

고려대학교 컴퓨터학과 특강

# 대학생 때 알았더라면 좋았을 것들

옥찬호

utilForever@gmail.com

# 발표자 소개

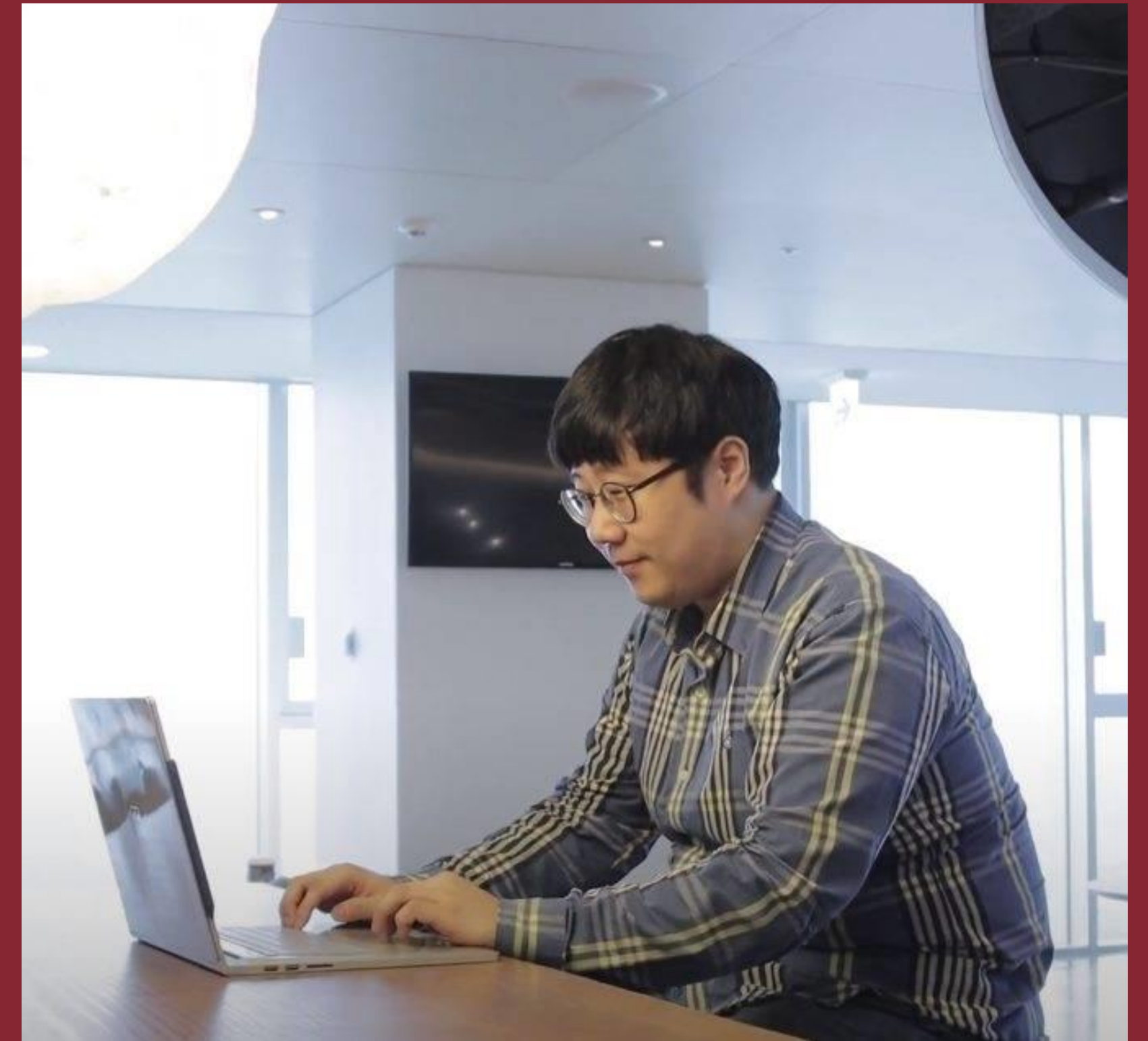
고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 옥찬호 (Chris Ohk)
  - (현) Momenti Engine Engineer
  - (전) Nexon Korea Game Programmer
  - Microsoft Developer Technologies MVP
  - C++ Korea Founder & Administrator
  - Reinforcement Learning KR Administrator
  - IT 전문서 집필 및 번역 다수
    - 게임샐러드로 코드 한 줄 없이 게임 만들기 (2013)
    - 유니티 Shader와 Effect 제작 (2014)
    - 2D 게임 프로그래밍 (2014), 러스트 핵심 노트 (2017)
    - 모던 C++ 입문 (2017), C++ 최적화 (2019)

utilForever@gmail.com



utilForever



1. 주어진 시간을 어떻게 써야 할까
2. 어떤 언어/분야를 공부해야 할까
3. 어떻게 공부해야 할까
4. 대학원을 갈까, 취업을 할까
5. ‘같이’의 ‘가치’
6. 후배의 성장을 도와줘라
7. 코이의 법칙
8. 마치며



# 주어진 시간을 어떻게 써야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



# 주어진 시간을 어떻게 써야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 시간을 얼마나 확보할 수 있는지부터 파악하자
  - 학기 중에는 생각보다 시간이 여유롭지 않다
  - 학교 수업을 들어야 한다 (학생들은 보통 6~7과목을 듣는다)
  - 수업마다 나오는 과제나 프로젝트를 해결해야 한다  
(처음 배우는 내용이라 시행 착오를 겪느라 시간이 많이 걸린다)
  - 동아리 활동을 해야 한다
  - 스터디에 참여해야 한다
  - 과외나 아르바이트를 해야 한다
  - 동기, 선배, 후배, 친구들과 놀아야 한다
  - ...



# 주어진 시간을 어떻게 써야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 중요한 일을 먼저 하자
  - 중요하지만, 급하지는 않은 일 vs 중요하지 않지만, 급한 일

	Urgent		Not Urgent	
	I		II	
	(MANAGE)		(FOCUS)	
Important	<ul style="list-style-type: none"><li>• Crisis</li><li>• Medical emergencies</li><li>• Pressing problems</li><li>• Deadline-driven projects</li><li>• Last-minute preparations for scheduled activities</li></ul>		<ul style="list-style-type: none"><li>• Preparation/planning</li><li>• Prevention</li><li>• Values clarification</li><li>• Exercise</li><li>• Relationship-building</li><li>• True recreation/relaxation</li></ul>	
	Quadrant of Necessity		Quadrant of Quality & Personal Leadership	
Not Important	III		IV	
	(AVOID)		(AVOID)	
	<ul style="list-style-type: none"><li>• Interruptions, some calls</li><li>• Some mail &amp; reports</li><li>• Some meetings</li><li>• Many "pressing" matters</li><li>• Many popular activities</li></ul>		<ul style="list-style-type: none"><li>• Trivia, busywork</li><li>• Junk mail</li><li>• Some phone messages/email</li><li>• Time wasters</li><li>• Escape activities</li><li>• Viewing mindless TV shows</li></ul>	
	Quadrant of Deception		Quadrant of Waste	

# 주어진 시간을 어떻게 써야 할까

---

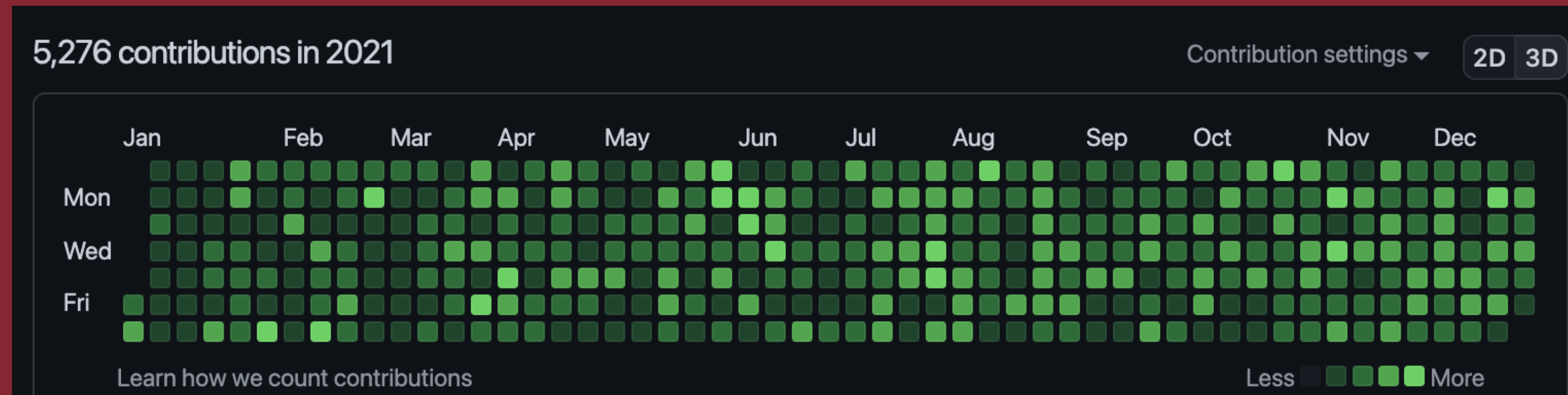
고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 다양한 경험을 해보자
  - 학기 중에는 주말을, 학기가 끝난 뒤에는 방학 기간을 활용
  - 개발과 관련된 행사에 참가해보자
    - 컨퍼런스, 세미나, 밋업, 스터디, 모각코 등 어떤 형태라도 좋다
    - 다양한 개발자들과 이야기를 나누며 새로운 지식들을 배울 수 있다
  - 개발과 관련 없는 행사에도 참가해보자
    - 독서, 산책, 여행, 전시회, 뮤지컬, 콘서트, 액티비티 활동 등 어떤 형태라도 좋다
    - 다른 전공을 공부하고 있는 학생들과 대화를 하며 다양한 아이디어를 얻을 수 있다
    - 사이드 프로젝트의 출발점이 되기도 한다

# 주어진 시간을 어떻게 써야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 좋아하는 게 있다면 몰두해보자
  - 무엇이든 좋다, 공부와 관련 여부는 중요하지 않다
  - 우선 좋아하는 일에 몰두할 수 있는 시간을 확보하자
  - 하루에 많이 하는 건 중요하지 않다, 조금씩이라도 꾸준히 하자





# 어떤 언어/분야를 공부해야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



# 어떤 언어/분야를 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 아직 무엇을 공부해야 될 지 모르겠다면 다양하게 해보자
  - 어렸을 때 재미를 느끼거나 적성에 맞는 분야를 찾기 위해 여러 학원을 다니던 시절...
  - 컴퓨터 공학에서 재미를 느끼거나 적성에 맞는 분야를 찾기 위한 여정도 마찬가지로
  - 요즘은 분야마다 쉽게 따라할 수 있는 튜토리얼과 예제가 있다
  - 간단한 사이드 프로젝트를 해보면서 만들기 위한 과정이 즐거운지 확인해보자
  - 서두르지 않아도 된다, 탐색하는 시간을 충분히 갖자



# 어떤 언어/분야를 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 개발자 로드맵을 참고하자
  - 각 분야마다 단계별로 어떤 지식들을 공부하면 되는지 정리되어 있다
  - 주의 : 로드맵에 있는 모든 지식을 공부해야 된다는 강박 관념을 가지지 말자
- 참고하면 좋은 로드맵 저장소 목록
  - <https://github.com/liuchong/awesome-roadmaps>
  - <https://github.com/kamranahmedse/developer-roadmap>
  - <https://github.com/utilForever/game-developer-roadmap>
  - <https://github.com/AMAI-GmbH/AI-Expert-Roadmap>
  - <https://github.com/floodsung/Deep-Learning-Papers-Reading-Roadmap>
  - <https://github.com/anacoimbrag/android-developer-roadmap>
  - <https://github.com/BohdanOrlov/iOS-Developer-Roadmap>



# 어떤 언어/분야를 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 하나의 프로그래밍 언어를 깊게 공부해보자
  - 어떤 분야를 제대로 공부해보기로 마음먹었다면,  
해당 분야에서 주로 사용하는 프로그래밍 언어를 깊게 공부하자
- 우리가 프로그램이나 라이브러리, 딥러닝 모델을 만들 때
  - 정상적으로 빌드가 되어 문제 없이 동작하게 만든다 (Make It Work)
  - 리팩토링을 통해 좀 더 나은 구조로 개선한다 (Make It Better)
  - 최적화를 통해 성능을 개선한다 (Make It Fast)
- 프로그래밍 언어를 잘 알고 있으면 위 과정을 좀 더 빠르게 진행할 수 있다
- 또한 다른 프로그래밍 언어를 배울 때 빠른 속도로 익힐 수 있다

# 어떻게 공부해야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



# 메타인지

## ‘자신의 생각에 대해 판단하는 능력



# 어떻게 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



知之爲知之 不知爲不知 是知也 - 孔子



# 어떻게 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



“아는 것을 안다고 하고 모르는 것을 모른다고 하는 것, 그것이 곧 얕이다.”

# 어떻게 공부해야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 무엇을 아는지 모르는지 어떻게 알 수 있을까?  
→ 내가 알고 있다고 생각하는 지식을 남에게 설명해본다
- 자기 교수 학습법 (Self-Instruction)
  - 배울 때는 깊이 있게 사고하기보다 기억하려는 쪽에 집중한다  
(수동적으로 지식을 받아들이기 때문 → **암기**)
  - 가르칠 때는 100% 이해하지 못하면 설명할 수가 없다  
(자신이 아는 정보를 다른 사람에게 전달하는 게 목적이기 때문 → **이해 + 암기**)
  - 이 과정에서 부족함을 느끼면 **자각(Perception)**이 일어난다



# 어떻게 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 공부할 때는 흐름을 따라가며 온전히 나의 지식으로 만든다
- 예를 들어, 자료구조 힙 (Heap)을 공부하고 싶다고 하자
  - 힙이 무엇인지 정의하자
  - 힙에서 새로운 값을 어떻게 삽입하는가
  - 힙에서 기존 값을 어떻게 삭제하는가
  - 삽입/삭제 동작의 시간 복잡도는 어떻게 되는가
  - 어떤 데이터가 들어올 때 Best-case인가, 그리고 Worst-case인가  
또한 그 때 시간 복잡도는 어떻게 되는가
  - 힙을 구현하는 방식에는 어떻게 있는가



# 어떻게 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 공부할 때는 흐름을 따라가며 온전히 나의 지식으로 만든다
- 예를 들어, 자료구조 힙 (Heap)을 공부하고 싶다고 하자
  - 힙이 무엇인지 정의하자
    - Complete Binary Tree이고, Heap Property를 만족하는 트리 기반 자료 구조 (Heap Property : 루트 노드가 최대/최소 값이고 Left/Right Sub-Tree가 Min/Max 힙이다.)
  - Complete Binary Tree가 무엇인지 정의하자
    - Tree가 무엇인지 정의하자
    - Graph가 무엇인지 정의하자
    - ...

# 어떻게 공부해야 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 점이 모여 선이 된다
  - 학교에서 이론 지식들을 공부하다 보면 생기는 궁금함  
“이렇게 배워서 나중에 개발하는데 도움이 될까?”
  - 하지만 여러분들은 개발할 때 이미 배운 이론 지식들을 활용하고 있다  
아직 이 사실을 인지하고 있지 못할 뿐이다
  - 우리가 프로그래밍 언어를 배워서 프로그램이나 라이브러리를 만들 때,  
그리고 컴퓨터 공학 이론을 배울 때는 각각 점으로 존재한다  
→ 이런 점들이 모여 선을 이루게 된다

# 어떻게 공부해야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 점이 모여 선이 된다

```
int values[100][20];
```

A

```
int sum = 0;
for (int y = 0; y < 100; ++y) {
    for (int x = 0; x < 20; ++x) {
        sum += values[y][x];
    }
}
```

B

```
int sum = 0;
for (int x = 0; x < 20; ++x) {
    for (int y = 0; y < 100; ++y) {
        sum += values[y][x];
    }
}
```



# 어떻게 공부해야 할까

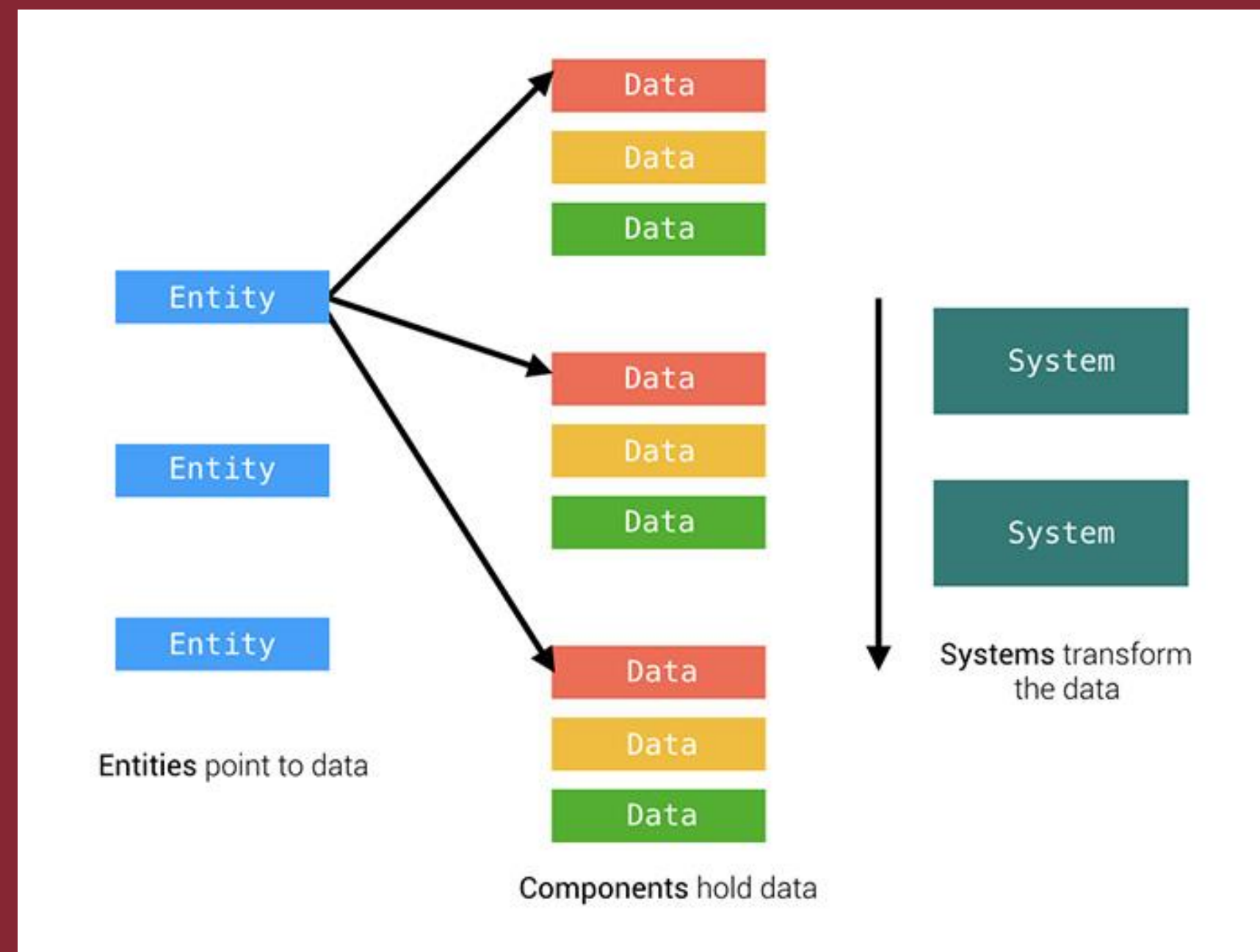
고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 점이 모여 선이 된다
  - 컴퓨터 구조 시간 때 배웠던 “지역성 (Locality)” 개념을 떠올려보자
  - 캐시가 효율적으로 동작하려면, 캐시에 저장할 데이터가 지역성을 가져야 한다
  - 지역성이란 데이터 접근이 시간적, 혹은 공간적으로 가깝게 일어나는 것을 의미한다
- 공간적 지역성
  - 특정 데이터와 가까운 주소가 순서대로 접근되었을 경우를 공간적 지역성이라고 한다
  - CPU/디스크 캐시의 경우 한 메모리 주소에 접근할 때 그 주소뿐 아니라 해당 블록을 전부 캐시에 가져오게 된다
  - 이 때 메모리 주소를 오름차순이나 내림차순으로 접근한다면, 캐시에 이미 저장된 같은 블록의 데이터를 접근하게 되므로 캐시의 효율성이 크게 향상된다

# 어떻게 공부해야 할까

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 점이 모여 선이 된다
  - 객체 지향 프로그래밍 (Object-Oriented Programming)을 대체하는 데이터 지향 디자인 (Data Oriented Design)이 지역성 개념을 활용한 대표적인 예다
  - 데이터 지향 디자인은 객체 지향 프로그래밍에 비해 성능이 뛰어나다



# 대학원을 갈까, 취업을 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들





# 대학원을 갈까, 취업을 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 대학원을 염두에 두고 있다면 성적 관리부터
  - 컴퓨터학과의 인기가 하늘을 찌르고 있다 → 대학원도 마찬가지로
  - 특히 AI 대학원의 경우 경쟁률이 매우 높다
  - 대학원 합격에 있어서 가장 중요한 지표는 **평점**
  - 학교마다 평점 계산 방법이 다르니 참고 (교양 포함 여부)
  - 평점 4.5 만점에 4.0 이상의 평점을 유지할 수 있도록 관리하기



# 대학원을 갈까, 취업을 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 대학원에서는 무엇을 하는가
  - 대학원 수업을 듣는다 (학부보다 지식의 깊이가 깊다)
  - 연구실에서 지도 교수님과 함께 논문 연구를 한다
  - 학부 또는 대학원 수업의 TA가 된다 (교수님이 강의하는 수업 또는 다른 수업)
  - 매달 연구실에서 연구비가 지급된다
  - 놀랍게도 방학은 없다 (방학에도 연구는 지속되어야 한다)
  - 학회 논문 제출 마감일이 다가오면 사람이 변한다 (?)

# 대학원을 갈까, 취업을 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- **체험판으로 맛부터 보자**
  - 컴퓨터 프로그램은 구입하기 전에 맛보기 위한 체험판이 있다
  - 둘 중 고민하고 있다면 현실에서도 체험할 수 있는 방법이 있다
    - 대학원 체험판 = 학부 연구생
    - 취업 체험판 = 인턴
  - 학부 연구생은 학기 중에 할 수 있다, 물론 방학 기간에도 할 수 있다
  - 인턴은 짧게는 방학 기간에, 길게는 1학기 동안 진행하기도 한다
  - 학부 연구생과 인턴 경험을 통해 어떤 길이 나한테 맞는지 직접 확인해보기 바란다

# 대학원을 갈까, 취업을 할까

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 간단하게 정리해 보자면
  - 개발을 하고 싶다 → 취업
  - 돈을 벌어서 여유로운 생활을 하고 싶다 → 취업
  - 연구를 하고 싶다 → 대학원 진학
  - 깊이 있는 공부를 하고 싶다 → 대학원 진학



# ‘같이’의 ‘가치’

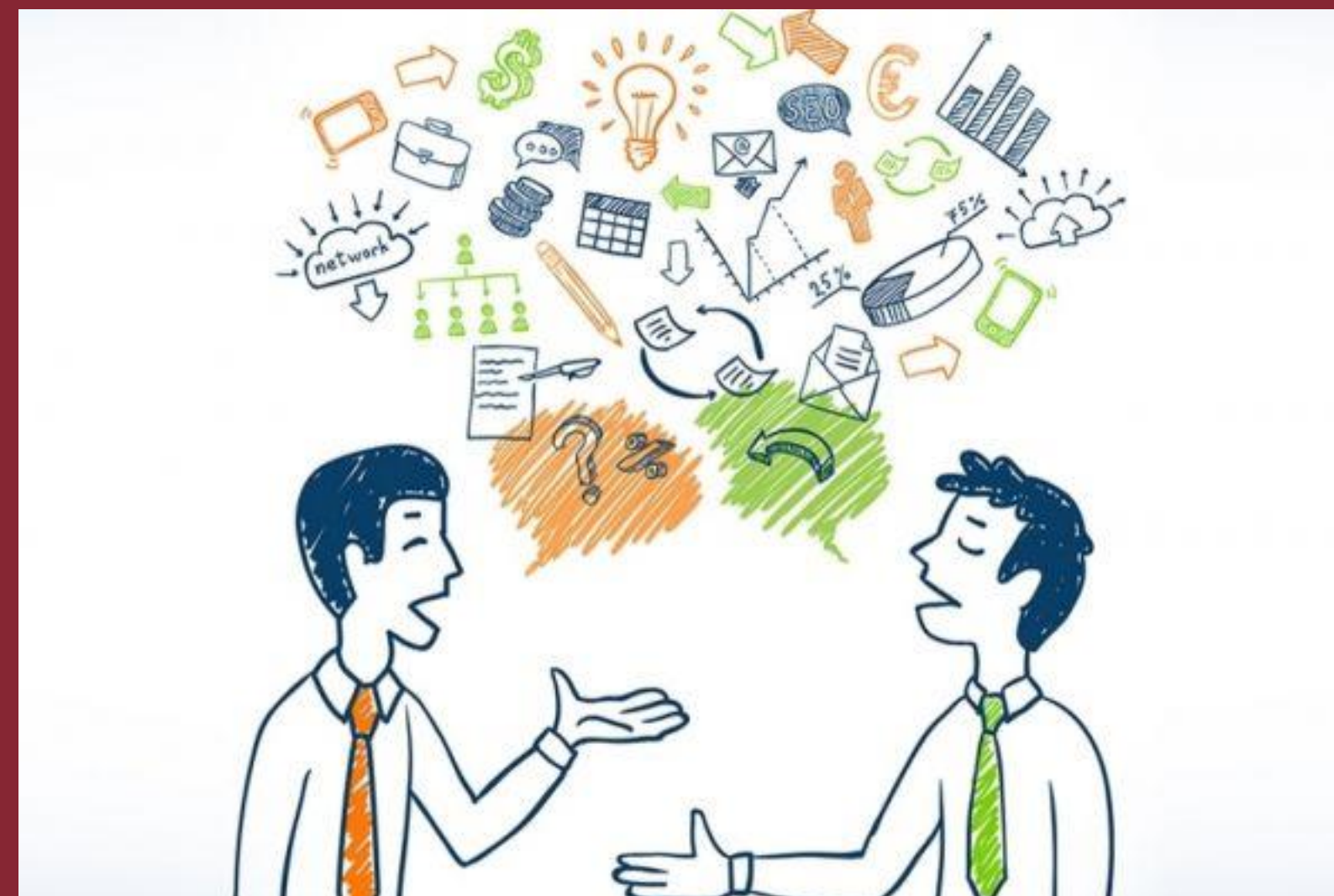
고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들





고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 



# ‘같이’의 ‘가치’

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 원활한 커뮤니케이션을 위해서는
  - 말하기 전에 상대방의 이야기를 끝까지 듣는다
  - 모르는 게 있을 때는 물어본다 (바보같은 질문은 없다)
  - 상대방은 나와 ‘틀린 게 아닌 다른’ 생각을 가질 수 있다
  - ‘공감’하는 태도를 보인다
  - 피드백을 할 때는 서로 존중하는 자세를 갖는다
  - 상대방의 자존감을 높일 수 있도록 격려한다
  - 알아야 할 내용을 미리 정리해 ‘문서’로 전달하면 많은 도움이 된다



# ‘같이’의 ‘가치’

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 원활한 개발을 위해서는
  - SNS나 메일로 코드를 전달하지 말자  
Git을 활용해 GitHub 또는 GitLab으로 관리하자
  - 회의한 내용을 반드시 기록하자
  - CI/CD를 통해 코드가 항상 정상적으로 컴파일 및 빌드가 되게 하자
  - 테스트 코드를 통해 작성한 코드가 정상 동작하는지 확인하자
  - PR을 통해 서로 코드 리뷰를 하는 시간을 갖자

# ‘같이’의 ‘가치’

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 같이 개발할 사람을 찾아보자
  - 가까운 동기, 선배, 후배, 친구 중에서 같은 분야에 흥미를 갖는 사람이 있는가
  - 세상은 넓다, 없다면 좀 더 반경을 넓혀보자
    - 개발자 컨퍼런스/세미나/밋업
    - 연합 개발 동아리
    - 온라인 커뮤니티
    - SNS
- 실제 사례
  - RosettaStone의 경우 학술 세미나와 SNS를 통해 알게된 동생들과 함께 개발
  - CubbyFlow의 경우 SNS를 통해 알게된 동생들과 함께 개발



# 후배의 성장을 도와줘라

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들





# 후배의 성장을 도와줘라

---

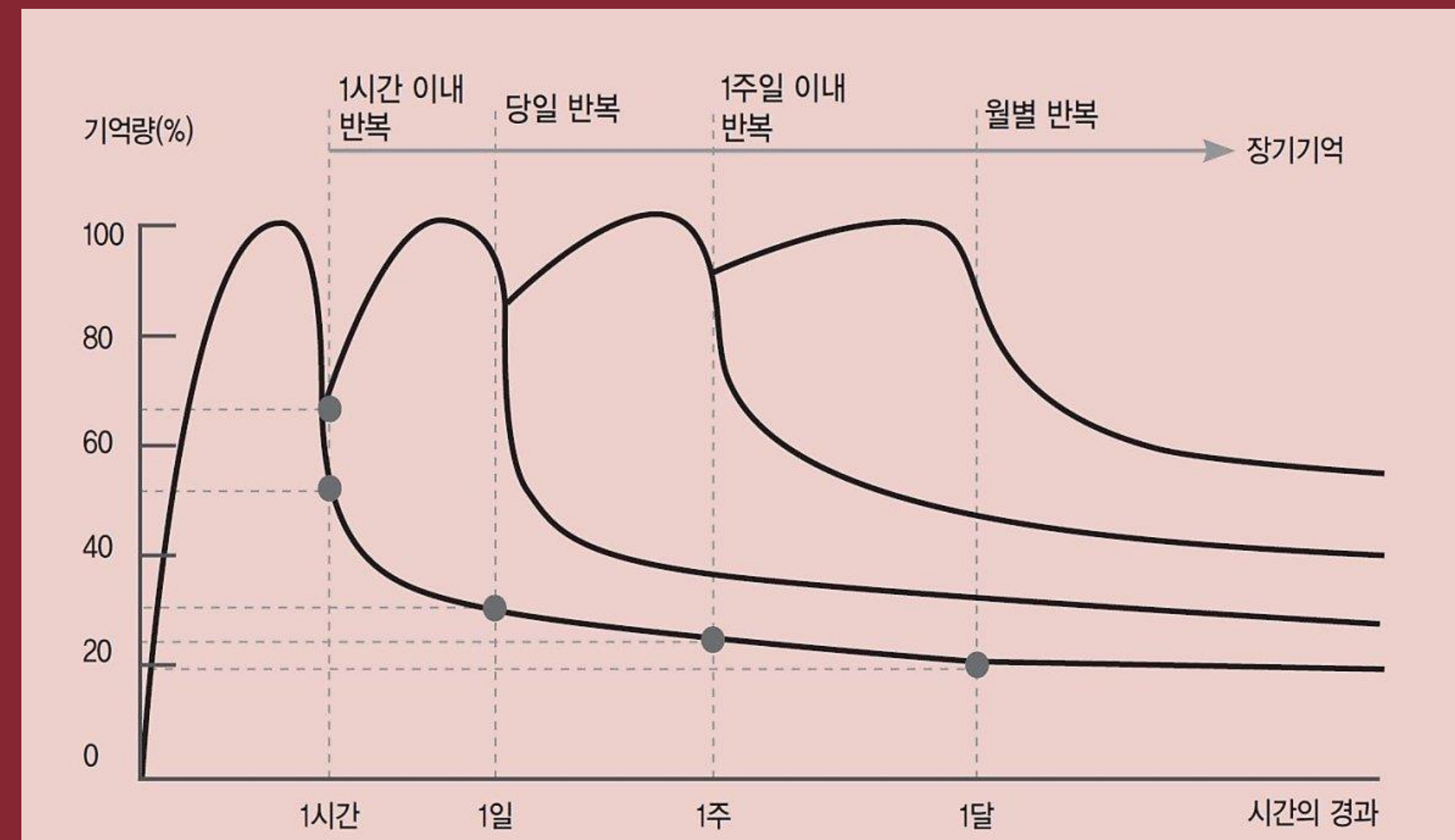
고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- **어미 호랑이와 아기 호랑이**
  - 아기 호랑이들은 어미 호랑이의 사랑과 가르침을 받으며 성장한다
  - 학교라고 다르지 않다, 후배들은 선배들의 사랑과 가르침을 받으며 성장한다
  - 사랑과 가르침을 주는 방법은 특별하지 않다
    - 밥이나 술을 사게 사랑과 가르침일 수도 있겠지만,
    - 후배들이 질문할 때 잘 받아주고 대답해주기
    - 후배들을 위해 스터디를 만들고 가르쳐주기
    - 같이 프로젝트를 하면서 이끌어주기

# 후배의 성장을 도와줘라

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

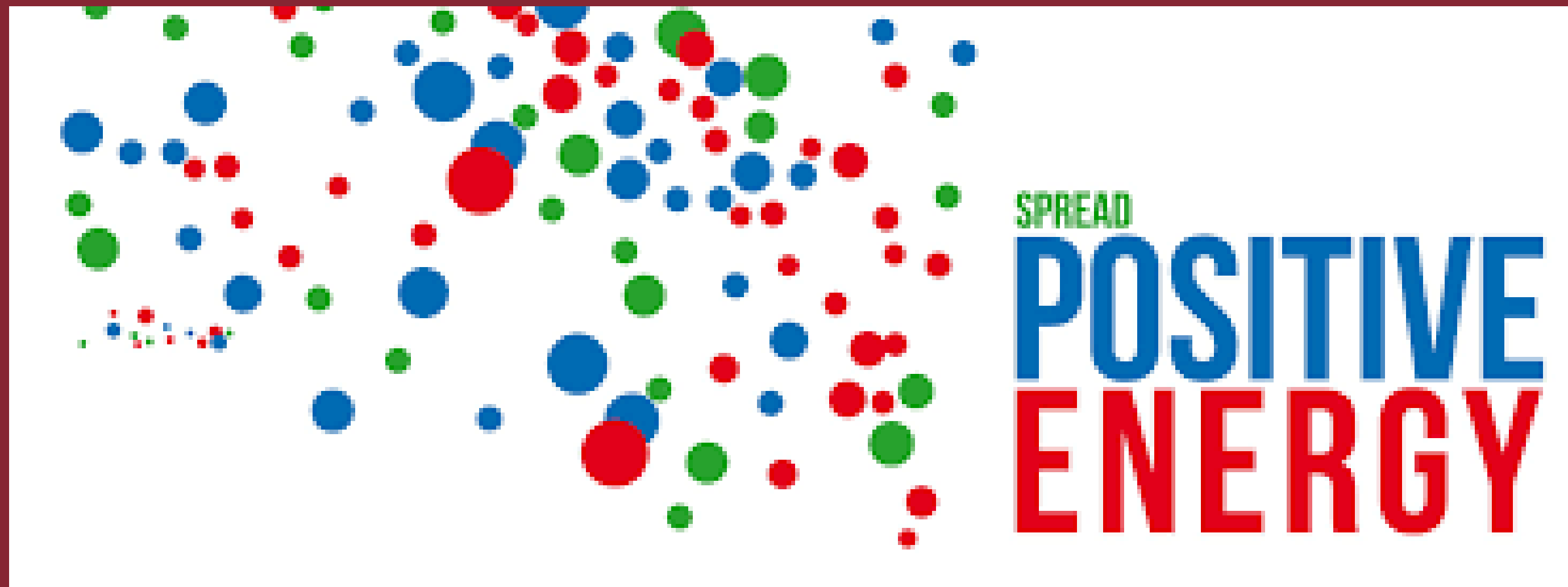
- 후배의 성장은 나의 성장이기도 하다
  - 어떤 지식을 공부하고 나면, 시간이 지남에 따라 기억량이 떨어진다
  - 후배들에게 어떤 지식을 알려줘야 할 때, 내 지식에 구멍이 없는지 확인할 수 있다
  - 구멍났던 지식을 다시 채우면서 나도 다시 성장할 수 있다
  - 그리고 다시 채운 지식은 더 오래 기억할 수 있다



# 후배의 성장을 도와줘라

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들

- 선순환의 고리
  - 선배가 후배들에게 사랑과 가르침을 준다
  - 사랑과 가르침을 받으며 성장한 후배들은 어느덧 선배가 되어 다음 후배들에게 사랑과 가르침을 준다
  - 선순환의 고리는 점점 퍼지면서 많은 사람들에게 긍정적인 에너지를 불어넣는다





# 코이의 법칙

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들





- 코이라는 물고기의 삶은 매우 특이하다
  - 작은 어항에 기르면 5~8cm 밖에 자라지 않는다
  - 커다란 수족관이나 연못에서 기르면 15~25cm까지 자란다
  - 강물에 방류하면 90~120cm까지 성장한다
- 같은 물고기지만 어항에서 기르면 피라미만하게 자라고,  
강물에 놓아두면 대어가 되는 신기한 물고기 → 코이의 법칙



# 코이의 법칙

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



사람은 믿어주는 만큼 자라고, 아껴주는 만큼 여물고, 인정받는 만큼 성장하는 법이야.  
내가 선택한 주변 환경과 생각들에 따라, 엄청난 결과의 차이를 만들 수 있다.

- 〈낭만닥터 김사부 2〉 중에서



# 마치며

---

고려대학교 컴퓨터학과 특강  
대학생 때 알았더라면 좋았을 것들



꿈은 ‘명사’가 아니라 ‘동사’여야 한다

- 인공지능 분야가 아직 블루 오션인지, 아니면 레드 오션인지 궁금합니다  
그리고 인공지능 대학원 진학이나 유학을 위한 조언을 자유롭게 듣고 싶습니다
- 성공적인 개발자가 되기 위해서 학부생으로 할 수 있는 것들이 궁금합니다
- 아직 1학년이라 정확히 무엇을 준비하고 해야할 지 모르겠습니다  
1학년으로 돌아간다면, 어떻게 살아갈 지 알려주셨으면 좋겠습니다 (노는 거 제외!)
- 대학교 1학년이라면 여름방학에 무엇을 하고 싶으신가요

- 대학원은 어떻게 들어가게 되었나요, 그리고 직업 선택에 도움이 됐나요
- 대학원 진학을 목표로 하고 있는데 준비하실 때 대략적으로 어떤 과정을 거쳤는지, 학부 1학년은 앞으로 어떤 험난한 길을 가야하는지 알려주시면 감사하겠습니다
- 개발자로서 일하게 될 때 대학원을 필수로 진학해야 한다고 생각하시나요  
필수라면 석사, 박사 중 어느 정도 수준까지 대학원을 진학하는게 좋을까요
- 석사나 박사를 통합 과정을 통해 빠르게 마칠 수 있는 길이 있을까요
- 랩실을 컨택할 때 꿀팁이 있다면 말해주세요
- 특별함을 갖추려면 꼭 대학원을 가야하는지 궁금합니다
- 학부 공부와 대학원 연구의 차이가 궁금합니다



- 구체적인 취업 방법이 궁금합니다
- 개발자 종류와 역할도 다양하다고 알고 있는데,  
데이터 분석 관련된 업무를 수행하는 개발자도 있나요
- 지금의 개발 분야에 이르기까지 어떤 진로 탐색 활동을 했나요
- 게임 개발 진로를 위해서 대학교 때 했으면 좋겠다 싶은 활동이 있나요  
그리고 게임 개발 업계의 전망과 대우도 궁금합니다
- 각 분야에서 필요한 수학 영역과 그 깊이가 궁금합니다
- 빅데이터 산업의 미래는 어떨까요

# 감사합니다.

utilForever@gmail.com

<https://github.com/utilForever>

Facebook, Twitter: @utilForever