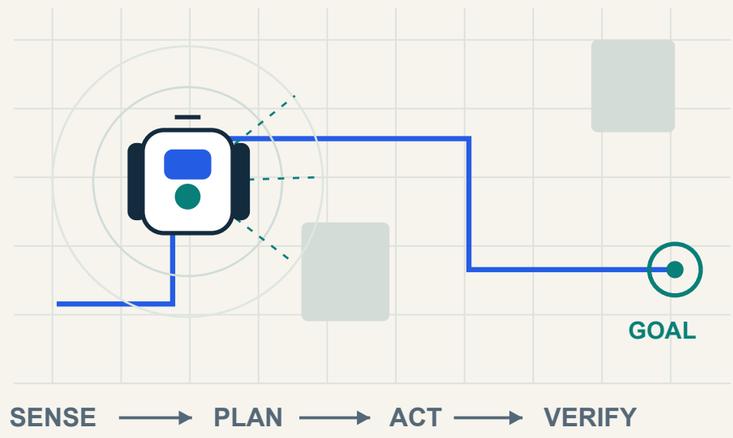
PHYSICAL AI

대학 기초에서 기업형 프로젝트까지

보유 장비로 시작하는 로봇 AI 학습·실험·검증 지도



12 개월 **8-10** 시간 / 주 **384-480** 총 학습시간

48주 학습 + 4주 복습·휴식 / 목표: 대학 핵심 기초 + 기업형 프로젝트 1개

01 보유 장비를 이렇게 연결합니다

추가 PC 구매 없이 시작

DEVELOP

RTX Spark

128GB · Windows

원격 개발 · 코드 편집

지원되는 로컬 AI 추론

REMOTE

SIMULATE / TRAIN / EVALUATE

64코어 워크스테이션

RTX PRO 6000 Blackwell × 2

GPU 0 시뮬레이션 · 통합 · 평가

GPU 1 정책 학습 · 독립 실험

96 GB / GPU

96 GB / GPU

메모리 자동 통합 아님

워크스테이션 권장: Ubuntu 24.04 · ROS 2 Jazzy · Isaac Sim 6.2 · Isaac Lab 3.0

워크스테이션 OS·RAM·SSD 미확인

02 작게 체험하고, 필요한 역할로 확장합니다

메모리 단위: GB

00	01	02	03	04	05	06	07
브라우저	**Pico 2**	**Pi 5**	**CPU PC**	**Jetson**	**RTX 4080**	**RTX 5080**	**PRO 6000 ×2**
기존 PC·모바일	마이크로컨트롤러	4GB / 8GB	T480급 · 16GB	Orin Nano Super	RAM32 / VRAM16	RAM64 / VRAM16	64코어 · 보유
MuJoCo 물리 체험	센서·모터 제어	Linux·센서 노드	MuJoCo·ROS 2	8GB · AI 추론	4코어 · SSD 50GB	8코어 · SSD 500GB	병렬 학습·평가
추가 구매 없음	개발 PC 별도	실물 제어용	학습 사양 제안	Sim 호스트 아님	Sim 공식 최소	Sim 공식 Good	현재 개발환경

역할에 따른 배열이며 절대 성능순은 아닙니다. CPU PC는 학습 제안, RTX 4080·5080은 공식 요구사양 기준입니다.

03 12개월, 하나의 로봇 시스템을 끝까지

기본 과제: 실내 이동로봇의 목표 도달·장애물 회피

Q1 기초를 코드로

01 Python · Linux · Git
README로 다시 실행

02 수학 · 좌표변환
운동 모델과 오차 그래프

03 운동학 · PID
초기조건 3개 비교

OUTPUT
운동 모델 + 제어 실험

Q2 로봇 시스템 연결

04 ROS 2 · TF · URDF
센서와 제어 노드 연결

05 잡음 · 상태추정
실패 사례 10개 분류

06 경로계획 · 주행
성공·충돌·시간 집계

OUTPUT
측정 가능한 기준 시스템

Q3 학습하고 비교

07 PyTorch · RL 기초
Cartpole 학습 재현

08 정책 학습 · PPO
동일 과제에서 제어기 비교

09 환경 변화 · 반복 실험
seed별 평균·산포

OUTPUT
동일 조건의 정책 성능 비교

Q4 재현하고 검증

10 환경 고정 · 자동 검사
새 환경에서 실험 재현

11 추론 지연 · 장애 대응
제어 주기·메모리 측정

12 독립 평가 · 기술보고서
코드·데모·결과 연결

OUTPUT
실행 가능한 포트폴리오

미적분 기초가 부족하면 4-8주 이상 보충합니다. 이미 익숙한 단계는 산출물로 진단한 뒤 통과할 수 있습니다.

04 무료 교육을 실습에 연결합니다

강좌명 선택 시 공식 자료로 이동

- **CS50P** — Python · 테스트
- **MIT 수학** — 선형대수 · 미적분 · 미분방정식
- **Modern Robotics** — 운동학 · 동역학 · 제어
- **MuJoCo** — 물리 · Python 제어 실습
- **ROS 2 Jazzy** — 노드 · 메시지 · 좌표계
- **Nav2** — 위치추정 · 경로계획
- **NVIDIA Isaac Sim** — 장면 · 센서 · 통합 검증
- **PyTorch Basics** — 모델 학습 · 저장 · 추론
- **NVIDIA Isaac Lab** — 강화학습 · 정책 · 실험 관리
- **Missing Semester** — Git · 디버깅 · 개발 도구
- **MIT Underactuated** — LQR · 최적제어 심화
- **KOCW 컴퓨터비전** — 한국어 · OpenCV 보완

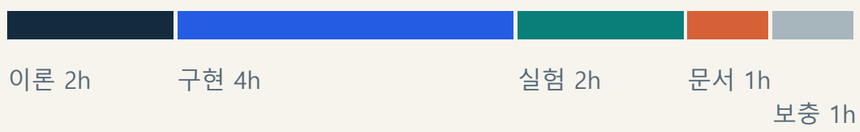
공개 강의·문서·코드 기준. 인증서·클라우드·하드웨어 비용은 별도이며, 전체 강좌 완주가 아닌 필요한 단원 선별 학습입니다.

05 매주 만들고, 수치로 검증합니다

요구사항 → 기준 제어 → AI 학습 → 동일 비교 → 장애 시험 → 재현·인수

성공률 · 충돌률 · 소요시간 · 추론 지연 · 메모리 · 실패 원인

주 10시간 예시



검증 예시: 5개 seed × 20회 평가. 교육용 설계이며 산업 인증 기준이 아닙니다.

실물 로봇 없이도 완성 가능. 실기체를 시험하지 않았다면 sim-to-real 검증 완료로 표현하지 않습니다.

하드웨어 요구사양 버전 호환표

기준일 2026.11.12