 계산해 보세요

학생 30명 × 한 명당 20회 도전
 쓰기 = 600건 (한도의 1.5%)
 읽기 ≈ 15,000~20,000건 (한도의 30~40%)
 → 하루 2~3개 학급이 안전선

읽기를 아끼는 법 ①

TOP 10만 구독합니다. 전체 순위를 실시간으로 받으면 읽기가 몇 배로 납니다.

읽기를 아끼는 법 ②

학년 전체를 하루에 돌린다면 실시간 대신 "게임이 끝날 때만 갱신"으로 바꿔 달라고 요청하세요.

한도를 넘겼다면

앱이 멈추지는 않고 저장·갱신만 실패합니다. 다음 날 자정 무렵 자동으로 풀립니다.

리더보드판 안전 수칙 — 학생에게 열기 전 6칸

5단원 점검표에 리더보드 전용 항목을 더했습니다

5단원의 여섯 칸(학생 이름·사진·API 키·저작권·닉네임·회수 절차)은 그대로 유효합니다. 여기에 데이터베이스가 붙으면서 새로 생긴 위험이 여섯 가지 더 있습니다. 첫 수업 전에 소리 내어 읽고 체크하세요.

□ 보안 규칙 확인

if true 로 열려 있지 않은지 규칙 탭에서 직접 눈으로 확인합니다. 이 단원에서 가장 중요한 칸입니다.

□ 닉네임 필터

학생은 부적절한 닉네임을 넣습니다. 반드시 넣습니다. 금지어 목록을 프롬프트에 넣고, 뚫리면 콘솔에서 지웁니다.

□ 실명·별명 주의

"김민준"은 물론 "3반민준이"도 실명입니다. 사물·동물·색깔 조합을 예시로 주세요.

□ 리전은 변경 불가

Enable Firebase 전에 위치를 정해야 합니다. 국내 asia-northeast3(서울)을 권장합니다.

□ 교사 계정으로만

Firebase 프로젝트는 만든 계정에 묶입니다. 업무용 계정으로 만들고, 학생에게 콘솔 주소를 알려주지 않습니다.

□ 수업 후 처리

학년이 끝나면 프로젝트를 정리하거나 데이터를 삭제합니다. 익명 데이터라도 남길 이유가 없으면 지우는 편이 낫습니다.

! 되짚기 — 우리 설계는 이메일도 이름도 받지 않습니다(익명 로그인 + 닉네임). 그래서 만 14세 미만 아동의 개인정보 처리에 해당하지 않도록 만든 것입니다. 이 전제가 무너지는 순간(로그인을 다시 켜거나, 실명 입력칸을 추가하는 순간) 법정대리인 동의 문제가 되살아납니다.

100분 미션 — 우리 반 순위표 완성하기

5단원에서 배포한 게임을 그대로 재료로 씁니다. 캡처를 옆에 두고 순서대로 따라오세요. 마지막 산출물은 "옆 선생님 기기에서 내 순위표가 실시간으로 바뀌는 장면"입니다.

0~8분	설계 먼저 정하기	4쪽 원칙 5가지 중 우리 반에 그대로 적용할 것 / 바꿀 것 표시 → 옆 선생님과 이유 나누기
8~20분	프로젝트 만들기	콘솔 접속 → 이름 → Gemini 선택 → 애널리틱스 끄기 → 프로젝트 만들기 (7~10쪽)
20~35분	Firestore 열기	컬렉션 scores 만들고 첫 문서 직접 입력 → 필드 유형 확인 (12~15쪽)
35~47분	익명 인증 켜기	Authentication → 로그인 방법 → 새 제공업체 추가 → 익명 → 사용 설정 → 저장 (18~21쪽)
47~65분	보안 규칙 검사	규칙 탭에서 현재 규칙 확인 → 프롬프트 ②로 고치기 → 게시 → 플레이그라운드 4가지 시험 (22~25쪽)
65~75분	앱에 붙이기	AI Studio → Integrations → fire 검색 → 리전을 서울로 → Enable Firebase (26~27쪽)
75~90분	리더보드 구현	프롬프트 ③ → TOP 10 / 내 최고 기록 / 내 순위 세 가지 확인 → 콘솔에서 필드 4개 확인
90~100분	회차 패널 + 공유화	프롬프트 ④ → Publish로 배포 → 옆 선생님 기기에서 접속해 동시에 점수 올려보기

제출 양식 ▶ 교과·학년: _____ | 프로젝트 ID: _____ | 리전: _____ | 회차 코드: _____
 배포 주소(Cloud Run): https://_____ | 규칙 시험 4종 통과: ☐☐☐☐ 동시 접속 확인: ☐

자주 막히는 지점 8가지

리더보드 실습에서 가장 많이 나오는 질문입니다. 절반은 "콘솔에서 스위치 하나를 안 켜다"에 해당합니다.

Q. 점수가 저장이 안 돼요

A. 가장 흔한 원인은 익명 로그인을 켜지 않은 것입니다. Authentication → 로그인 방법 → 익명 → 사용 설정(21쪽). 그래도 안 되면 보안 규칙을 봅니다.

Q. 프로젝트를 더 못 만든대요

A. 계정당 프로젝트 한도입니다. 안 쓰는 프로젝트를 정리하거나, 기존 프로젝트에 Firestore만 추가해 쓰세요(8쪽).

Q. 1000점이 999점보다 아래예요

A. score를 string으로 저장했기 때문입니다. 글자 순서로 정렬됩니다. int64로 바꿔 달라고 요청하세요(15쪽).

Q. 이상한 닉네임이 올라왔어요

A. 콘솔 Firestore에서 해당 문서를 삭제합니다. 앱에서는 지울 수 없게 막아 두었으므로 교사만 지울 수 있습니다(30쪽).

Q. 오류는 없는데 순위표가 비어요

A. 규칙은 조용히 거부합니다. 콘솔의 Firestore에서 문서가 실제로 쌓이는지 직접 확인하는 것이 가장 빠릅니다.

Q. 리전을 잘못 골랐어요

A. 위치는 한 번 정하면 바꿀 수 없습니다. 실습 단계라면 새 프로젝트로 다시 만드는 편이 빠릅니다(27쪽).

Q. GitHub Pages에서 순위표가 안 보여요

A. 정상입니다. 정적 호스팅에는 서버가 없습니다. 리더보드 버전은 Publish로 만든 Cloud Run 주소를 학생에게 나눠 주세요(5쪽).

Q. 규칙을 고쳤는데 그대로예요

A. "게시"를 누르지 않았을 가능성이 큼니다. 편집만으로는 반영되지 않습니다. 왼쪽 버전 기록에 새 날짜가 생겼는지 확인하세요(24쪽).

6단원 클리어 — 그리고 전체 과정 완주

이론에서 출발해 도구를 익히고, 교과 게임을 만들고, 주소를 붙이고, 마지막으로 우리 반의 기록을 한곳에 모았습니다. 이제 선생님은 학생용 학습 게임을 처음부터 끝까지 혼자 만들어 배포할 수 있습니다.

WORLD 1·2

게임리터러시 이론

게임을 보는 눈 · MDA · AI 에이전트 사례

WORLD 3

AI Studio 첫걸음

접속부터 첫 게임 생성·수정까지

WORLD 4

교과 맞춤 게임

성취기준 → 메커닉 → 강화 → 플레이테스트

WORLD 5

온라인 출판

GitHub 저장소 · Pages · 무료 영구 주소

WORLD 6

실시간 리더보드

Firebase 콘솔 · 익명 인증 · 보안 규칙 · 회차 초기화

남은 숙제

만든 순위표를 다음 차시에 실제로 한 번 써 보고, 4쪽의 원칙 다섯 가지 중 무엇이 지켜졌고 무엇이 무너졌는지 한 문장씩 기록해 두세요.

더 나아가려면

학급 간 협동 모드, 교사 대시보드(어느 문항에서 많이 틀렸는지), 학생이 직접 문항을 만드는 편집 모드 — 모두 같은 방식으로 붙일 수 있습니다.



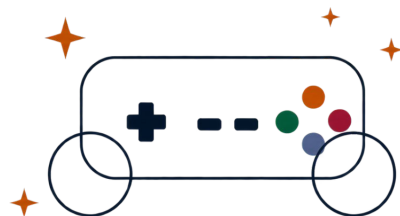
ALL WORLDS CLEAR

이제 선생님이 만드는 차례입니다 — 교실에서 실제로 한 번 돌려 보고, 그 기록을 다음 선생님에게 넘겨주세요.

교사를 위한 게임리터러시 · 연수 자료

AI 에이전트와 함께하는 교실 맞춤형 게임 제작 마스터

게임을 읽는 눈에서
우리 반 게임을 만드는 손까지



발 행 일 | 2026년 9월 14일

발 행 처 | 게임문화재단

발 행 인 | 유병한

지 은 이 | 이재호, 장준형, 정지원

감 수 | 김현경, 강세진

주 관 | 게임문화교육원

운 영 | (사)한국창의정보문화학회

주 소 | 서울특별시 서초구 방배로 114 다이치빌딩 2층 (게임문화재단)

전 화 | 02-586-3558

이 메 일 | 3cgameliteracy@gmail.com

I S B N | 979-11-24801-07-9(13370) [비매품]

- 이 책의 내용에 대한 저작권은 게임문화재단에 있으며, 2026 교사 게임리터러시 사업의 재정 지원에 의해 제작된 것입니다.
- 이 책을 게임리터러시 관련 교육 및 연수에 사용하시는 출처를 명시하여야 합니다.
- 이 책의 내용을 무단으로 복제 또는 전재하는 것은 법적으로 금합니다.

교사 한 사람이 세 가지를 갖고 돌아갑니다

1

교과 내용을 담은 학습 웹 게임 한 편

2

학생이 링크와 QR로 바로 여는 배포본

3

우리 반의 기록이 실시간으로 보이는 순위표

교재 「AI 에이전트와 함께하는 교실 맞춤형 게임 제작 마스터」(게임문화재단, 2026) · ISBN 979-11-24801-07-9 · 비매품

이 자료의 저작권은 게임문화재단에 있으며, 2026 교사 게임리터러시 사업의 재정 지원으로 제작되었습니다. 교육·연수에 사용할 때 출처를 밝혀 주세요.

게임문화재단

게임문화교육원
Game Culture Education Institute

(사)한국창의정보문화학회

