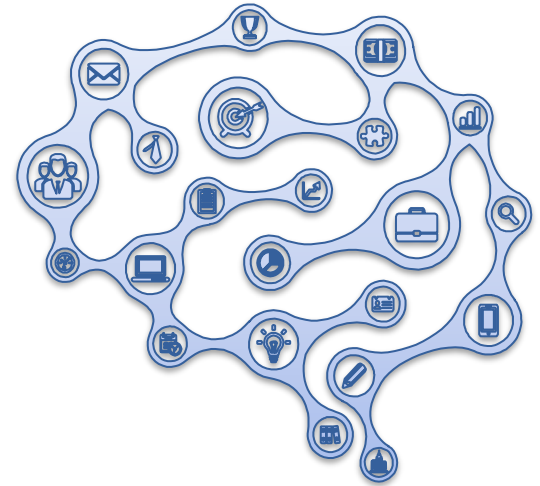


의료 Artificial Intelligence

문제와 탐색 (chap3)

2022.03.24



인공지능 이론

의료 빅데이터

의료정보

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

진료정보

의약품
정보

치료재료
정보

의료자원
정보

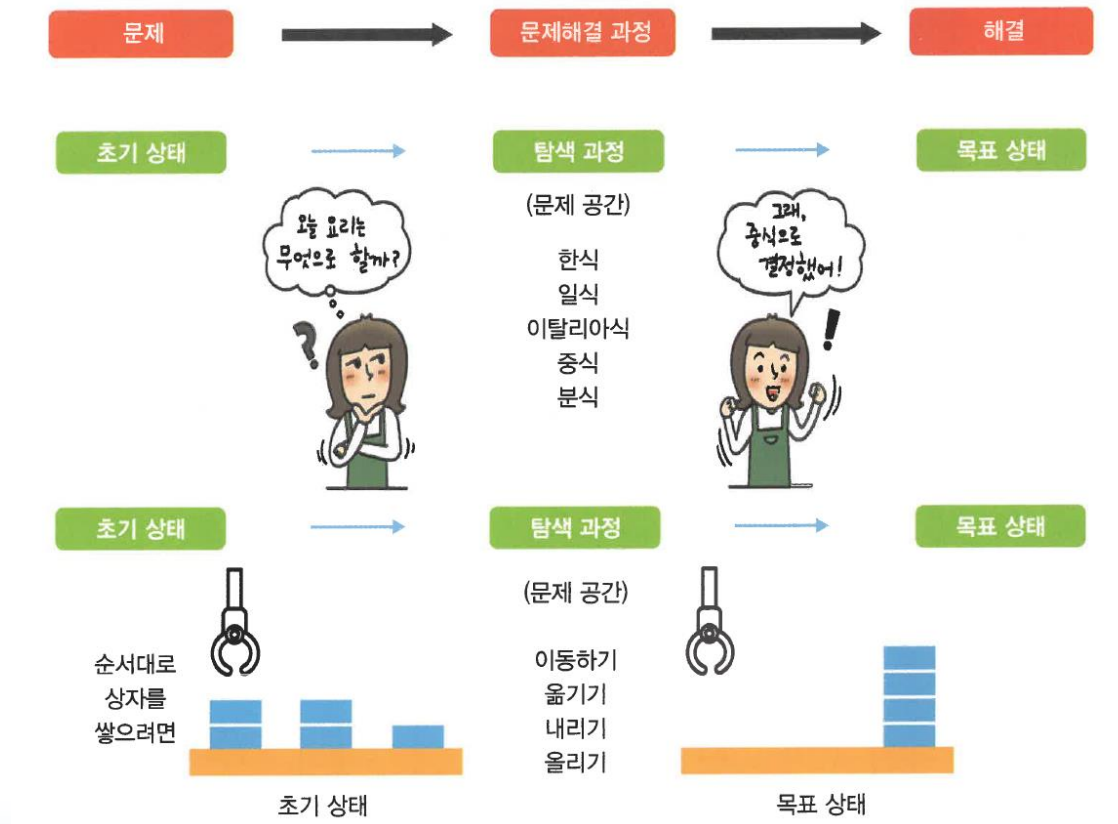
비급여
정보

의료 질 평가
정보

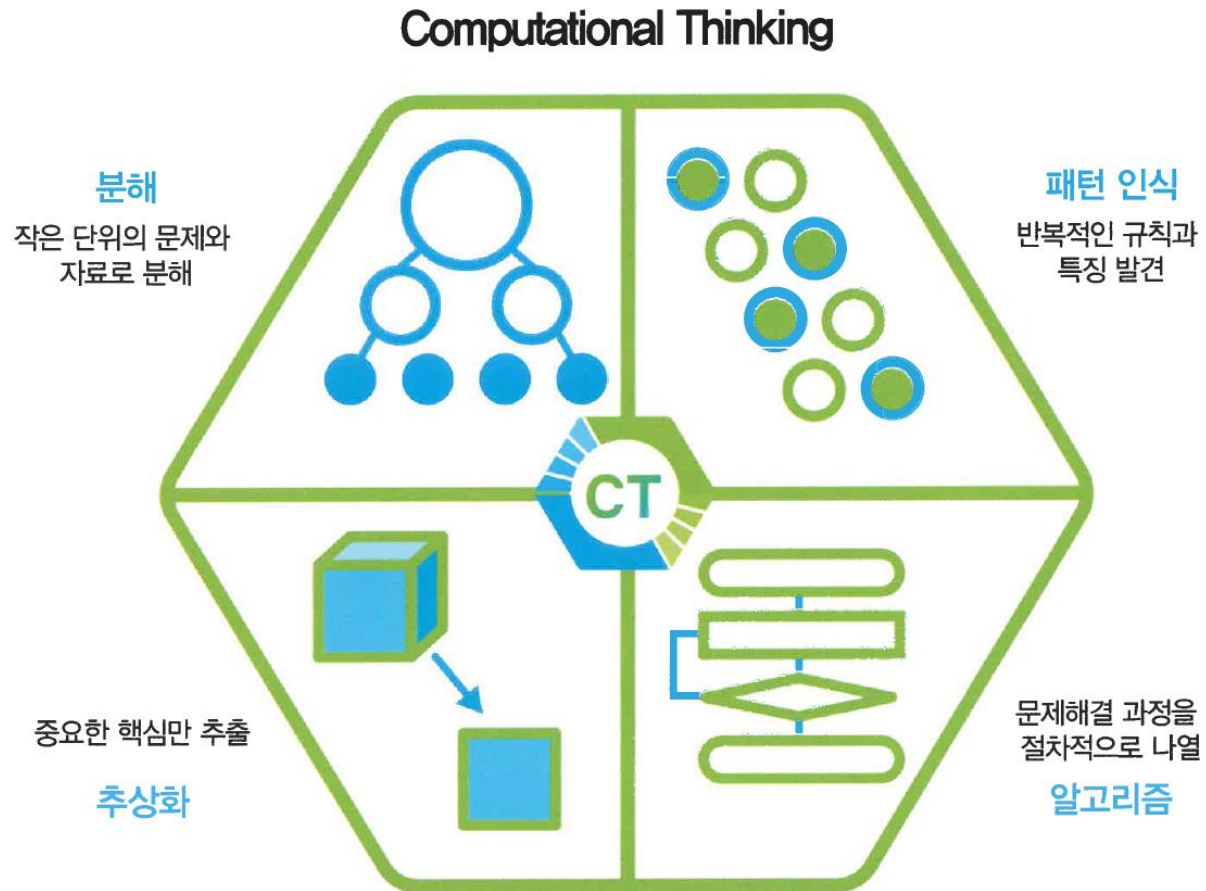
연계 정보

문제와 문제해결

- 문제 : 현재 상태와 목표 상태간의 간격
- 문제해결 : 문제공간(problem space)에서 목표하는 결과를 찾을 때까지 탐색하는 과정
- 문제해결 확인 : 초기 상태와 목표 상태의 일치 여부



컴퓨팅 사고 = 문제해결



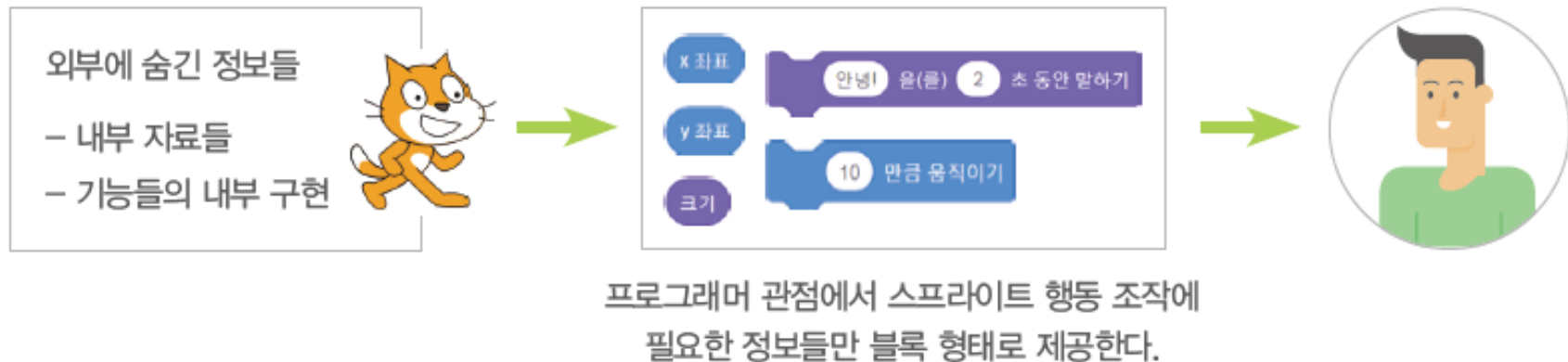
추상화

추상화(abstraction)는

1. 여러 사물이나 개념에서 공통된 속성들을 추출하여
문제 해결에 필요한 속성들만으로 간단하게 정리하기

필요한 것을 정의 하는 것 → 필요한 데이터와 기능을 찾고 구성하는 것

2. 컴퓨터가 이해할 수 있는 논리로 재해석하기



추상화

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

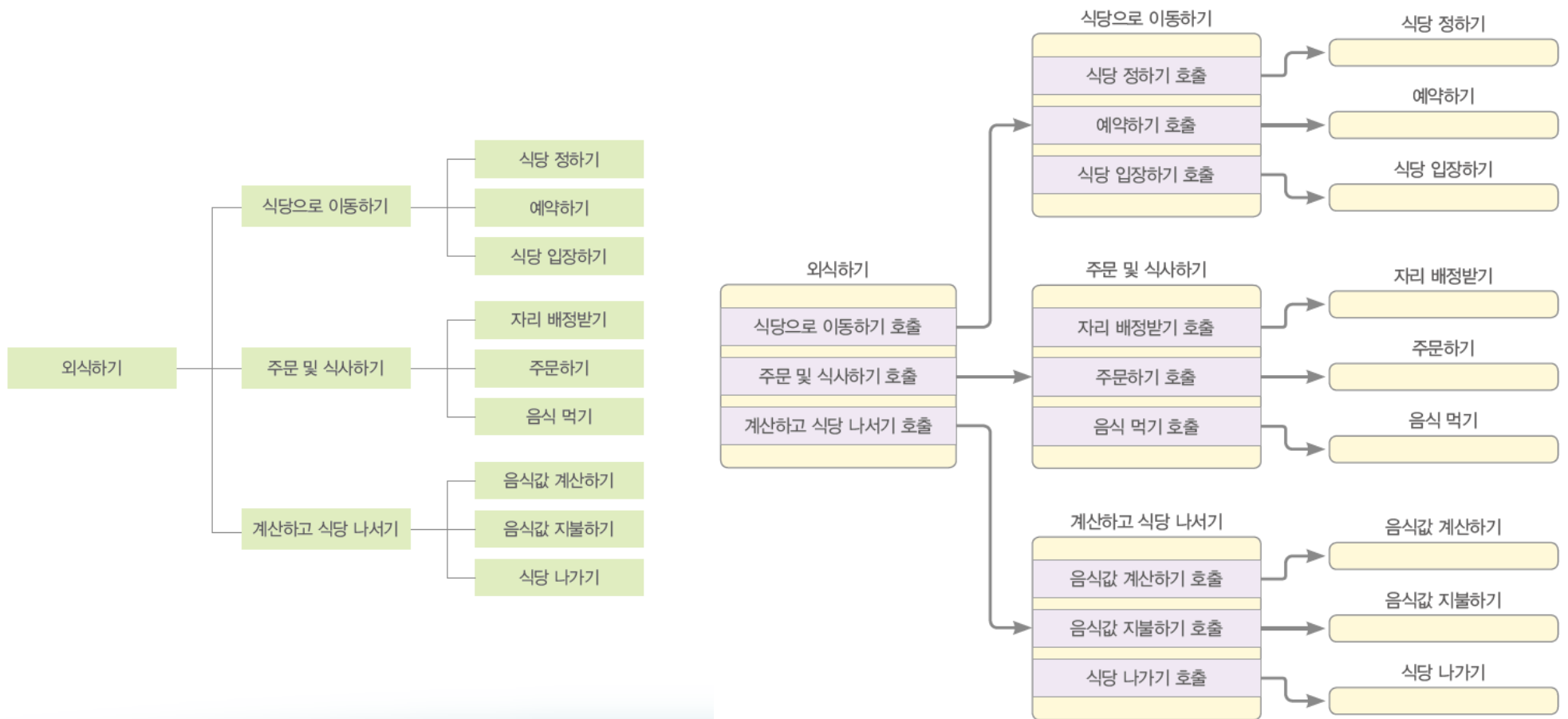
데이터 : 수 2개, 사용자가 입력한 답, 정답

기능 : 덧셈문제출제, 사용자 입력, 정답 확인 , 결과 출력

분해

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

작은 독립적인 문제로 분해하는 사고 → 모듈화



패턴인식

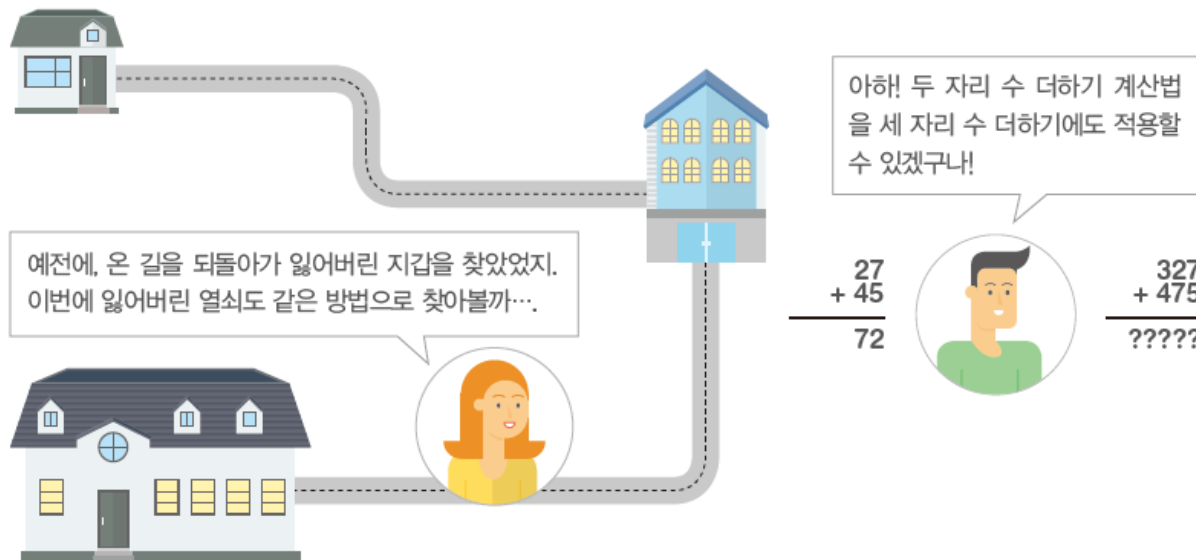
패턴인식(pattern recognition)은

1. 특징별로 나누어 이미 알고 있는 방법(패턴)과 매칭하는 과정

→ 어떤 문제를 이미 해결 방법을 알고 있는 다른 문제와 연관시켜 응용

2. 반복되어 나타나거나 쓰이는 것을 발견하여 패턴을 정의하는 과정

→ 반복적으로 순차처리 되는 부분을 패턴화



알고리즘

알고리즘(algorithm)은 어떤 문제를 해결하기 위한 절차나 방법을 의미한다.

누군가 해결한 검증된 방법을 적용하는 것

유클리드의 최대 공약수 구하기 알고리즘

두 정수 a , b 의 최대공약수 구하기 알고리즘

(가정 : 두 정수 a 와 b 는 0 이상의 값이고 a 가 b 보다 크거나 같다.)

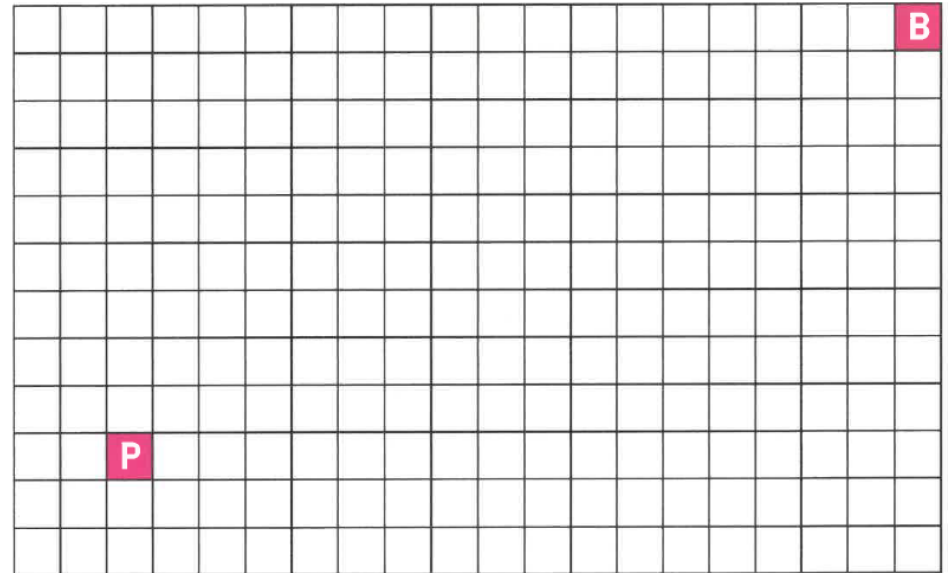
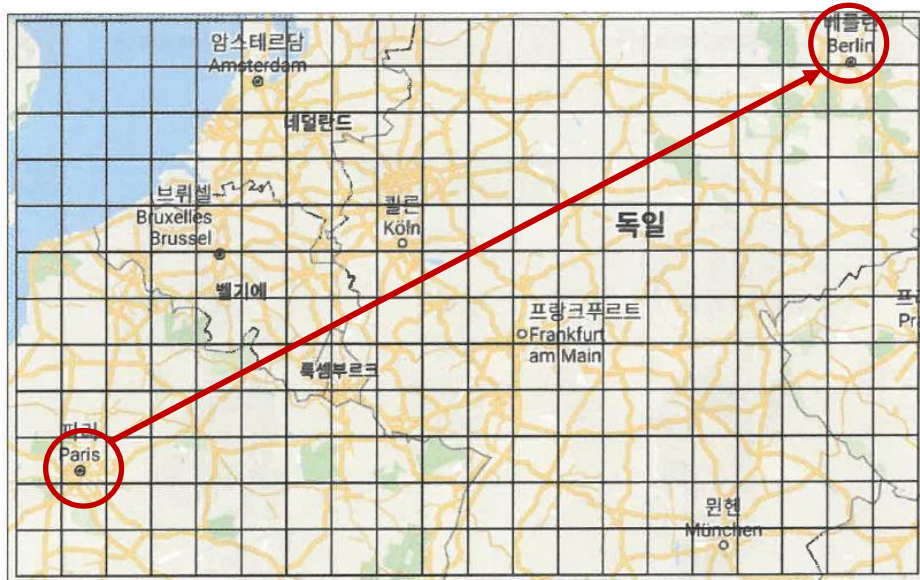
1단계 : b 가 0이면 a 를 최대공약수로 정하고 알고리즘을 종료한다.

2단계 : b 가 0이 아니라면 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

문제의 추상화

컴퓨터가 이해할 수 있는 논리로 재해석하기

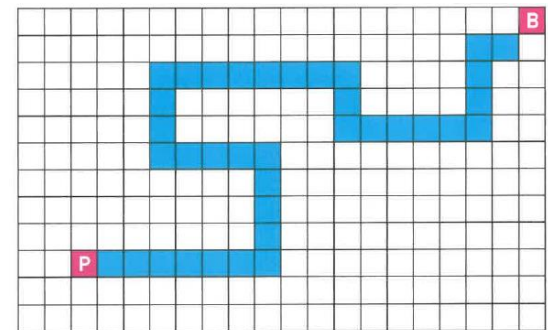
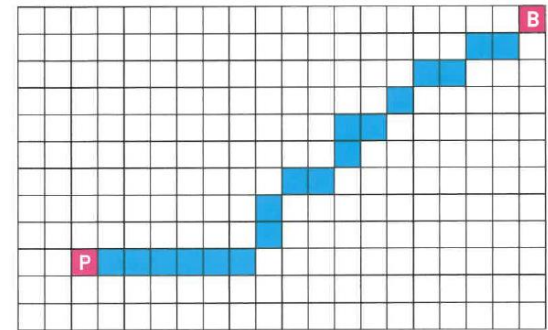
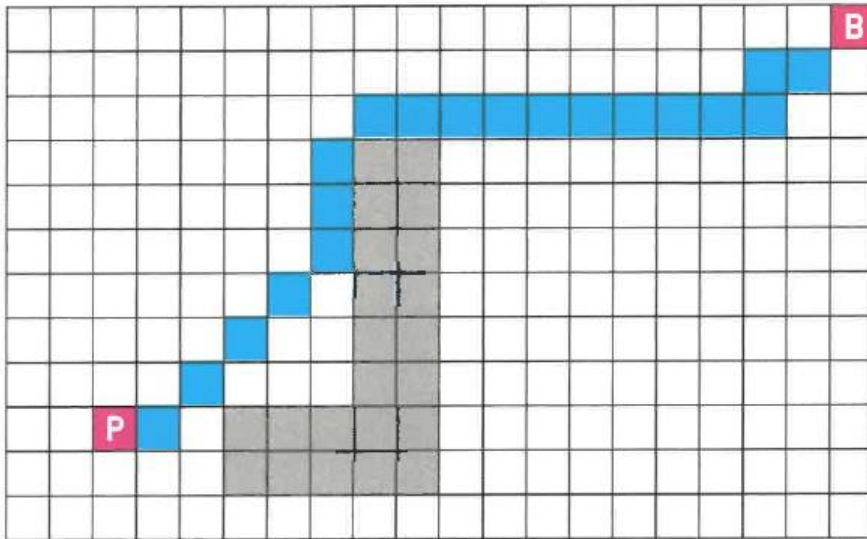


휴리스틱 알고리즘

휴리스틱(경험적) 알고리즘 :

적당한 탐색(모든 방법을 다 확인해 보지 않고)을 통해 문제를 해결 하는 방법

- 해당 문제를 해결할 수 있는 방법이 증명되지 않았을 때, 시행착오를 거치며 충분히 효율적인 해답을 유추해 나가는 기법을 의미합니다.
- 유전자 알고리즘(Genetic Algorithm)은 정말 대표적인 휴리스틱 알고리즘

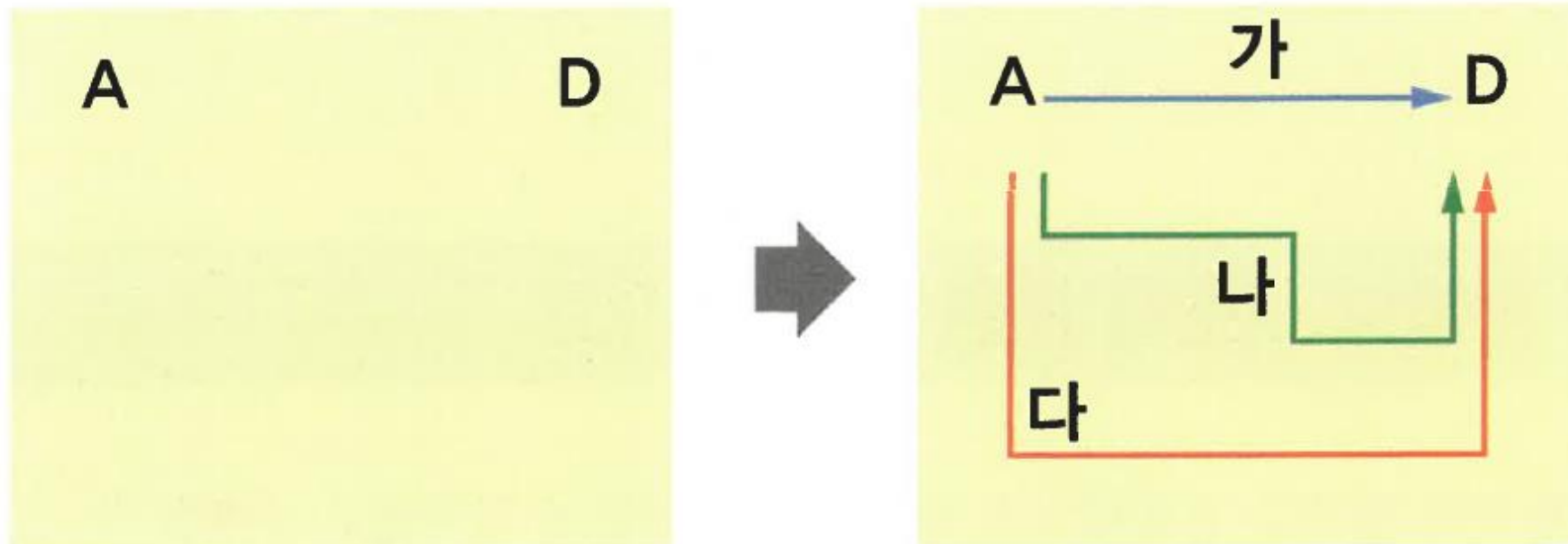


휴리스틱 알고리즘

- 시행 : 한 칸씩 이동
- 착오확인 : 목적지까지 남은 거리로 확인
 - 충분한 결과인지 확인 : 평가함수

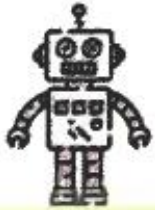
길찾기에서 휴리스틱 :

일반적으로 가장 짧은 거리를 찾으려고 한다는 특징을 응용



휴리스틱 알고리즘

길찾기의 경우 격자로 추상화



A

D



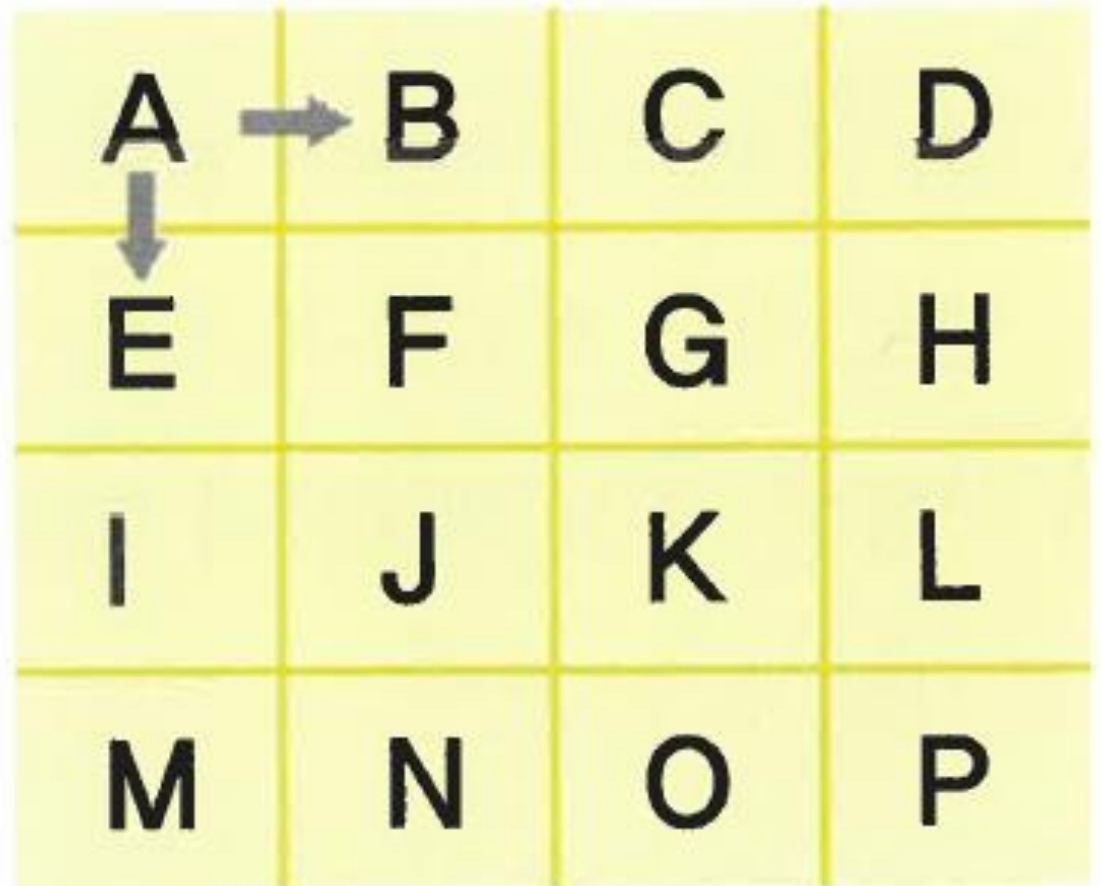
A

D



휴리스틱 알고리즘

판단(평가함수)으로 방향과 지점을 결정



지능 만들기 - 논리설계 실습

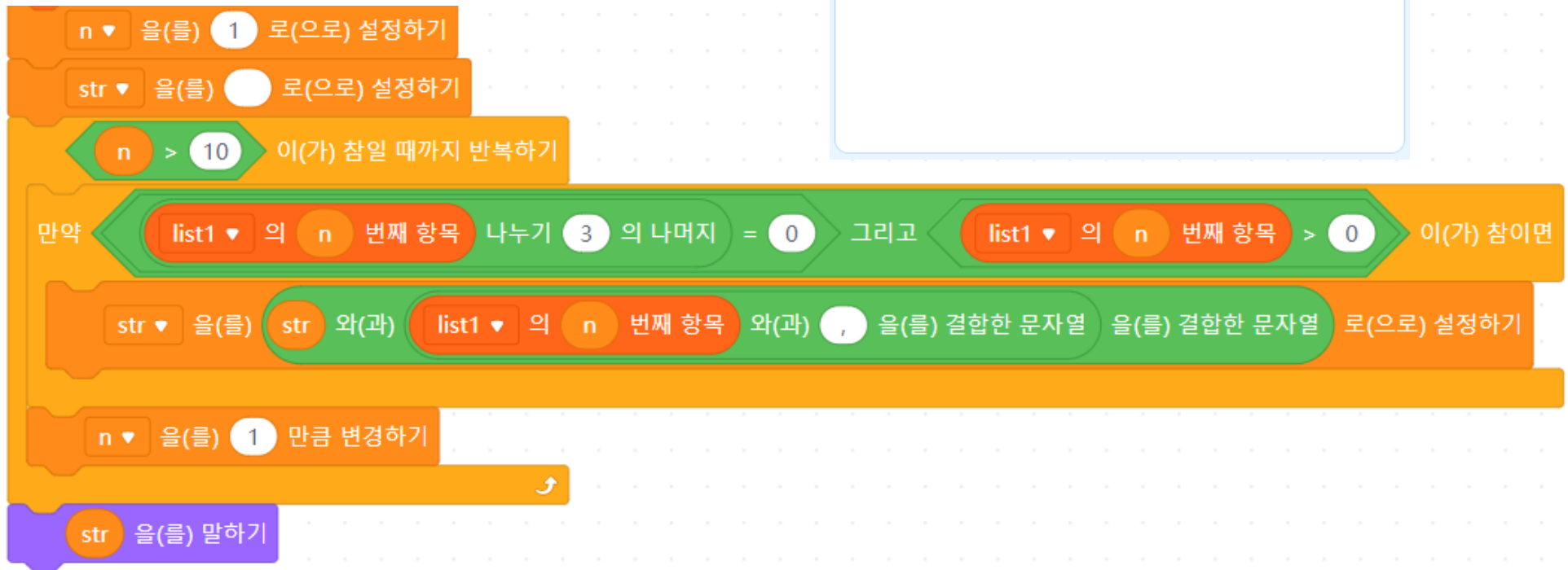
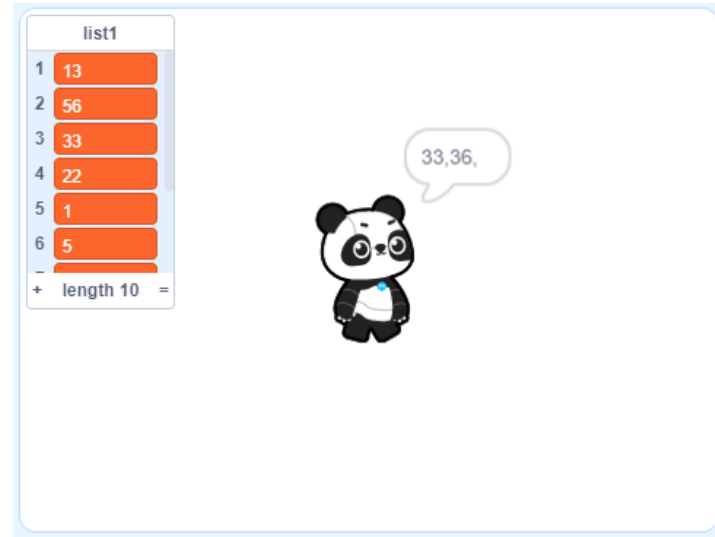
논리설계 실습 - 1

다음 블록프로그램을 보고 출력결과를 써 보시오. (펜더가 하는 말)

The image shows a Scratch script on a grid background. The script starts with a 'when clicked' event block. It then initializes a list 'list1' and adds ten numbers: 13, 56, 33, 22, 1, 5, 36, 7, -3, and 23. It sets a counter 'n' to 1 and an empty string 'str'. A loop 'repeat until n is greater than 10' contains the following logic: an 'if-then' block that checks if the nth item of 'list1' is divisible by 3 (remainder 0) and if it is greater than 0. If both conditions are true, it concatenates the item to 'str' followed by a comma. After the loop, it increments 'n' by 1 and says 'str'.

```
클릭했을 때
모두 삭제 list1
list1에 13 항목을(를) 추가하기
list1에 56 항목을(를) 추가하기
list1에 33 항목을(를) 추가하기
list1에 22 항목을(를) 추가하기
list1에 1 항목을(를) 추가하기
list1에 5 항목을(를) 추가하기
list1에 36 항목을(를) 추가하기
list1에 7 항목을(를) 추가하기
list1에 -3 항목을(를) 추가하기
list1에 23 항목을(를) 추가하기
n을(를) 1로(으로) 설정하기
str을(를) 로(으로) 설정하기
반복하기 until n > 10
  만약 list1의 n번째 항목 나누기 3의 나머지 = 0 그리고 list1의 n번째 항목 > 0 이(가) 참이면
    str을(를) str와(과) list1의 n번째 항목와(과) ,을(를) 결합한 문자열을(를) 결합한 문자열로(으로) 설정하기
  n을(를) 1만큼 변경하기
str을(를) 말하기
```

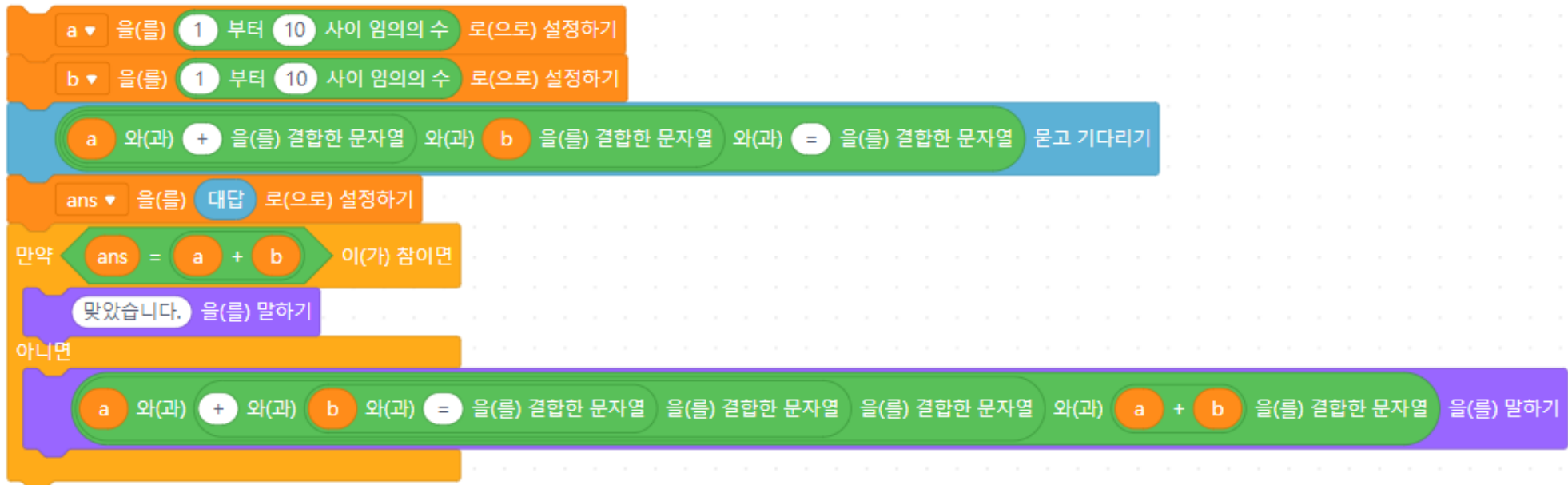
논리설계 실습 - 1



논리설계 실습 - 2

덧셈게임 : 3주차 실습 덧셈 게임을 확장해서
총 10문제를 내고 정답을 맞춘 점수를 출력하는 프로그램

- 덧셈 문제 10문제를 출제
- 사용자 답을 받아 답이 맞으면 점수를 증가시킴
- 점수가 6점이상이면 '훌륭합니다' / 6점미만이면 '노력이 필요합니다' 출력



논리설계 실습 - 2



- ☒ a
- ☒ ans
- ☒ b
- ☒ score



MBlock 실습

스프라이트 동작 제어 블록

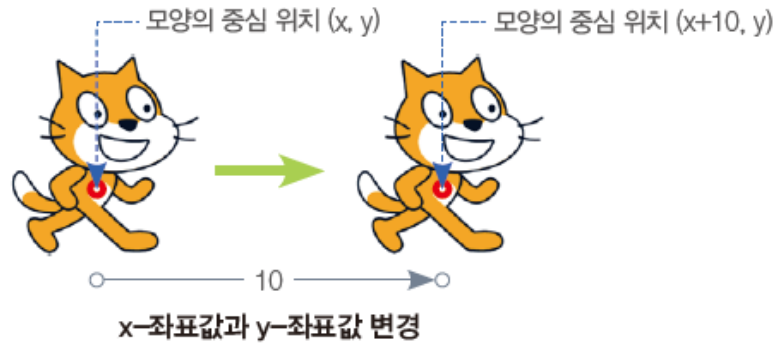
블록	설명
	설정된 값만큼 움직인다. 양수이면 전진, 음수이면 후진한다.
	설정된 각도만큼 시계 방향으로 회전한다.
	설정된 각도만큼 시계 반대 방향으로 회전한다.
	마우스 포인터나 다른 스프라이트 위치 또는 무작위 위치로 이동한다.
	좌표(x, y)로 이동한다.
	설정된 시간 동안 마우스 포인터, 다른 스프라이트, 무작위 위치로 이동한다.
	설정된 시간 동안 좌표(x, y)로 이동한다.
	설정된 방향으로 향한다. 0 : 위쪽, 90 : 오른쪽, 180 : 아래쪽, -90 : 왼쪽

스프라이트 동작 제어 블록

블록	설명
 마우스 포인터 ▾ 쪽 보기	마우스 포인터나 다른 스프라이트 위치로 향한다.
 x 좌표를 10 만큼 바꾸기	x좌표를 설정한 값만큼 변경한다.
 x 좌표를 -98 (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.

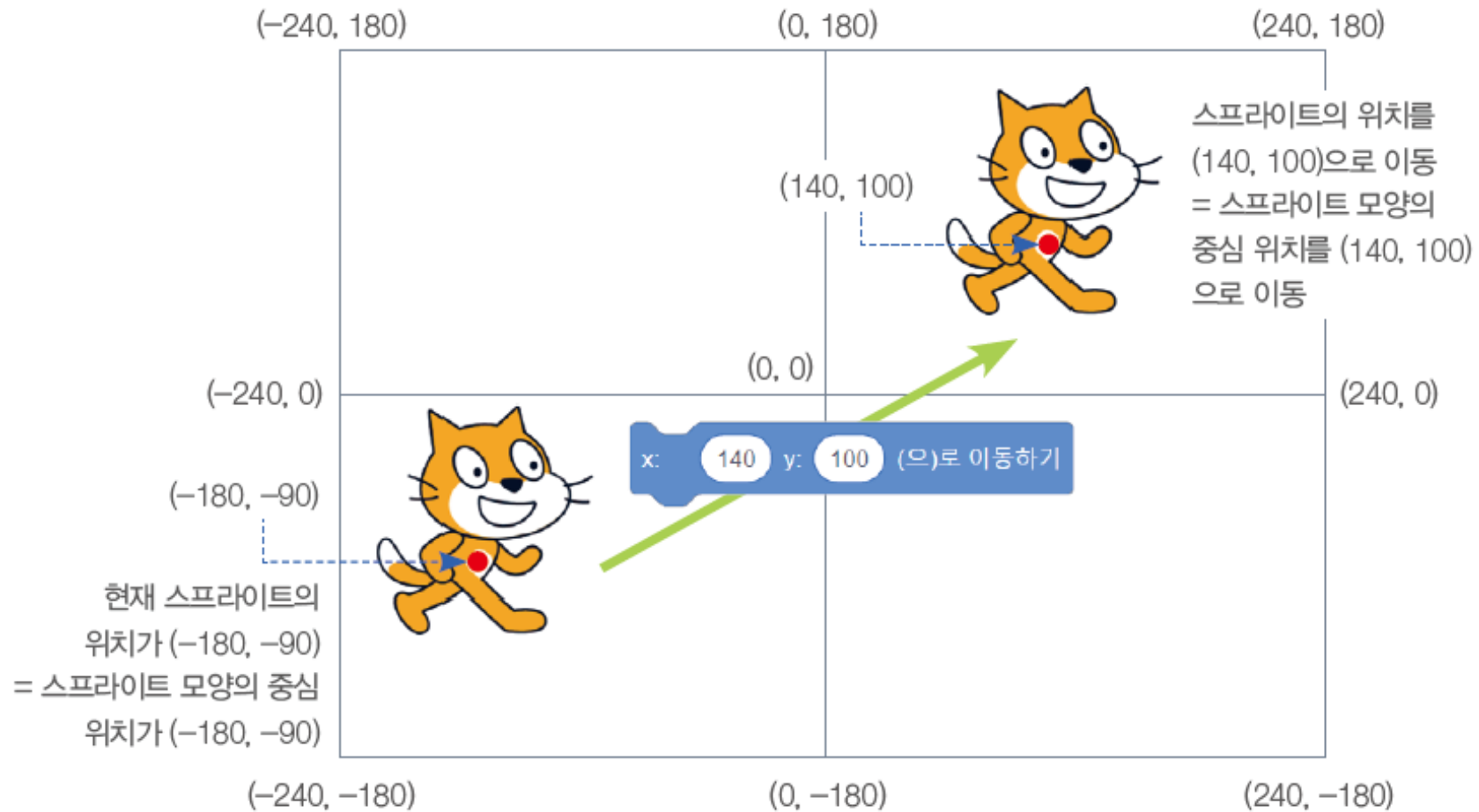
스프라이트 동작 구현

위치 이동



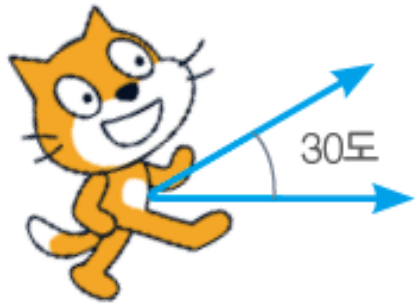
스프라이트 동작 구현

좌표에 의한 이동

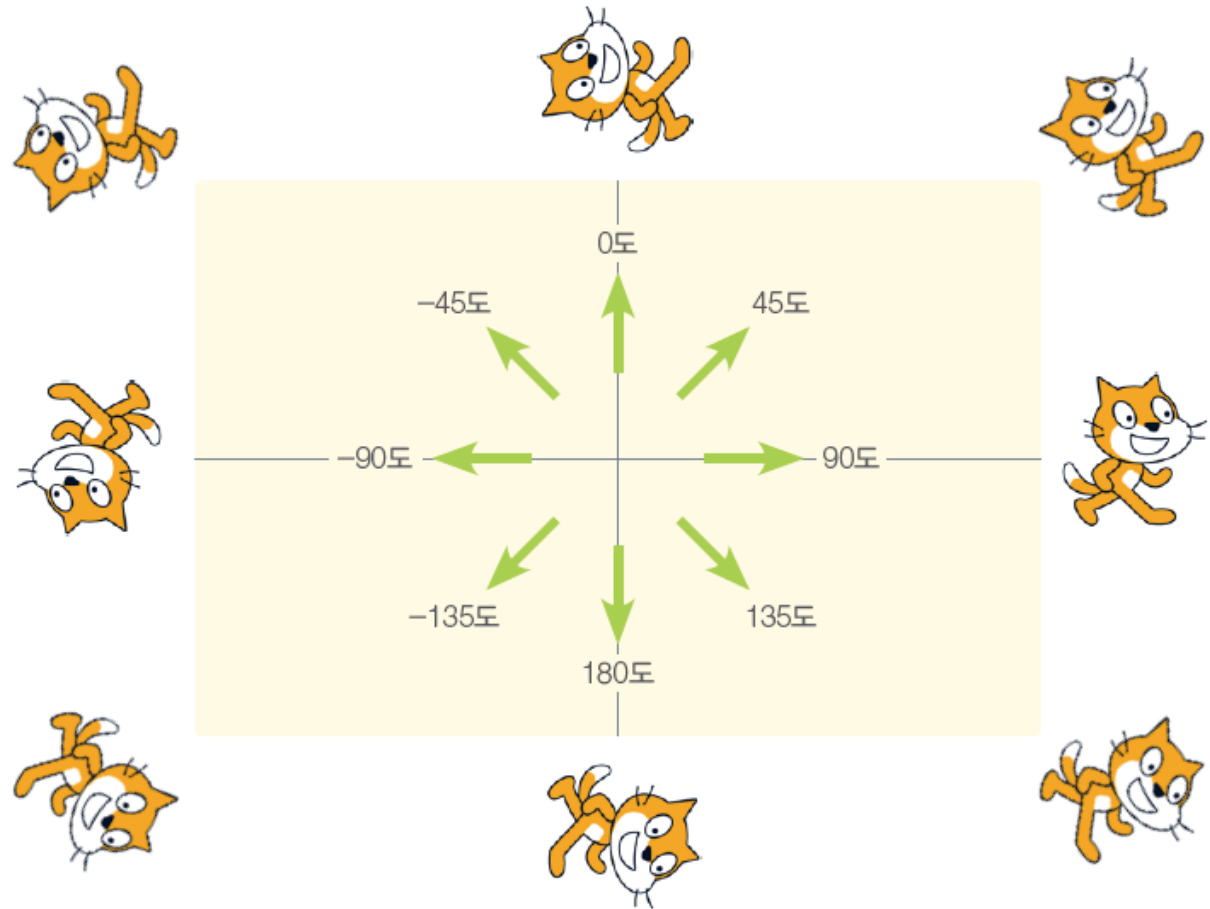


스프라이트 동작 구현

방향 회전 방식















회전할 각도를 정해줌



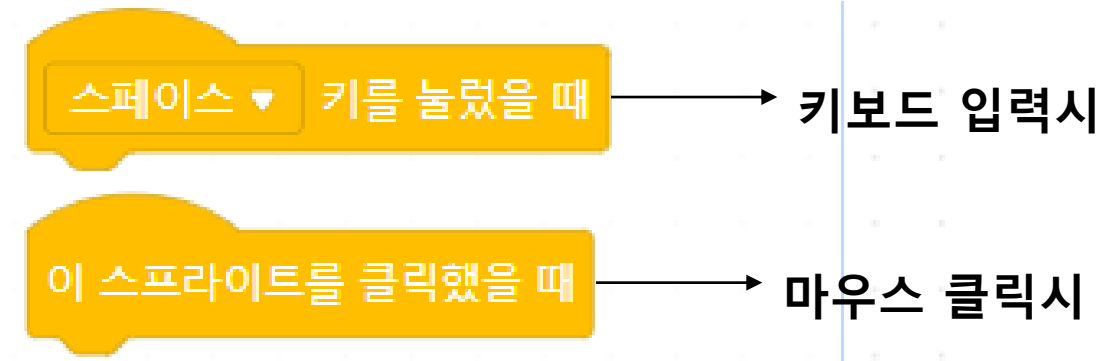
스프라이트 동작 구현

회전 방식 모드 옵션

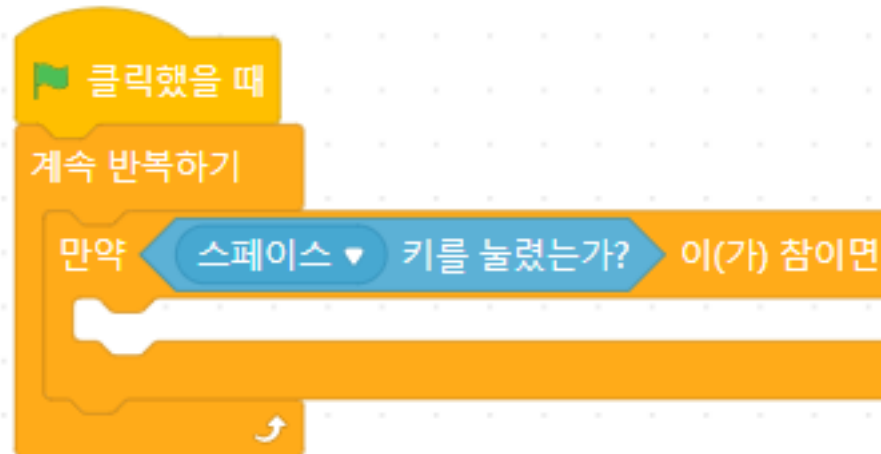
회전 방식 모드	0도	90도	180도	-90도
회전 방식을 회전하기 ▾ (으)로 정하기				
회전 방식을 왼쪽-오른쪽 ▾ (으)로 정하기				
회전 방식을 회전하지 않기 ▾ (으)로 정하기				



키, 마우스 반응

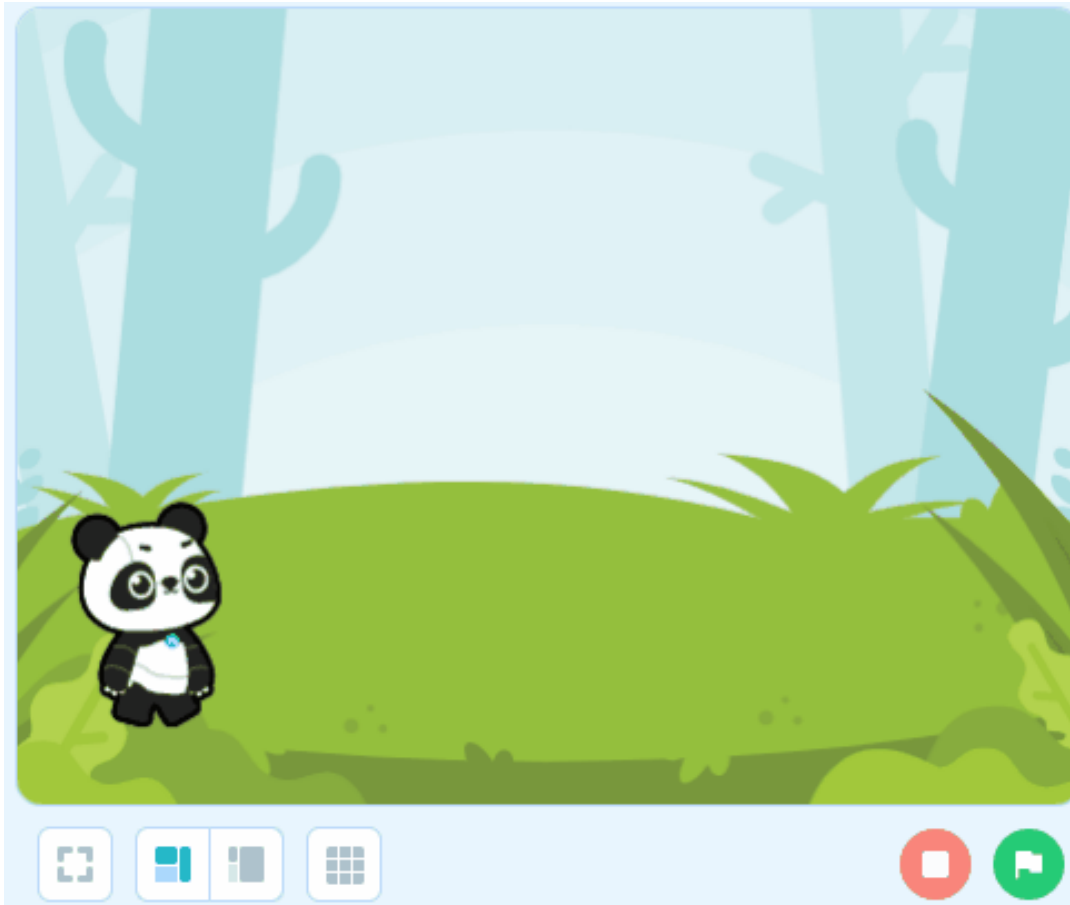


객체를 특정 키에 계속 반응하게 함



실습 - 1

화살표키 방향에 따라 펜더가 한 걸음씩 이동하는 프로그램 구현



클릭했을 때

x: -189 y: -75 로(으로) 이동하기

회전 방식을 왼쪽-오른쪽 ▾ 로 정하기

90 도 방향 보기

오른쪽 화살표 ▾ 키를 눌렀을 때

회전 방식을 왼쪽-오른쪽 ▾ 로 정하기

90 도 방향 보기

10 만큼 움직이기

다음 모양으로 바꾸기

실습 - 1



위쪽 화살표 ▼ 키를 눌렀을 때

회전 방식을 회전하기 ▼ 로 정하기

0 도 방향 보기

10 만큼 움직이기

다음 모양으로 바꾸기

아래쪽 화살표 ▼ 키를 눌렀을 때

회전 방식을 회전하기 ▼ 로 정하기

180 도 방향 보기

10 만큼 움직이기

다음 모양으로 바꾸기

클릭했을 때

x: -189 y: -75 로(으로) 이동하기

회전 방식을 왼쪽-오른쪽 ▼ 로 정하기

90 도 방향 보기

왼쪽 화살표 ▼ 키를 눌렀을 때

회전 방식을 왼쪽-오른쪽 ▼ 로 정하기

-90 도 방향 보기

10 만큼 움직이기

다음 모양으로 바꾸기

오른쪽 화살표 ▼ 키를 눌렀을 때

회전 방식을 왼쪽-오른쪽 ▼ 로 정하기

90 도 방향 보기

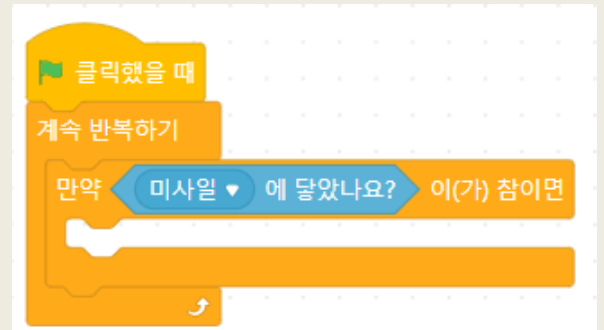
10 만큼 움직이기

다음 모양으로 바꾸기

스프라이트 감지 이벤트

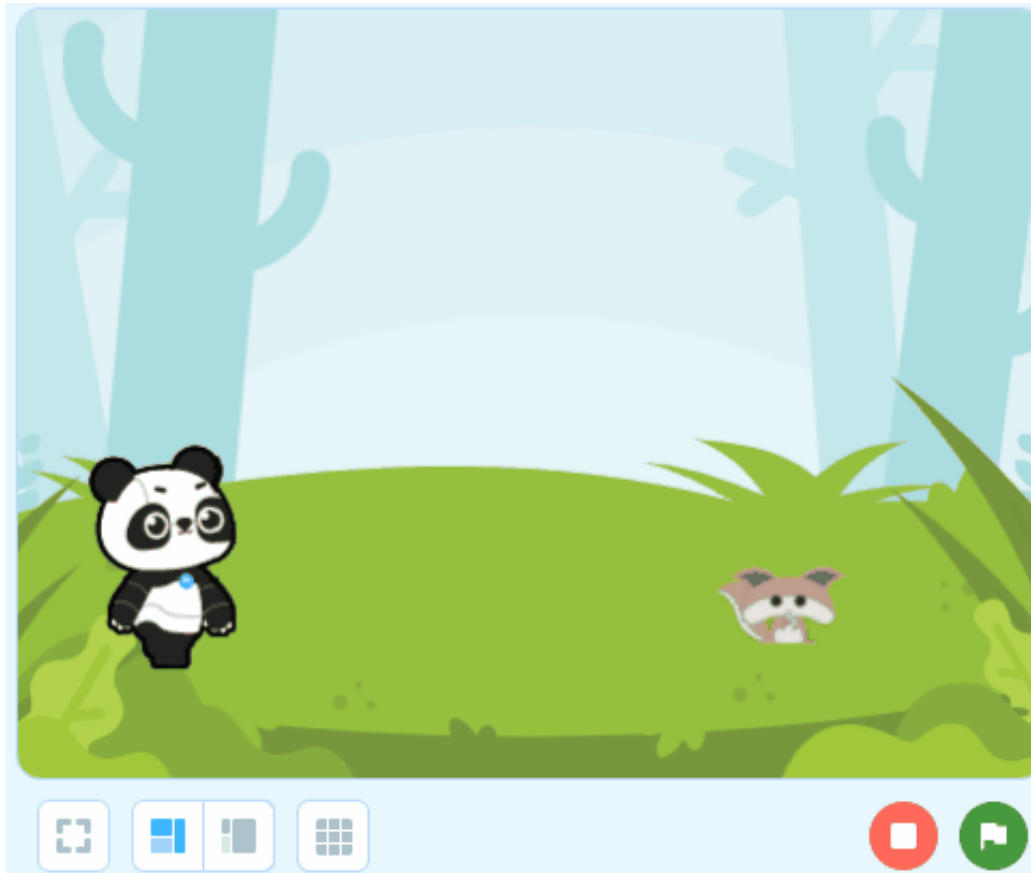
블록	설명
★  에 닿았는가?	스프라이트가 마우스 포인터, 벽, 다른 스프라이트 등에 닿았는지 확인한다.
★  색에 닿았는가?	스프라이트가 설정한 색에 닿았는지 확인한다.
 색이  색에 닿았는가?	스프라이트에 있는 왼쪽에 설정한 색이 오른쪽에 설정한 색에 닿았는지 확인한다.
 까지의 거리	선택한 스프라이트나 마우스 포인터까지의 거리를 확인한다.
 라고 묻고 기다리기	묻고 사용자가 답할 때까지 기다린다.
	사용자가 답한 내용을 저장한다.
★  키를 눌렀는가?	설정한 키가 눌렸는지 확인한다.
	마우스를 클릭했는지 확인한다.
	마우스 포인터의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

객체가 다른 객체에 닿을 때



실습 - 2

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



클릭했을 때

x: -189 y: -75 로(으로) 이동하기

회전 방식을 왼쪽-오른쪽 ▾ 로 정하기

90 도 방향 보기

오른쪽 화살표 ▾ 키를 눌렀을 때

회전 방식을 왼쪽-오른쪽 ▾ 로 정하기

90 도 방향 보기

10 만큼 움직이기

다음 모양으로 바꾸기

만약 Cat17 ▾ 에 닿았나요? 이(가) 참이면

미안 ~~ 을(를) 2 초 동안 말하기

실습 - 2



```
왼쪽 화살표 ▾ 키를 눌렀을 때
회전 방식을 왼쪽-오른쪽 ▾ 로 정하기
-90 도 방향 보기
10 만큼 움직이기
다음 모양으로 바꾸기
만약 Cat17 ▾ 에 닿았나요? 이(가) 참이면
미안 ~~ 을(를) 2 초 동안 말하기
```

```
오른쪽 화살표 ▾ 키를 눌렀을 때
회전 방식을 왼쪽-오른쪽 ▾ 로 정하기
90 도 방향 보기
10 만큼 움직이기
다음 모양으로 바꾸기
만약 Cat17 ▾ 에 닿았나요? 이(가) 참이면
미안 ~~ 을(를) 2 초 동안 말하기
```

```
위쪽 화살표 ▾ 키를 눌렀을 때
회전 방식을 회전하기 ▾ 로 정하기
0 도 방향 보기
10 만큼 움직이기
다음 모양으로 바꾸기
만약 Cat17 ▾ 에 닿았나요? 이(가) 참이면
미안 ~~ 을(를) 2 초 동안 말하기
```

```
아래쪽 화살표 ▾ 키를 눌렀을 때
회전 방식을 회전하기 ▾ 로 정하기
180 도 방향 보기
10 만큼 움직이기
다음 모양으로 바꾸기
만약 Cat17 ▾ 에 닿았나요? 이(가) 참이면
미안 ~~ 을(를) 2 초 동안 말하기
```

```
클릭했을 때
x: -189 y: -75 로(으로) 이동하기
회전 방식을 왼쪽-오른쪽 ▾ 로 정하기
90 도 방향 보기
```

실습 - 2



Cat17

클릭했을 때

계속 반복하기

만약 Panda ♥ 에 달았나요? 이(가) 참이면

아야 !!! 을(를) 2 초 동안 말하기