

CHAPTER 01

컴퓨팅사고 교과목 소개

01. 컴퓨팅사고 교과목 소개



■ 학습목표

- 본 강의는 교양 과목 수준으로 소프트웨어와 관련된 사전 지식이 없는 학생(비전공자)들도 주어진 문제를 소프트웨어적으로 해결할 수 있는 컴퓨팅 사고력 향상을 목표로 한다. 일상생활에서 접할 수 있는 문제에 숨어 있는 규칙을 찾고 스크래치 프로그래밍을 이용하여 문제를 논리적인 순서에 따라 해결할 수 있는 **컴퓨팅 사고 능력**을 키운다.

■ 평가방법

- 중간평가는 실습평가로 진행하며 30% 반영(온라인 시험)
- 기말평가는 이론평가로 진행하며 30% 반영 (온라인 시험)
- 기타 20%는 실습과제로 진행 (예 : 연습문제 풀이 및 퀴즈)
- 출석점수는 전체의 20% 반영

■ 강의소통 방법

- 에델바이스 게시판, 질의응답
- 중간고사, 기말고사는 2주전 공지
 - 시험일정, 시험범위, 시험방법 등을 공유

01. 컴퓨팅사고 교과목 소개



■ 주요 공지사항

- 본 강의는 이론1시간 실습1시간으로 진행합니다.
- 평가방법은 학교 평가기준에 따라 **상대평가** 합니다.
 - A~B학점 : 70%
- 중간고사와 기말고사는 이론 및 실기 평가로 실시합니다.
 - 기말고사 **미응시자**는 F처리 (중간평가 미응시자는 0점 처리)
- 출석은 1시수 결석 당 1점 씩 감점하며, **1/3 이상 결석은 F 처리**합니다.
 - 종합정보시스템의 전자 출결 기준
 - 지각처리 1회 (-0.5점)
 - 4주 결석 (총 8시간)
- 공결(군입대, 행사참여, 면접 등) 및 병결(코로나, 교통사고 등)의 경우 반드시 종합정보시스템을 이용하여 확인서를 제출합니다.
- 수업진행 과정에 필요한 실습환경은 각자 설치하고 관리하도록 합니다.
- 수업에 방해되는 행위 발생 시 **퇴실조치**되며 **결석처리**됩니다.
 - 소음발생 및 실습용 컴퓨터를 이용하여 게임 실행
 - 휴식시간 반드시 엄수
- 수업진행에 질문이나 문제점이 발생하면 에텔바이스(질의응답), 전화, 문자를 이용합니다.

CHAPTER 02

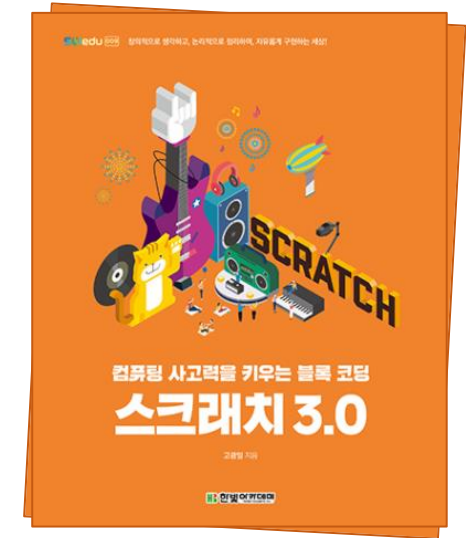
교재 소개

02. 교재 소개

■ 주교재



■ 부교재



- **도서명** : 창의적 문제 해결을 위한 컴퓨팅사고
- **저자** : 교양대학 김시정 교수, 공과대학 김봉한 교수
- **출판사** : 청주대학교 출판부
- **페이지** : 250p

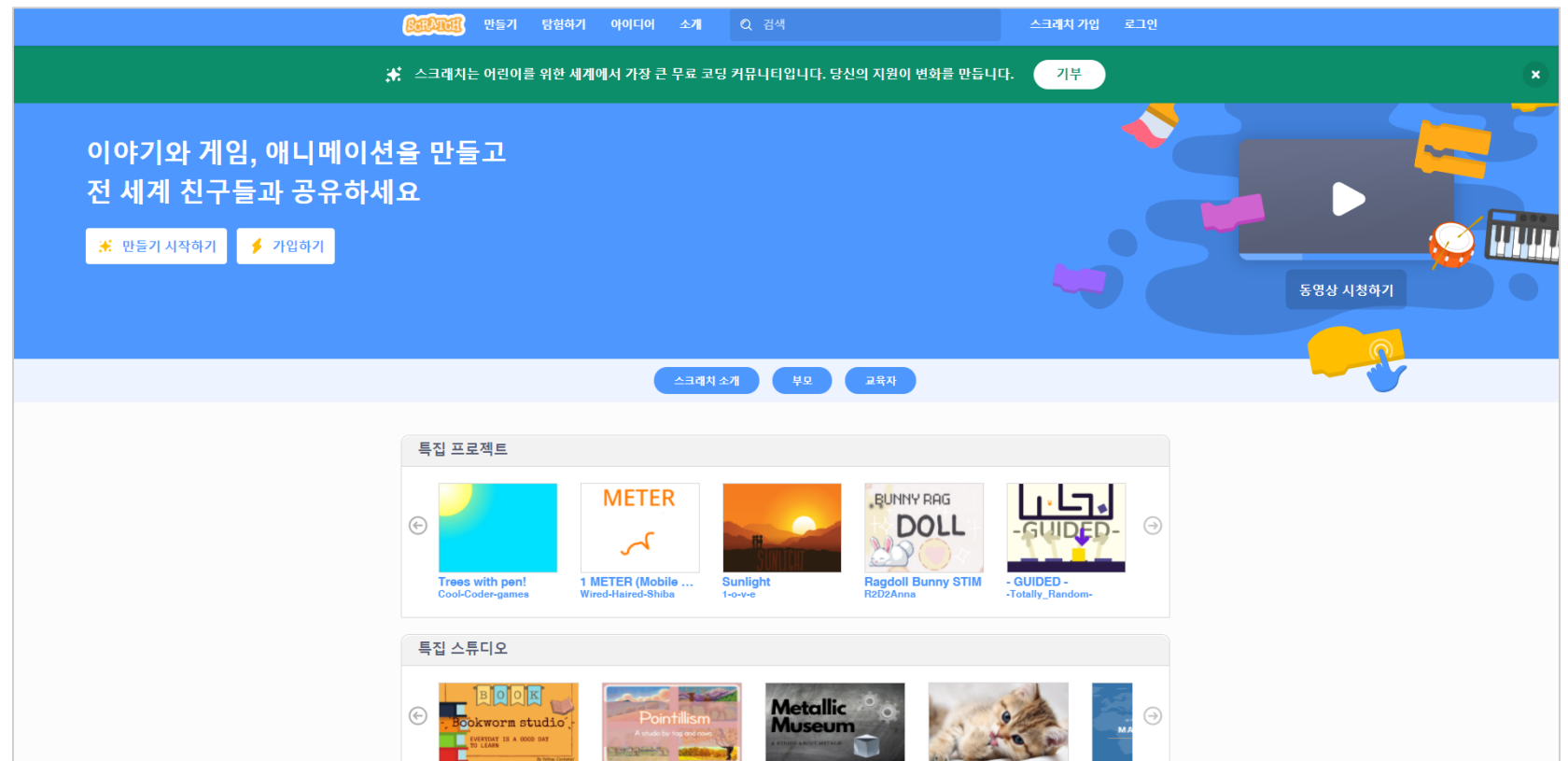
CHAPTER 03

주요 강의진행 내용

03. 주요 강의진행 내용

■ 스크래치의 실행 (Web)

- 실행 환경 : 크롬, 엣지, 파이어폭스, 사파리 (익스플로러 지원하지 않음)
- <http://scratch.mit.edu>

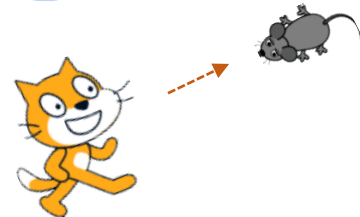


03. 주요 강의진행 내용

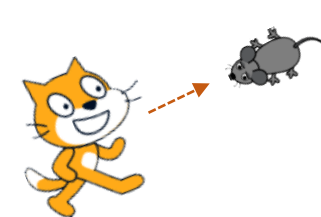
■ 스크래치의 특징

- 스크래치는 스크립트 언어이다.
 - 컴파일을 하지 않고 바로 실행이 가능함
 - 단순하고 쉬운 문법으로 구성됨
 - 프로그램이 따라야 할 규칙 또는 명령어의 집합으로 구성되어 있음
- 스크래치는 레고 블록을 조립하듯이 프로그래밍을 할 수 있다.
 - 학습자들에게 익숙한 드래그 앤 드롭 방식으로 스크립트를 조합함
 - 반복/제어문과 같은 복잡한 프로그램을 블록들을 사용하여 구현할 수 있음
- 스크래치는 프로그래밍 작업을 통합적으로 관리할 수 있다.
 - 프로그램의 전체적인 구조를 파악하고 관리하는데 편리한 환경을 갖추고 있음

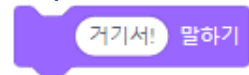
Step : 1



Step : 2

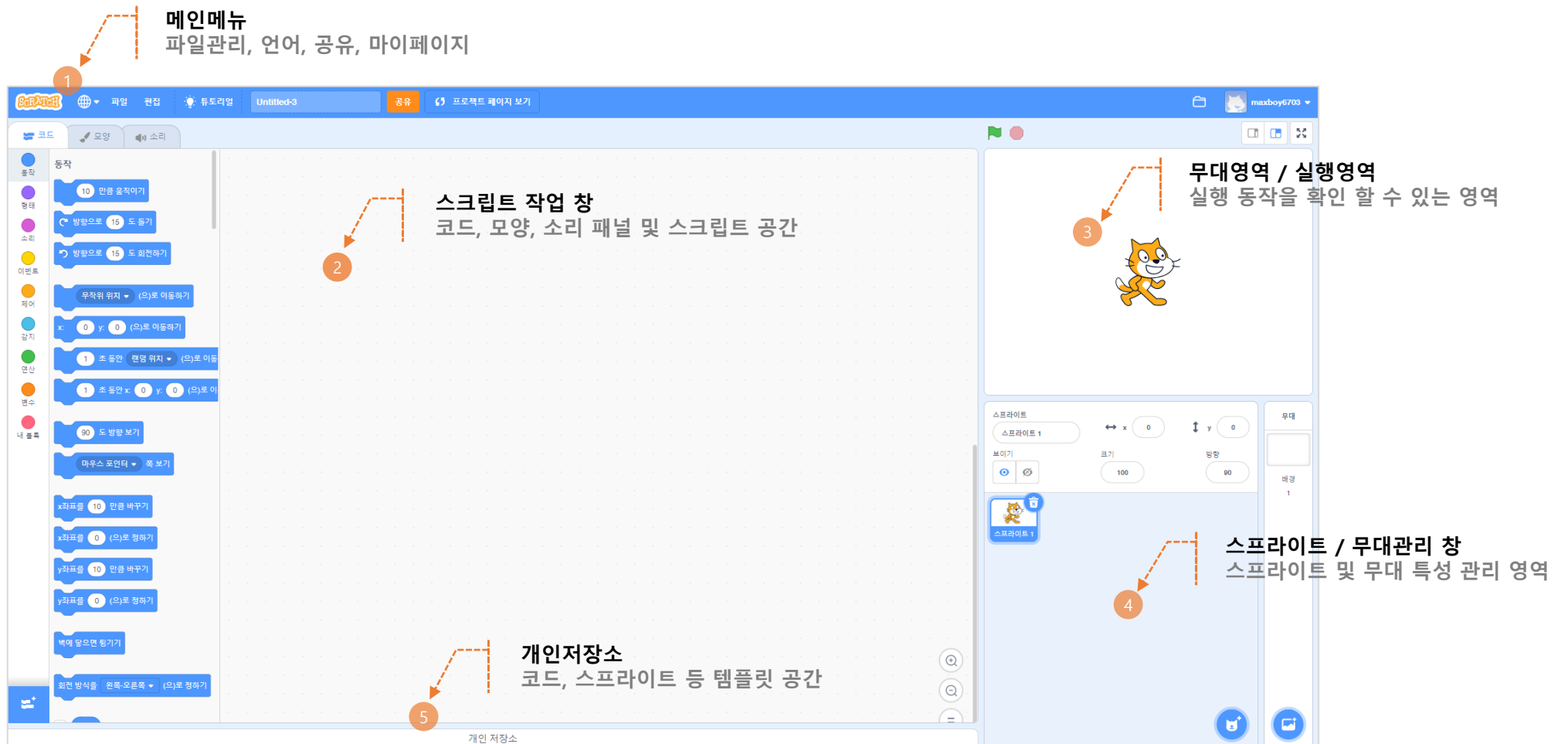


Step : 3



03. 주요 강의진행 내용

■ 스크래치의 프로그래밍 환경 (Web)



CHAPTER 04

강의 계획표

04. 강의 계획표

주차	강 의 내 용
1	강의 오리엔테이션 및 컴퓨팅사고 개념
2	동작 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
3	형태 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
4	소리 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
5	이벤트 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
6	제어 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
7	변수와 리스트의 개념을 학습한다. 변수와 리스트를 활용한 실습 예제를 구현한다.
8	중간평가를 위한 과제를 수행한다. : 스크래치의 기본 구성 블록을 이용한 종합 예제를 학습한다.
9	감지 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
10	확장기능의 추가 방법을 안다. 확장기능의 블록의 사용법을 학습하고 블록을 이용한 실습 예제를 구현한다.
11	나만의 블록 기능 사용법을 학습한다, 컴퓨팅사고의 과정을 통한 모듈화를 적용한 실습 예제를 수행한다.
12	메시지 전달 기능을 구현 방법을 학습한다. 메시지 전달의 개념을 활용한 실습 예제를 수행한다.
13	복제기능의 개념과 활용을 학습한다. 복제 기능을 활용한 실습 예제를 수행한다
14	종합 실습 예제 구현 : 컴퓨팅사고의 단계별 설계를 통한 문제해결을 프로젝트를 통해 수행한다.
15	기말 평가

창의적 문제 해결을 위한

컴퓨팅 사고

Computational thinking

CHEONGJU UNIVERSITY



1주차 2교시

교양대학 전 완 식

CHAPTER 05

컴퓨팅 사고의 개념

05. 컴퓨팅사고의 개념

■ 컴퓨팅 사고의 정의

- 컴퓨팅 사고 **Computational Thinking, CT**는 컴퓨터 **사람이나 기계**가 효과적으로 수행할 수 있도록 **문제를 정의**하고 그에 대한 **답을 기술**하는 것이 포함된 사고 과정의 일체를 일컫는다. 또한, 정답이 하나가 아니라 여러 가지일 수 있는 문제 **Open-ended Problem**는 다양한 변수에 기반한 포괄적이며 유의미한 해답의 도출이 필요한데, 컴퓨팅 사고를 통해서 문제를 해결할 수 있는 알고리즘을 도출 한다.

■ 컴퓨팅 사고의 시작

- 1980년 MIT대학의 시모어 페퍼트 **Seymour Papert** 교수가 처음으로 언급
- 2006년 카네기 멜론 대학의 자넷 윙 **Jeannette Wing** 교수가 컴퓨터 분야의 학술지 ACM에 “**Computational Thinking**” 논문 발표
 - 논문 내용 : <https://www.cs.columbia.edu/~wing/ct-korean.pdf>

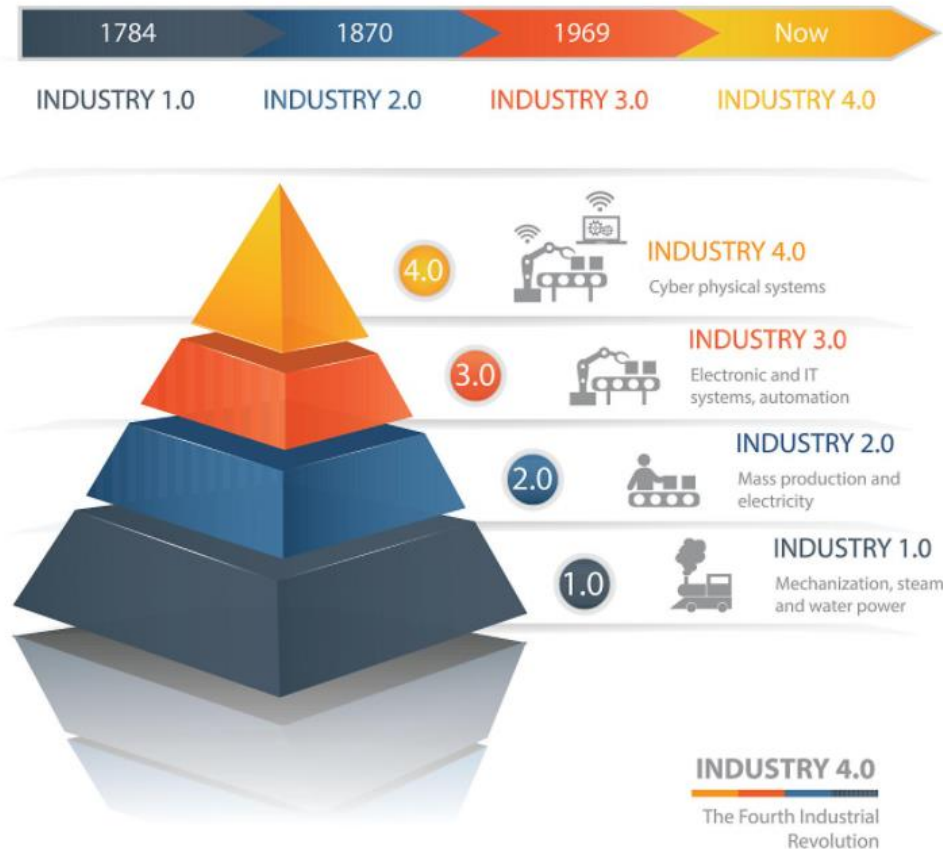


컴퓨팅 사고는 우리가 복잡한 문제들을 해결하기위해 무슨 문제인지 이해하고, 가능한 해결책들을 개발할 수 있게 해준다. 그 다음, 우리는 컴퓨터가, 인간이 또는 이 둘 다 이해 할 수 있는 해결책들을 제시할 수 있다.

05. 컴퓨팅사고의 개념

■ 4차산업혁명과 컴퓨팅사고

- '제4차 산업혁명' 용어는 2016년 세계경제포럼에서 언급되었으며, 정보통신 기술 기반의 새로운 산업시대를 대표하는 용어



+ 지능정보기술



05. 컴퓨팅사고의 개념

■ 컴퓨팅사고의 구성

- 컴퓨팅 사고는 복잡한 문제를 컴퓨터가 효과적으로 처리할 수 있는 단위로 **분해 decomposition**, 문제 간 유사성을 찾는 **패턴 인식 pattern recognition**, 문제의 핵심만 추려 복잡한 문제를 단순화하는 **추상화 abstraction**, 일련의 규칙과 절차에 따라 문제를 해결하는 **알고리즘 Algorithm** 기법 등을 사용하여 사람과 컴퓨터 모두가 문제를 처리할 수 있는 형태로 표현한다.
- CSTA : Computer Science Teachers Association의 컴퓨팅 사고 특성 정리



05. 컴퓨팅사고의 개념

■ 컴퓨팅사고의 구성 요소

- 분해 : 주어진 문제를 해결하기 쉬운 작은 단위의 문제로 나누는 것.
- 추상화 : 문제에서 불필요한 부분을 제거하고 중요한 특징만 남겨 단순화시키는 것.
- 패턴인식 : 주어진 데이터를 특징별로 분류하여 의미 있는 패턴이 있는지 찾는 것.
- 알고리즘 : 주어진 문제를 해결하기 위한 일련의 절차나 방법을 공식화한 형태로 표현하는 것.



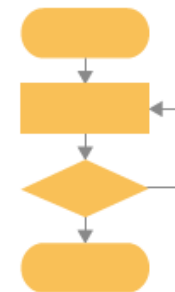
+ 분해



+ 추상화



+ 패턴인식



+ 알고리즘

05. 컴퓨팅사고의 개념

■ 분해 단계

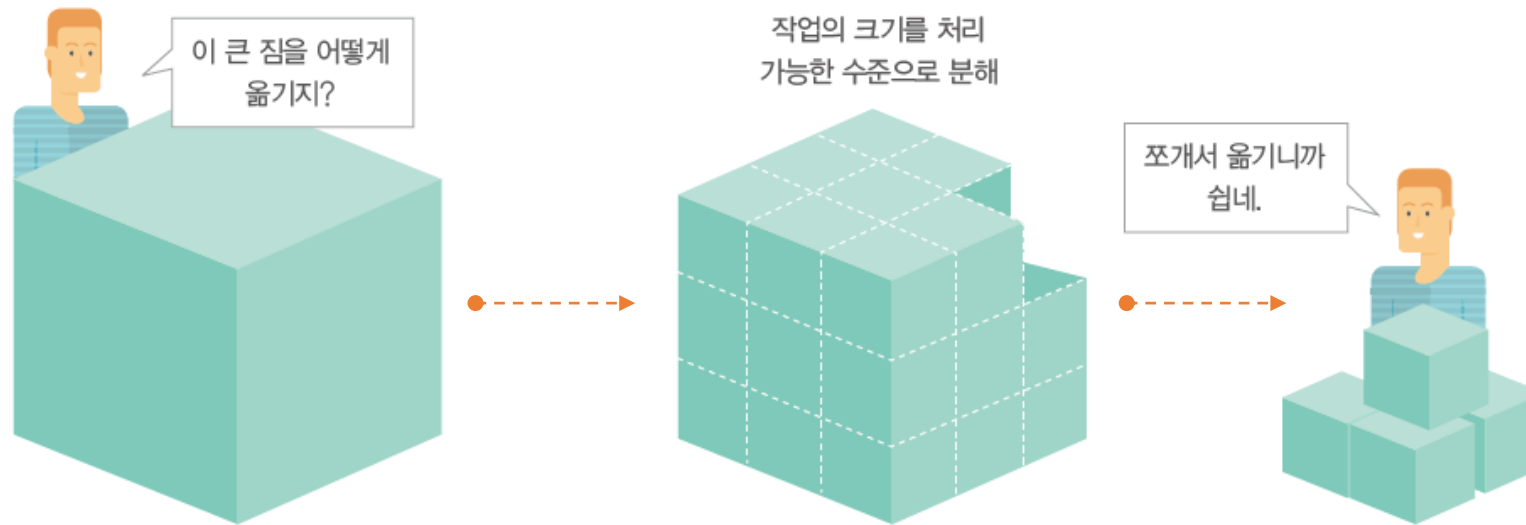
- 분해(decomposition)는 문제를 해결하기 쉬운 작은 단위의 문제로 나누는 것.



05. 컴퓨팅사고의 개념

■ 분해의 예

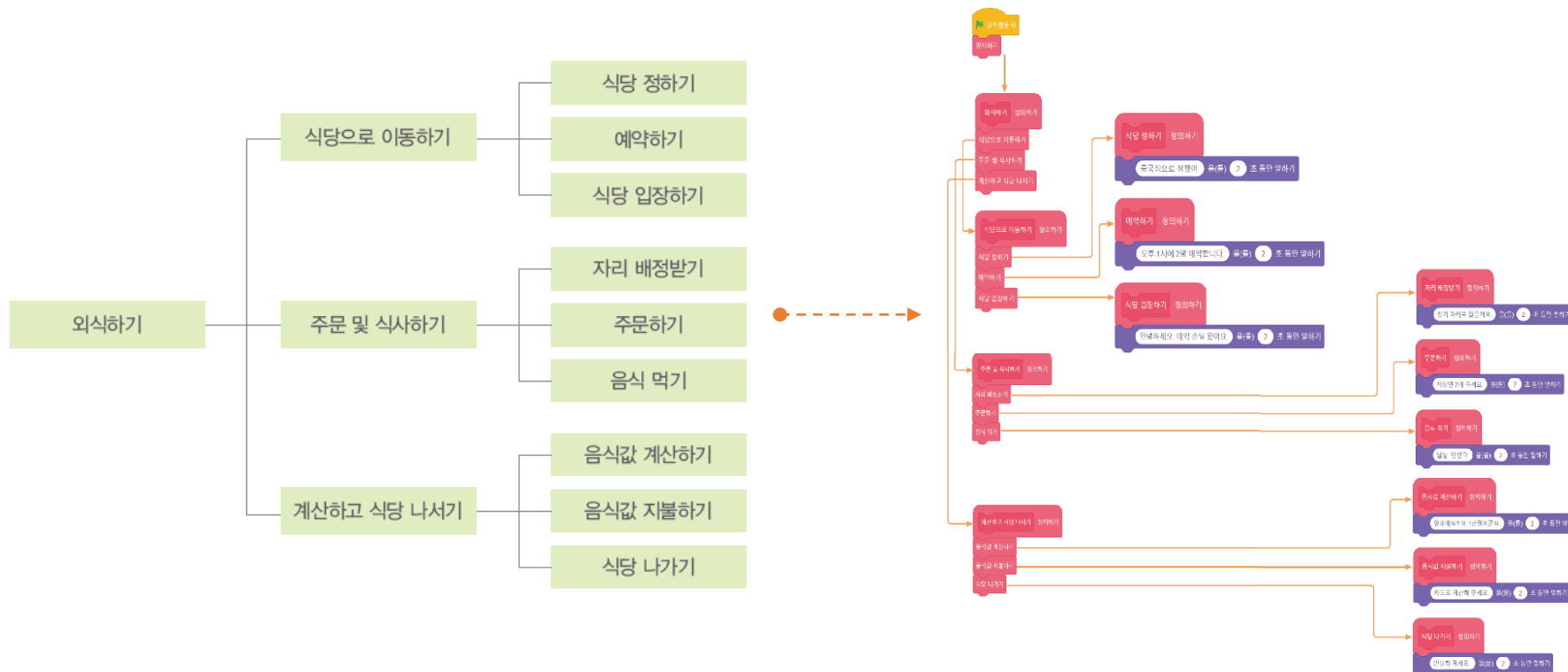
- 컴퓨팅 사고에서 말하는 분해란 큰 규모의 복잡한 작업을 **처리 가능한 수준의 작업 단위가 될 때까지 계속 작게 쪼개는 작업**을 의미한다.



05. 컴퓨팅사고의 개념

■ 일상생활에서의 분해

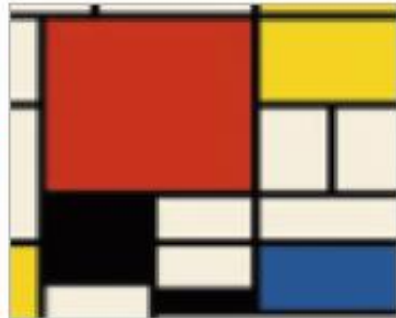
- 분해 작업을 할 때 큰 단위에서 작은 단위로 세분화하는 것을 **하향식 설계방식**이라고 한다.
- 보통 하향식 설계로 분해된 작업은 계층적 구조를 띠기 때문에 **계층적 분해**라고도 한다.



05. 컴퓨팅사고의 개념

■ 추상화 단계

- 추상화(abstraction)는 여러 사물이나 개념에서 공통된 속성들을 추출하여 보다 일반화된 개념을 만들거나 복잡한 대상을 문제 해결에 필요한 속성들만으로 간단하게 정리하는 개념.

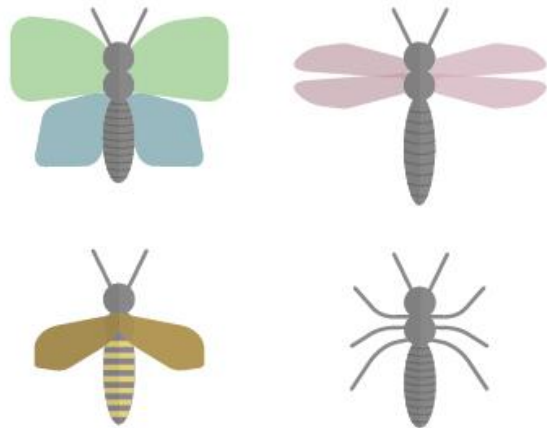


세상을 면과 색으로만 바라볼까?

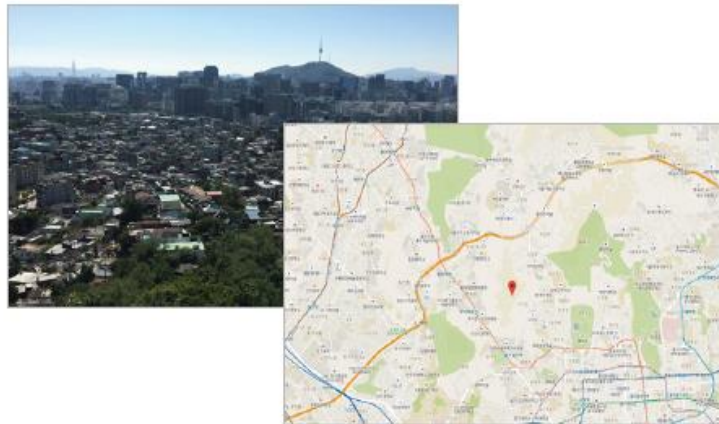
05. 컴퓨팅사고의 개념

■ 추상화의 예

- 여러 사물이나 개념에서 공통되는 특성이나 속성을 추출하여 보다 일반화된 개념을 만들어내는 유형, 일반화(generalization).
- 사물을 표현할 때 불필요한 요소를 없애거나 감추고 관심 있는 것들만 표현하는 유형.



+ 유형1 : 절지동물의 모습을 일반화

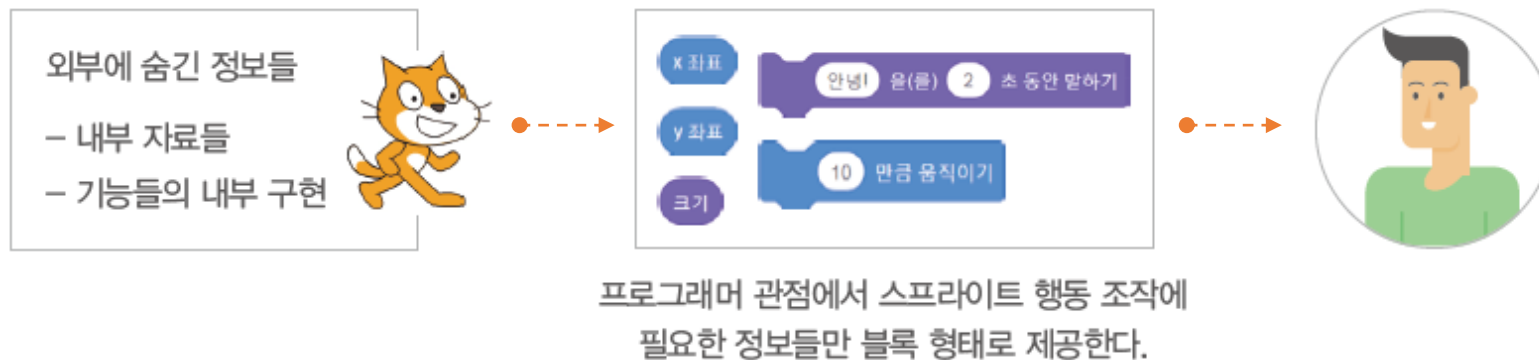


+ 유형2 : 필요요소 들로만 표현

05. 컴퓨팅사고의 개념

■ 스크래치에서 추상화의 예

- 스크래치는 스프라이트를 조작하는 데 필요한 블록들을 프로그래머에게 제공하지만 그 외 스프라이트의 내부 모습은 볼 수 없도록 되어 있다.



05. 컴퓨팅사고의 개념

■ 패턴 인식 단계

- 패턴인식(pattern recognition)은 주어진 사물을 특징별로 나누어 그 속에서 의미 있는 패턴을 찾는 과정

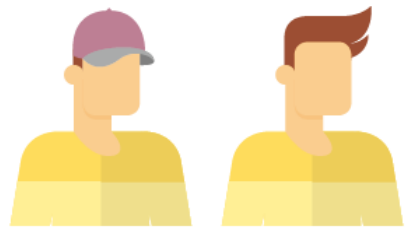
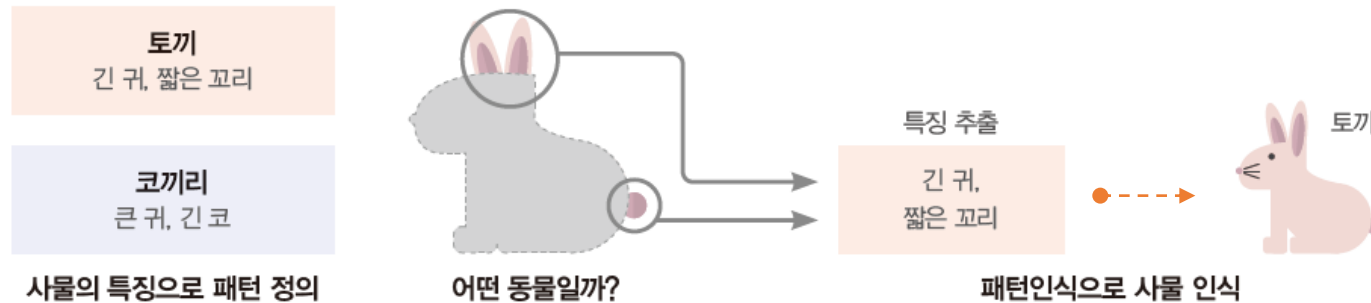


수직 이착륙 비행기 개발에 대나무
프로펠러 원리를 이용할 수 있겠구나!

05. 컴퓨팅사고의 개념

■ 일상생활에서 패턴 인식의 예

- 패턴인식(pattern recognition)은 사물을 구별할 수 있는 특징을 모아 패턴을 정의하고 어떤 사물이 어떤 패턴에 속하는지 결정하는 단계
- 사물 인식이나 현상 예측시스템(기상예측) 등에 널리 활용.



모자와 상관없이 같은 사람으로 인지함



디자인과 상관없이 의자라고 인지함

05. 컴퓨팅사고의 개념

■ 알고리즘 단계

- 알고리즘(algorithm)은 어떤 문제를 해결하기 위한 **절차나 방법**을 의미 : 공식화
- 알고리즘은 **입력, 출력, 명확성, 유한성, 효율성**이라는 성질을 만족해야 하며, 일반적으로 **자연어**(natural language), **의사 코드**(pseudo code)와 **순서도**(flow chart)로 표현한다.
 - **입력** : 외부에서 제공되는 자료가 0개 이상 존재해야 한다.
 - **출력** : 적어도 1개 이상의 결과를 출력해야 한다.
 - **명확성** : 수행과정은 명확하고 모호하지 않은 명령어로 구성되어야 한다.
 - **유한성** : 유한 번의 명령어를 수행하고, 유한 시간 내에 종료되어야 한다.
 - **효율성** : 모든 과정은 명백하게 실행 가능하고, 검증 가능해야 한다.
- 최초의 알고리즘은 B.C. 300년경 **유클리드의 최대 공약수**를 구하는 방법으로 알려져 있다.

두 정수 a, b 의 최대공약수 구하기 알고리즘 (가정 : 두 정수 a 와 b 는 0 이상의 값이고 a 가 b 보다 크거나 같다.)

1단계 : b 가 0이면 a 를 최대공약수로 정하고 알고리즘을 종료한다.

2단계 : b 가 0이 아니면 a 에서 b 를 빼고 그 결과값을 c 로 정한다.

3단계 : b 와 c 중 큰 값을 a 로, 작은 값을 b 로 정하여 위 작업을 1단계부터 다시 수행한다.



CHAPTER 06

소프트웨어의 개념과 블록코딩

06. 소프트웨어의 개념과 블록코딩

■ 소프트웨어의 개념

- 컴퓨터를 움직이게 하는 중요 요소로, **하드웨어의 상대적 개념**이다.
- 소프트웨어는 많은 프로그래밍 언어로 구현된 컴퓨터가 이해할 수 있는 명령어들의 집합 프로그램을 지칭한다.
- 컴퓨터 또는 스마트폰에서 제공되는 모든 기능은 소프트웨어로 제공된다.
- 우리는 많은 문제해결의 방법으로 **알고리즘화**한 내용을 프로그래밍을 통하여 표현하고 테스트하여 구현한다.
- 소프트웨어에는 시스템 프로그램, 운영체제와 응용프로그램 등이 있다.



06. 소프트웨어의 개념과 블록코딩

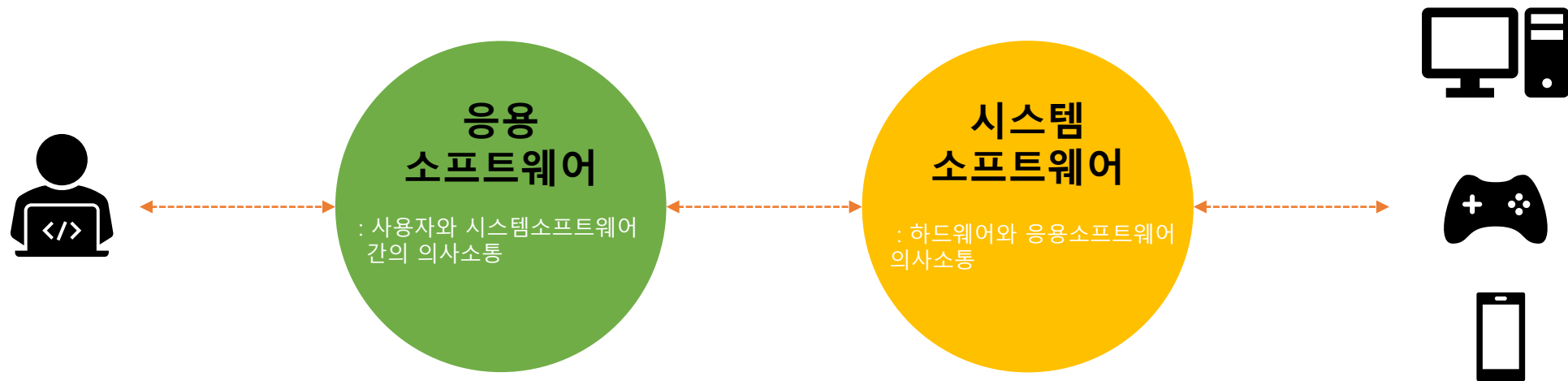
■ 하드웨어와 소프트웨어의 개념



- 하드웨어 : 컴퓨터를 구성하는 모든 기계 장치를 통틀어 하드웨어라고 함
- 소프트웨어 : 소프트웨어는 하드웨어의 상대적 개념으로 구체적인 사물이 아니며, 하드웨어 속에 내장된 프로그램을 의미

06. 소프트웨어의 개념과 블록코딩

■ 소프트웨어의 종류



- 소프트웨어는 크게 **응용소프트웨어**와 **시스템 소프트웨어**로 분류됨
- 시스템 소프트웨어에는 윈도우, IOS, 안드로이드OS 등과 같이 하드웨어를 제어하는 프로그램을 말함
- 응용 소프트웨어는 게임, 그래픽툴, 문서편집도구 등과 같이 사용자와 직접적인 커뮤니케이션을 하는 프로그램을 말함

■ 넓은 의미의 소프트웨어 개념

넓은 의미의 소프트웨어

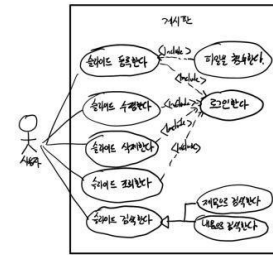
일반적인 의미의 소프트웨어

```

1 string input;
2 int length, in;
3 double dblLen;
4 bool again = true;
5
6 while (again) {
7     in = -1;
8     again = false;
9     getline(cin, input);
10    system("cls");
11    stringstream(input) >> dblLen;
12    stringLen = input.length();
13    if (length < 4) {
14        again = true;
15        continue;
16    } else if (stringLen - 3 != ".") {
17        again = true;
18        continue;
19    } while (++in < iLength) {
20        if (isdigit(input[in])) {
21            continue;
22        } if (in == (length - 3)) {
23
24
25
26
27
28
29
30
31
32
33
34
35

```

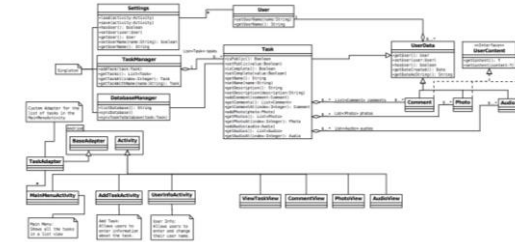
프로그램



- 사용자 요구분석

[illegible]

- 테스트 문서



- 프로그램 설계 문서



- 사용자 가이드

- 프로그램 개발과 관련된 문서, 그래픽디자인, 기획 및 설계 단계의 아이디어 까지도 소프트웨어의 개념에 포함됨
- 소프트웨어의 가장 대표적인 예가 프로그램이기 때문에 소프트웨어와 프로그램을 동일한 개념으로 사용

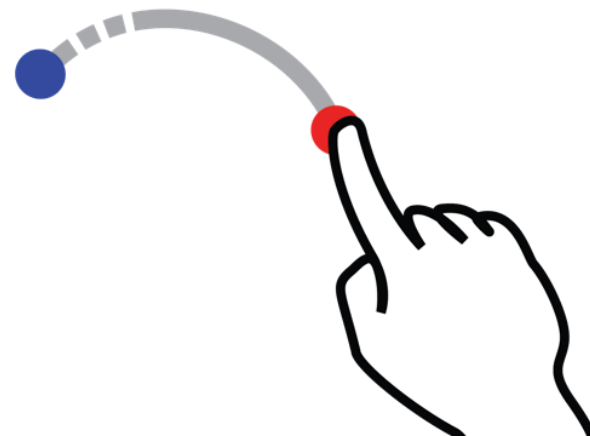
06. 소프트웨어의 개념과 블록코딩

■ 다양한 블록코딩 프로그램

- 스크래치 : <https://scratch.mit.edu/>
- 엠블록 : <https://www.makeblock.com/>
- 엔트리 : <https://playentry.org>
- 앱인벤터 : <http://www.appinventor.mit.edu/>

■ 블록코딩 프로그램의 장점

- 드래그 앤 드롭 방식으로 스크립트(코딩)를 할 수 있다.
- 블록 모양으로 스크립트(코드)가 작성되어 있어 문법이 직관적이다.
- 문제해결을 하기위한 사고를 스크립트로 표현하기에 용이하다.
- 스크립트의 전체적인 문법적인 구조를 해석하기가 쉽다.
- 단순한 조합만으로 프로그램을 만들 수 있다.

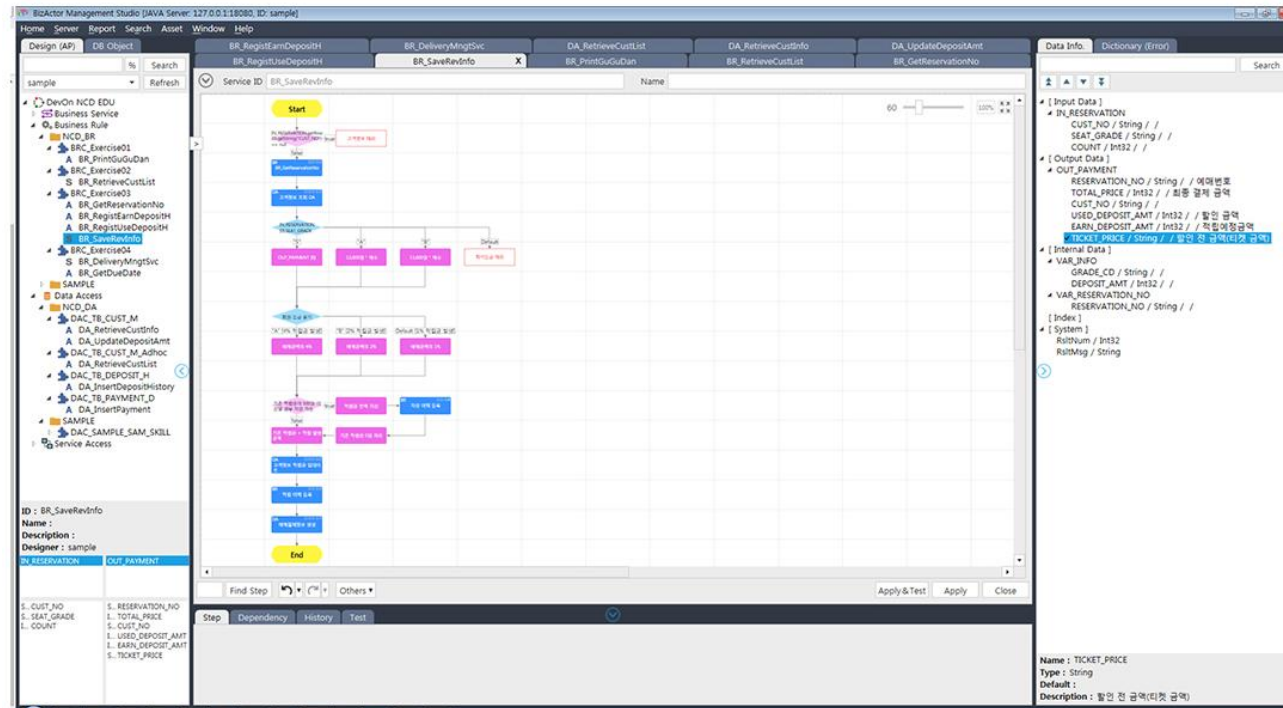


“ 어려운 컴퓨터 언어와 문법을 몰라도 컴퓨팅 사고력 중심의 효과적인 학습이 가능하다. ”

06. 소프트웨어의 개념과 블록코딩

■ 블록 코딩의 이슈

- 블록 코딩은 프로그램 입문자나 초등학생들을 위한 것이 아니다.
- 현재 중요 SW 개발이 블록 코딩들의 자동화 코딩 플랫폼이 사용되고 있다.
- LG CNS의 DevOn NCD : <https://www.lgcns.com/Solution/DevOn-NCD>



CHAPTER 07

인공지능 시대의 소프트웨어

07. 인공지능 시대의 소프트웨어

- **인공 지능 기술은 다양한 분야에 적용되고 있다.**
 - 로봇산업, 헬스케어, 자율 주행 자동차, 인공지능 교육, 스마트 그리드, 가전, 게임, 스마트 공장, 스마트 농장 뿐만 아니라 창작의 영역까지 그 범위가 확대되고 있다.
- **자동차의 혁명 : 벤츠의 AI 서비스 살펴보기**



그 버튼에 대해 설명합니다

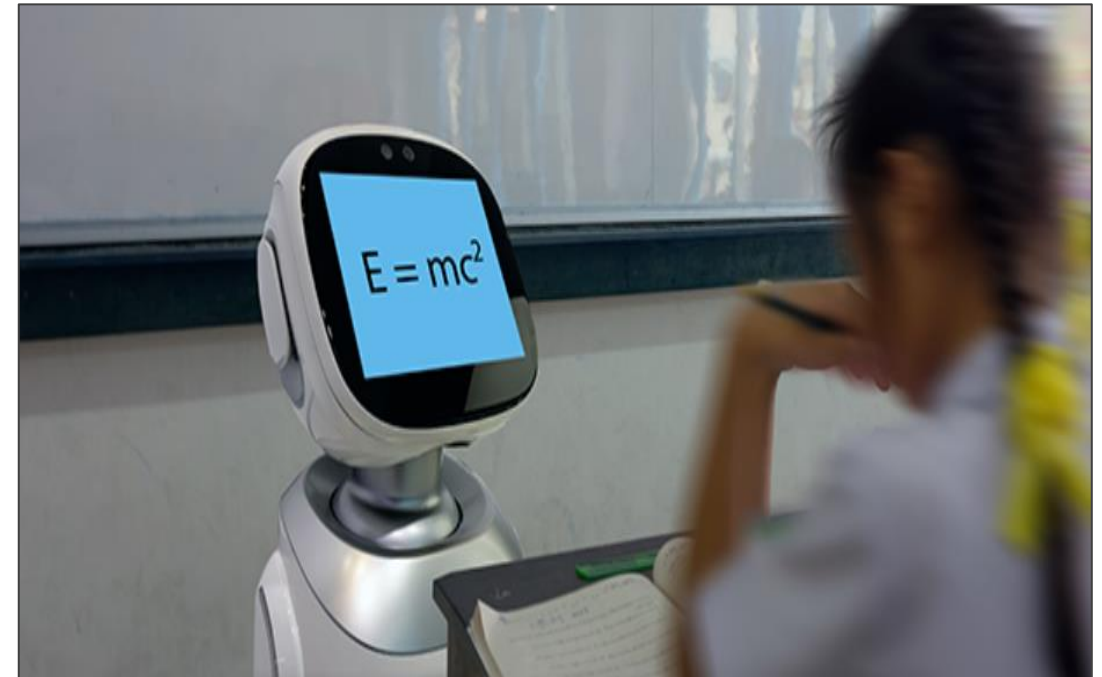
차량 취급 설명서
스마트 폰의 앱에서 자동차의 버튼을
인식하고 기능을 설명한다.

07. 인공지능 시대의 소프트웨어

■ 소프트뱅크 AI 로봇 페퍼



환자의 건강 상태를 상담하는 로봇
현대인의 우울증 전조 증세 탐지를 위한 연구 진행



2030 미래에는 로봇 교사가 교육을 담당

07. 인공지능 시대의 소프트웨어

■ Teachable Machine 사용하기

- 인공지능 서비스를 위한 용어들
 - 학습 : 컴퓨터가 공부를 하는 과정 (데이터를 인식)
 - 모델 : 학습과정을 수행 한 결과물
 - 추론 : 학습한 결과를 확인 하는 과정
 - 기계학습(머신러닝, 딥러닝)을 도와주는 SW (ex : tensorflow)

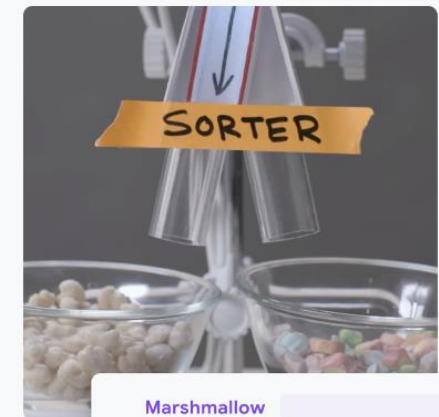
<https://teachablemachine.withgoogle.com/>

Teachable Machine

이미지, 사운드, 자세를 인식하도록 컴퓨터를 학습시키세요.

사이트, 앱 등에 사용할 수 있는 머신러닝 모델을 쉽고 빠르게 만들어 보세요. 전문지식이나 코딩 능력이 필요하지 않습니다.

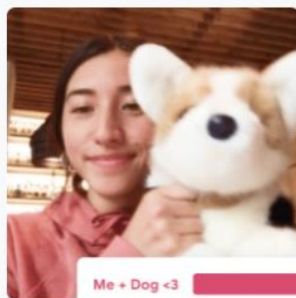
시작하기



07. 인공지능 시대의 소프트웨어

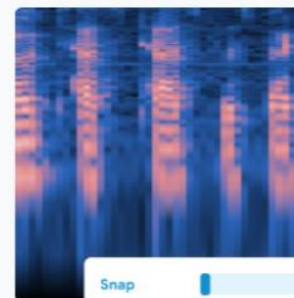
■ Teachable Machine으로 무엇을 학습시킬 수 있을까?

- 이미지 또는 영상
- 음악, 소리와 같은 사운드
- 인체 동작



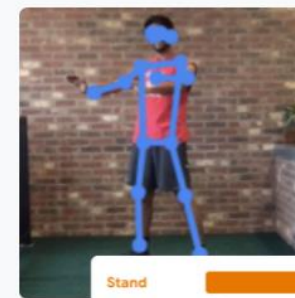
이미지

파일 또는 웹캠을 사용해 이미지를 분류하는 방법을 모델에 학습시킵니다.



사운드

짧은 사운드 샘플을 녹음하여 오디오를 분류하도록 모델을 학습시키세요.



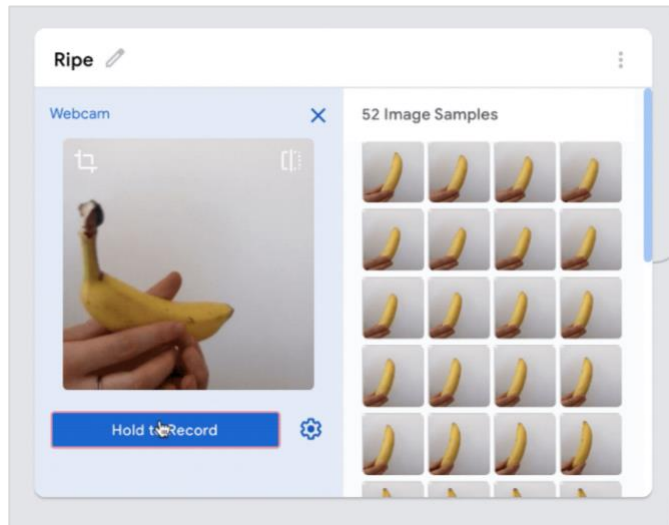
자세

파일을 사용하거나 웹캠에서 자세를 취하여 몸의 자세를 분류하도록 모델을 학습시키세요.

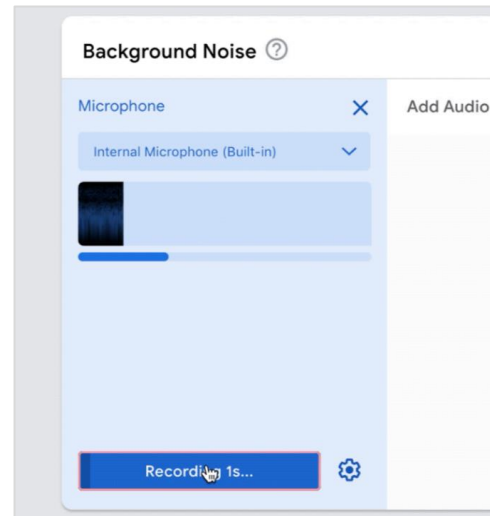
07. 인공지능 시대의 소프트웨어

■ Teachable Machine 사용법 1단계

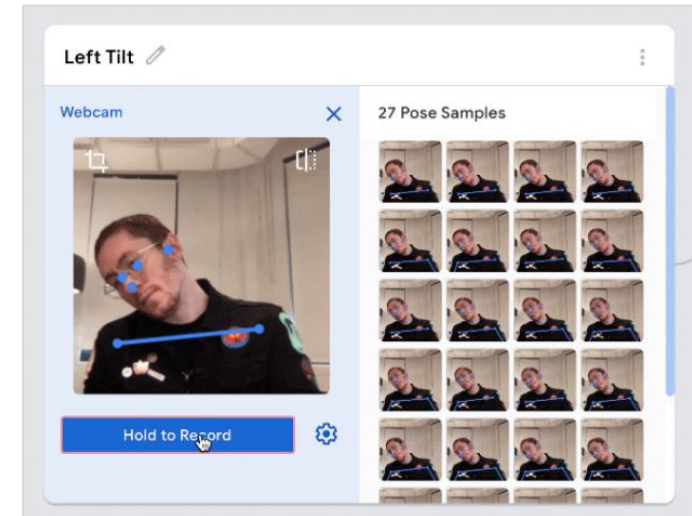
- 데이터 수집 과정 수행
 - 컴퓨터에 클래스를 정해 놓고, 그 클래스에 해당될 수 있는 예시들(데이터)을 제공 하고 학습하도록 한다.
 - 현재 서비스가 지원하는 데이터는 이미지/소리/자세 이다.
 - 미리 저장해둔 사진/음성파일을 첨부할 수도 있고, 카메라 또는 마이크를 통해서 캡처 방식으로 데이터를 생성할 수 있다.



+ 이미지



+ 소리



+ 동작

07. 인공지능 시대의 소프트웨어

■ Teachable Machine 사용법 2단계

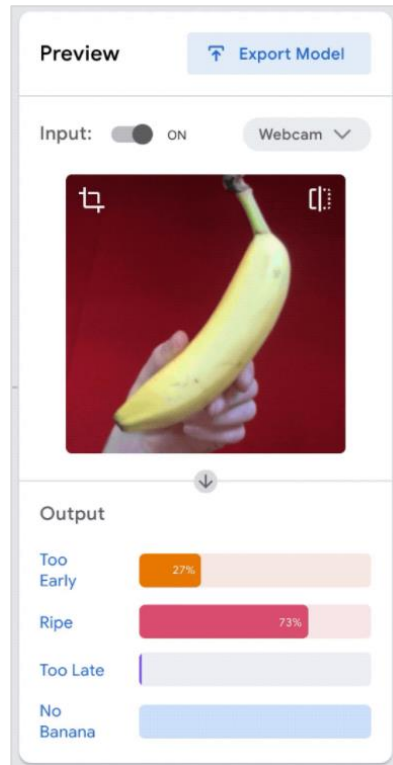
- 학습 과정 수행
 - 컴퓨터(모델)이 제시된 클래스와 예시들을 통해 학습하는 과정을 진행.
 - 학습된 컴퓨터(모델)가 정답을 맞추지 못한다면, 다양한 피드백을 반영. (새로운 예시들, 학습규칙 등)



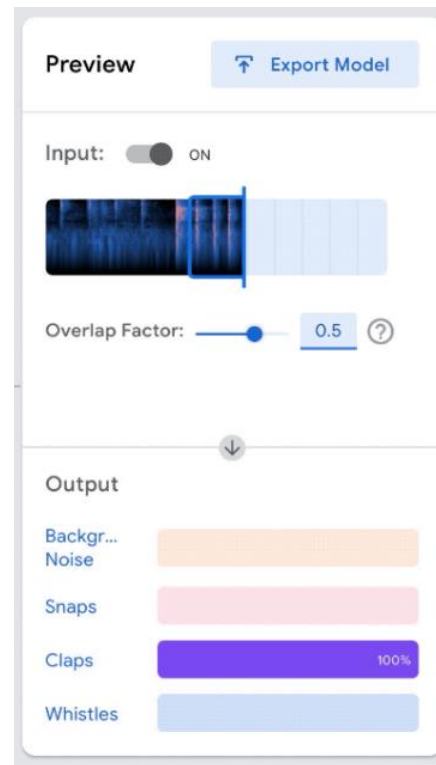
07. 인공지능 시대의 소프트웨어

■ Teachable Machine 사용법 3단계

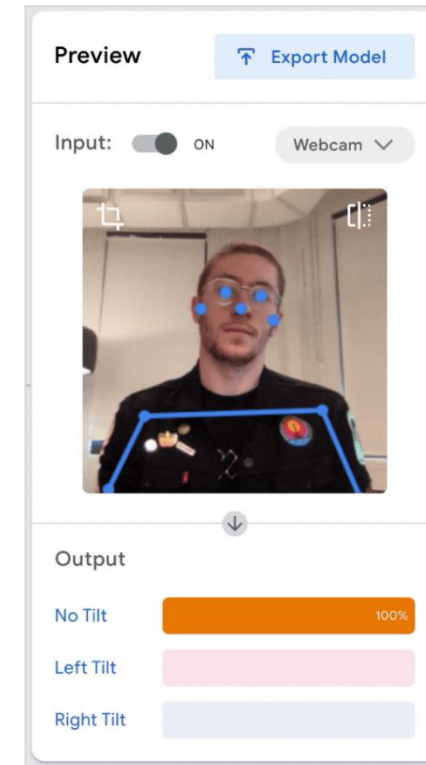
- 인식과정 (또는 Export Model 과정) 수행
 - 학습된 인공지능이 의도한 대로 새로운 데이터를 분류하는지 알아보기 위해 웹캠이나 기타 업로드의 방식으로 데이터를 입력하여 결과를 확인한다.



+ 이미지



+ 소리



+ 동작