

# 알고리즘사고 기반의 SW기초교육 과정 개발



단국대학교 SW중심대학사업단  
서응교

[eungkyosuh@dankook.ac.kr](mailto:eungkyosuh@dankook.ac.kr)



지능 한국의 미래는,  
檀檀[단단]한 SW교육혁신에 있다.

4차 산업혁명 지능정보사회를 리드하는  
글로벌 SW인재허브 실현

# 목차

## 1. 단국대학교 SW중심대학

## 2. SW교육과 ADD Thinking

- SW교육과 ADD Thinking
- DAD Processing

## 3. 대학기초SW입문

- 단국대학교 SW필수교양
- 대학기초 SW입문
- 대학기초 SW입문 - 강의안 예시

## 4. 단국대학교 SW기초 교육 대표 성과



# 1. 단국대학교 SW중심대학

# 단국대학교 SW중심대학

비전

지능 한국의 미래, **檀檀[단단]한 SW교육 혁신에 있다.**  
4차 산업혁명 지능정보사회를 리드하는 글로벌 SW인재허브 실현

목적



SW중심화



혁신적 SW교육프로그램 운영



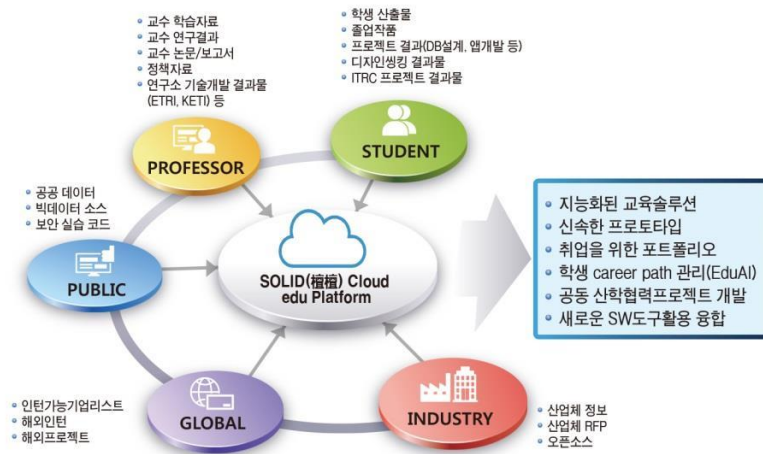
글로벌 인재 양성

SW교육  
혁신 방안

**ADD Thinking**  
"Algorithmic Thinking + Design Thinking  
+ Deep Learning Thinking"

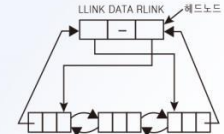
**POSE**  
(Project, OpenSource, English)

SOLID Cloud edu Platform



Algorithmic Thinking

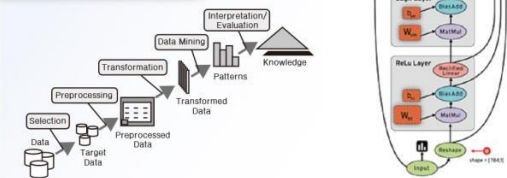
```
1. index := 1;
2. while index ≤ n and L[index] ≠ x do
3.   index := index + 1
4. end { while };
5. if index > n then index := 0 end
```



Design Thinking



Deep Learning Thinking



대학 · 기업 합동 문제 해결형  
SW교과과정 개발

**Project**

- 1인 1신작프로젝트 의무화
- 산업체 요구를 반영하여 프로젝트 진행

**Open Source**

- 기업이 사용하는 실무중심의 오픈소스 활용 교육강화

**English**

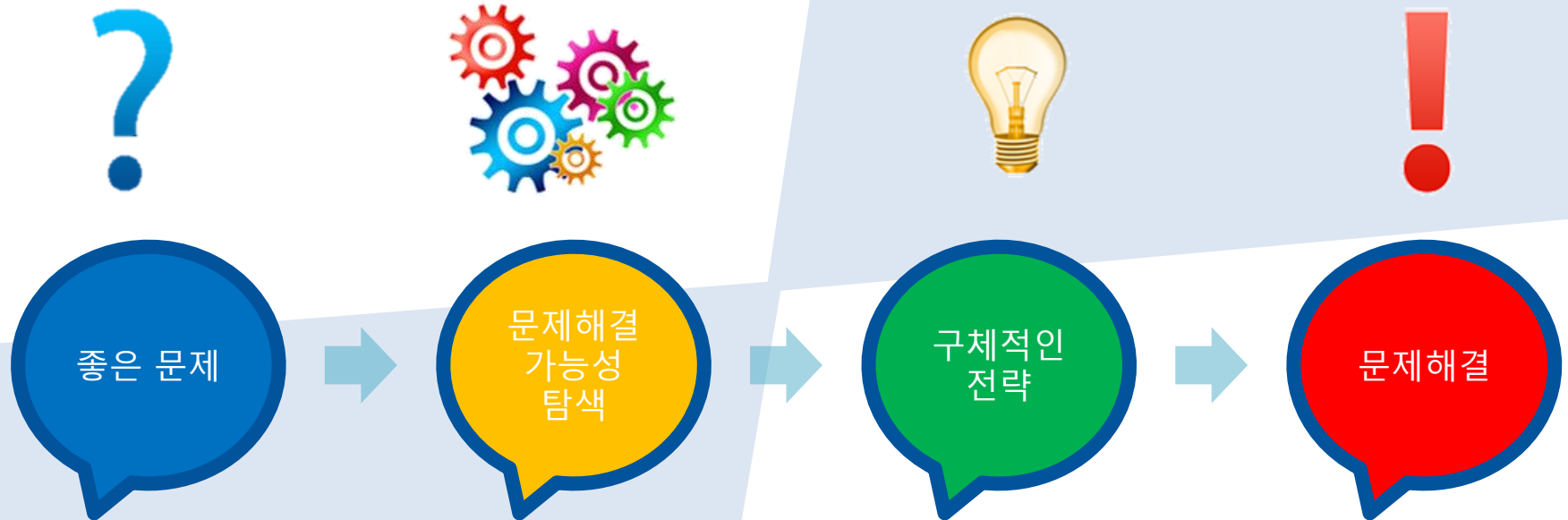
- 공학도의 글로벌 역량과 산업체 업무수행을 위한 실무영어 교육강화



단국대학교  
SW중심대학

## 2. SW교육과 ADD Thinking

소프트웨어 교육 = 창의적인 **문제 해결력**



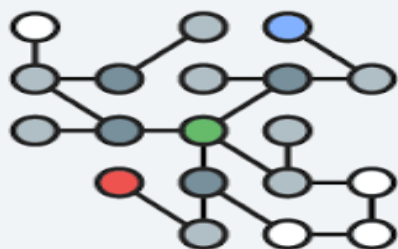
# SW교육과 ADD Thinking

# 창의적 소프트웨어 개발을 위한 ADD Thinking

**ADD Thinking 교육실현으로 컴퓨터과학 기본 개념과 원리, 컴퓨팅 기술을 바탕으로 실생활 및 다양한 학문 분야의 문제를 창의적이고 효율적으로 해결하는 능력과 태도의 함양.**

# Algorithm Thinking

## 컴퓨팅사고기반의 논리적 사고



## ALGORITHM

# Design Thinking

사용자의 불편함에 공감하고  
문제를 재 정의하는 생산자  
의 역할을 경험하는 사고



# Deep Learning Thinking

다양한 아이디어를 결합함  
으로 창의적으로 문제를 해  
결할 수 있는 **확장적 사고**





# SW교육과 ADD Thinking

## ADD Thinking 실현을 위한 DAD Processing

**D**esign Thinking

창의적인 문제 해결책 제시



**A**lgorithm Thinking

컴퓨터처럼 생각하는 방법

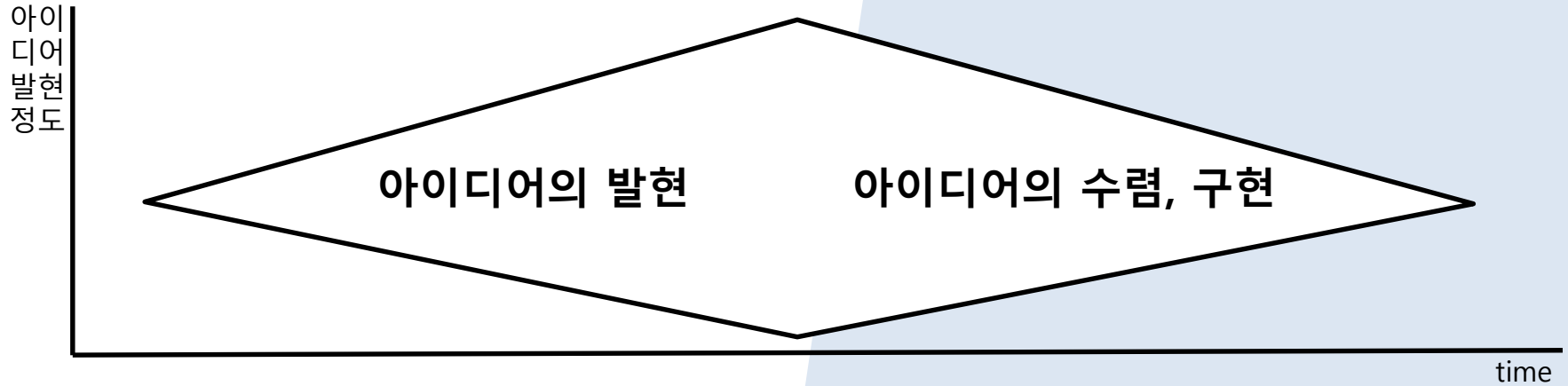


**D**eep Learning Thinking

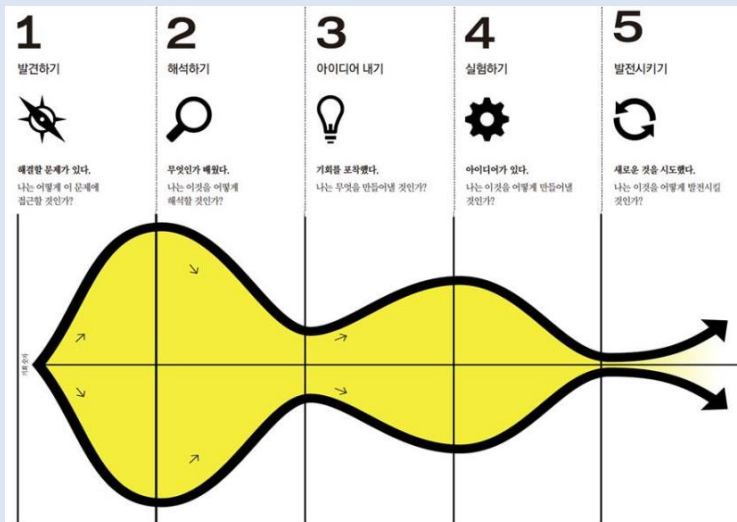
다양한 아이디어를 결합



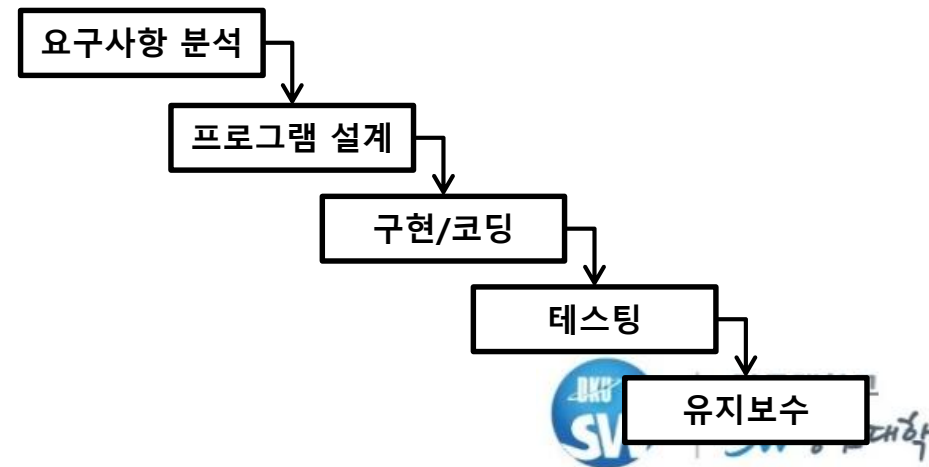
# SW교육과 ADD Thinking



## 창의적 아이디어의 발산 : Design Thinking



## 논리적 아이디어의 구현 : Computer Coding



## 1. Design Thinking

- ✓ 현재 상태 : 실내 놀이터에 아이들이 없다.
- ✓ 목표 상태 : 실내놀이터를 활성화 하도록 한다.
- ✓ 불편사항
  - ✓ 실내 놀이터는 아이들만 이용하기 때문에 부모나 청소년은 함께 놀 수 없어 불편하다.
  - ✓ 보호자의 입장에서는 실내놀이터는 아이들을 돌봐야 한다는 생각때문에 피곤함이 몰려온다.

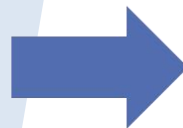
연령대에 상관없이 놀 수 있는 공간을 만든다.  
어린이/청소년/ 어른이 놀 수 있는 공간을 배치한다.



## 2. Algorithm Thinking

- ✓ 문제의 원인 : 모든 연령대가 이용하기 때문에 복잡하다
- ✓ 문제 해결을 위한 조건 제시 : 인원수 제한/ 2시간 체험 시간 주기
- ✓ 해결방법
  - ✓ 입장 인원을 실시간으로 카운팅해서 보여주기

티켓을 판매할 때마다 카운팅 할 수 있는 알고리즘을 만들어 프로그램 개발

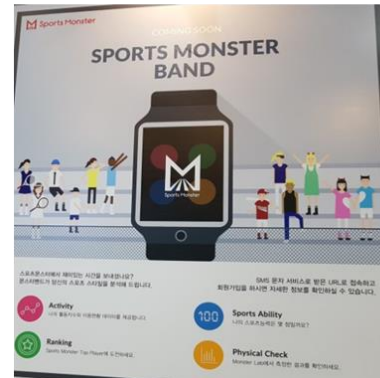


## 3. Deep Learning Thinking

- ✓ 어린이와 청소년과 어른이 다 같이 놀 수 있는 체험 공간에 좀 더 재미있는 아이디어 결합

스마트 밴드 : 출입 정보 + SNS + 바이오 정보

*‘이 몬스터 밴드는 운동량을 측정해주는 기능을 갖췄다. 퇴장 전 몬스터 밴드 반납 시 핸드폰번호를 남기면 전송된 SMS를 통해 자신의 운동량과 활동지수를 확인할 수 있다.’*



## 2. 단국대학교 SW기초 교육 내용 개발



# 단국대학교 SW필수교양

- 전공별 사회에서 요구되는 **SW역량에 따라 6개 그룹으로 분류**하여 SW언어와 수준을 차별화하여 교육함

- 비전공자 SW기초교육 요구 도출

## 전문가 간담회

- ▶ SW교육센터 SW교육 전문가(학교, 업체 등) 간담회 개최를 통한 의견수렴
- ▶ 전문가 집단 델파이분석을 통한 교육목표, 프로그래밍 언어, 교수법 등을 도출

## 문헌 조사 / 의견 조사

- ▶ Stack Overflow, Quora, ResearchGate, CoderZ 검색을 통한 전공별 졸업생 직업 그룹이 필요로 하는 컴퓨터 프로그래밍 언어 도출
- ▶ 각 단과대학별 교양필수수업 프로그래밍 언어 및 커리큘럼 작성 요청(공문)
- ▶ 해당 학과 교수 의견 수렴(교무위원회)

계열	인문(HLL)	사회(RPL)	예체능(ART)	자연(SCI)	공학(ENG)	SW
단과대학 (죽전)	문과대학, 법과대학, 사범대학	사회과학대학, 상경대학, 국제대학	예술디자인대학, 음악대학	해당 없음	건축대학, 공과대학	SW융합대학 (소프트웨어학과, 응용컴퓨터공학과, 모바일시스템공학과)
단과대학 (천안)	인문대학, 외국어대학	공공인재대학, 보건과학대학	예술대학, 스포츠과학대학	자연과학대학, 생명자원과학대학, 의과대학, 치과대학, 간호대학	융합기술대학	해당 없음

## 비이공계

## 이공계

2017년 14개 단과대학  
2,612명 수강

2018년 전체전공 22개 단과대학  
8,238명 수강

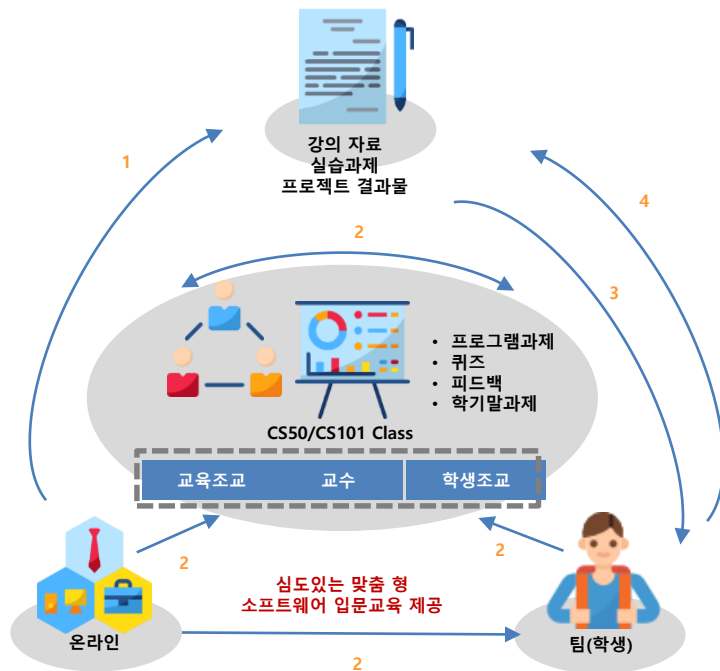
비SW전공

SW전공

단국대학교

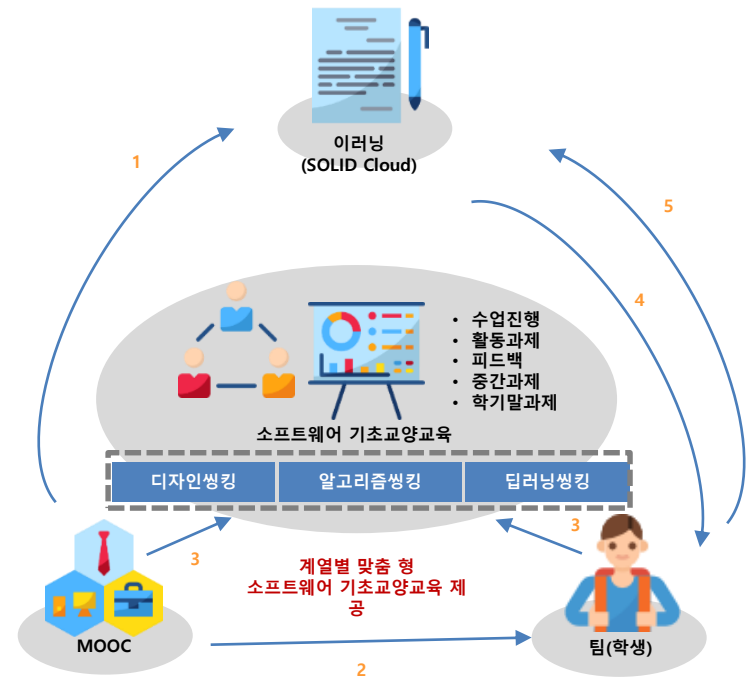
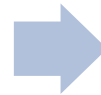
중심대학

## CS50/CS101 기반의 소프트웨어 기초교양교육 모델 개발



**CS50**

- ▶ 현실에 밀접한 문제해결기반 강의
- ▶ 모든 자료 온라인화
- ▶ 실력별 차등 과제



**소프트웨어 기초 교양교육**

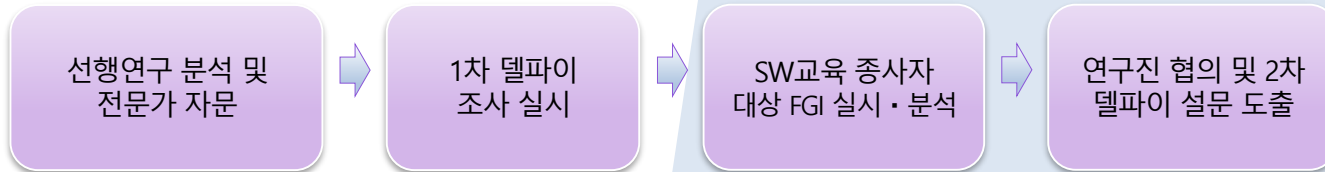
- ▶ 현실에 밀접한 문제해결기반 강의
- ▶ SOLID Cloud기반의 교육자료 온라인화
  - 계열별 맞춤형 자료 제공
  - 학습결과물 공유
- ▶ 계열별 차등 과제



# 단국대학교 SW필수교양



전공별 특성에 맞도록 전공영역별로 다양한 프로그래밍 언어를 학생



## • 비전공자 SW기초교육 운영 실적

과목	학점	교육내용	학기	분반수	실적
창의적사고와 코딩	2	SW비전공 계열학생들이 창의적 사고를 코딩을 통해 구현	1/2	115	4,159명(죽전 : 2,209명, 천안 : 1,950명)
대학기초SW입문	2	계열학생들에 필요한 알고리즘사고를 코딩을 통해서 구현	2	120	4,079명(죽전 : 1,919명, 천안 : 2,160명)
합계					8,238명

SW 의무 교육을 필수 이수하도록 하여 4차 산업혁명시대에 맞추어 단순 코딩교육이 아닌

**「창의적·실제적 문제해결」**을 모토로 코딩을 통해 문제해결 역량제고를 목표로 실시

# 대학기초 SW입문

## 알고리즘씹킹 기반의 계열별 [대학기초SW입문] 프로그래밍 언어 선정 결과

### 인문계열: App Inventor

- 인문학생들은 직관적으로 이해할 수 있으며, 실생활에서 활용이 가능한 앱을 중심으로 수업을 진행하는 것이 집중이 높음
- GUI형 프로그래밍 언어는 인문계 전공자가 알고리즘을 표현하기에 적합한 언어
- Blockly보다 실제 사용 및 흥미를 줄 수 있는 앱인벤터가 학습에 도움이 됨

### 사회계열: App Inventor

- 사회과학은 인문계열보다는 데이터를 다룰 줄 알아야 하나 저학년 일수록 생활밀착형 툴이 좋음
- 코드형 프로그래밍 언어는 일부 사회과학 전공자만 유용하다고 판단되어 척도가 낮음.
- Blockly보다 실제 사용 및 흥미를 줄 수 있는 앱인벤터가 학습에 도움이 됨

### 자연계열: Python

- 자연과학은 데이터를 다루기 위한 텍스트 언어나 실생활 밀접한 앱에 관한 내용이 적합함. 또한, 실험 데이터와 휴대폰과 연결이 가능함
- 코드형 프로그래밍 언어는 일부 자연계 전공자만 유용하다고 판단되어 척도가 낮음.
- 자연계열은 전공에 대한 응용까지 고려한다면 파이썬이 필요하다고 생각함.

### 공학계열: Python

- 공학 기초인 C언어나 최근 각광을 받는 파이썬이 좋음
- 공학계 전공자라도 코드형 프로그래밍 언어를 쉽게 이해하기 어려울 수 있고 입문 과정에서는 적절하지 않다고 판단됨
- 블록기반언어는 전공 연계가 어려움.
- 실제 해보면 러플이 파이썬보다 더 복잡함
- 공학계열에는 러플보다는 파이썬을 배우는 것이 더 활용도가 높음
- JavaScript는 웹 프로그래밍할 때 주로 사용하는 언어로 창의적인 알고리즘을 작성하는 방법을 배우는 언어로는 적절해 보이지 않음.

### 예체능계열: Scratch

- 예체능 학생들 중 음악과 미술 계열은 음과 미디어 아트 등을 주제로 하는 스크래치 류의 툴이 적합함.
- GUI형 프로그래밍 언어는 예체능 전공자가 알고리즘을 표현하기에 적합한 언어로 판단됨

## 인문계열: App Inventor

내용 영역	내용 요소	하위 요소
문제해결과 알고리즘	문제해결과정	일반적 문제해결과정
		컴퓨팅사고기반의 문제해결과정
	알고리즘과 절차	알고리즘 정의
		논리적 문제해결 절차
	알고리즘 특성 및 표현	알고리즘 특징과 표현
알고리즘과 문제 분해	문제 발견 및 정의	문제 발견
		문제 이해
		문제 정의
		문제 해결 계획 수립 (시간계획과 필요한 자원)
	문제 분석과 표현	현재 상태
		목표 상태
		(문제해결에 필요한) 요소 구별
		(시간과 자원과 같은) 조건
	문제 나누기	문제 분해의 특징
		자료 중심 분해
		절차 중심 분해
		(목표 상태에 도달하기 위한) 작업 나열
알고리즘과 추상화	패턴인식	공통 요소 찾기
	기능 추상화	핵심 기능 찾기 (문제 해결에 꼭 필요한 기능 찾기)
		논리적 절차

## 사회계열: App Inventor

내용 영역	내용 요소	하위 요소
문제해결 과 알고리즘	문제해결과정	일반적 문제해결과정
		컴퓨팅사고기반의 문제해결과정
	알고리즘 개념	알고리즘 정의
		논리적 문제해결 절차
	알고리즘 특성 및 표현	알고리즘 특징 알고리즘 기술 언어
알고리즘과 문제 분해	문제 발견 및 정의	문제 발견
		문제 이해
		문제 정의
		문제 해결 계획 수립 (시간계획과 필요한 자원)
	문제 분석	현재 상태
		목표 상태
		(문제해결에 필요한) 요소 구별
		(시간과 자원과 같은)조건
	문제 나누기	문제 분해의 특징
		자료 중심 분해
		절차 중심 분해
		작업 나열
알고리즘과 추상화	패턴인식	공통 요소 찾기
	기능 추상화	핵심 기능 찾기
		논리적 절차
	데이터 추상화	핵심 데이터 찾기
		데이터 추상화의 특징
		변수 데이터 타입
알고리즘과 프로그래밍	알고리즘 표현	알고리즘 표현 방법 알고리즘 설계 기법
	프로그래밍	프로그래밍 환경의 이해

## 자연계열: Python

내용 영역	내용 요소	하위 요소
문제해결과 알고리즘	문제해결과정	일반적 문제해결과정 컴퓨팅사고기반의 문제해결과정
	알고리즘과 절차	알고리즘 정의 논리적 문제해결 절차
	알고리즘 특성 및 표현	알고리즘 특징 알고리즘 기술 언어
		문제 발견/문제 이해 /문제 정의
알고리즘과 문제 분해	문제 발견 및 정의	문제 해결 계획 수립
	문제 분석	현재 상태 목표 상태 (문제해결에 필요한) 요소 구별 (시간과 자원과 같은)조건
		문제 분해의 특징
		자료 중심/절차 중심 분해
	문제 나누기	작업 나열
알고리즘과 추상화	패턴인식	공통 요소 찾기
	기능 추상화	핵심 기능 찾기 논리적 절차
	데이터 추상화	핵심 데이터 찾기 (컴퓨팅으로 구현하기 위한) 데이터 추상화의 특징 변수 데이터타입
알고리즘 설계	알고리즘 구조	순차/선택/반복구조 중첩제어구조
	알고리즘 표현	알고리즘 표현 방법 알고리즘 설계 기법 선형구조 비선형 구조
	논리	대입/산술/비교/논리연산 프로그래밍 환경의 이해 변수
알고리즘 구현	프로그래밍	제어 구조 구현 함수 구현 이벤트 구현
		모델과 시나리오
		시뮬레이션 구현
		시뮬레이션 실행 및 평가
	시뮬레이션	

## 공학계열: Python

내용 영역	내용 요소	하위 요소
문제해결과 알고리즘	문제해결과정	일반적 문제해결과정 컴퓨팅사고기반의 문제해결과정
	알고리즘 개념	알고리즘 정의 논리적 문제해결 절차
	알고리즘 특성 및 표현	알고리즘 특징 알고리즘 기술 언어
		문제 발견,문제 이해,문제 정의
알고리즘과 문제 분해	문제 발견 및 정의	문제 해결 계획 수립
	문제 분석	현재 상태 목표 상태 요소 구별 (시간과 자원과 같은)조건
		문제 분해의 특징
		자료 중심/절차 중심 분해 (목표상태에 도달하기 위한) 작업 나열
	문제 나누기	
알고리즘과 추상화	패턴인식	공통 요소 찾기
	기능 추상화	핵심 기능 찾기 논리적 절차
	데이터 추상화	핵심 데이터 찾기 데이터 추상화의 특징 변수 데이터 타입
알고리즘 설계	알고리즘 구조	순차구조, 선택구조, 반복구조, 중첩제어구조
	알고리즘 표현	알고리즘 표현 방법 알고리즘 설계 기법 선형/비선형 구조
	논리	대입/산술/비교/논리연산
	함수 정의	함수의 개념, 기능정의, 매개변수, 반환값
알고리즘구현	알고리즘 분석	효율성 분석
	프로그래밍	구현
	시뮬레이션	모델과 시나리오 시뮬레이션 구현 시뮬레이션 실행 및 평가
	디버깅	코드 오류 발견 코드 오류 수정

내용 영역	내용 요소	하위 요소
문제해결과 알고리즘	문제해결과정	일반적 문제해결과정
		컴퓨팅사고기반의 문제해결과정
	알고리즘과 절차	알고리즘 정의
		논리적 문제해결 절차
알고리즘과 문제 분해	문제 발견 및 정의	문제 발견
		문제 이해
		문제 정의
		문제 해결 계획 수립 (시간계획과 필요한 자원)
	문제 분석	현재 상태
		목표 상태
		(문제해결에 필요한) 요소 구별
		(시간과 자원과 같은)조건
	문제 나누기	자료 중심 분해
		절차 중심 분해
		(목표 상태에 도달하기 위한) 작업 나열
알고리즘과 추상화	패턴인식	공통 요소 찾기

결과를 놓고 보니 전체적으로 교육 범위(양)나 난이도(질)가 다소 높게 책정된 것으로 보입니다. 대부분의 학생들이 초중고 때 SW교육을 받아본 적이 없을 것이므로 코포자가 없도록 내용을 쉽게 구성해야 할 것으로 사료됩니다.

일반적인 문제해결 과정과 컴퓨팅 사고 기반을 비교하여 생각해보는 것은 대학생의 기본 소양으로 매우 좋은 과정으로 보입니다

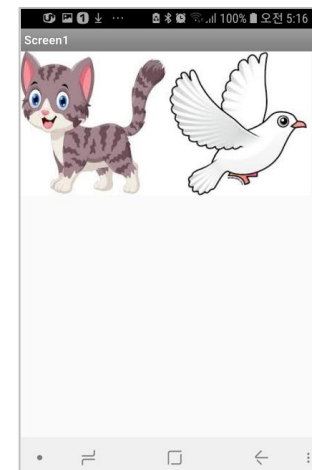
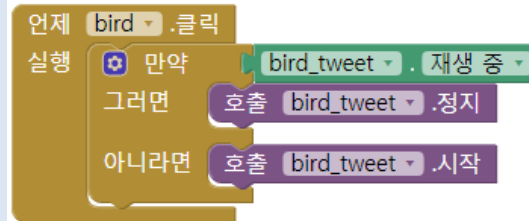
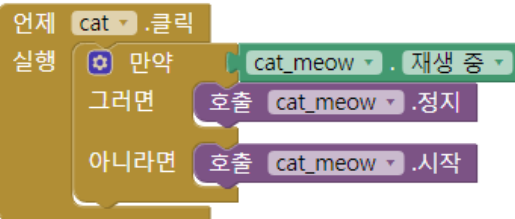
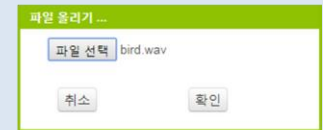
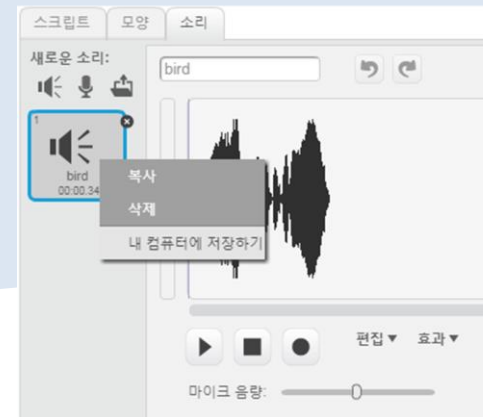
인문계에게는 생각하는 방식의 다양성을 배울 수 있으며 이과에게는 좀 더 논리적으로 생각하는 법을 다시 한번 정리할 좋은 기회로 보입니다 더 나아가 실제 컴퓨터 프로그래밍을 하여 컴퓨팅 사고를 이해하면 더 깊이 해당 내용을 느낄 수 있을 것으로 보입니다. 좋은 커리큘럼 만들어 주시기 바랍니다

# 대학기초SW입문 – 강의안 예시

## 인문 계열 – App Inventor

선택적 문제해결 : 플레이어 두 개를 이용하여 사진을 클릭하면 소리가 재생되고 다시 클릭하면 멈추는 앱 만들기

■ 컴포넌트를 [팔레트]에서 꺼내 오기

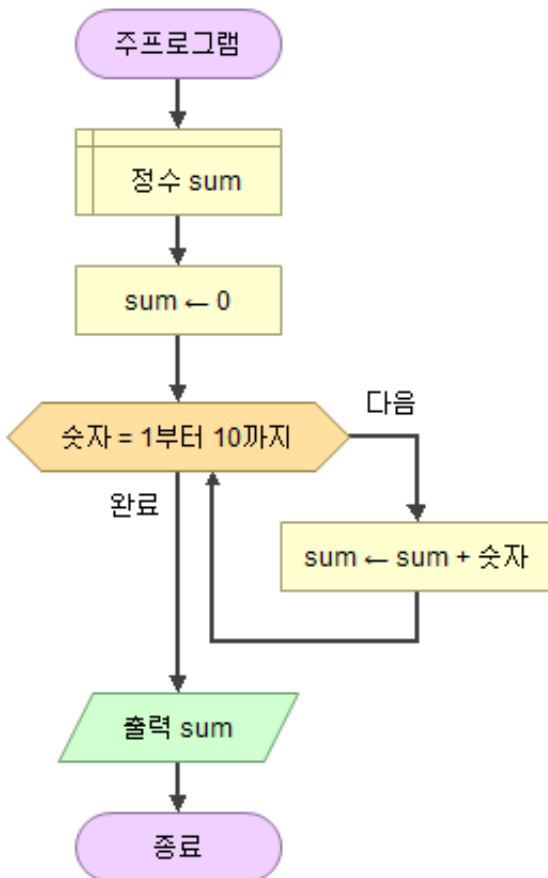


# 대학기초SW입문 – 강의안 예시

## 사회 계열 – App Inventor

반복 표현 : 0부터 10까지의 합계 구하기

- 레이블 1개, 수평배치1  
(버튼 2개: 횡수 반복\_실행, 조건반복\_실행)
- 수평배치2(레이블 2개: 레이블2, 정답(높이30,너비 200))



The App Inventor interface shows the implementation of the sum calculation. The main screen displays the title "1부터 10까지의 합은?", two buttons "횡수 반복 실행" and "조건 반복 실행", and a text input field for the answer. The components list on the left includes a button, a text box, and a label. The properties pane on the right shows the text box's width set to 200 pixels. The code block on the right implements the logic: when the "횡수 반복 실행" button is clicked, it sets a global variable "sum" to 0, then enters a loop from 1 to 10 with an interval of 1. Inside the loop, it adds the current number to "sum". After the loop, it sets the text of the "정답" label to the value of "sum".

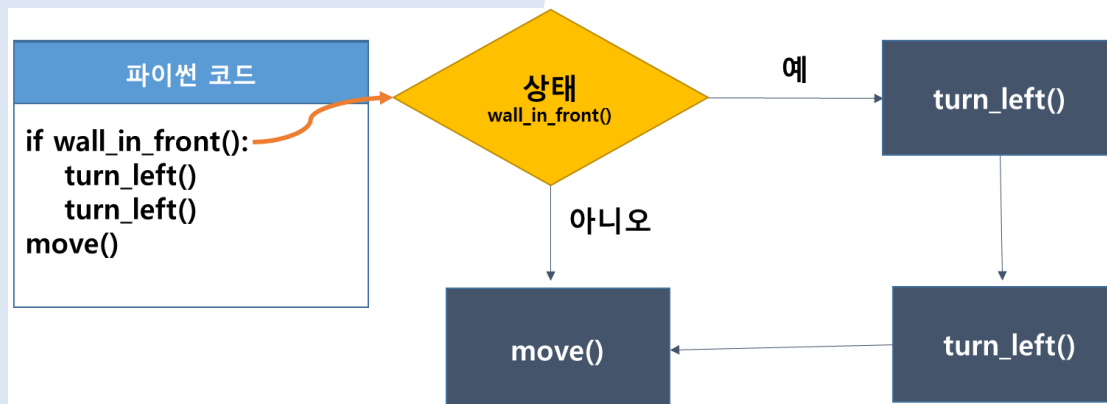
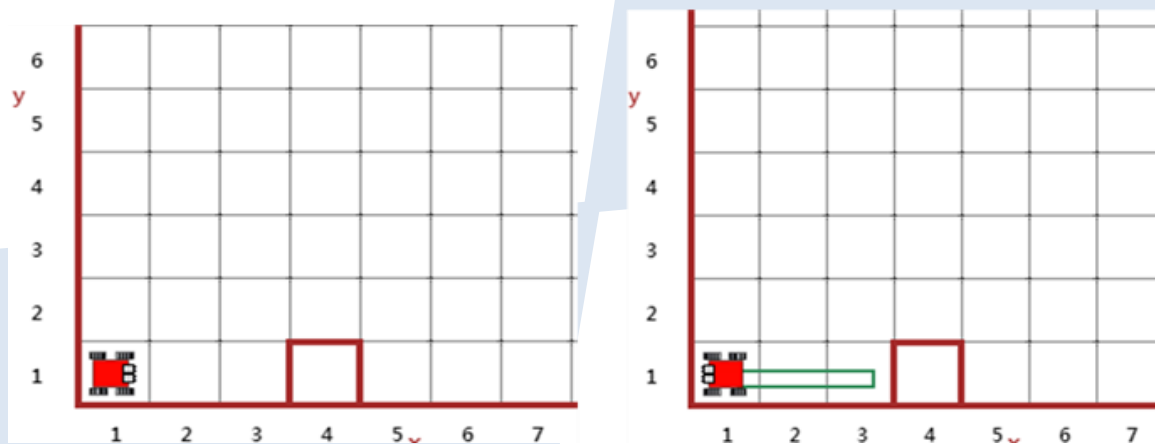
# 대학기초SW입문 – 강의안 예시

자연 계열 – Python(RUPL)

선택과 알고리즘적 사고

로봇이 벽을 만나면 되돌아오는 프로그램을 만들어 보자.

(wall\_in\_front(): 앞에 벽이 있으면 true)





# 대학기초SW입문 – 강의안 예시

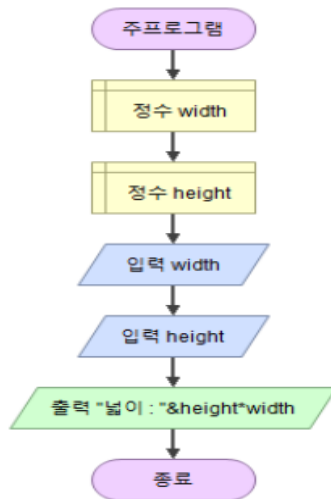
공학 계열 – Python

## 몫과 나머지 연산

두 수를 입력 받아서 몫과 나머지를 구하고자 한다.  
필요한 데이터는?                      필요한 연산자는?



### 1 순서도 작성하기



### 2 파이썬으로 코딩하기

```
In [3]: width=int(input("넓이를 입력하세요 : "))
height=int(input("높이를 입력하세요 : "))
print("사각형의 넓이는? : ", width * height)
```

넓이를 입력하세요 : 3  
높이를 입력하세요 : 5  
사각형의 넓이는? : 15

# 대학기초SW입문 – 강의안 예시

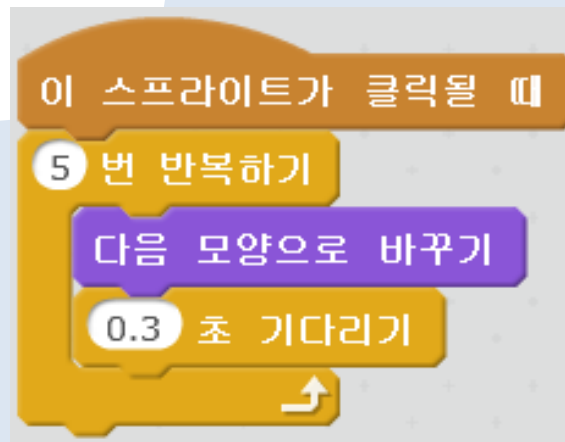
예체능 계열 - scratch

프로젝트에서 작성한 프로젝트에 좌우로만 반복하여 움직이는 꽃게와 집게를 움직이는 꽃게를 표현하는 프로그램 작성

## 1. 꽃게1

- **이벤트: 이 스프라이트가 클릭할 때 실행**

기능 : 모양바꾸기

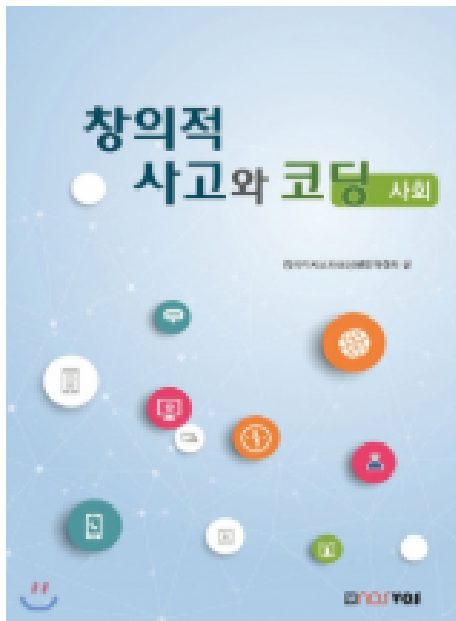


반복적 처리는 동일한  
처리 과정을  
정해진 횟수만큼  
반복하거나 조건에 따라  
반복

## 4. 단국대학교 SW기초 교육 대표 성과

# 대표 성과

- **창의적 사고와 코딩**
  - 교재 5유형 출판
- '창의적 사고와 코딩(사회)' 교재 (사)한국대학출판협회 '**2018 올해의 우수도서**' 선정
  - SAP Fiori를 활용한 '창의적 사고와 코딩' 교양교육

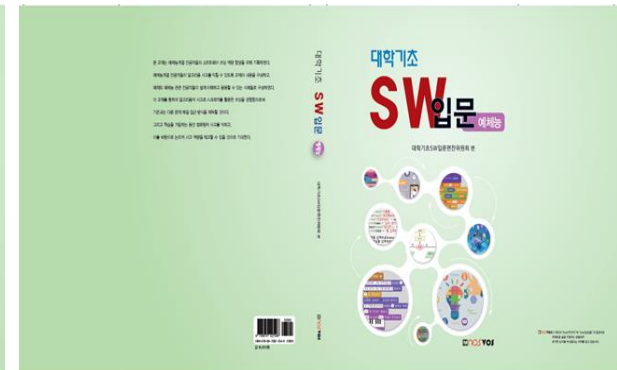
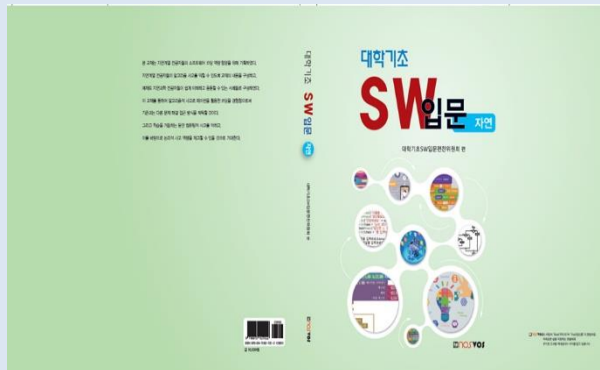


창의적 사고와 코딩(사회)(서응교 외 지음, 단국대출판부)  
크게 창의적 사고와 소프트웨어 코딩, 소프트웨어 코딩의 기초, 창의적 사고의 발현과 소프트웨어 설계, 소프트웨어를 통한 창의적 사고의 구현에 관한 내용이 주를 이룬다. 부록으로 앱 인벤터로 처음 코딩해보기 또한 담겨있다.

# 대표 성과

- **대학기초 SW입문**  
- 교재 5유형 출판

교재명	ISBN
대학기초소프트웨어입문(인문)	978-89-7092-700-8 03560
대학기초소프트웨어입문(사회)	978-89-7092-701-5 03560
대학기초소프트웨어입문(자연)	978-89-7092-702-2 03560
대학기초소프트웨어입문(공학)	978-89-7092-703-9 03560
대학기초소프트웨어입문(예체능)	978-89-7092-704-6 03560



단국대학교  
SW중심대학

# 대표성과 – 학생들의 의견

- 😊 앱인벤터를 이용해 간단한 앱을 만들어 보는 것이 좋은 경험이 되었다고 생각한다
- 😊 앱을 직접 만들어볼 수 있어서 좋았습니다
- 😞 앱을 만드는 것은 처음 접해보는 것이었는데 기본지식이 깔려있지 않은 상황에서 하기에는 난이도가 있었다고 생각한다.

## 인문(HLL)

- 😊 실습중심이라서 나중에 유용하게 쓰일것 같아요 감사합니다 교수님 ㅎㅎ
- 😊 실습위주로 해서 그런지 지루하지 않았습니다.
- 😞 컴퓨터가 불량인건 학교에서 바로바로 수리해주세요.
- 😞 컴퓨터가 느려서 진행이 지체된다.

## 사회(RPL)

- 😊 스크래치에 대한 흥미도를 높일 수 있었다.
- 😊 예체능 학생들에게는 조금 어려운 수업이었으나 다른 것과 더 융합해서 sw를 만들수 있어서 좋았습니다.
- 😞 기말고사가 너무 어려웠습니다
- 😞 스크래치등의 프로그램을 다루는 이유를 잘모르겠음

## 예체능(ART)

- 😊 학생들의 눈높이에 맞춘 수업
- 😞 컴퓨터 점검이 필요
- 😞 인터넷 속도가 너무 느려서 수업 진도를 따라가는데 힘든적이 많다

## 자연(SCI)

- 😊 조별활동으로 협동심이 길러짐
- 😊 혼자 학습해볼 시간을 주셨던 것이 좋았습니다.
- 😞 수강생의 규모가 조금 많은 것 같다.
- 😞 진도가 너무 빠르다.

## 공학(ENG)



감사합니다.

