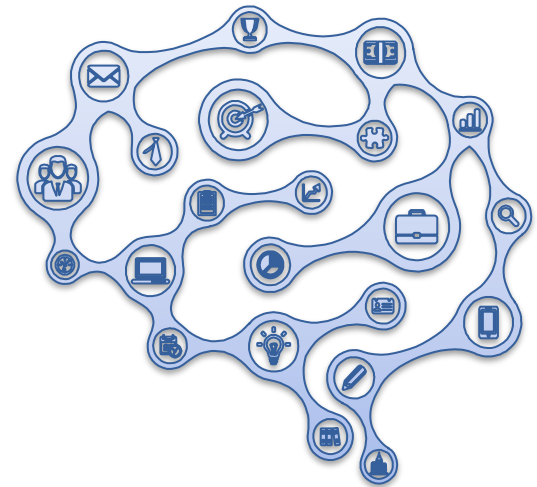


# 의료 Artificial Intelligence

4차산업혁명과 의료 빅데이터 (chapter 14)

2022.03.17



# 오늘 배울 내용 ...

## 1. 4차산업혁명

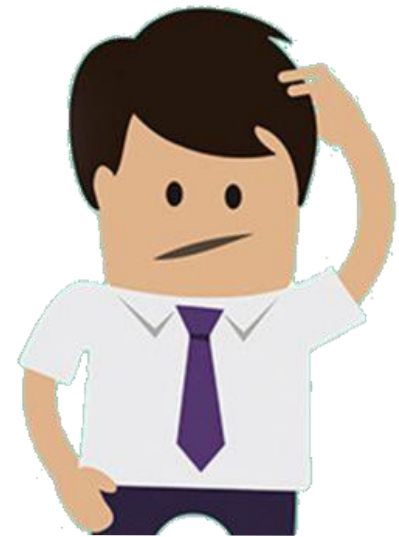
2. 빅데이터와 사물인터넷

3. 의료 부문 빅데이터

4. 논리설계 - 간단 논리 실습 2

5. mblock : 이벤트 응용

어렵지 않다  
쉬운 것도 아니다



---

# 인공지능 이론

---

# 혁신 / 기술혁명 / 산업혁명 ...

- 혁신 : 기존에 존재하는 기술을 가치 있는 방식으로 재조합하거나 대체
  - 기술혁명 : 신기술이 기존의 기술을 단기간에 대체하며 세상을 혁신적으로 변화시키는 것 (게임체인저)
  - 산업혁명 : 발명과 기술의 발전 등으로 정치·경제·사회·문화 등에 급격한 변화를 가져오는 것. (기술이 사회를 변혁시키는 것)
- \* 산업혁명에서 발생하는 변화는 산업 분야는 물론 사회 전체의 구조와 성격에도 격변하는 대전환을 가져옴



# The Fourth Industrial Revolution ...



1차 산업혁명(1770~1850년) 시대의 기술혁명은  
증기기관의 발명으로  
기계화 시대와 증기·철도의 시대를 열었음



2차 산업혁명(1870~1914년) 시대의 기술혁명은  
철강·전기를 기반으로 하는 대량생산체제로의 전환,  
이어서 석유와 자동차가 등장



3차 산업혁명(1970~최근) 시대의 기술혁명은  
반도체와 전자로,  
디지털혁명 또는 자동화혁명과 함께 진행되고 있음

# The Fourth Industrial Revolution ...

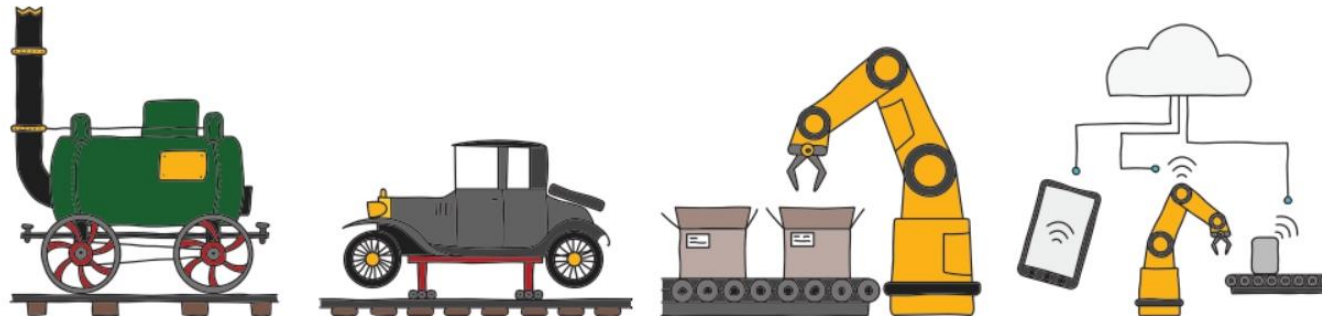
**빅데이터 + 인공지능 + 초연결**



단계	시기	기반기술	파급 효과	분야
1차 산업혁명	1769년	증기기관	기계화, 대량생산, 직조직기	기계혁명
2차 산업혁명	19세기 말	전기에너지, 내연기관	효율적인 대량생산, 조립라인	에너지혁명
3차 산업혁명	1960년대	자동화, 전자공학, 인터넷	컴퓨터 기반의 정보혁명	디지털혁명
4차 산업혁명	2020년대	가상 물리 시스템, 5G, AI, IoT	기술발전과 첨단화의 융합	기술혁명

# The Fourth Industrial Revolution ...

1. 1, 2, 3차 산업혁명이 기계로 인간의 노동력을 절감하는 손과 발의 역할을 수행했다면,  
4차 산업혁명에서는 인간의 두뇌 역할을 기계가 분담하거나 대체
2. AI 기반의 만물 · 초지능 혁명으로, 사람·사물·공간을 초연결 · 초지능화하여  
산업 구조와 사회 시스템을 급격히 변화시키는 혁신적인 기술 혁명임
3. 과거 조합이 불가능했던 대상들의 연결/조합이 디지털과 ICT 혁신으로 가능해지면서  
새로운 가치를 창출하게 됨



# 세상은 어떻게 바뀌는 가?

**보이지 않는 것이 보이는 것을 움직인다.**

1. HW 회사 → 서비스 회사

2. 기계 → 컴퓨터

3. 제품 기획자 → 데이터 과학자

4. 인간 의존적 생산 → SW에 의한 생산

5. 인간에 의한 서비스 → 로봇에 의한 서비스

6. P2P 연결 → 플랫폼에 의한 연결(서비스에 대한 연결)





# ICBMSA 플랫폼

---

- 4차 산업혁명을 촉발하는 핵심 기술  
: ICBMSA (사물인터넷, 클라우드, 빅데이터, 모바일, 보안, 인공지능)
- ICBMSA 플랫폼은 서로 융합해 상호작용하는 통합 플랫폼으로, 4차 산업혁명 시대를 이끌 것으로 전망
- 이미 미국, 독일 등 주요 선진국은 ICBMSA 기반의 지능정보기술을 제조업 등 다양한 산업 분야와 융합해 이전에는 없던 새로운 형태의 제품·서비스·비즈니스를 창출
- 인터넷을 기반으로 수많은 IoT 디바이스로 수집한 데이터가 클라우드에 저장되고, AI로 빅데이터를 분석함으로써 유용한 정보가 생성

# IoT : 사물인터넷

- 좁은 의미 : 유비쿼터스 환경에서 사물끼리 통신을 주고받는 것

넓은 의미 : 인간, 사물, 서비스가 인간의 명시적 개입 없이 상호 협력해 센싱, 네트워킹, 정보 처리 등의 지능적 관계를 형성 하는 사물 공간 연결망

- IoT의 요소

. 연결 : 기기와 서비스 인프라를 묶는 역할을 하는 유·무선통신 혹은 네트워크 인프라

. 정보 : 정보의 생산은 곧 주변 환경과 상태의 정보를 얻어내는 센싱 기술, 정보는 생성되는 지점에 의미가 있음

. 서비스 : 사람에게 제공을 목적으로 정보를 묶고 가공해 제공하는 서비스, 서비스를 제공하기 위한 서비스 연동 인터페이스 기술

# IoT : 사물인터넷

1. IoT는 4차 산업혁명을 위한 기반기술 중에서 실생활의 변화와 가장 밀접한 기술
2. 사람·디바이스·공간 등 모든 것이 네트워크를 기반으로 연결되는 초연결망으로 실현
3. 스마트 홈, 스마트시티에서 인간의 모든 활동에 반응하는 IoT 서비스가 상용화될 것
4. 스마트 공장에서도 IoT는 핵심임



# Cloud : 클라우드

---

1. 클라우드 컴퓨팅은 네트워크를 통해 원격으로 필요한 작업을 수행하는 기술.
2. 클라우드 컴퓨팅은 IoT 기반으로 연결된 센서에 의해 수집된 정보를 빅데이터로 분석할 수 있는 저장 공간이자 기반시설 역할
3. IoT를 활성화하려면 클라우드 서비스를 먼저 구축해야 함. IoT를 통해 수집되는 수많은 데이터를 저장할 수 없다면 IoT 비즈니스가 불가능하기 때문
4. HW 대상을 의식하지 않고 컴퓨팅 시스템, 서비스를 이용하는 것

# Big Data : 빅데이터

1. 대량의 정형·반정형·비정형 데이터로부터 의미 있는 가치를 추출하고 유용한 결과를 도출하는 기술
2. 빅데이터의 속성 : Volume, Variety, Velocity + Veracity, Value
3. 연결 기기의 보급으로 인해 데이터가 폭증. 구조화되지 않은 데이터, 저장 방식이 다른 데이터, 중복 데이터 등이 발생하면서 관리 대상이 크게 늘어남
4. 비정형 데이터를 정형화 하여 분석하는 시도가 이루어지는 중



그림 8-6 유튜브의 1분당 비디오 업로드 양과 매일 시청 시간(출처: 유튜브)

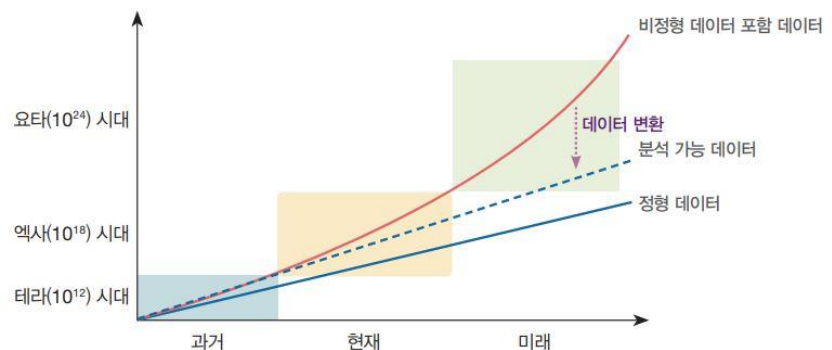


그림 8-7 비정형 데이터의 폭증으로 인한 분석의 한계

# Big Data : 빅데이터

## 빅데이터 처리 과정

- 수집
  - 기기나 컴퓨터로부터 데이터를 모음
  - 자체적으로 보유한 데이터베이스에 적재되는 **정형 데이터**
  - 문서 파일, 이미지, 영상 등의 **비정형 데이터**
  - 목적에 맞는 소스 데이터를 수집하는 과정은 주로 프로그래밍이나 툴을 이용해 자동으로 이루어짐
- 저장
  - 대량의 데이터를 파일 형태로 저장할 수 있는 기술과 **비정형 데이터를 정형화된 데이터 형태로 저장하는 기술**이 필요
- 처리
  - 수집한 데이터는 실시간으로 일괄 처리하는 **정제 과정**을 거침
  - **필요 없거나 손실된 데이터는 제거**하고 분석에 필요한 데이터는 효율적으로 재배치함
- 분석
  - 저장한 데이터를 **의미 있는 지표로 분석**하는 단계
  - 분석 알고리즘이 내장된 인공지능이나 분산 병렬 처리 알고리즘, 분석 스크립트 등을 통해 **분석 결과를 만들어냄**
- 표현 (서비스화 / 시각화)
  - **시각화** : 분석이 가능하도록 변환된 데이터는 **사람이 인지할 수 있는 형태로 표현**  
→ 시각적인 도표나 그래프로 표현
  - **서비스화** : 응용을 위해 애플리케이션이나 **기계가 필요한 형태로 데이터를 제공**

# 의료 빅데이터

## 의료정보

전국민의 진료정보와 의료기관, 제약회사, 유관기관 등 다양한 경로에서 수집한 정보를 분석·정제한 데이터

진료정보

의약품  
정보

치료재료  
정보

의료자원  
정보

비급여  
정보

의료 질 평가  
정보

연계 정보

---

## 지능 만들기 - 논리설계 실습

---



# mblock 논리 설계

---

1. 문제를 분석해서 입력 - 처리 - 출력으로 구분
  2. 입력과 로직에 필요한 변수에 대해 변수 블록 만들기
  3. 처리 로직에 필요한 블록 확인 : 연산, 값 입력, 동작, 형태 등
  4. 순차, 조건, 반복 논리에 맞도록 블록을 구성
  5. 출력은 스프라이트(캐릭터)의 생각하기나 말하기 블록으로 표현
- \* 반복 사용되는 논리는 사용자 블록(내 블록)을 만들어 호출

# [mblock 논리] 데이터의 출력

안녕! 을(를) 2 초 동안 말하기

안녕! 을(를) 말하기

→ 문자열 출력

sum 을(를) 2 초 동안 말하기

sum 을(를) 말하기

→ 변수값 출력

결과 = 와(과) sum 을(를) 결합한 문자열 을(를) 2 초 동안 말하기

→ 조합 출력

결과 = 와(과) sum 을(를) 결합한 문자열 을(를) 말하기

# [mblock 논리] 데이터의 입력

변수 : 사용자가 입력한 값, 초기에 필요한 값, 처리에 필요한 값, 결과값

변수 만들기

변수 생성

count

count 을(를) 0 로(으로) 설정하기

변수값 설정/변경

count 을(를) 1 만큼 변경하기

변수값 증가  
 $\text{변수값} = \text{변수값} + x$

count 값을 묻기 묻고 기다리기  
count 을(를) 대답 로(으로) 설정하기

변수값을 사용자가 입력

b 을(를) 1 부터 10 사이 임의의 수 로(으로) 설정하기

랜덤 값을 입력

# [mblock 논리] 연산

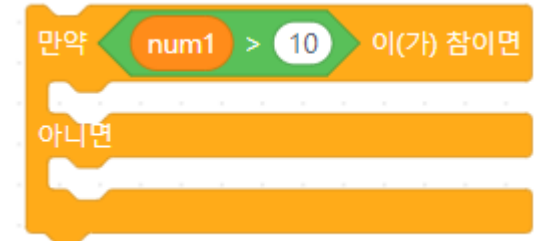
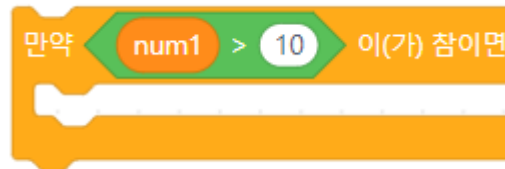
num1 = 1 + 2



num1 = num2 - 2



[조건]  
num1이 10보다 크면



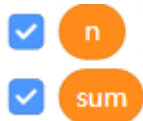
[반복]  
조건이 참일때까지 반복



# 논리설계 실습 - 1

1~100까지 3의 배수만 누적해서 더한 결과 출력

변수



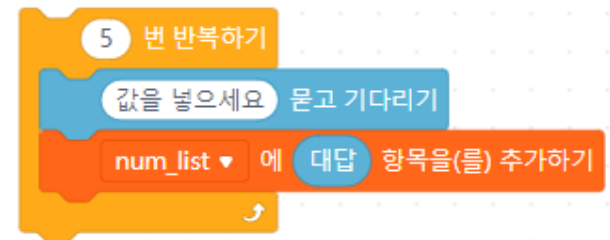
# [mblock 논리] 리스트 다루기

묶음 데이터는 리스트를 만들어 제어한다.

**리스트 : 순서가 필요한 변수들 필요시, 속성이 같은 값을 묶어서 처리할 때**



값을 입력하는 블록



값을 사용하는 블록



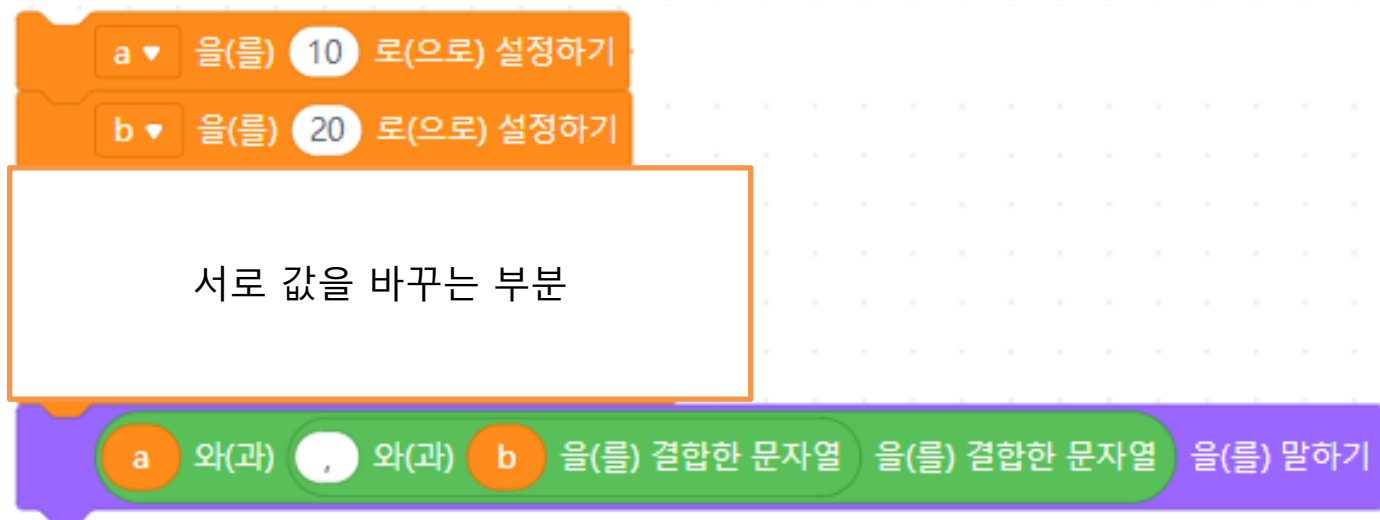
## 논리설계 실습 - 2

13,56,33,22,1,5,36,7,-3,23 를 담은 리스트에서 최대값을 찾아 출력



## 논리설계 실습 - 3

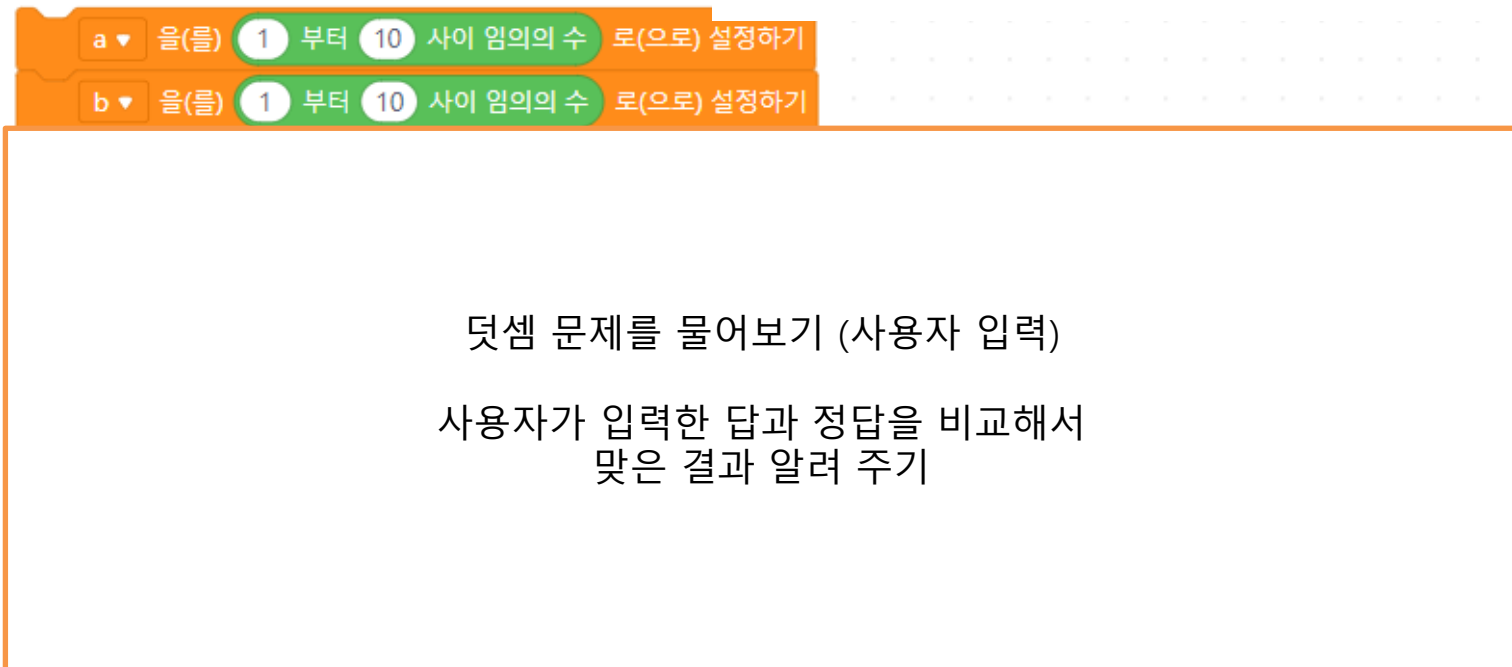
a=10, b=20일 때 두 변수값을 바꾼 후 a,b를 출력하기





## 논리설계 실습 - 4

덧셈게임 : 1~10사이의 임의의 수 2개를 만들어 덧셈문제를 출제하고  
사용자가 입력한 답이 맞으면 '맞았습니다'  
틀리면 문제와 정답을 알려주는 프로그램



## 논리설계 실습 - 5

덧셈게임 : 실습 4의 덧셈 게임을 확장해서  
총 10문제를 내고 정답을 맞춘 점수를 출력하는 프로그램

클릭했을 때

덧셈 문제 10문제를 출제해서  
사용자 답을 받음.  
사용자 답이 맞으면 점수를 증가시킴

점수가 6점 이상이면 '훌륭합니다'  
점수가 6점 미만이면 '노력이 필요합니다' 출력

---

# MBlock 실습

---

# 스프라이트 제어 블록

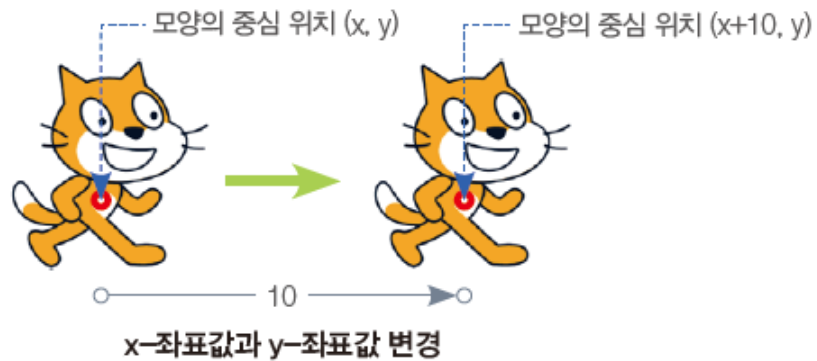
블록	설명
	설정된 값만큼 움직인다. 양수이면 전진, 음수이면 후진한다.
	설정된 각도만큼 시계 방향으로 회전한다.
	설정된 각도만큼 시계 반대 방향으로 회전한다.
	마우스 포인터나 다른 스프라이트 위치 또는 무작위 위치로 이동한다.
	좌표(x, y)로 이동한다.
	설정된 시간 동안 마우스 포인터, 다른 스프라이트, 무작위 위치로 이동한다.
	설정된 시간 동안 좌표(x, y)로 이동한다.
	설정된 방향으로 향한다. 0 : 위쪽, 90 : 오른쪽, 180 : 아래쪽, -90 : 왼쪽

# 스프라이트 제어 블록

블록	설명
 마우스 포인터 ▾ 폭 보기	마우스 포인터나 다른 스프라이트 위치로 향한다.
 x 좌표를 10 만큼 바꾸기	x좌표를 설정한 값만큼 변경한다.
 x 좌표를 -96 (으)로 정하기	x좌표를 설정한 값으로 변경한다.
 y 좌표를 10 만큼 바꾸기	y좌표를 설정한 값만큼 변경한다.
 y 좌표를 26 (으)로 정하기	y좌표를 설정한 값으로 변경한다.
 벽에 닿으면 튕기기	벽에 닿으면 반대 방향으로 전환한다.
 x 좌표 y 좌표 방향	각각 x좌표 값, y좌표 값, 방향 값이다.

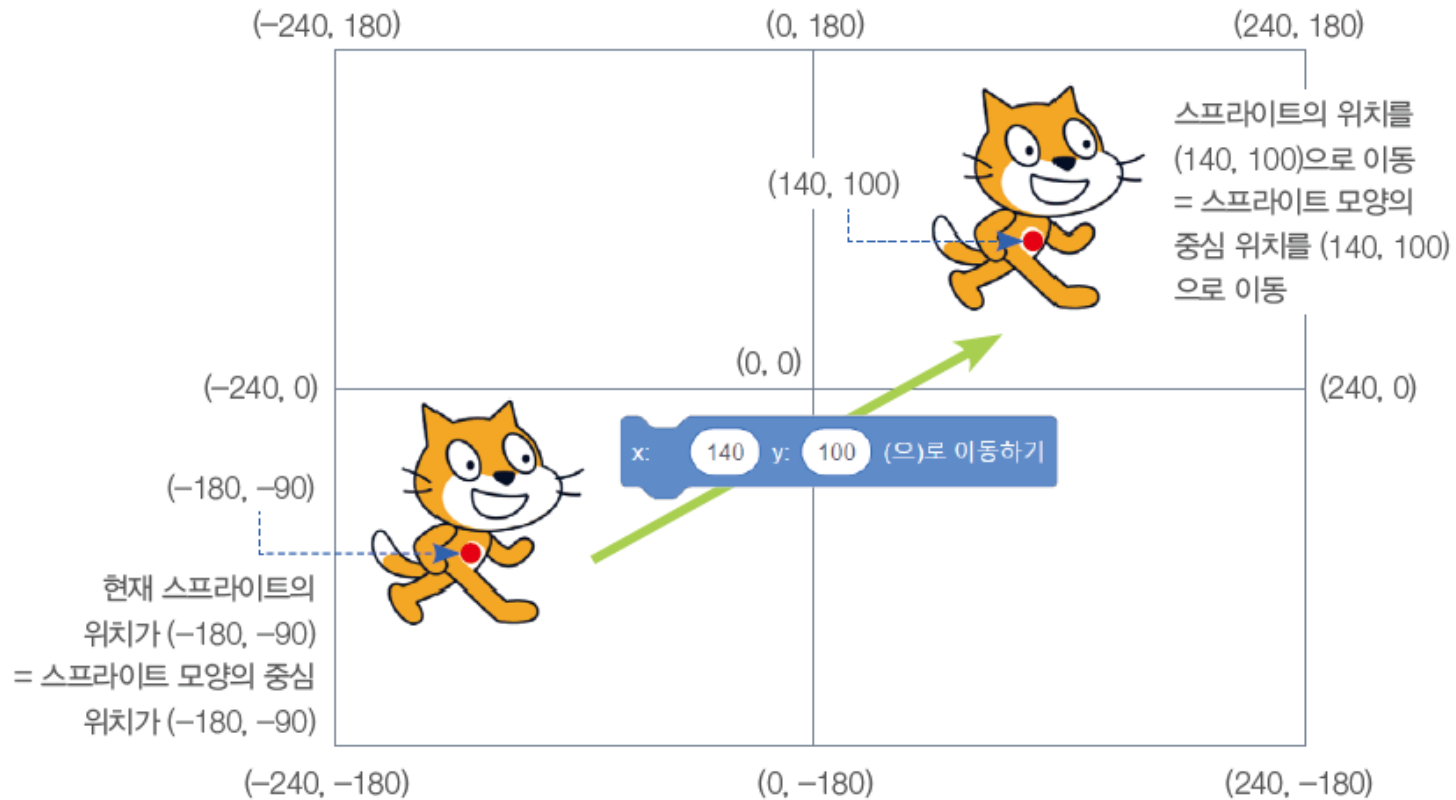
# 스프라이트 동작 구현

## 위치 이동



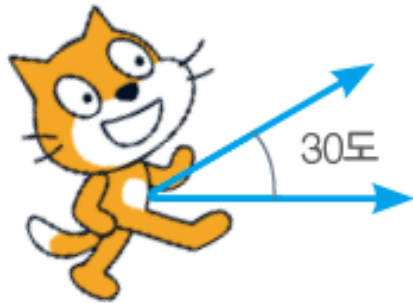
# 스프라이트 동작 구현

## 좌표에 의한 이동

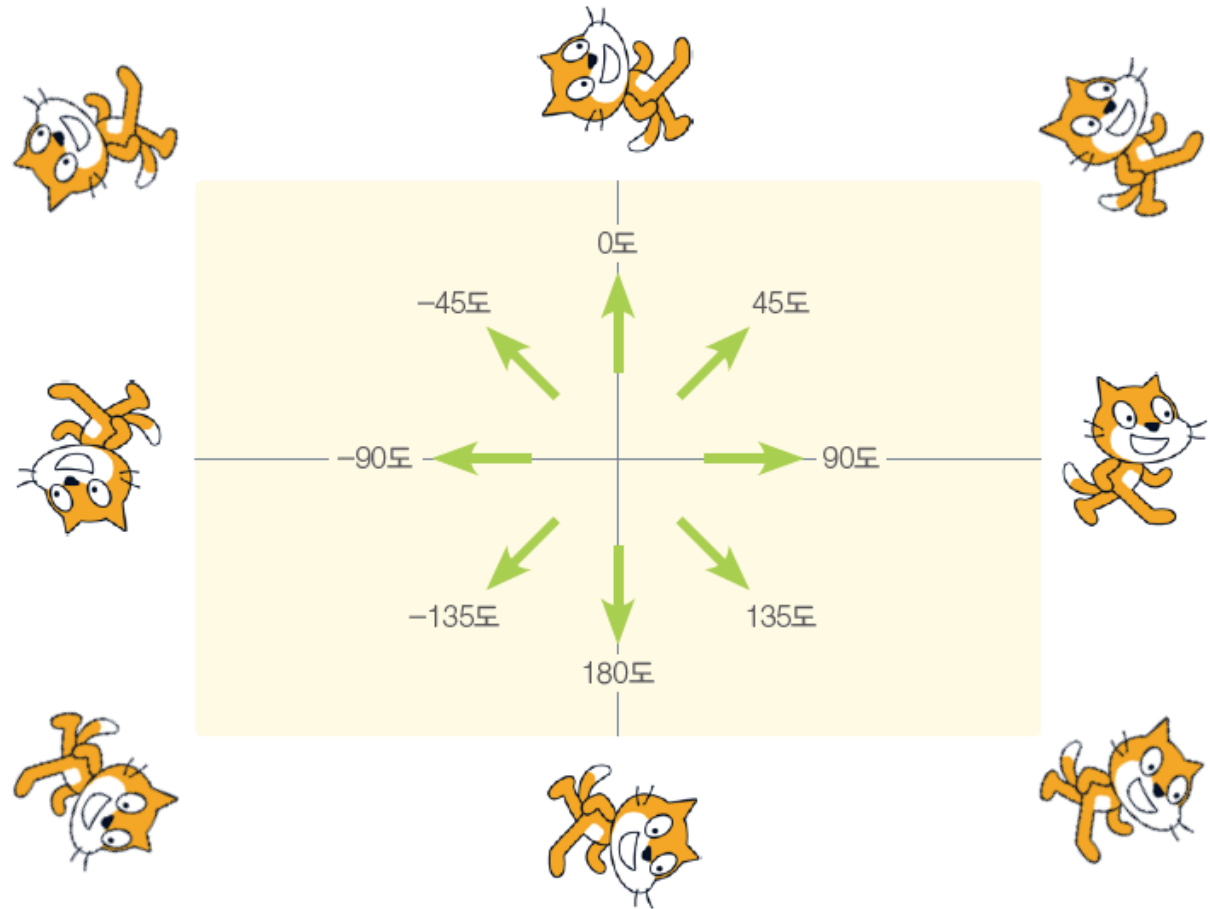


# 스프라이트 동작 구현

## 방향 회전 방식















(a) 회전할 각도를 정해줌





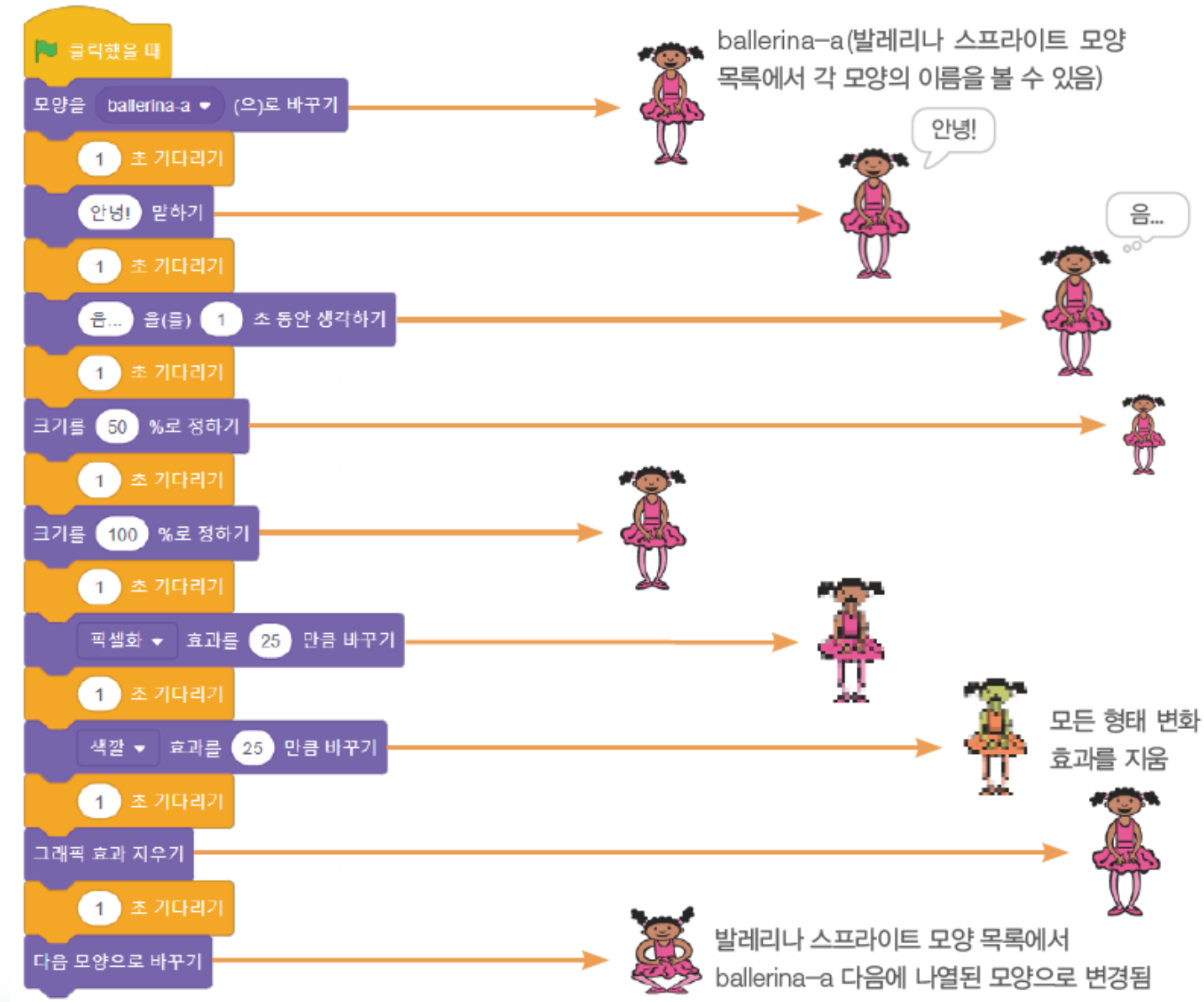
# 스프라이트 동작 구현

## 회전 방식 모드 옵션

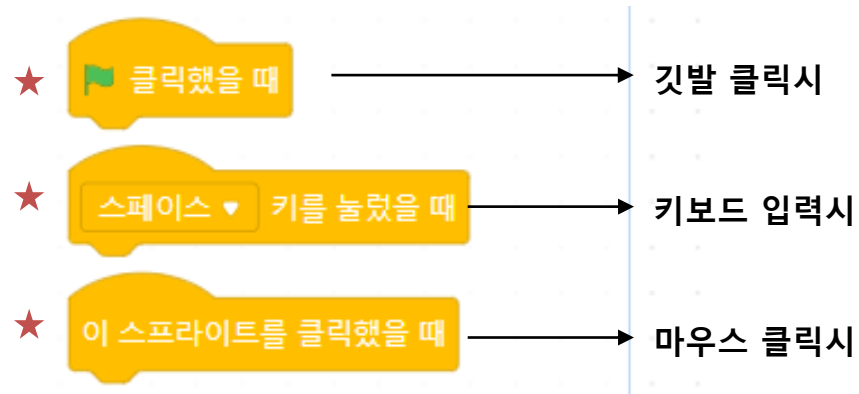
회전 방식 모드	0도	90도	180도	-90도
회전 방식을 회전하기 ▾ (으)로 정하기				
회전 방식을 왼쪽-오른쪽 ▾ (으)로 정하기				
회전 방식을 회전하지 않기 ▾ (으)로 정하기				



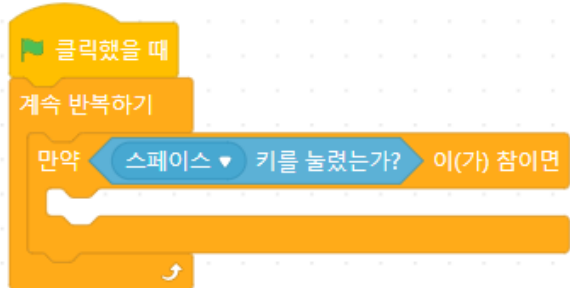
# 스프라이트 모양 변경



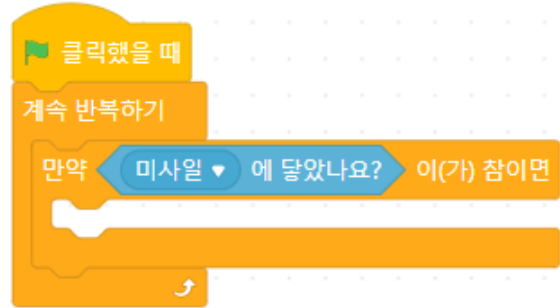
# 스프라이트 감지 로직



객체를 특정 키에 계속 반응하게 함



객체가 다른 객체에 닿을 때 반응하게 함



# 스프라이트 감지 이벤트

블록	설명
★  에 닿았는가?	스프라이트가 마우스 포인터, 벽, 다른 스프라이트 등에 닿았는지 확인한다.
★  색에 닿았는가?	스프라이트가 설정한 색에 닿았는지 확인한다.
 색이  색에 닿았는가?	스프라이트에 있는 왼쪽에 설정한 색이 오른쪽에 설정한 색에 닿았는지 확인한다.
 까지의 거리	선택한 스프라이트나 마우스 포인터까지의 거리를 확인한다.
 너 이름이 뭐니? 라고 묻고 기다리기	묻고 사용자가 답할 때까지 기다린다.
 대답	사용자가 답한 내용을 저장한다.
★  키를 눌렀는가?	설정한 키가 눌렸는지 확인한다.
 마우스를 클릭했는가?	마우스를 클릭했는지 확인한다.
 마우스의 x좌표	마우스 포인터의 x좌표를 확인한다.

# 실습 - 1

화살표키 방향에 따라 펜더가 한 걸음씩 이동하는 프로그램 구현

다음 모양으로 바꾸기

회전 방식을 회전하기 ▼ 로 정하기

180 도 방향 보기



## 실습 - 2

실습1에 이어서, 펜더가 고양이에 닿으면 고양이는 '아야!'하고 말하고  
펜더는 '미안' 하고 말하기

Panda ▼ 에 닿았나요?

