

누가 접속했을까? 어디서 일하고, 무엇을 남길까?

로그인 · 소스코드 · 서버 · 저장소 · 배포
함께 배우고, AI 공유 서버 탐정에게 도전하기

개념 15분 + 실습 25분 · 기본 40분

생활 속 비유로 핵심을 배우고, 강사 서버 1대에 각자의 AI 에이전트로 접속해 경쟁합니다. 실습은 5~10분 연장할 수 있습니다.

1교시 · 15분

도입 1분 · 로그인 5분
코드·서버·저장소 4분
접속 주소 2분 · 배포 3분

2교시 · 25분

접속·본인 확인 5분 · 전략 준비 3분
API 자료 읽기와 장소 선택
결과 확인·회고 5분

실습 목표: 행사 조건과 장소 자료를 비교해 알맞은 장소를 선택하고 근거를 설명하기.
수업 본문 1~11쪽 · 선택 활동과 참고 12~15쪽

로그인은 '내 기록의 주인'을 찾는 기준

“오늘 온 민수와 어제 온 민수는
같은 사람일까요?”

가장 익숙한 로그인은 **ID와 비밀번호(PW)**를 입력하는 방식입니다. ID는 “어느 계정인가”, 비밀번호는 “그 계정을 사용할 사람인가”를 확인하는 데 씁니다. 앱 안에서는 학생 번호처럼 **고유한 계정 번호**로 기록을 구분합니다.

왜 가장 기본적인 기능일까요?

앱이 '누구의 것인지'를 알아야 내 글, 내 기록, 내가 할 수 있는 일을 제대로 연결할 수 있습니다. 개인별 기록을 다루는 서비스에서는 다른 기능들이 로그인 위에 쌓입니다.

주요 기능

계정을 구분하면 가능한 일

내 글·사진·주문 내역

누가 만든 기록인지 찾아서 내 것만 보여 주기

학습 진도·즐거찾기·개인 설정

다시 방문하거나 다른 기기를 써도 내 기록 이어 보기

수정·삭제·관리자 기능

내 글을 고칠 사람과 운영 기록을 확인하고 초기화할 사람 구분

점수·사용 횟수·활동 이력

누구의 도전인지 기록하고 같은 기준으로 집계하기

ID·PW에서 OAuth 소셜 로그인으로



ID/PW: 가게마다 새 회원증 만들기 | OAuth 소셜 로그인: 기존 학생증으로 확인받기

ID/PW 방식에서는 우리 앱이 아이디와 비밀번호를 확인합니다. 소셜 로그인에서는 **Google 같은 곳이 본인을 확인하고, 우리 앱이 그 결과를 받아 입장시킵니다.** Google 비밀번호는 Google 화면에 입력합니다.

비교	ID / PW	소셜 로그인
가입·재방문	서비스마다 가입하고 비밀번호 기억	기존 계정으로 몇 번의 클릭
사용자 장점	외부 계정 없이 독립적으로 사용	새 비밀번호를 만들 부담이 적음
만드는 쪽	비밀번호 보관·분실 재설정 등을 준비	본인 확인 부담 일부를 외부에 맡김
주의할 점	비밀번호 분실·재사용·유출에 대응	외부 계정 정지·장애의 영향, 제공할 정보 확인

소셜 로그인 버튼을 많이 제공하는 이유

가입 도중 포기할 번거로움을 줄이고, 사용자는 관리할 비밀번호를 줄일 수 있습니다. 만드는 쪽도 인증 기능을 활용할 수 있습니다. 서비스 성격에 따라 ID/PW와 함께 제공하기도 합니다.

입장한 뒤에는? 앱이 로그인 상태를 유지해 주는 것은 ‘입장 팔찌’, 관리자 기능을 따로 제한하는 것은 ‘직원 전용 출입구’입니다. 소셜 로그인을 붙여도 **내 기록과 권한은 우리 앱이 관리합니다.**

흔히 ‘OAuth 소셜 로그인’이라고 부릅니다. OAuth는 접근 권한을 맡기는 약속이고, Google 로그인 같은 사용자 확인에는 OIDC도 사용합니다. 소셜 로그인이 모든 보안 문제를 대신 해결하는 것은 아닙니다.

출처: [Google · 소셜 로그인 개요](#), [OIDC Core](#). 회원증·학생증은 이해를 돕는 비유입니다.

코드는 레시피, 로컬은 우리 집 주방, 서버는 가게



소스코드 = 레시피 → 로컬 실행 = 우리 집 주방 → 서버 운영 = 손님을 받는 가게

IT 용어	요리에 비유하면	실제로는
소스코드	무엇을 어떻게 만들지 적은 레시피	앱이 할 일을 적어 둔 것
로컬 실행	우리 집 주방에서 요리해 맛보기	내 PC에서 앱을 켜고 시험
서버 운영	가게를 차려 손님 주문 받기	다른 사람의 요청을 받아 처리하고 응답
코드 저장소 · Git	레시피와 수정 이력 보관함	코드와 이전 버전을 보관
데이터 저장소 · 파일 · DB	누가 무엇을 주문했는지 적는 장부	회원·답안·점수 같은 기록 보관

레시피를 보관했다고 가게가 열리지는 않아요

GitHub에 코드를 올리는 일은 레시피를 보관하는 일입니다. 앱을 실행하고 사람들이 접속할 수 있게 준비해야 함께 쓸 수 있습니다.

오늘의 경기: 강사가 혼자 시험하면 우리 집 주방에서 맛보기, 참가자들의 질문을 받으면 가게 운영입니다. 질문은 주문, 응답은 음식, 경기 기록은 장부에 해당합니다. 오늘은 강사 PC가 서버 역할을 합니다.

서버를 끄면 가게가 문을 닫듯 접속이 멈춥니다. 파일로 보관한 코드와 경기 장부는 다시 켜도 이어 읽을 수 있습니다.

출처: [MDN · 웹 서버](#), [Git · 버전 관리](#)

IP는 실제 주소, 도메인은 기억하기 쉬운 이름

가게의 숫자 주소를 외우기보다 이름으로 찾는 편이 쉽습니다. 인터넷에서도 **사람은 도메인 이름을 쓰고, 컴퓨터는 IP 주소를 찾아 접속**합니다.

말	쉬운 비유	예시
IP 주소	찾아갈 실제 주소	192.168.0.2
도메인	주소 대신 기억하는 가게 이름	game.example · 설명용 이름
포트	도착한 곳의 담당 창구	:3000
localhost	"우리 집"이라고 부르기	지금 쓰는 내 컴퓨터 자신

DNS = 이름으로 주소를 찾는 주소록

도메인 입력 → DNS에서 연결할 IP 찾기 → 그 주소의 서버에 접속하기.

도메인을 사기만 해서는 앱이 실행되지 않습니다. 주소를 연결하고 서버도 켜 두어야 합니다.

오늘은 같은 공유기 안에서 접속합니다

http://192.168.0.2:3000/

오늘은 별도 도메인 없이 강사가 안내한 IP와 포트로 접속합니다. `game.example`은 설명용이며 실제 게임 주소가 아닙니다. 친구 기기에서 localhost를 열면 **친구 자신의 컴퓨터**로 갑니다.

`192.168.0.0/24`는 오늘 사용하는 대역 이름입니다. `192.168.0.`까지가 같은 건물, 마지막 숫자가 방 번호라고 생각해 보세요. `192.168.0.1`은 공유기 주소로 자주 쓰입니다.

현재 실제 IP를 확인해 사용합니다. 같은 공유기여도 게스트망·연결 차단 설정 때문에 접속이 막힐 수 있습니다. /24의 마스크는 255.255.255.0이며 자세한 숫자 계산은 오늘 생략합니다.

출처: [MDN · 도메인과 DNS](#), [Microsoft · 주소와 서브넷](#)

배포의 마지막은 다른 사람 기기에서 확인하기

03에서 레시피·주방·가게가 무엇인지 배웠고, 04에서 찾아갈 주소를 배웠습니다. 이제 **만든 앱을 다른 사람이 실제로 쓰게 하는 순서**를 봅니다. 이것이 배포입니다.

01 앱을 실행할 자리를 정한다

오늘은 강사 PC입니다. 인터넷 서비스라면 계속 운영할 컴퓨터나 서비스를 준비합니다.

02 코드와 필요한 설정을 넣고 실행한다

가게에 레시피와 재료를 준비하고 불을 켜듯, 앱을 켜고 저장소·클라우드 AI 연결을 확인합니다.

03 사람들이 찾아올 주소를 안내한다

오늘은 강사 IP와 포트를 보냅니다. 도메인을 쓰는 서비스라면 이름이 올바른 서버로 연결되게 설정합니다.

04 다른 사람 기기에서 끝까지 써 본다

참가자가 별명으로 들어가 질문하고 응답을 받는지, 자기 기록이 남는지 확인합니다.

“내 화면에서는 되는데요?”

우리 집에서 맛본 것과 손님이 가게에서 주문해 먹는 것은 다릅니다. **참가자의 기기에서 입장 → 질문 → 응답 → 기록까지** 확인해야 함께 쓸 준비가 된 것입니다.

운행을 계속하려면 서버를 켜 두고, 데이터 보관과 사용자별 권한도 준비합니다. 인터넷으로 공개할 때는 HTTPS 주소 같은 추가 준비가 필요합니다.

한 문장으로 · “배포는 실행하고, 주소를 알려 주고, 다른 사람이 실제로 쓸 수 있는지 확인하는 일이다.”

오늘은 같은 공유기 안에서 사용합니다. 실제 소셜 로그인 연결과 인터넷 공개 배포는 별도 실습에서 다룹니다.

출처: [Render](#) · [웹 서비스 배포](#), [데이터 영구 저장](#)

에이전트에게 맡기는 공유 서버 탐정 미션

에이전트가 서버에 있는 행사 조건과 장소 정보를 읽고, 모두 만족하는 장소를 골라 제출합니다. 여러 학생이 **서버 한 대**에 각자 에이전트로 접속해 언제든지 자기 속도로 도전합니다.

학생 A의 에이전트

조건 읽기
장소 비교
선택 제출

공유 서버 한 대

같은 행사 조건
같은 장소 목록
참가자별 기록 저장

학생 B의 에이전트

조건 읽기
장소 비교
선택 제출

오늘의 목표

로그인 정보가 요청을 누구의 기록으로 연결하는지, API가 프로그램끼리 정보를 주고받는 약속인지, 서버와 저장소가 어떤 일을 하는지 직접 확인합니다.

가상의 행사 조건과 장소 자료로 API 요청과 응답, 참가별 기록을 익히는 연습입니다.

별명으로 참가하고, 에이전트가 서버 자료 읽기

01 실습 화면을 열고 별명을 입력합니다

서버가 참가 번호와 개인 입장 키를 발급합니다. 같은 별명을 써도 참가 번호와 기록은 따로 만들어집니다.

02 에이전트 요청문을 복사합니다

참가자 본인의 키만 사용하고, 다른 사람의 기록이나 키를 찾지 말라고 안내합니다.

03 에이전트가 API 안내를 읽습니다

`/api/docs`에서 서버와 대화하는 주소와 방법을 확인합니다.

04 행사 조건과 장소 목록을 비교합니다

`/api/challenge`와 `/api/venues` 자료를 읽고 조건을 모두 만족하는 장소를 찾습니다.

05 선택을 제출하고 기록을 확인합니다

`/api/submit`으로 장소 ID를 제출합니다. 참가자마다 최대 10번까지 시도할 수 있습니다.

언제든 열려 있는 연습

강사가 경기를 시작하거나 종료하지 않습니다. 제한시간도 없습니다. 서버 한 대를 모두 함께 쓰며, 필요할 때 다시 접속해 계속 연습할 수 있습니다.

정답을 찍기보다 자료를 확인하고 근거를 설명하기

화면의 요청문을 복사해 사용하세요. 아래는 에이전트가 맡을 일의 예입니다.

너는 공유 서버 탐정 미션을 돕는 에이전트야.

내가 제공한 참가자 키만 사용해. 다른 참가자의 정보나 기록을 찾거나 사용하지 마.

1. /api/docs를 읽고 API 사용법을 확인해.
2. /api/challenge에서 행사 조건을 읽어.
3. /api/venues에서 장소 목록을 읽어.
4. 인원, 예산, 실내 여부, 계단 없는 접근, 종료 시각을 하나씩 비교해.
5. 조건을 모두 만족하는 장소가 있으면 그 이유를 간단히 설명하고 /api/submit에 장소 ID를 제출해.
6. 확실하지 않은 조건은 추측하지 말고 어떤 자료가 부족한지 말해.

내가 직접 제공한 입장 정보와 실습 API만 사용해. 서버 파일이나 다른 참가자의 기록을 조사하지 마.

학생은 자기 입장 정보로 접속하고, 에이전트는 공개된 문서·조건·장소 목록을 읽습니다. 참가 키는 다른 사람과 공유하지 마세요.

한 걸음씩 읽고 근거를 확인합니다

단계	참가자와 에이전트	서버
입장	별명으로 참가	고유 참가 번호와 입장 정보 발급
자료 읽기	API 안내, 행사 조건, 장소 목록 확인	같은 공개 자료를 모든 참가자에게 제공
판단	조건마다 장소를 비교하고 근거 설명	정답을 직접 알려 주지 않음
제출	선택한 장소 ID 제출	내 참가 기록에 결과와 시도 횟수 저장

서두를 필요 없습니다

수업에서는 15분 개념과 15분 실습으로 시작해도 좋고, 실습은 더 길게 이어갈 수 있습니다. 웹 실습은 수업 뒤에도 계속 열려 있고 시간 제한이 없습니다.

정답을 빠르게 맞히는 것보다, 어떤 조건을 읽고 왜 그 장소를 골랐는지 설명할 수 있는지가 중요합니다.

방금 해 본 일에서 서버와 API를 찾아봅시다

실습에서 본 것

연결되는 개념

별명으로 참가했더니 참가 번호가 생김

로그인·식별: 요청과 개인 기록을 구분

`/api/docs`에서 사용법을 읽음

API: 프로그램 사이의 요청·응답 약속

서버가 조건과 장소를 돌려줌

서버: 요청을 받아 자료를 보내고 처리

다시 접속해도 제출 기록이 남음

저장소: 기록을 보관해 나중에 다시 사용

다른 사람도 같은 주소에 접속함

배포·네트워크: 여러 기기에서 쓰도록 공개

서버는 여러 사람의 요청을 받고, 각자 기록을 따로 보관하는 공동 작업대와 같습니다.

작은 변경으로 서버의 일을 살펴봅니다

활동 A · 새 장소 추가

장소 하나를 목록에 더하고, 정원·가격·접근성 정보를 서버가 돌려주는지 살펴봅니다.

활동 B · 조건 하나 바꾸기

인원이나 예산을 바꿔 정답 장소가 달라지는지 확인합니다. 판단 기준은 어디에 들어 있나요?

활동 C · 응답을 기록으로

새 참가자가 입장하고 제출할 때 저장소의 참가별 기록이 어떻게 늘어나는지 확인합니다.

에이전트는 판단을 돕고, 서버는 규칙과 기록을 관리합니다.

실제 서비스를 만들 때도 각 역할을 나누면 동작을 이해하고 고치기 쉽습니다.

화면의 안내를 보고 한 가지씩 확인합니다

참가가 되지 않아요

별명을 1~20자로 입력하고 다시 시도합니다. 이미 참가한 적이 있으면 기존 참가 정보를 사용합니다.

401 · 입장 정보 오류

에이전트 요청문에 내 참가 키가 포함되어 있는지 확인합니다. 키를 다른 사람과 공유하지 않습니다.

요청 형식 오류

API 안내에서 주소·요청 방식·필드 이름을 다시 읽습니다. JSON 요청에는 장소 ID만 보냅니다.

서버에 연결할 수 없어요

안내된 사이트 주소가 맞는지 확인하고, 잠시 뒤 다시 시도합니다. 계속되면 강사에게 오류 문구를 전달합니다.

실습은 항상 열려 있습니다. 강사의 시작 신호를 기다릴 필요가 없습니다.

요청이 어디를 거쳐 결과가 되었을까요?

내가 확인할 것	나의 메모
내 참가 번호	별명과 참가 기록을 연결하는 번호
에이전트가 읽은 API 주소	/api/docs · /api/challenge · /api/venues
중요하게 비교한 조건	인원 · 예산 · 실내 · 계단 없는 접근 · 종료 시각
제출한 장소와 근거	왜 모든 조건에 맞는지 설명해 보기
요청과 응답의 흐름	내 에이전트 → 서버 → 저장 기록 → 결과 응답

같은 별명이 있어도 참가 번호가 다르면 기록을 구분할 수 있습니다. 실명이나 실제 비밀번호는 실습에 필요하지 않습니다.

오늘 다룬 범위와 참고 자료

1. [MDN · HTTP 개요](#) · 브라우저와 서버가 요청과 응답을 주고받는 방식
2. [MDN · 웹 서버란?](#) · 서버가 요청을 받아 자료를 보내는 기본 역할
3. [MDN · API](#) · 프로그램끼리 정해진 약속으로 정보를 교환
4. [Git · 버전 관리](#) · 소스코드 변경 기록을 보관하는 방법

실습은 가상의 행사 조건과 장소 정보를 사용합니다. 실제 개인정보를 입력하지 마세요. 한 서버를 공유하지만 참가자별 입장 정보와 제출 기록은 분리됩니다.

SMALL GROUP · WEEK 03 · OAuth 로그인, 서버, 배포

비전공자를 위한 생활 속 비유 · 개념 15분 + 실습 25분 · 예시 주소는 수업 환경에 맞게 변경