

# 재미있게 교육하는 소프트웨어

장윤재 ([janggoons.kr](http://janggoons.kr)), SW융합교육원, 삼육대학교



# 목차

—  
재미있게 교육하는 SW

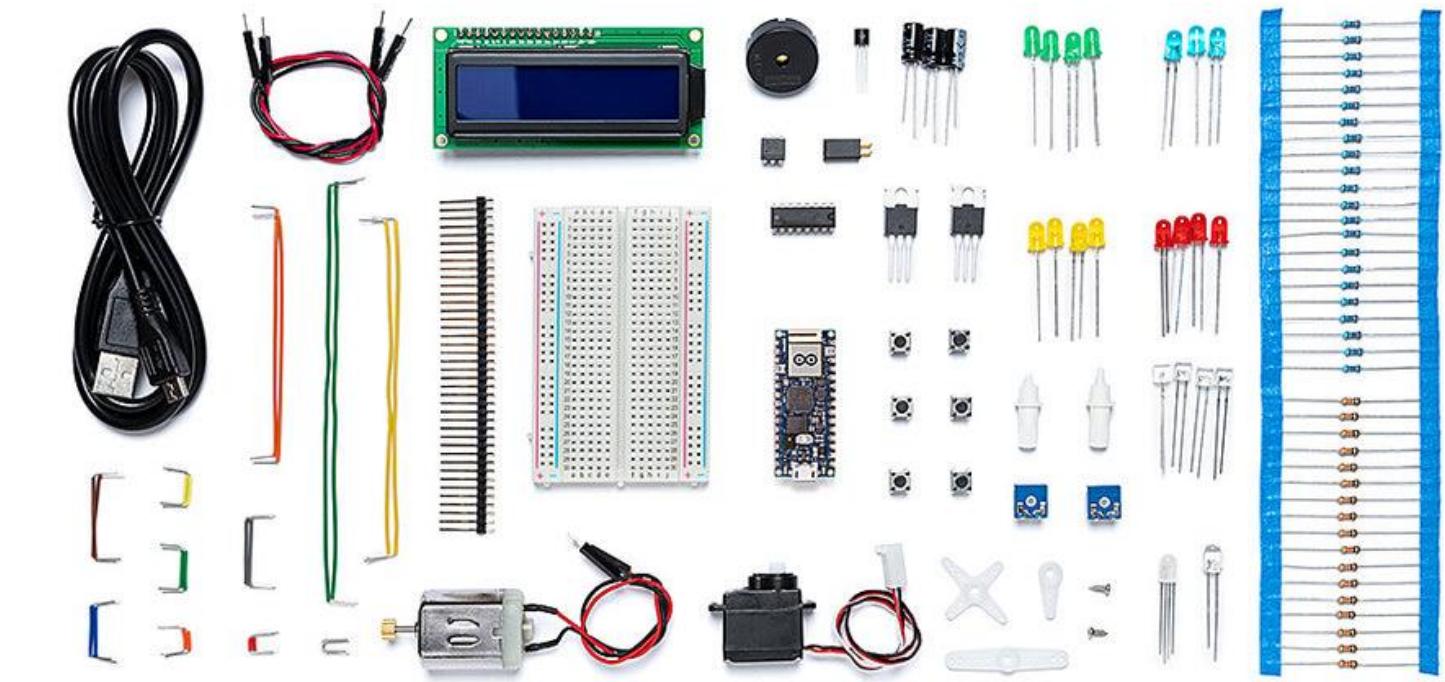
- 01 재미있게 SW를 가르칠 수 있을까?
- 02 재미있게 SW를 가르치는 방법
- 03 각자의 고민

01

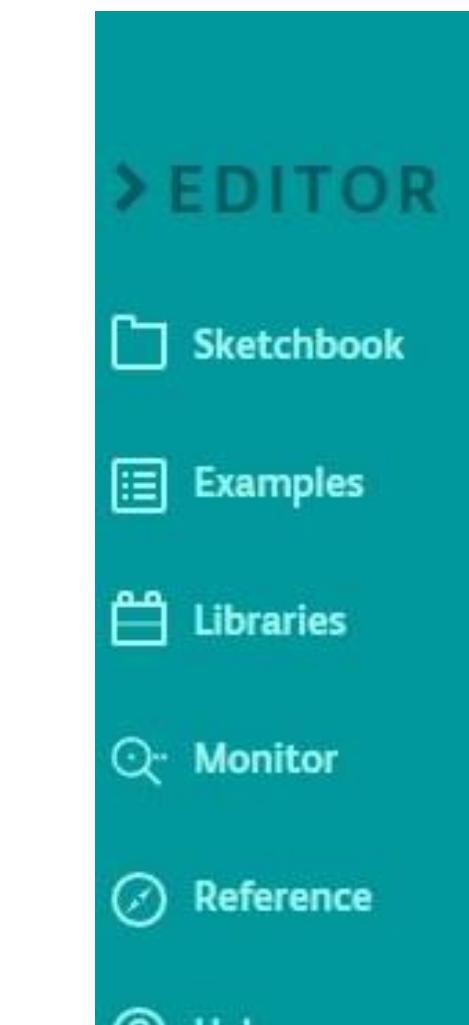
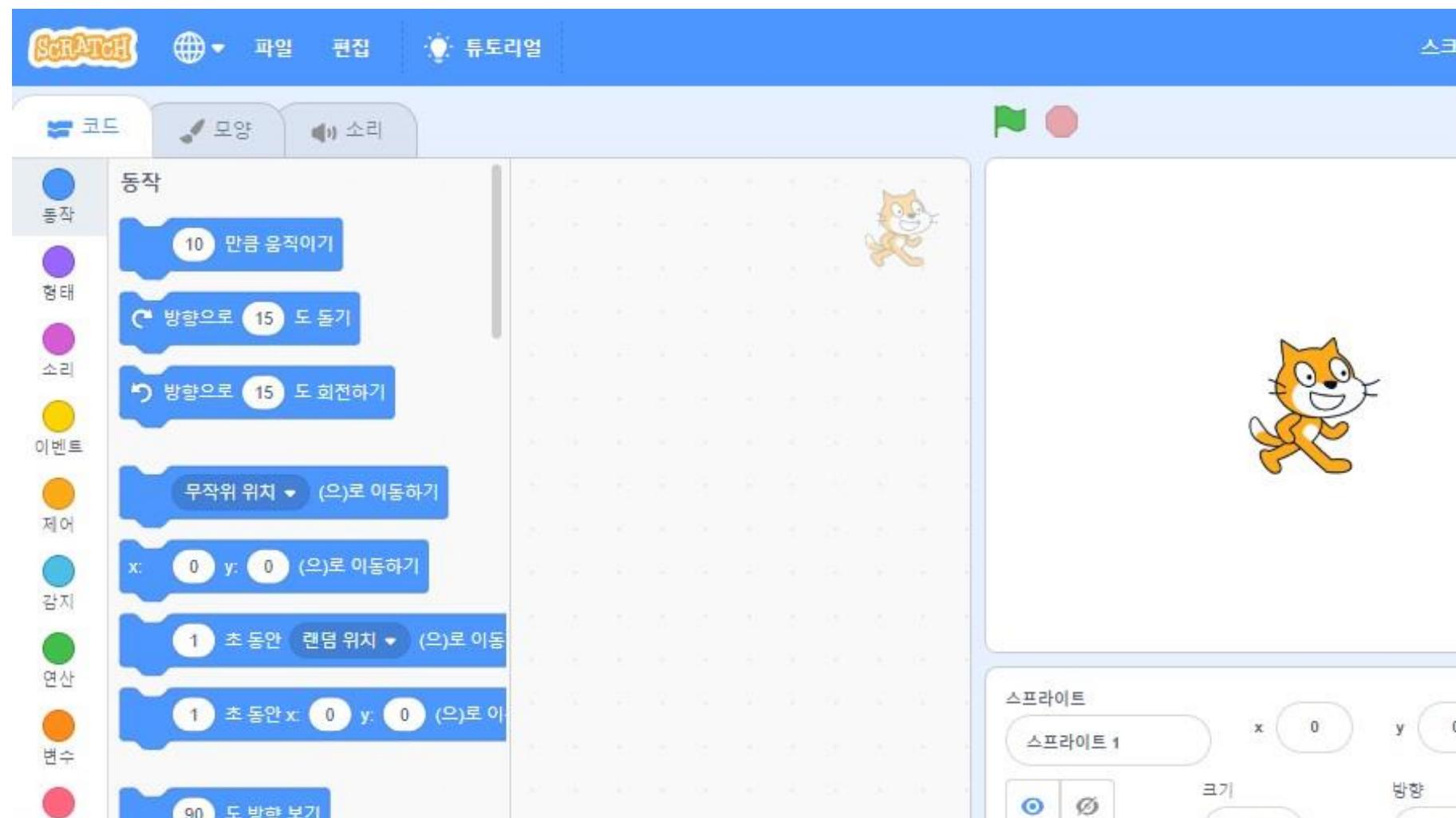
# 재미있게 SW를 가르칠 수 있을까?

## ✓ 두 가지 교육 사례

- **스크래치를 배우는 두 가지 방법**
  - 1. 블록을 순서대로 배우고 간단한 프로그램을 따라서 만들기
  - 2. 학생들이 블록을 탐험하면서 각자 원하는 간단한 프로젝트 만들기
- **아두이노를 배우는 두 가지 방법**
  - 1. 부품별로 배우고 간단한 프로그램을 따라서 만들기
  - 2. 다양한 부품을 탐험하면서 각자 원하는 간단한 프로젝트 만들기



source: [https://store-usa.arduino.cc/products/iot-bundle?pr\\_prod\\_strat=copurchase&pr\\_rec\\_id=91d02758d](https://store-usa.arduino.cc/products/iot-bundle?pr_prod_strat=copurchase&pr_rec_id=91d02758d)



```
sketch_oct20a
✓ → ✗ Arduino Nano
sketch_oct20a.ino ReadMe.adoc
1 void setup(){
2   pinMode(11, OUTPUT);
3   Serial.begin(9600);
4 }
5
6 void loop(){
7   bool buttonVal = digitalRead(3);
8   Serial.println("buttonVal");
9   if(buttonVal){
10     digitalWrite(11, HIGH);
11   } else {
12     digitalWrite(11, LOW);
13 }
```

## ① 용어에서 시작

- "재미있게"

- 교수자가 재미있는가? 학습자가 재미있는가?
- 어떻게 가르쳐야 할까?
- 재미란 무엇일까?

- "SoftWare"

- 우리가 가르치는 것은 SW인가? 프로그래밍언어인가? 컴퓨터인가? 컴퓨터 과학인가? 정보인가?
- 무엇을 가르쳐야 할까?

## ⑤ 무엇을 가르쳐야 할까?

- 우리가 하는 교육의 목표는 무엇인가?

- 1) SW 기술을 잘 이해하고 활용하기

- 무어의 법칙(1965년) : 반도체에 집적하는 트랜지스터 수는 적어도 18개월 마다 2배로 증가한다.
    - “무어의 법칙” 보다 7배 빠르다, 질주하는 인공지능” [[링크](#)]
    - SW 개발자를 양성하는 것이 목표인가?
    - AI 기술을 활용(융합)하여 문제 해결하기, AI융합교육

- 2) SW(또는 Computer Science)의 핵심 개념과 원리를 이해하여 문제 해결에 활용하기

- SW 제작을 위한 리터러시, 역량을 키운다.
      - 프로그래밍 언어를 통해 문제 해결 과정을 만들기
      - HW를 제어하는 SW 만들기
    - 컴퓨팅 사고력을 키우는 것이 목표인가?

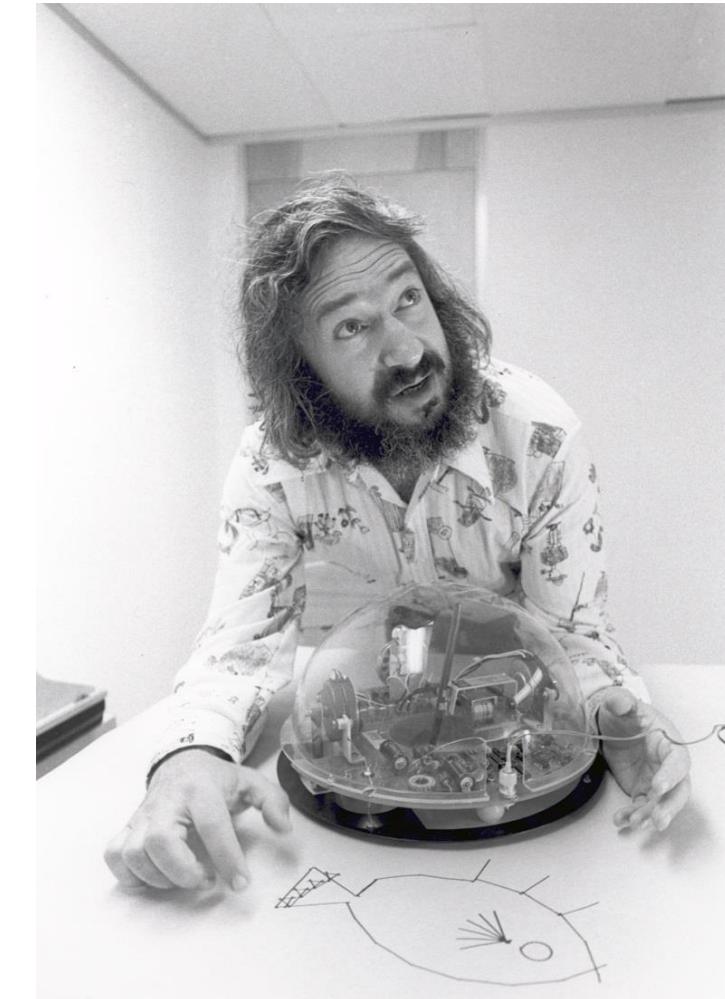
## ✓ 무엇을 가르쳐야 할까?

### • 컴퓨팅 사고력 Computational Thinking

- 컴퓨팅 Computing 을 이용하여 문제를 해결할 수 있는 능력
- 다른 사람/기계에게 일을 시키는 방법
- 명확하고, 논리적이어야 한다.
- 패턴을 찾고, 반복되는 구조를 찾아서, 자동화해야 한다.

### • 인공지능의 등장!

- 문제 해결 과정에서 패턴을 찾고, 반복되는 구조를 찾아서, 자동화한 결과를 도출하는 것은 이제 사람이 아닌 기계(인공지능)
- 프로그램을 사람이 아닌 기계가 만든다.
- 사람은 단지 필요한 데이터를 제공할 뿐이다.
- 컴퓨팅 사고력은 누가 발휘하고 있는가?
  - 기계인가 사람인가?
  - 프로그래머인가, 프롬프트 엔지니어인가?



source: [https://en.wikipedia.org/wiki/Seymour\\_Papert](https://en.wikipedia.org/wiki/Seymour_Papert)

Jeannette M. Wing  
Computational Thinking and  
Thinking About Computing  
Evening Lecture Series  
09.02.2009  
ihmc

source: <https://www.youtube.com/watch?v=C2Pq4N-iE4I>

## ✓ 무엇을 가르쳐야 할까?

- 교수자는 무엇을 가르치는 명확히 인식해야 한다.
  - SW를 만드는 방법, AI 기술을 활용하는 방법, 데이터 분석을 하는 방법을 가르치는가?
  - 컴퓨팅 도구를 이용하여 아이디어를 구현하는 방법, 문제를 해결하는 방법을 가르치는가?

- 기반이 되는 분야
  - 대학의 학문 Disciplines
    - 컴퓨터 과학 Computer Science
    - 데이터 과학 Data Science, 인공지능 Artificial Intelligence
    - 정보학 Informatics
  - 초중등교육의 교과 Subject
    - 국내 : 정보, 인공지능과 기초
    - 해외 : 컴퓨팅, 컴퓨터 과학, 정보

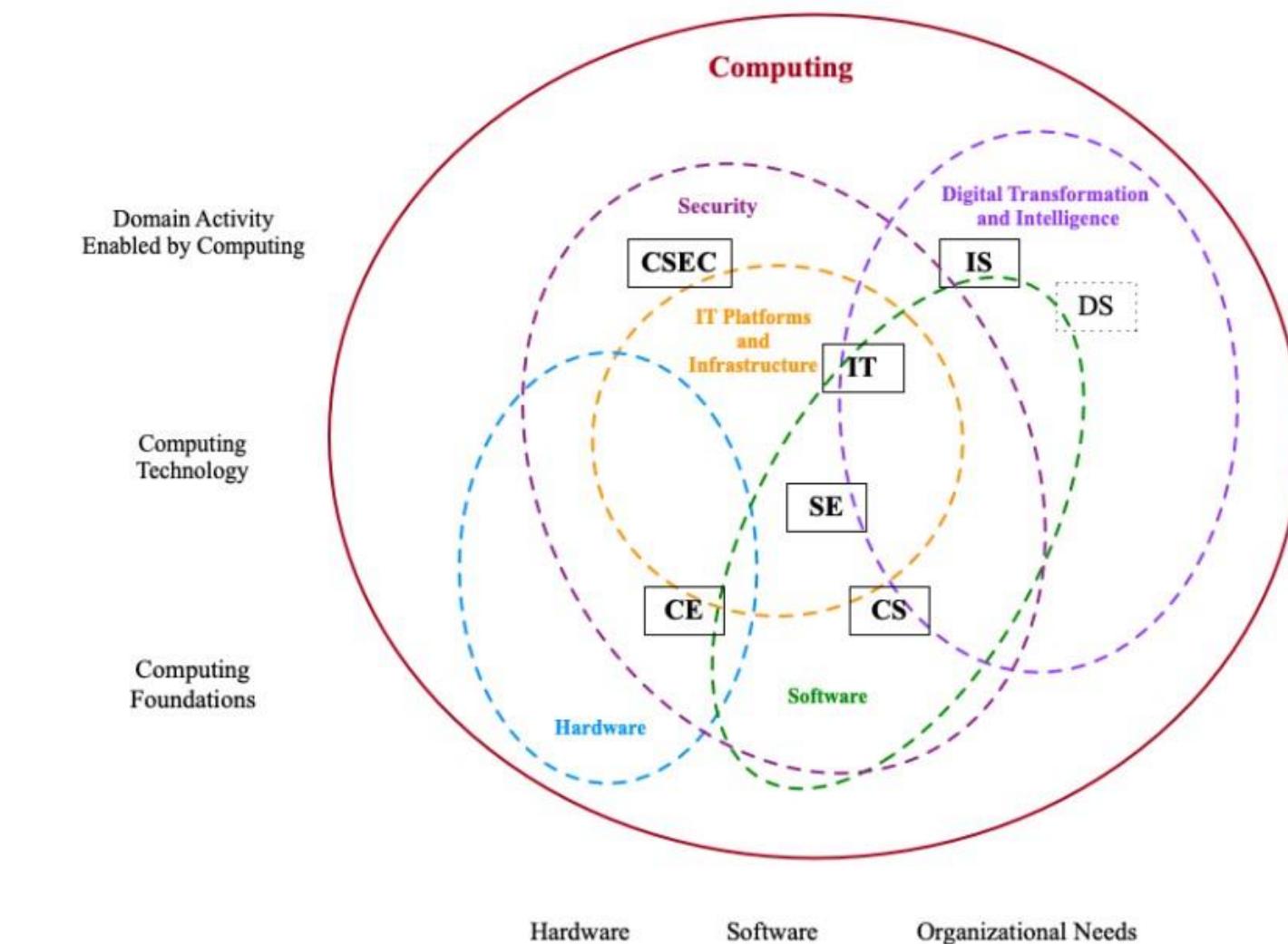


Figure 2.2. A contemporary view of the landscape of computing education  
Legend: Curricular reports: CE=computer engineering; CS=computer science; CSEC=cybersecurity;  
IS=information systems; IT=information technology; SE=software engineering; DS=data science (under development).

source: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>

## ⑤ 어떻게 가르쳐야 할까?

- SW 교육에서 사용할 수 있는 수업 방법
  - 이론 수업, 실습 수업, 이론 + 실습 수업
  - P(roject)BL, P(roblem)BL
  - 따라하기 > 수정/응용하기 > 창안하기
- 수업 방법
  - 가르치는 사람 중심 -> 교수 방법 <- 교수 이론 Instructional Theory
  - 배우는(학습하는) 사람 중심 -> 학습 방법 <- 학습 이론 Learning Theory

## ⑤ 어떻게 가르쳐야 할까?

- 교수 방법

- 정해진 수업 시간 내에서, 정해진 교육 목표를 달성하기 위해, 정해진 내용을 효과적으로 전달하기 위한 방법을 정하는 것
- 주로, 교수 설계 과정 Instructional Design Models에서 방법을 결정한다.
- ADDIE 모형(분석 > 설계 > 개발 > 실행 > 평가), 백워드 설계 Backward Design 모형(목표 > 평가 > 내용)

- 학습 방법

- 인간은 어떻게 학습하는지를 중심으로, 행동주의, 인지주의, 구성주의 등의 이론을 바탕으로 학습 방법을 정하는 것
- 주로, 교수 방법 내에서 학습 전략 Learning Strategy 등으로 제시된다.
- 컴퓨터 과학 교육 분야에서의 학습 이론, 페퍼트의 구성주의 Papert's Constructionism

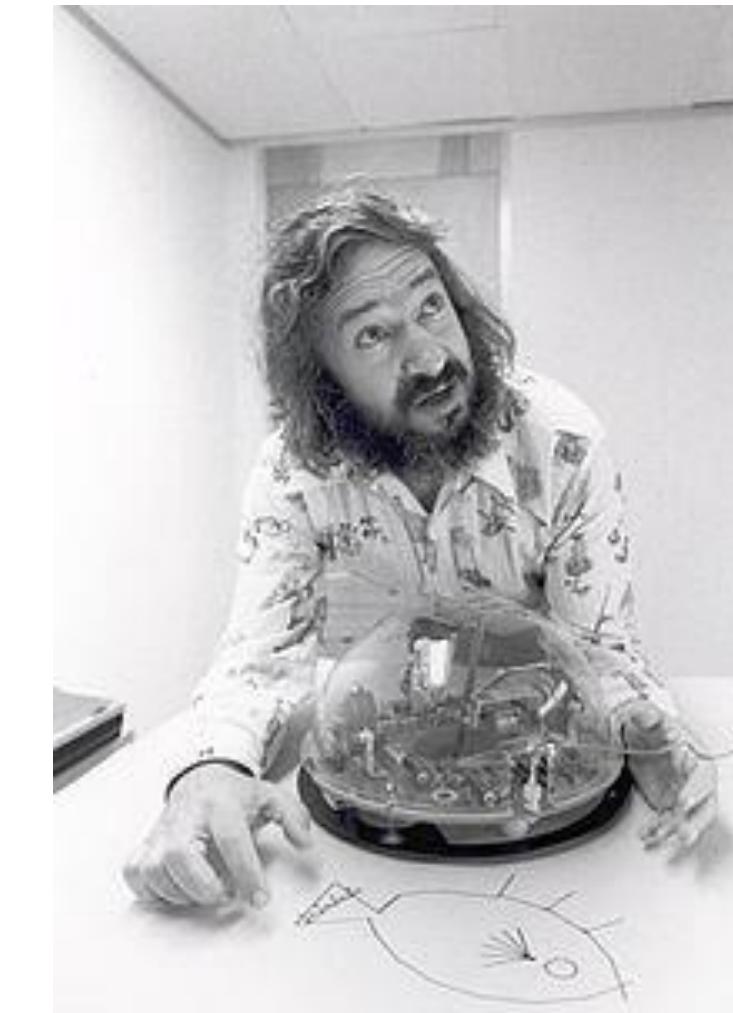
## ⑤ 어떻게 가르쳐야 할까?

- 지식은 어떻게 구성되는가?
  - 피아제 Piaget는, 자신의 인지구조에 새롭게 추가되거나 변형되면서 (지식이) 구성된다.
  - 페퍼트 Papert는, 자신에게 의미있는 (개념적) 도구를 통해 만들어보면서 (지식이) 구성된다.



Jean Piaget (1896~1980)

source: [https://ko.wikipedia.org/wiki/%EC%9E%A5\\_%ED%94%BC%EC%95%84%EC%A0%9C](https://ko.wikipedia.org/wiki/%EC%9E%A5_%ED%94%BC%EC%95%84%EC%A0%9C)



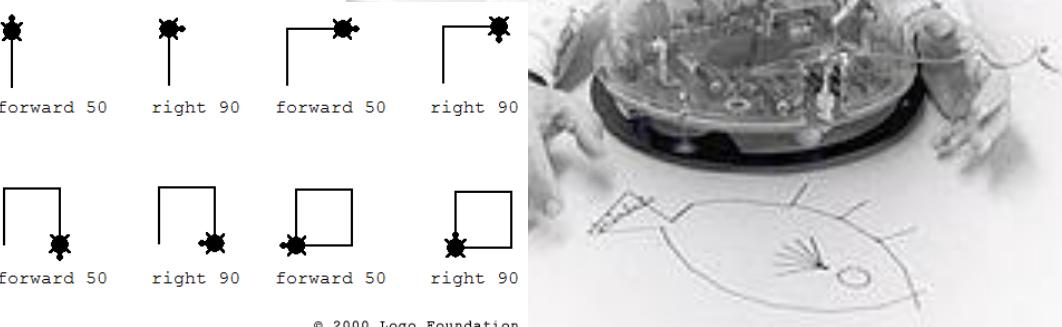
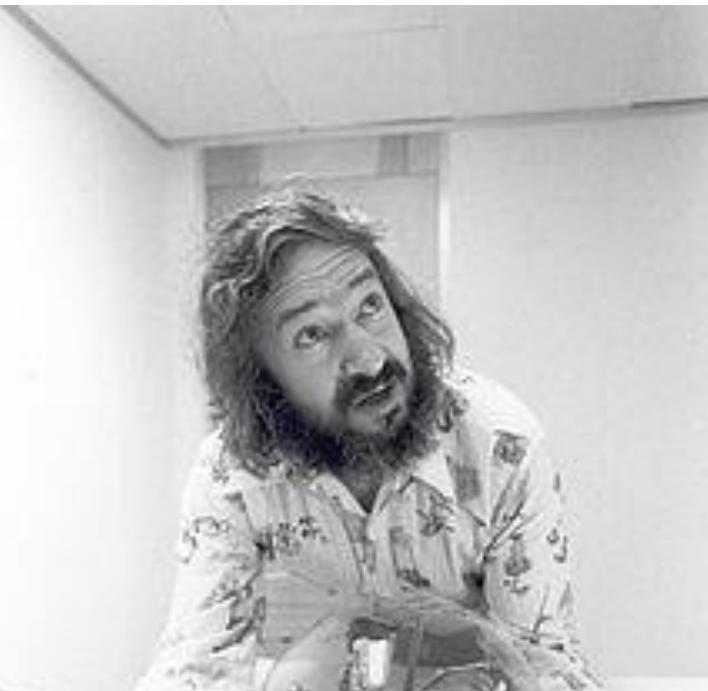
Seymour Papert (1928~2016)

source: [https://en.wikipedia.org/wiki/Seymour\\_Papert](https://en.wikipedia.org/wiki/Seymour_Papert)

## ⑤ 어떻게 가르쳐야 할까?

- 페퍼트의 구성주의는 어떻게 발전하였나?

- 앨런 케이 Alan Kay, 모든 아이들을 위한 컴퓨팅 환경, 다이나북 Dynabook, 스퀵 이토이 Squeak Etoys
- 미첼 레스닉 Mitchel Resnick, 창의 컴퓨팅 Creative Computing, 스크래치 Scratch



Seymour Papert (1928~2016)

source: [https://en.wikipedia.org/wiki/Seymour\\_Papert](https://en.wikipedia.org/wiki/Seymour_Papert)

source: <https://techcommunity.microsoft.com/t5/small-basic-blog/small-basic-the-history-of-the-logo-turtle/ba-p/337073>



Alan Kay (1940~)

source: <https://en.wikipedia.org/wiki/Dynabook>



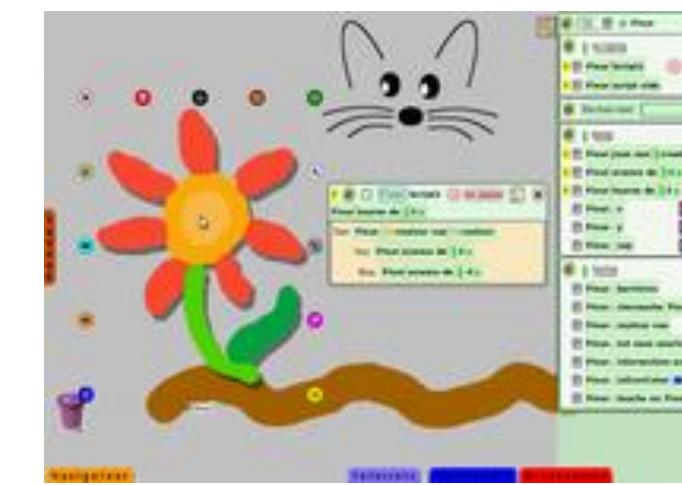
Mitchel Resnick (1956~)

source: [https://en.wikipedia.org/wiki/Mitchel\\_Resnick](https://en.wikipedia.org/wiki/Mitchel_Resnick)



시모어 페퍼트로부터 배우기 (2014, MIT)

source: <https://www.hankookilbo.com/News/Read/201608270461393704>



Squeak Etoys

source: [https://en.wikipedia.org/wiki/Etoys\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Etoys_(programming_language))



Scratch

창의적 배움의 나선

## ⑤ 어떻게 가르쳐야 할까?

### • 나는 언제 제대로 학습했을까?

- 책을 본다, 교강사의 설명을 듣는다, 강의 영상을 본다.
- 자격증 시험을 준비한다. 관련 대학(학과)에 진학한다. 교육기관에 참여한다.
- 다른 사람의 프로젝트를 분석한다. 토이 프로젝트를 만든다.

### • 나는 언제 즐겁게 학습했을까?

## ⑤ 재미란 무엇일까?

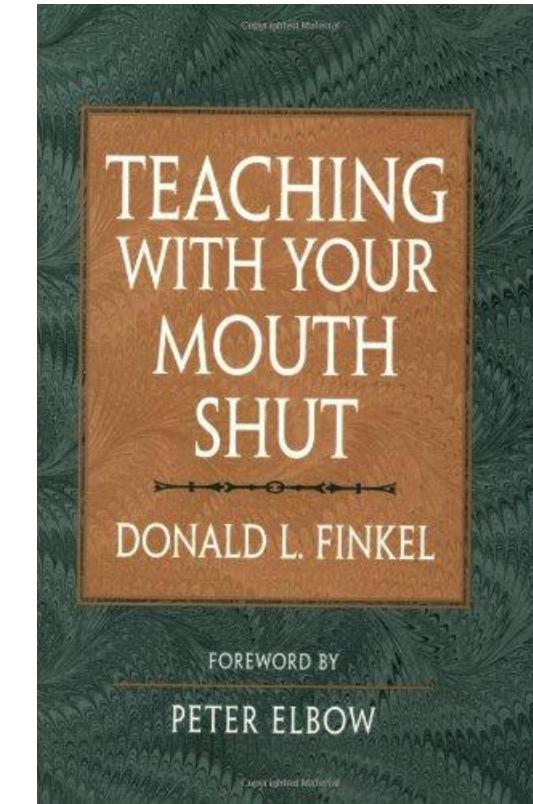
- 즐거움, 재미는 누가 느끼는 걸까?
  - 학습자가 즐거운가? 교수자가 즐거운가?
- 재미를 확인하는 방법
  - 남이 즐거운지 아닌지를 확인하는 방법
  - 내가 즐거운지 아닌지를 확인하는 방법

## ⑤ 재미란 무엇일까?

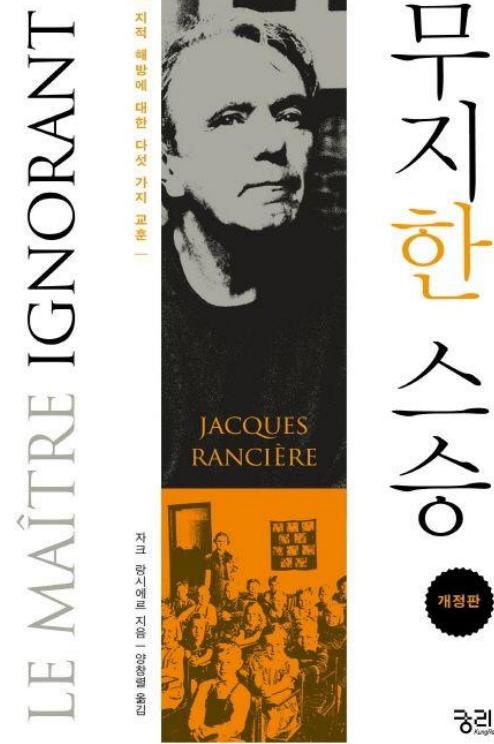
- 질문을 바꿔 보자.
  - 어떻게 하면 (학습자가) 재미 fun 있는 수업을 할 수 있을까?
  - > 어떻게 하면 (학습자가) 참여 engagement하는 수업을 할 수 있을까?
  - > 어떻게 하면 (학습자에게) 의미 meaning 있는 수업을 할 수 있을까?

## ✓ 재미란 무엇일까?

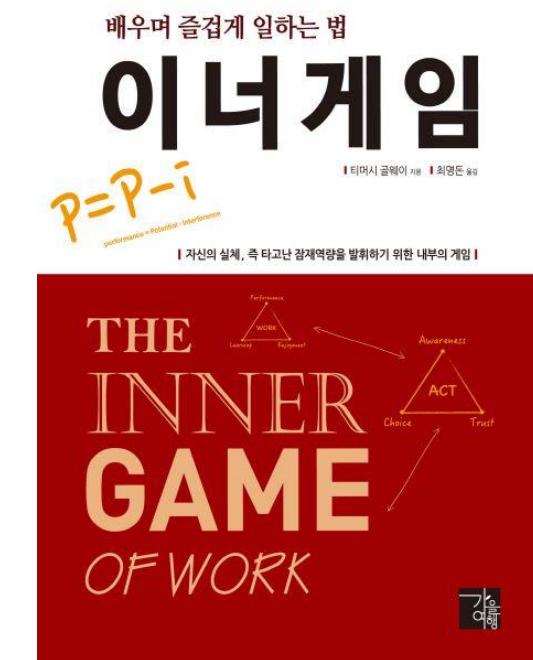
- 학습자가 참여하는 수업
  - 학습자가 졸지 않는 수업
    - 교수자는 말을 줄이고, 학습자가 말을 많이 하는 수업
    - 동료 학습자와 함께 참여하는 수업
    - 서로 설명하고 이해하기
  - 학습자가 주도하는 수업
    - 해야 할 일을 직접 정하기
    - 만든 결과물을 자랑하기
    - 스스로 관찰, 신뢰, 선택하기
    - 놀이를 참여하기, 만들기



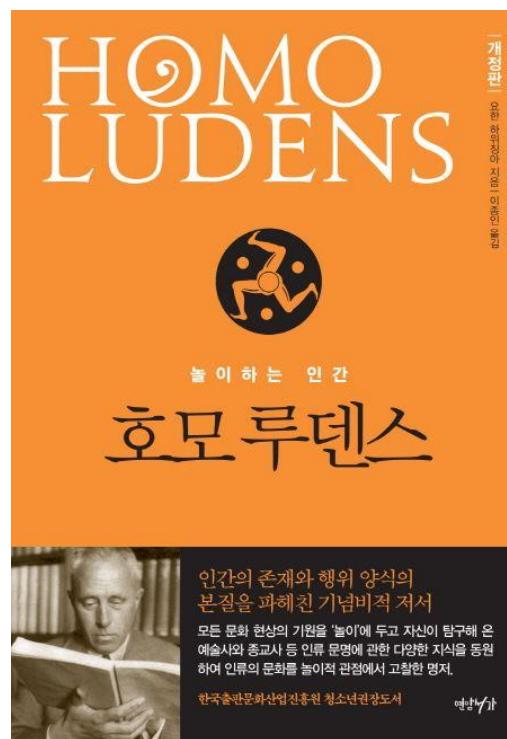
source: <https://www.amazon.com/Teaching-Your-Mouth-Donald-Finkel/dp/0867094699>



source: <https://product.kyobobook.co.kr/detail/S000000867477>



source: <https://product.kyobobook.co.kr/detail/S000001975255>



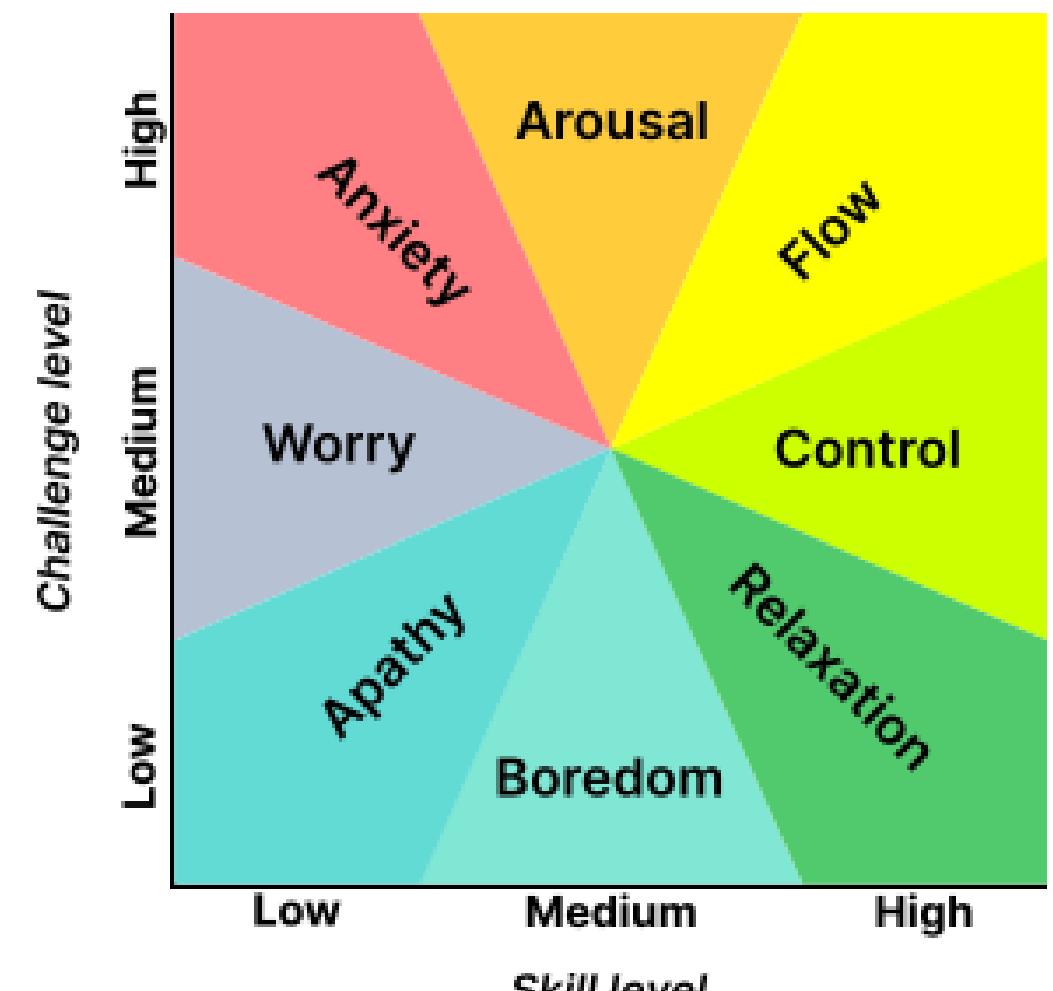
source: <https://product.kyobobook.co.kr/detail/S000001796677>

## ✓ 재미란 무엇일까?

- 학습자에게 의미있는 수업
  - 교육 목표 - 내용 - 평가를 일치하기
  - 학습자의 요구사항을 파악하기
  - 어렵지만 재미있는 도전 과제를 제시하기
  - 관심사를 연결하고 학습 흥미를 유도하기



source: [https://twitter.com/phil\\_carew/status/903048957424615424](https://twitter.com/phil_carew/status/903048957424615424)



Flow Theory

source: [https://en.wikipedia.org/wiki/Flow\\_\(psychology\)](https://en.wikipedia.org/wiki/Flow_(psychology))

## ⑤ 정리하자면,

- 무엇을 가르쳐야 할까?
  - 소프트웨어, 인공지능, 데이터과학
  - 배경은 컴퓨터 과학 (또는 정보학)
  - 초중등 학생은 학문이 아닌 교과 (정보 교과)
  - 기술을 이해하고 활용하기, 아이디어를 표현하고 문제를 해결하기
- 어떻게 가르쳐야 할까?
  - 학습자가 즐겁게 지식을 구성할 수 있는 수업
  - 학습자가 참여하고 의미를 느끼는 수업

02

## 재미있게 SW를 가르치는 방법

### ⑤ 어떤 SW 수업을 해야 즐거울까?

- 학습자를 참여시키기
  - 이론 > 이론+실습 > 실습+이론 > 실습
    - 교수자의 말을 줄이기
    - 교수자가 아닌 코치가 되기
  - 혼자 > 페어 프로그래밍
    - 학습자가 서로 가르치고 배우고 격려하기
    - 교수자는 설명하는 방법, 질문하는 방법을 알려주기



## ⑤ 어떤 SW 수업을 해야 즐거울까?

- 학습자에게 의미 있는 수업 만들기

- 스스로 문제 또는 주제를 정하기

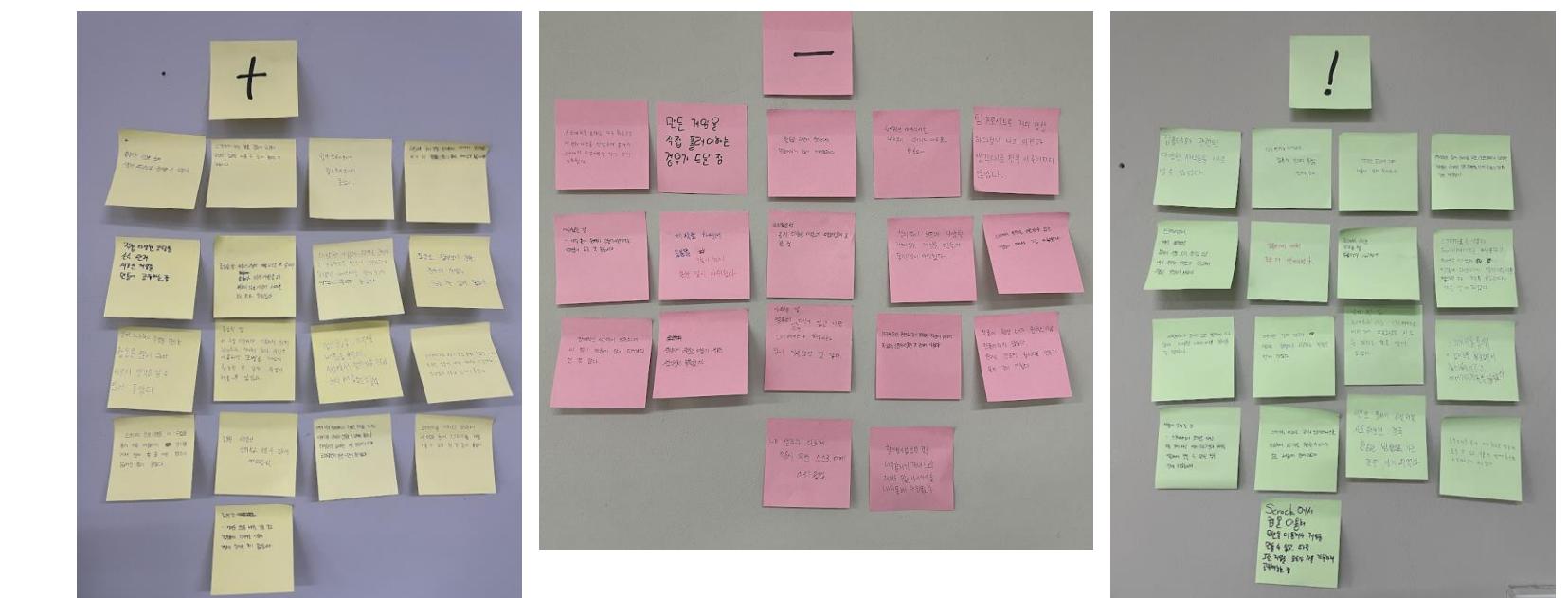
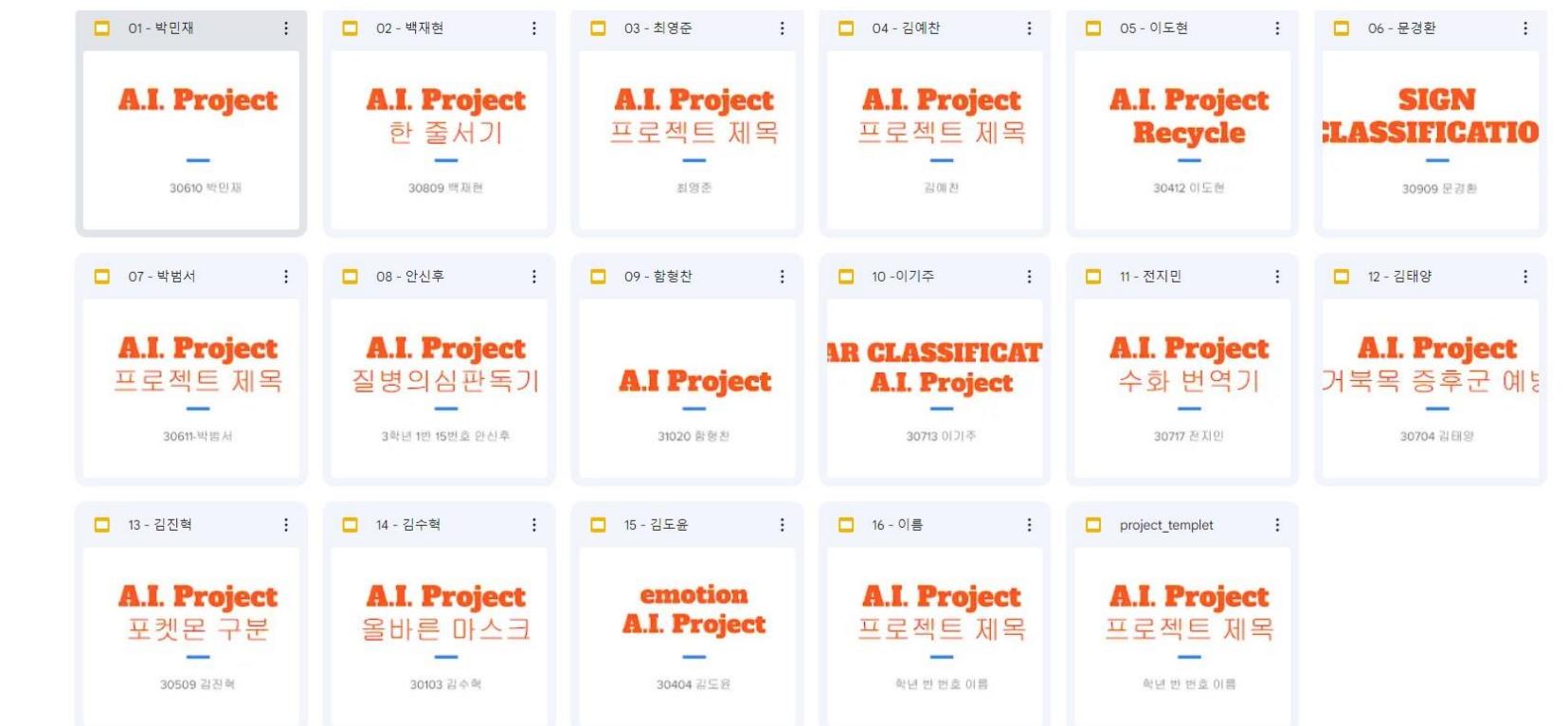
- 실습해본 내용을 토대로 학습자가 만들고 싶은 것을 정하기
    - 혼자 정하기, 함께 정하기

- 학습자가 만든 것을 자랑하기

- 아이디어부터 결과물까지 과정을 기록하고, 공유하기
    - 긍정적인 피드백, 동료 평가
    - 공동편집 문서/프레젠테이션

- 학습 과정을 성찰(회고) 하기

- Plus-Minus-Interest(Insight)
    - 시작하기 전으로 돌아가서 나에게 한 마디 한다면?
    - 온라인 설문, 포스트잇



### ✓ SW 수업을 도와주는 도구들

- 프로그램 코드를 공유(바로 실행)할 수 있는 도구
  - [github](#), [glitch](#), [replit](#), [codepen](#), [codesandbox](#), [google colab](#)
- 프로젝트 내용을 공동 작업하고 확인할 수 있는 도구
  - [Google Drive](#), [Microsoft OneDrive](#)
- 학습자의 의견을 수집할 수 있는 도구
  - [Google Forms](#), [Microsoft Forms](#), [포스트잇 앱](#)
  - [Mentimeter](#), [Trello](#)

### ⑤ [활동] 한 개를 주고, 여러개를 받기

- 준비 : 5~6명이 한 팀, 포스트잇, 필기도구
- 규칙
  - 내가 알고 있는(경험한), 재미있게 SW를 가르치는 방법 한 가지를 포스트잇에 적기
  - 포스트잇을 모아서 비슷한 것과 다른 것을 구분하기
  - 각자 돌아가면서 이야기하기

03

## 각자의 고민

## ⑤ 현재 강의에서 고민하는 점

- 코드 제작이 목표인가 콘텐츠(프로그램, 프로젝트, 작품) 제작이 목표인가?

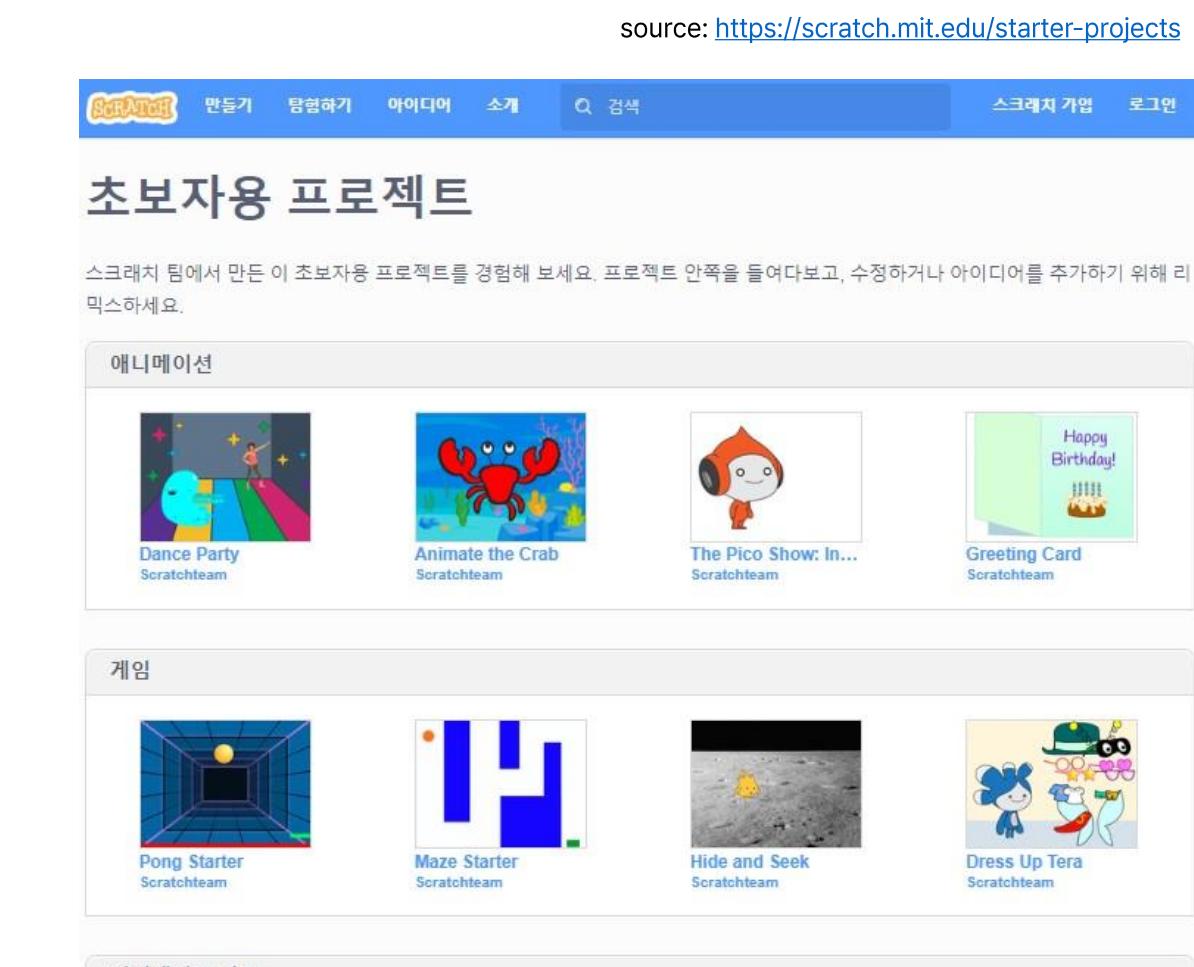
“쉬운 코드로 다양한 콘텐츠”

“처음 체험하는 친구들에게 쉽고 재밌게 학습하는 방법”

“프로젝트주제를 설정하여 파이썬기초를 강의후 주제별로 활동을 마무리까지 학교나 센터에서에서 현장수업 시 수업주차가 짧은 맞보기 강의 요청이 많은데 기초를 즐겁게 배우면서 프로젝트수업으로 연계하는 강의커리큘럼알려주세요~”

## • 참고

- 씨앗 프로젝트에서 시작하기 >
- 수업사례 : [인공지능과 미래사회 진로 특강](#)



### ⑤ 현재 강의에서 고민하는 점

- 어디부터 어디까지 가르쳐야 할까?

“학생들이 따라오기 어려워하는 부분이 많습니다. 일주일이 지나면 개념이 초기화되는 경우가 많습니다.”

“학생들 수준에 맞는 맞춤형 수업”

“학생들이 더 탐구하고 집중할 수 있는 방법이 무엇일까?”

- 참고

- 교육과정 재구성(국가수준 > 지역수준 > 학교수준)
- 학습자의 관심사와 연결하기

## ⑤ 현재 강의에서 고민하는 점

### • 교강사는 교육 콘텐츠 제작자인가 전달자인가?

“강의 교안대로 하고싶은데 가끔씩 뜯금없는 부분이 나온다면 설명이 충분하지 않아..당황스러울때가 있습니다. 설명이 직관적이지 않아서 아쉽습니다.”

“실전 웹 프로그래밍과 인공지능 프로그래밍의 경우 중간과정부터 1차시에 학생들이 인지해야 하는 내용이 많아지면서, 어떻게 쉽게 전달해 갈지에 대한 고민이 있습니다.”

“너무 기초없이 진행하는 느낌이라 별도 교안 제작하는 점”

“교안의 내용이 처음 접하는 경우 다소 어렵다고 판단되는 경우 개인적으로 작업을 하여 학생들에게 쉽게 전달 할 수 있는 방법들을 확인하고 있습니다. 학생들에게 교안의 내용을 기준으로 쉽게 이해 가능한 방법들을 고민하고 있습니다.”

### • 참고

- 교육 콘텐츠 재구성, 현장 교사의 내용교수지식(PCK)
- 내용을 모두 전달해야 하나? 목표를 달성해야 하나?

## ⑤ 현재 강의에서 고민하는 점

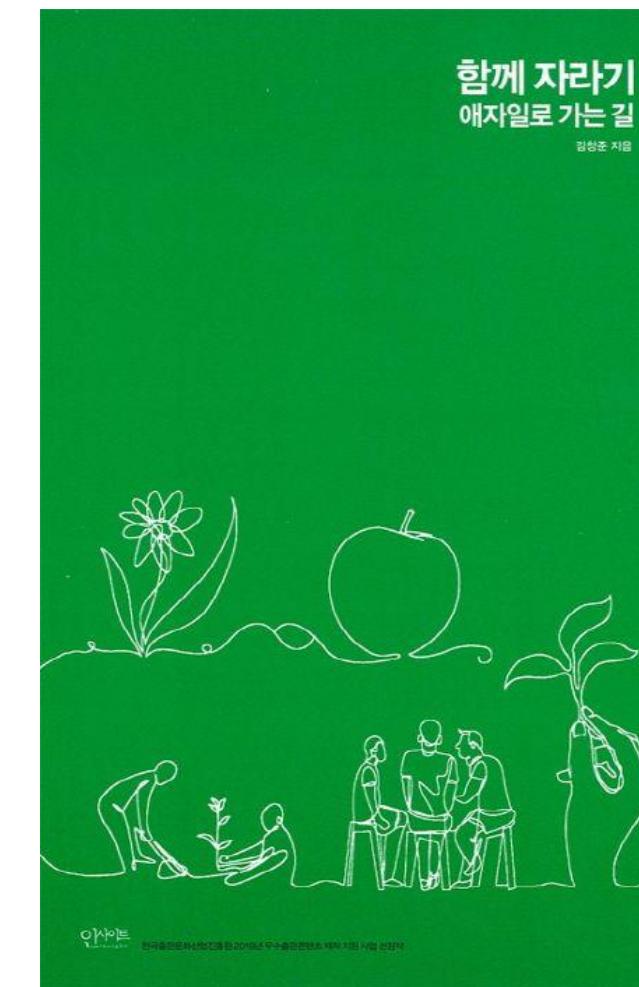
- 서로 다른 수준의 학생들이 한 반에 모두 있다면?

“흥미도와 학습 수준이 다른 학생들이 같은 시간 같은 교실에서 수업할 때 균형 잡힌 수업을 어떻게 할 것인가”

“고등학교에 따라 학생들의 수준과 편차가 심해서~ 관심이 없는 친구들의 경우 어떻게 동기부여를 해 줄 수 있을지 고민됩니다.”

- 참고

- 동료학습, 페어 프로그래밍, [함께 자라기](#) >



source: <https://product.kyobobook.co.kr/detail/S000001033071>

### ⑤ 현재 강의에서 고민하는 점

- 영어(제2언어)를 재미있게 배우는 방법?

“언어를 재미있게 수업하는 방법”

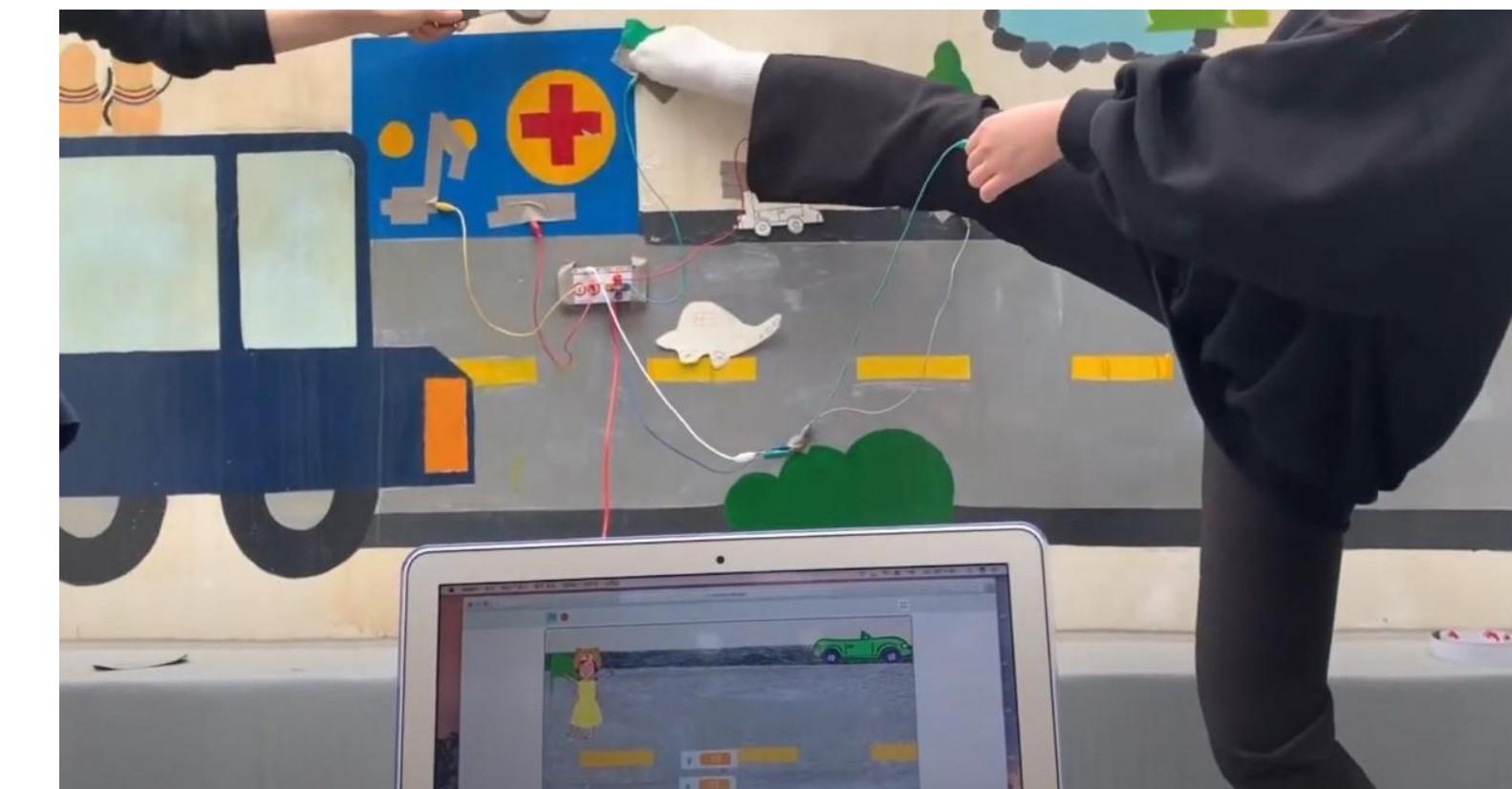
“언어 수업시 학생들이 재미있게 수업하는 방법”

“언어수업시 학생들이 재미있게 수업하는 방법”

“텍스트코딩을 어려워하는 아이들에게 보다 쉽고 재미있게 교수하는방법”

- 참고

- 언어는 목적이 아닌 수단!
- 수업사례 : [컴퓨팅으로 예술 작품 만들기](#) >
  - 평가요소 : 독창성, 공감성, 재미



source: <https://www.youtube.com/watch?v=KD-v9WyOiHU>

## ⑤ 현재 강의에서 고민하는 점

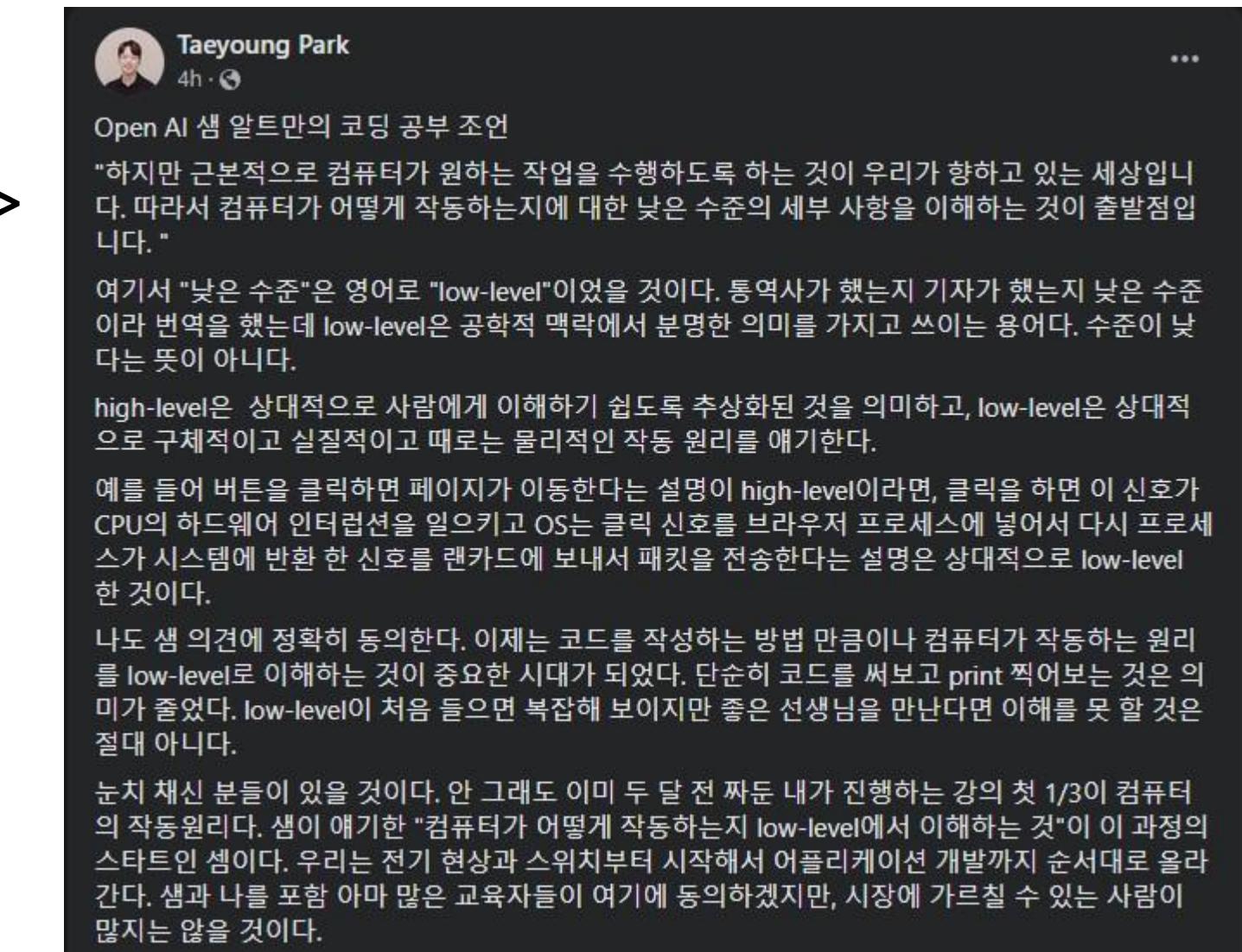
- 복잡하고 어렵고 추상적인 개념을 이해하려면?

“프로그램에 대한 개념을 쉽고 재미있게 알려주기 위한 방법에 대해 알고 싶습니다.”

“역전파 cnn 등 어려운 내용을 학생들에게 더 쉽게 전달하고 싶습니다.”

- 참고

- [CS언플러그드](#), [AI언플러그드](#)
- 인공지능 시대에서 코딩 교육, 샘 알트만의 조언 >



Taeyoung Park  
4h ·  ...  
[Open AI 샘 알트만의 코딩 공부 조언](#)  
"하지만 근본적으로 컴퓨터가 원하는 작업을 수행하도록 하는 것이 우리가 향하고 있는 세상입니다. 따라서 컴퓨터가 어떻게 작동하는지에 대한 낮은 수준의 세부 사항을 이해하는 것이 출발점입니다."  
여기서 "낮은 수준"은 영어로 "low-level"이었을 것이다. 통역사가 했는지 기자가 했는지 낮은 수준이라 번역을 했는데 low-level은 공학적 맥락에서 분명한 의미를 가지고 쓰이는 용어다. 수준이 낮다는 뜻이 아니다.  
high-level은 상대적으로 사람에게 이해하기 쉽도록 추상화된 것을 의미하고, low-level은 상대적으로 구체적이고 실질적이고 때로는 물리적인 작동 원리를 얘기한다.  
예를 들어 버튼을 클릭하면 페이지가 이동한다는 설명이 high-level이라면, 클릭을 하면 이 신호가 CPU의 하드웨어 인터럽션을 일으키고 OS는 클릭 신호를 브라우저 프로세스에 넣어서 다시 프로세스가 시스템에 반환 한 신호를 랜카드에 보내서 패킷을 전송한다는 설명은 상대적으로 low-level한 것이다.  
나도 샘 의견에 정확히 동의한다. 이제는 코드를 작성하는 방법 만큼이나 컴퓨터가 작동하는 원리를 low-level로 이해하는 것이 중요한 시대가 되었다. 단순히 코드를 써보고 print 찍어보는 것은 의미가 줄었다. low-level이 처음 들으면 복잡해 보이지만 좋은 선생님을 만난다면 이해를 못 할 것은 절대 아니다.  
눈치 채신 분들이 있을 것이다. 안 그래도 이미 두 달 전 짜둔 내가 진행하는 강의 첫 1/3이 컴퓨터의 작동원리다. 샘이 얘기한 "컴퓨터가 어떻게 작동하는지 low-level에서 이해하는 것"이 이 과정의 스타트인 셈이다. 우리는 전기 현상과 스위치부터 시작해서 어플리케이션 개발까지 순서대로 올라간다. 샘과 나를 포함 아마 많은 교육자들이 여기에 동의하겠지만, 시장에 가르칠 수 있는 사람이 많지는 않을 것이다.

## ✓ 현재 강의에서 고민하는 점

- 학교 교육과정과 연계하는 방법?

“AI와 피지컬 수업을 어떻게 교과에 녹여서 수업 할지를 고민중입니다.”

“실생활과의 접목”

“실제 적용가능한 상상력을 높이는 코딩과 인공지능교육 전달”

- 참고

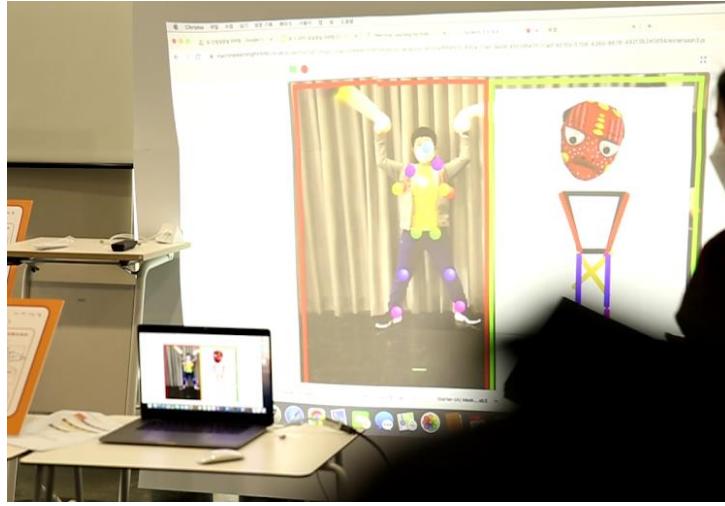
- 국가수준 교육과정, 정보 교과서

- 교수자의 관심사, 학습자의 관심사에서 주제를 찾기

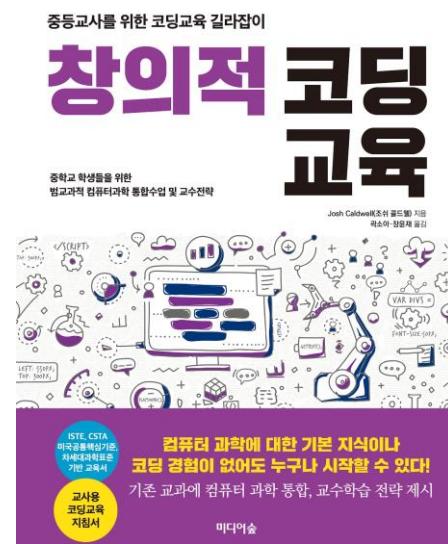
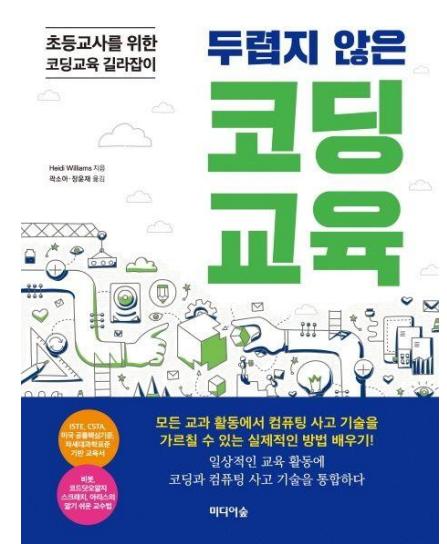
- 수업사례 : 둠칫둠칫 AI 아뜰리에 \_ 인공지능×움직임×놀이 융합교육 콘텐츠 [[소개영상](#)][[소개자료](#)]



source: <https://sites.google.com/view/kmu-sw-camp/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%A8-%EC%86%8C%EA%B0%9C>



source: <https://sites.google.com/view/ai4funplay/home>



source: <https://sites.google.com/view/coding-as-a-playground/home>

source: <https://sites.google.com/view/no-fear-coding/home>

source: <https://sites.google.com/view/creativecodingbook/home>

작성자: 김서진, 양정현, 이현주, 장은재 | 저작권 © 2021 Jpub

### ⑤ 현재 강의에서 고민하는 점

- 중요한 정보를 주어진 시간내에 전달하는 방법?

“강의 스킬”

“학생들에게 중점적으로 전달해야 할 사항들과 효과적인 전달방법”

“강의를 잘 하고 싶어요”

- 참고

- 발성, 표정, 동작을 의도적으로
- 선택과 집중, 반복, 질문, (긍정적) 피드백

### ⑤ 현재 강의에서 고민하는 점

- 교육 효과를 높이는 방법?  
“플랫폼으로 수업시 아이들의 집중을 높여 이해시킬 방법”

- 참고

- 교육이란 교수자와 학습자의 상호작용, [줄탁동시](#)
- 블랜디드 러닝(온라인 + 오프라인)
- “온라인 영상은 오프라인 학습을 위한 보조자료”

### ⑤ [활동] 질문하기

- 준비 : 5~6명이 한 팀, 포스트잇, 필기도구

- 규칙

- 지금 머리속에 떠오르는, 재미있게 SW를 가르치는 방법에 대한 질문(또는 아이디어) 한 가지를 포스트잇에 적기
- 포스트잇을 모아서 비슷한 것과 다른 것을 구분하기
- 질문에 따른 답을 서로 이야기하기
- 답하기 어려운 질문을 선정하여 다 함께 이야기하기

---

# Thanks!

"The best way to predict the future is to invent it." - [Alan Kay](#)