

Computer Hardware & Industry Perspective

1주차 - 오리엔테이션 + 전체 분해 실습

- 이론:
 - 하드웨어와 소프트웨어의 관계, 시스템 전체를 보는 눈의 필요성
 - 컴퓨터의 주요 부품(메인보드, CPU, GPU, RAM, SSD, 파워, 팬 등) 개괄
 - 본체 내부의 구조와 데이터 흐름 개요 (전원 → 연산 → 저장 → 출력)
- 실습:
 - 데스크탑 컴퓨터 본체를 실제로 열어 내부 구조 확인
 - 부품별 명칭과 역할 짚어가며 관찰: 메인보드, 전원부, 슬롯, 케이블 등
- 토론:
 - “컴퓨터를 이해한다는 건 무엇일까?”
 - 기술을 다루는 사람에게 기계의 실체란 어떤 의미인가?
 - 나에게 컴퓨터는 지금까지 어떤 존재였는가, 앞으로는 어떤 존재여야 할까?
- 1주차 선각 주제 :
 - 조립 시장의 종말? 완제품 시장이 장악하는 이유
 - 모듈형 컴퓨터는 왜 실패하고, 왜 다시 시도될까?
 - Framework Laptop, Raspberry Pi 5 - 탈인텔 시대의 DIY 철학

시작하기에 앞서

**왜
이 동아리에
오셨나요?**

**이 동아리에서
무엇을
얻고 싶나요?**

**지금까지 컴퓨터를
‘뜯어본’ 적이 있나요?**

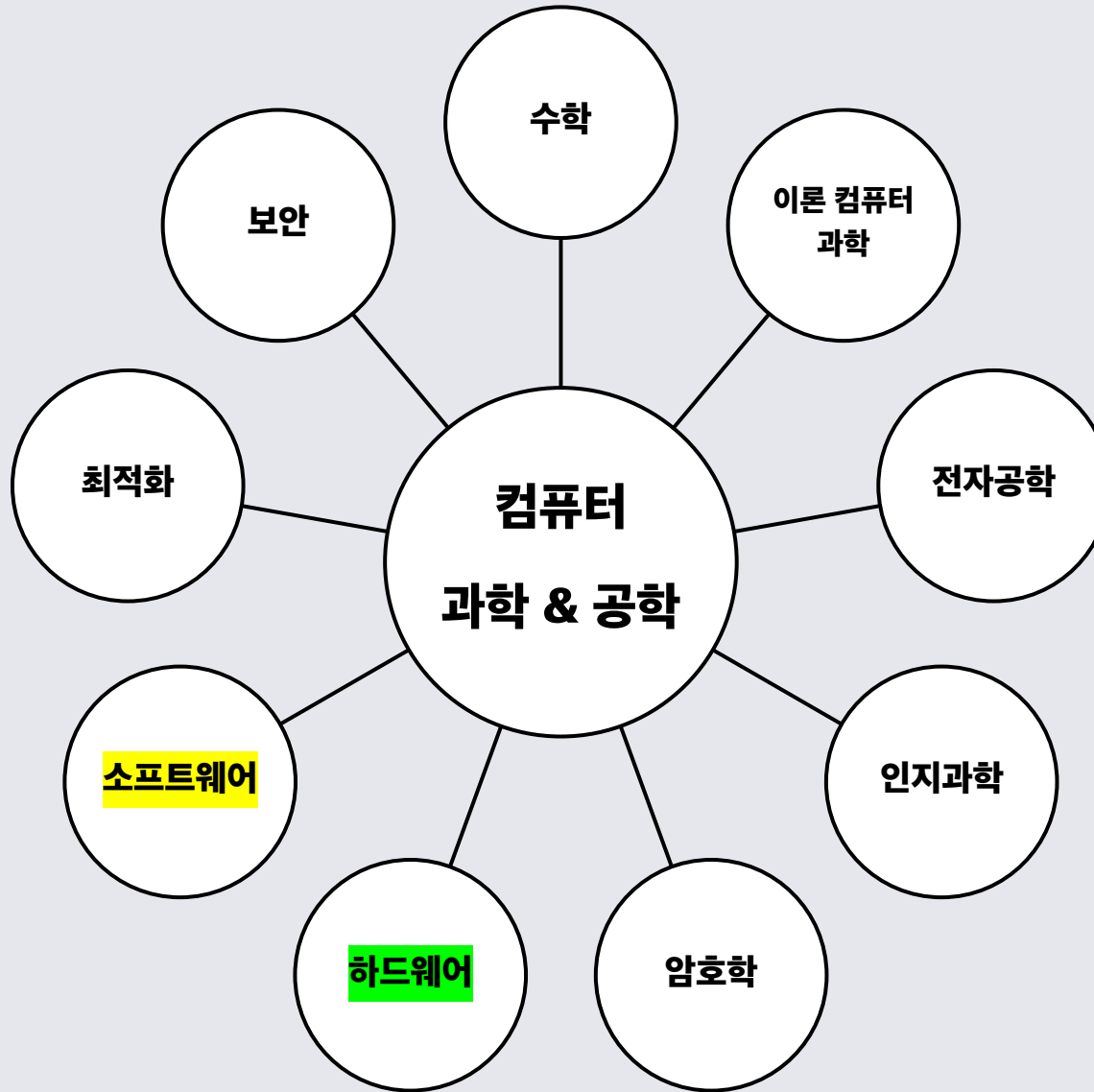
언제 모일까?

얼마나 모일까?

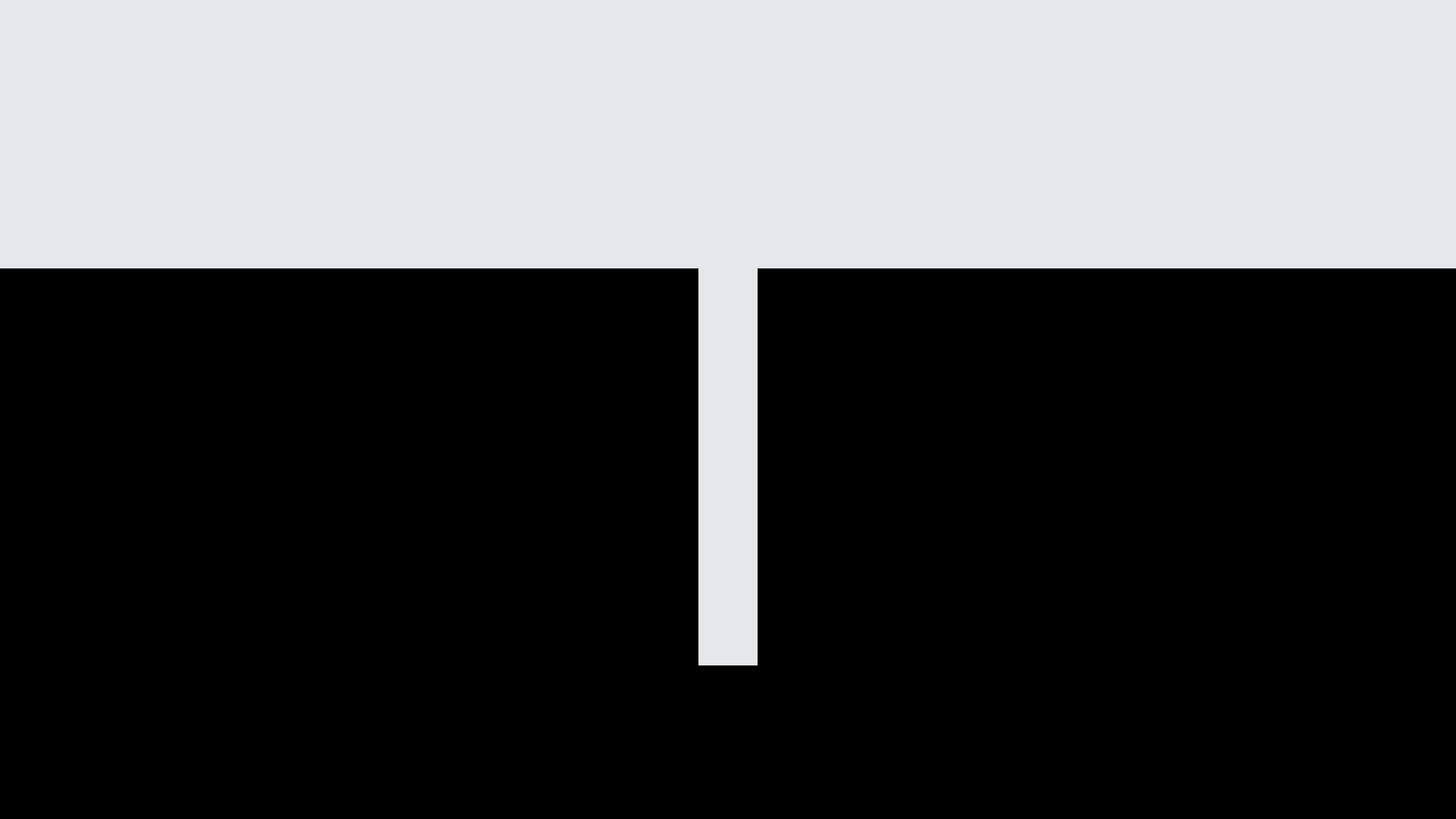
이론

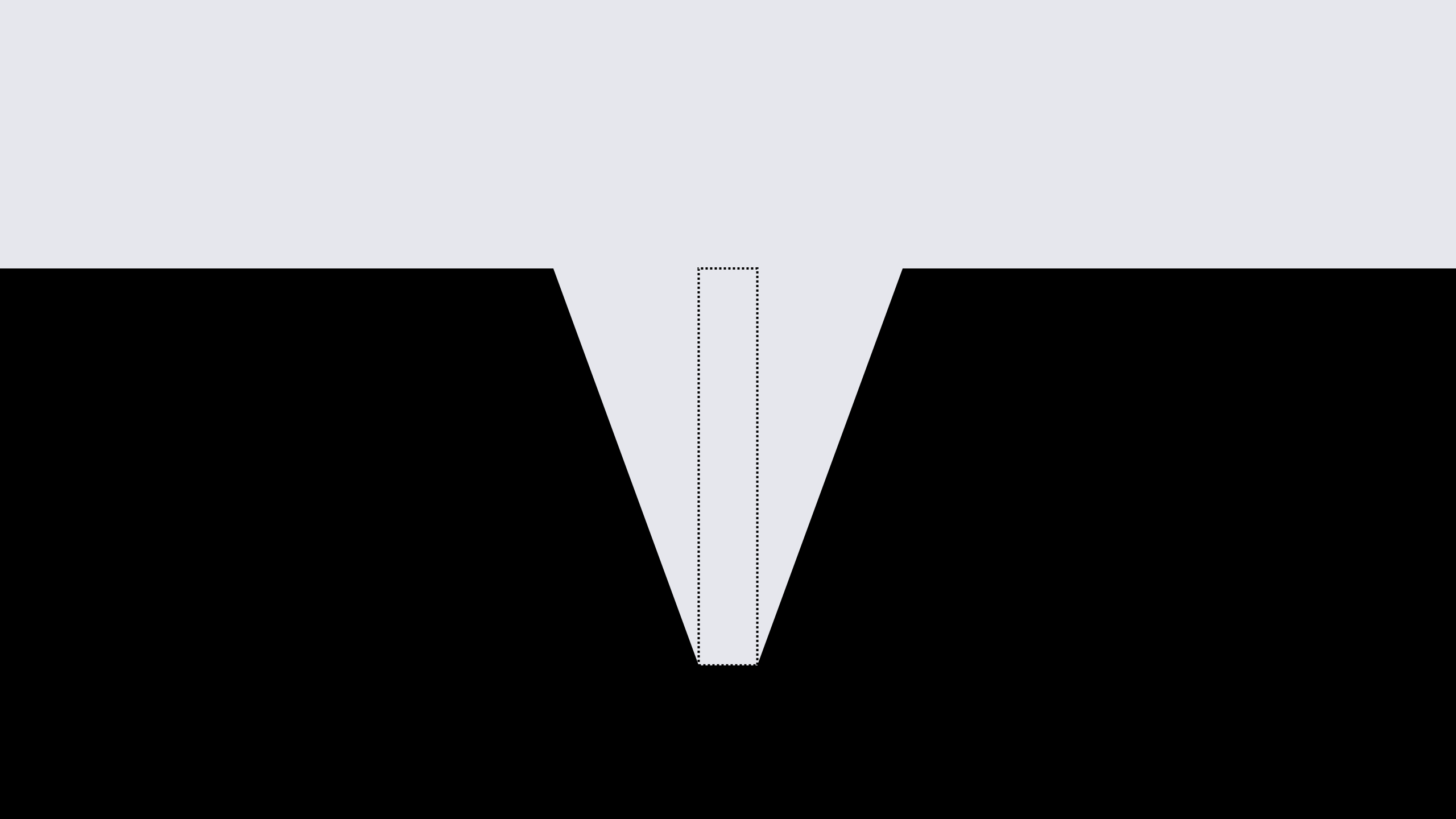
**하드웨어와 소프트웨어의 관계,
그리고 시스템 전체를 보는 눈의 필요성**

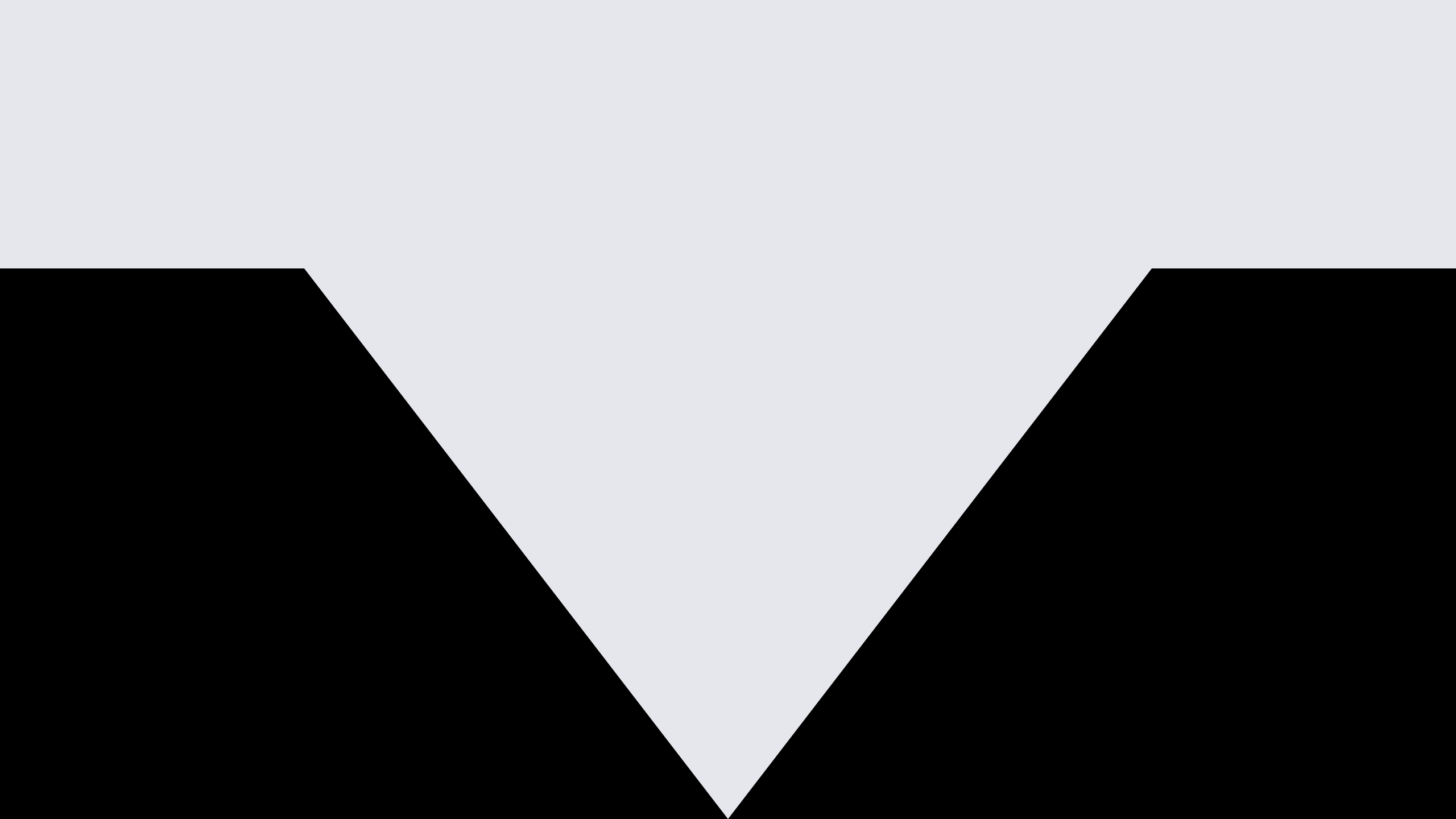
도메인 넓히기



컴퓨터 과학 & 공학







“TSMC는 왜 삼성보다 높은 점유율을 유지할까?”

“콘솔 vs PC, 칩 설계 철학은 어떻게 다른가?”

“하드디스크는 진짜로 완전히 사라질까?”

“오픈소스 하드웨어(RISC-V)는 성공할 수 있을까?”

“뉴로모픽 칩이 실제로 '뇌'와 닮았다는 말은 무슨 의미인가?”

“왜 애플은 M시리즈를 직접 설계하기 시작했는가?”

**CHIP이 단순히 하드웨어 공부를 넘어서
기술 감각을 가진 개발자를 만드는 공간이 되도록**

소프트웨어는 혼자서 존재할 수 없다.
실행되는 '어딘가'가 필요하다. 그게 바로 하드웨어다.

우리가 짜는 코드는 결국 전기 신호를 어떻게 흐르게 할지에 대한 지시문이다.

하드웨어는 전기, 회로, 트랜지스터라는 물리적인 실체이고
소프트웨어는 그 위에서 돌아가는 논리적 사고의 결정체다.

우리는 평소엔 이 둘을 분리해서 보지만, 시스템은 이 둘이 함께 작동할 때 완성된다.

전체를 본다는 건... “내 코드가 지금 어디쯤을 지나고 있을까?”를 상상할 수 있다는 것.

우리는 보통 “코드만” 본다
소프트웨어 개발자는 코드 → 결과의 흐름에 익숙

그런데 이 코드는 어디에서, 어떻게, 무엇 위에서 실행될까?

`int x = 10;`
이건 변수를 만든 게 아니라,
→ 레지스터나 메모리 어딘가에 전기를 흐르게 한 것

소프트웨어와 하드웨어는 어떻게 연결되어 있을까?

- **하드웨어: 물리적인 연산과 저장이 일어나는 땅**
- **소프트웨어: 그 땅 위에서 살아가는 로직과 의미**

둘 사이를 잇는 것이 운영체제, 어셈블리, 명령어 집합(IS)

**코드는 "말 걸기 위한 언어"이고,
그 대상은 결국 하드웨어다.**

시스템 전체를 본다는 건 무슨 뜻일까?

- 단편적 시야:
 - “코드가 느리다.” →
 - 변수 이름 바꿔볼까?
- 시스템적 시야:
 - “코드가 느리다.” →
 - → 캐시 미스일 수도?
 - → CPU 점유율 높음?
 - → I/O 병목?
 - → 쓰레드 스케줄러가 컨텍스트 스위칭 너무 자주?

문제를 추적하는 눈,
전체 흐름을 연결짓는 시야,
그게 바로 “시스템 전체를 보는 눈”

왜 이게 우리에게 필요할까?

- **우리는 코드만 쓰는 사람이 아니라,**
 - **→ 시스템을 이해하는 사람이 되기 위해 여기에 왔다.**

이해의 깊이 = 선택의 폭

- **→ 어떤 구조를 고를지**
- **→ 어떤 최적화를 택할지**
- **→ 어떤 직무로 진출할지**

컴퓨터의 주요 부품 개괄

- 메인보드: 모든 부품이 연결되는 중앙 통로
- CPU: 모든 계산과 판단을 내리는 두뇌
- GPU: 병렬 작업 특화된 연산 보조장치, AI·그래픽·게임에 강함
- RAM: 작업 중인 데이터를 잠시 담는 작업 책상
- SSD: 오래 보관하는 서랍
- 파워: 부품에 전기를 공급하는 심장과 혈관
- 팬: 열을 식히는 냉각 시스템

관찰 팁:

“이 부품이 없으면 어떤 문제가 생길까?”를 상상하며 각 부품을 뜯어보자

본체 내부의 구조와 데이터 흐름 개요 (전원 → 연산 → 저장 → 출력)

- 파워가 전기를 공급함 (24핀 메인보드, 8핀 CPU 전원 등)
 - CPU가 명령어를 받고 연산을 시작함
 - 필요한 데이터는 RAM에서 가져오고, 결과를 SSD에 저장함
 - 결과는 화면(GPU → 모니터)이나 소리(사운드카드 → 스피커) 등으로 출력됨
-
- 이걸 왜 알아야 할까?
 - 오류가 났을 때 어디부터 점검해야 하는지, 이 흐름을 모르면 막막하다.
 - 성능 병목이 생겼을 때 어디서 생기는지도 모르면 그냥 “느리다”고만 느껴진다.

실습

1단계

외부 관찰 - 전체 구조 파악

“이 선은 어디서 어디로 가고 있을까?”

“이건 데이터를 전송하는 선일까, 전력을 공급하는 선일까?”

2단계

저장장치가 없다면 어떻게 될까

“왜 저장장치에는 전기와 데이터가 따로 들어갈까?”

“이 장치가 데이터를 ‘기억’하는 방식은 뭐가 있을까?”

3단계

RAM이 없다면 어떻게 될까

“이걸 통해 CPU와 어떤 데이터를 얼마나 빨리 주고받을까?”

“RAM이 빠지면 컴퓨터는 어떤 오류를 낼까?”

4단계

GPU가 없다면 어떻게 될까

“왜 GPU는 CPU보다 전기도 더 많이 먹고, 덩치도 클까?”

“이 안에 뭐가 들어 있길래 그렇게 복잡할까?”

5단계

쿨러가 없다면 어떻게 될까

“열은 어디서 나오고, 어디로 사라지는 걸까?”

“써멀그리스는 어떤 역할을 할까?”

6단계

파워서플라이가 없다면 어떻게 될까

“전기는 어떤 경로로 흐르는가?”

“12V, 5V, 3.3V 선들이 왜 구분되어 있는가?”

토론

**단순히 사용법을 안다는 거랑
구조를 안다는 건 뭐가 다를까요?**

기계의 속을 몰라도 괜찮을까요?

**나는 이 기계를 통해
어떤 능력을 얻고,
또 잃고 있나요?**

**오늘, 우리는 컴퓨터를 ‘사용’하지 않고
처음으로 직접 마주했습니다.**

[과제 안내] 1주차 선각 리포트

이번 주 우리가 처음 컴퓨터의 속을 직접 들여다봤다면,
이제는 산업의 흐름 속에서 그 바깥을 바라보는 시간입니다.

다음 주 모임 전까지 아래 중 관심 있는 주제 하나를 선택해
‘선각 리포트’를 A4에 작성해 주세요.

1주차 선각 주제 (택1)

1. “조립 시장의 종말? 완제품 시장이 장악하는 이유”
2. “모듈형 컴퓨터는 왜 실패하고, 왜 다시 시도될까?”
3. “Framework Laptop, Raspberry Pi 5 - 탈인텔 시대의 DIY 철학”

다음 시간엔?

**“정보는 어디에, 어떻게 저장되어야 하는가?”
우리가 데이터를 다루는 방식을,
기계의 입장에서 바라봅니다.**

- 메모리와 저장장치의 세계로 -