

초

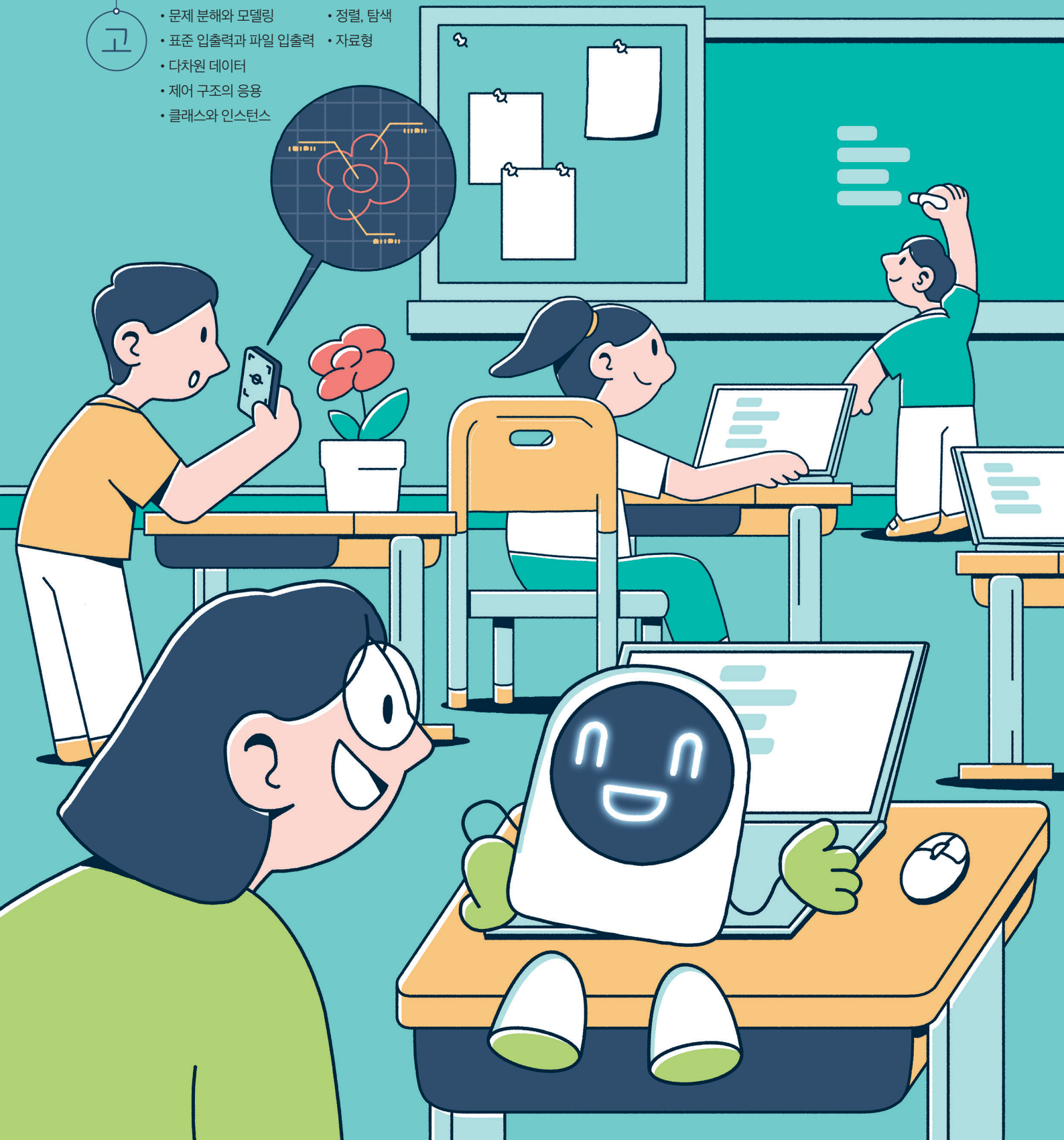
- 문제 찾기와 문제 해결 절차
- 일상생활의 문제를 해결하기 위한 알고리즘 구상
- 문제를 해결하는 기초적인 프로그래밍

중

- 문제 추상화
- 순차적인 데이터 저장
- 중첩 제어 구조
- 알고리즘 표현 방법
- 논리 연산
- 함수와 디버깅

고

- 문제 분해와 모델링
- 표준 입출력과 파일 입출력
- 다차원 데이터
- 제어 구조의 응용
- 클래스와 인스턴스
- 정렬, 탐색
- 자료형



III

알고리즘과 프로그래밍

- 1 알고리즘
- 2 프로그래밍

이 단원을 배우면

실생활에서 문제를 발견하고 추상화하여 알고리즘으로 해결책을 표현할 수 있고, 프로그램을 설계·구현하는 과정에서 자동화의 필요성과 중요성을 이해하고 실천하는 능력을 기를 수 있다.

기억해 볼까요?

초등학교에서 배웠던 내용을 정리해 보자.

컴퓨터를 이용한 문제 해결 과정

문제 해결 과정은 문제 탐색, 문제의 현재 상태와 목표 상태 정의, 알고리즘 작성, 프로그래밍으로 이루어진다.

알고리즘

- 문제를 해결하는 방법이나 절차를 말한다.
- 알고리즘을 표현하는 방법에는 자연어, 의사 코드, 순서도가 있다.

1

알고리즘



가수의 춤 동작을 살펴보면 알고리즘 제어 구조와 유사하게 순차, 반복, 선택이 담겨 있는 것을 발견할 수 있다. 이와 같이 일상생활 속에서 경험할 수 있는 다양한 알고리즘을 생각해 보자.



나의 하루 일과를 알고리즘으로 표현해 볼까?

01

문제의 이해와 추상화

학습 목표

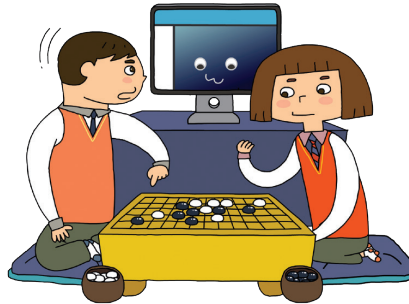
- 문제의 상태를 정의하고 수행 가능한 형태로 구조화할 수 있다.
- 문제 해결을 위한 추상화의 중요성을 이해하고, 핵심 요소를 중심으로 문제 해결 과정을 표현할 수 있다.

미리 보는 학습 흐름도

문제 이해

문제 정의

추상화



컴퓨팅 시스템은 어떤 문제를 해결할 수 있을까?

1 문제의 이해와 정의

문제는 해결해야 하는 일이나 과제를 말한다. 문제의 상태는 문제가 발생한 초기 상태, 문제를 풀어 가는 현재 상태, 문제가 해결된 목표 상태로 구분한다. 초기 상태에서 목표 상태로 가기 위해 현재 상태를 계속 바꾸어 가는 과정을 문제 해결 과정이라 한다.

예를 들어 이삿짐을 엘리베이터로 옮기는 상황을 생각해 보자. 엘리베이터는 최대 200kg까지만 실을 수 있고, 주민들의 불편을 줄이기 위해서는 최소한의 운행 횟수로 이삿짐을 옮겨야 한다.

개념을 잇다

50쪽

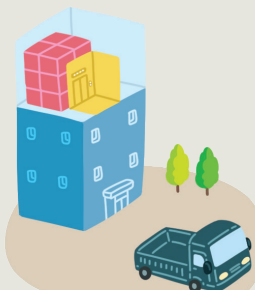
문제에 필요한 데이터의 수집 방법을 생각해 보자.

최소한의 운행 횟수로 이삿짐을 옮기는 문제의 정의

“이삿짐 옮기는 문제의 상태를 어떻게 정의할까?”

초기 상태

이삿짐을 옮기기 전의 상황



현재 상태

엘리베이터로 짐을 옮기는 상황



목표 상태

이삿짐을 모두 옮긴 상황



② 추상화

추상화는 복잡한 것을 간결하고 이해하기 쉽도록 단순화하는 과정을 말한다. 추상화를 통해 복잡한 문제를 해결할 수 있으며, 더 효율적이고 정확하게 문제를 해결하는 데에도 도움이 된다. 추상화를 하려면 문제 발견, 상태 정의, 핵심 요소 추출 등의 과정이 필요하다.

핵심 요소 추출은 문제 해결을 위해 필요한 정보나 요소를 찾아내는 것을 말한다. 핵심 요소를 추출하면 문제를 단순화하고 명확하게 표현할 수 있다. 또한 핵심 요소를 바탕으로 알고리즘을 만들 수 있다.

개념을 잇다 86쪽

알고리즘의 개념을 생각해 보자.



Q 우리가 일상생활에서 발견할 수 있는 추상화의 예에는 어떤 것이 있을까?

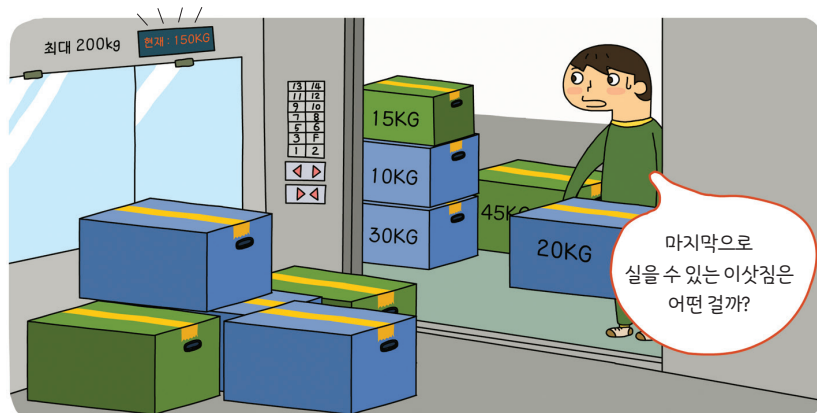
A 수학의 공식, 스마트폰의 아이콘, 지하철 노선도, 데이터의 구조화 등이 추상화된 사례이다.



최소한의 운행 횟수로 이삿짐을 옮기는 문제의 추상화

“엘리베이터로 이삿짐을 옮길 때 어떤 점을 고려해야 할까?”

- 이삿짐의 색상, 종류, 엘리베이터의 제조 회사, 이사 복장 등은 중요하지 않다.
- 엘리베이터에 실을 수 있는 무게나 크기에 맞춰 몇 개의 이삿짐을 옮길 수 있는지만 고려한다.



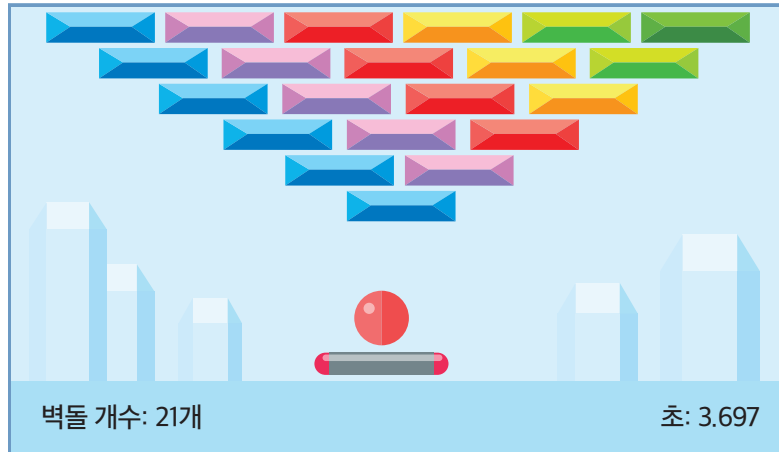
핵심 요소 추출하기

- 다음 문제의 해결에 필요한 핵심 요소를 찾고 그렇게 생각한 이유를 친구들과 이야기해 보자.

1. 벽돌깨기 게임을 만들기 위해 필요한 핵심 요소를 <보기>에서 세 가지 골라 보자.

·보기·

벽돌, 배경 화면, 음악, 공, 벽돌 모양, 공의 크기, 막대, 공의 색깔



데이터의
테이블 구조를
생각해 보자.



2. 학생들의 평균 성적을 구하려고 할 때, 다음 테이블(표)에서 핵심 요소를 찾아보자.

학번	이름	전화번호	취미	국어 점수	정보 점수	수학 점수	영어 점수
1	박◆민	○○○-○○○○	춤	100	100	80	70
2	김태●	△△△-△△△△	음악 감상	50	60	50	90
3	전○국	□□□-□□□□	요리	60	50	70	50
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

스스로 확인하기

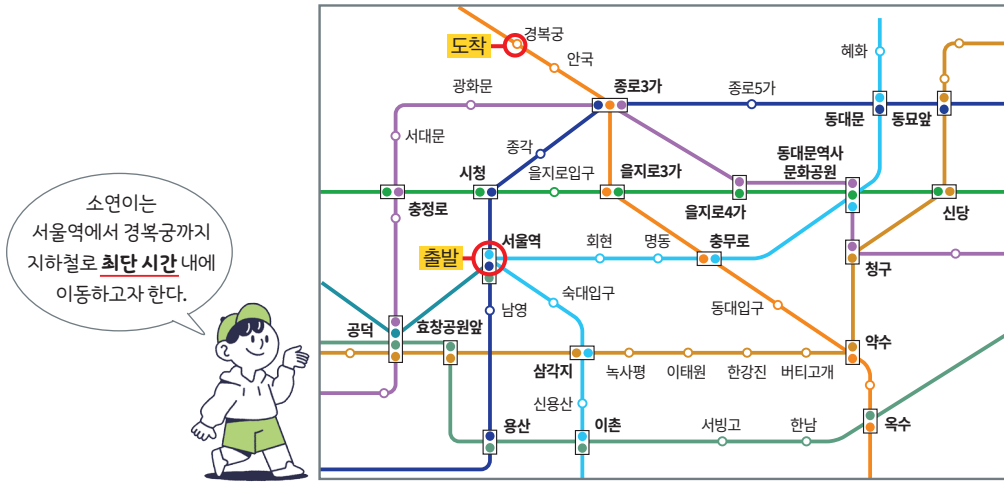
1 문제의 상태는 [], [], [] (으)로 구분할 수 있다.

2 []은/는 복잡한 것을 간결하고 이해하기 쉽도록 단순화하는 과정을 말한다.

복합성 2, 비유 표현, 비유 사용, 비유 사용 1

문제를 추상화하여 해결하기

다음 문제 상황을 읽고 추상화하여 해결해 보자.



※ 역 간 이동 시간은 2분으로 동일하며, 환승에 걸리는 시간은 4분이다.

1

주어진 상황에서 초기 상태, 현재 상태, 목표 상태를 정의해 보자.

- 초기 상태:
- 현재 상태: 도착역으로 이동하는 상태
- 목표 상태:

2

다음 중 문제를 해결하기 위한 핵심 요소에 해당하는 것을 찾아 표시해 보자.

- | | | | | |
|------------------------------------|---------------------------------|--------------------------------|------------------------------------|---------------------------------|
| ☐ 이동 역 개수 | ☐ 역 이름 | ☐ 노선 색깔 | ☐ 역 간 이동 시간 | ☐ 환승 시간 |
| ☐ 역의 위치(순서) | ☐ 지하철 번호 | ☐ 환승 횟수 | ☐ 환승역의 규모 | ☐ 지하철 구조 |
| ☐ 도착역 | ☐ 출발역 | ☐ 날씨 | ☐ 경복궁의 역사 | ☐ 신발 크기 |

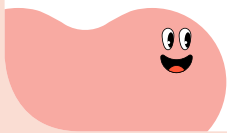
3

출발역에서 도착역까지 소요되는 시간을 정리하고 최적 경로와 이동 시간을 찾아보자.

→ 최적 경로와 이동 시간:

활동 평가하기

항목	평가
• 문제의 초기 상태, 현재 상태, 목표 상태를 정의하였는가?	☺ ☹ ☹
• 문제 해결에 필요한 핵심 요소를 적절하게 찾았는가?	☺ ☹ ☹
• 핵심 요소를 바탕으로 문제를 분석하여 최적의 경로와 이동 시간을 찾았는가?	☺ ☹ ☹



“문제를 추상화해서 문제 해결 과정을 표현해 볼까?”

초등학교에서 문제 해결
과정을 자연어, 의사 코드, 순서도로
표현하는 방법을 배웠어.



숫자를 입력받아 홀짝을 맞히는 프로그램을 생각해 보자.

- 초기 상태: 홀짝을 모르는 상태
- 목표 상태: 홀짝을 맞힌 상태

자연어

특별한 형식 없이 우리가 사용하는 일상적인 언어로 알고리즘을 표현하는 방법이다.

“임의의 숫자를 입력받아 입력받은 수를 2로 나눈 나머지가 0이면 짝수,
아니면(나머지가 1이면) 홀수를 출력한다.”

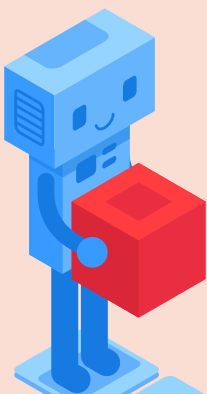
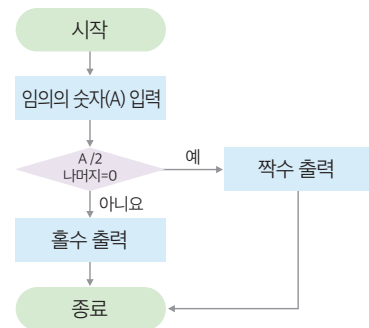
의사 코드

프로그래밍 언어와 비슷한
형태로 알고리즘을 표현하는 방법으
로, 프로그래밍 언어로 변환하기 쉽다.

- 1 임의의 숫자(A)를 입력
- 2 A 나누기 2의 나머지가 0이면
짝수 출력
- 3 그렇지 않으면
홀수 출력
- 4 종료

순서도

정해진 기호를 이용하여
알고리즘을 표현하는 것으로 전체 흐름을
한눈에 알 수 있다.



02

알고리즘 기초

학습 목표

- 알고리즘의 중요성을 이해할 수 있다.
- 문제를 해결하는 다양한 알고리즘을 비교·분석할 수 있다.

미리 보는 학습 흐름도

알고리즘의 개념

알고리즘의 중요성

알고리즘의 비교·분석



🔍 섞여 있는 카드를 어떻게 하면 빠르게 정리할 수 있을까?



1 알고리즘의 개념과 중요성

개념을 잇다

82쪽

추상화의 개념을 생각해 보자.

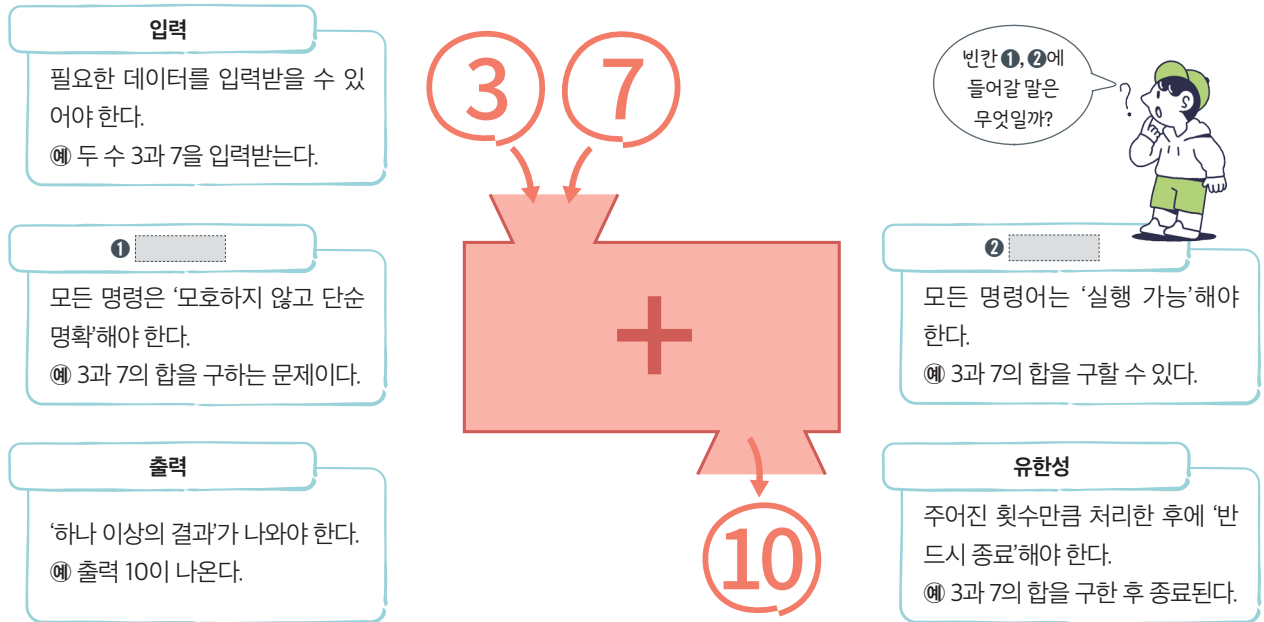
알고리즘은 특정 문제를 해결하는 절차나 방법이다. 알고리즘으로 표현하기 위해서는 추상화 과정을 거쳐 추출한 핵심 요소를 초기 상태에서 목표 상태로 변화시키는 과정이 필요하다.

일상생활의 문제를 해결할 때도 알고리즘은 중요한 역할을 한다. 요리하는 순서, 로그인하는 과정, 자동판매기의 작동, 길 찾기 문제 등에도 알고리즘이 활용되고 있다.

일상생활 속의 다양한 알고리즘



주어진 문제를 빠르고 효율적으로 해결하기 위해 필요한 알고리즘은 프로그래밍에서 매우 중요한 역할을 한다. 컴퓨팅 시스템에 활용하는 알고리즘은 ‘입력’, ‘출력’, ‘명확성’, ‘수행 가능성’, ‘유한성’ 등의 조건을 갖추어야 한다.



▲ 그림 III-1 알고리즘의 조건

유익한 글씨 © 유익한 글씨



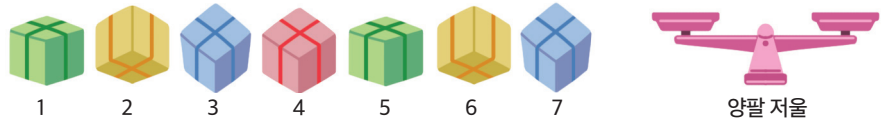
● 효율적인 알고리즘

컴퓨팅 시스템에서 효율적인 알고리즘을 설계하기 위해서는 처리 시간 또는 저장 공간이 적게 소요되는 알고리즘을 설계해야 한다.

② 알고리즘의 비교·분석

주어진 문제를 해결하는 알고리즘은 여러 가지일 수 있다. 어떤 알고리즘을 사용하느냐에 따라 효율성이 달라질 수 있으므로 여러 가지의 알고리즘을 비교·분석한 후 효율적인 알고리즘을 선택해야 한다.

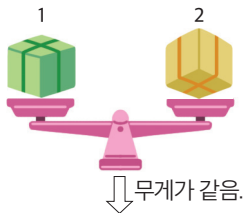
7개의 제품 중에서 불량으로 무게가 가벼운 제품이 하나 있다. 양팔 저울을 사용하여 가벼운 제품 하나를 빠르게 찾아내는 알고리즘을 생각해 보자.



알고리즘 1 두 개씩 비교하기

양팔 저울에 제품을 순서대로 두 개씩 올려 비교하면서 가벼운 것을 찾아본다.

● 1단계 1번, 2번 제품 비교



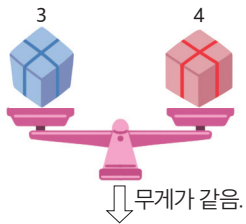
무게가 다름.



1번 만에 찾기 완료

한 번 비교하는데 걸리는 시간이 같다면 비교 횟수가 적을 때 시간이 적게 걸리니 더 효율적이라고 할 수 있어.

● 2단계 3번, 4번 제품 비교

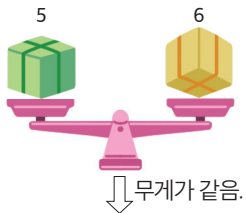


무게가 다름.



2번 만에 찾기 완료

● 3단계 5번, 6번 제품 비교



무게가 다름.



3번 만에 찾기 완료

최대 비교 횟수 3회

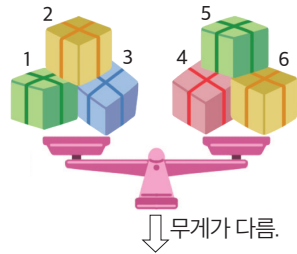


→ 최대 3번 비교해야 가벼운 제품을 찾을 수 있다.

알고리즘 2 반으로 나누어 비교하기

양팔 저울에 제품의 반씩 나누어 올려 비교하면서 가벼운 것을 찾아본다.

1단계 세 개씩 나누어 비교(1~3, 4~6번 제품 비교)



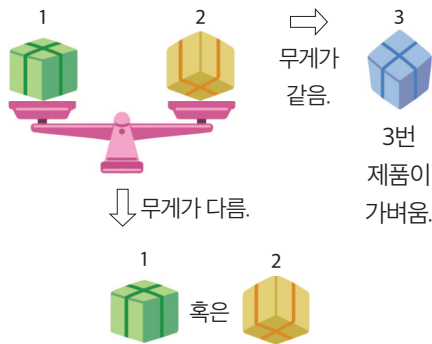
무게가 같음.



1번 만에
찾기 완료

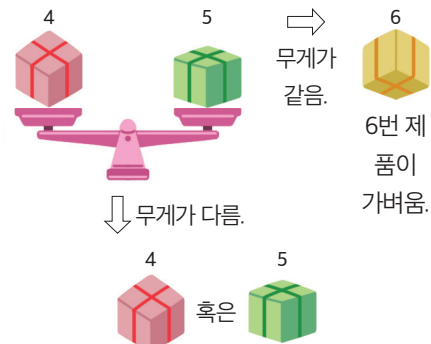
2단계 1단계에서 가벼운 쪽 제품 중 두 개의 제품 무게 비교

• 1~3번 제품이 가볍다면



양팔 저울이 올라간 쪽이 가벼운 제품임.

• 4~6번 제품이 가볍다면



양팔 저울이 올라간 쪽이 가벼운 제품임.

2번 만에
찾기 완료

최대 비교
횟수 2회

→ 최대 2번 비교해야 가벼운 제품을 찾을 수 있다.

알고리즘 1 은 최대 3번의 비교가 필요하며, 알고리즘 2 는 최대 2번의 비교가 필요하다. 제품의 개수가 늘어나면 알고리즘 1 과 알고리즘 2 의 비교 횟수는 더 많은 차이를 보인다. 여러 개의 제품 중에서 한 개의 가벼운 제품을 찾는 문제에서는 알고리즘 2 가 더 좋은 알고리즘이다.



100개의 제품에서 한 개의 가벼운 제품 찾기

• 제품이 100개 있을 때 위의 알고리즘 1 과 알고리즘 2 를 참고하여 비교 횟수를 적어 보자.

→ 알고리즘 1 의 최대 비교 횟수 _____ 번

→ 알고리즘 2 의 최대 비교 횟수 _____ 번





프로그래밍으로 알고리즘의 효율성 살펴보기

1부터 100까지 더하는 알고리즘의 효율성을 프로그래밍으로 비교해 볼 수 있다.

알고리즘 1 1부터 순서대로 더하기

$$1 + 2 + 3 + 4 + \dots + 99 + 100$$

3 6 10 ... ? ?
1번 2번 3번 99번



| 실행 시간 |



알고리즘 2 패턴을 발견하여 더하기

$$1 + 2 + 3 + \dots + 50 + 51 + \dots + 98 + 99 + 100$$

1+100=101
2+99=101
3+98=101
...
50+51=101



| 실행 시간 |



알고리즘 1의 방법으로 프로그래밍한 후 실행한 결과는 1.6초이고
알고리즘 2의 방법으로 프로그래밍한 후의 실행 결과는 0.0초로 측정되었어.
따라서 시간 측면에서 알고리즘 2가 더 효율적이야. 이와 같이 결과가
같더라도 어떤 알고리즘을 선택하느냐에 따라 효율성이 달라질 수 있어.



- 1 은/는 문제를 해결하는 방법과 절차를 말한다.
- 2 섞여 있는 숫자에서 한 개의 숫자를 찾을 때 일반적으로 순서대로 찾는 방법과 반씩 나누어 찾는 방법 중 순서대로 찾는 방법이 더 효율적이다. (○, X)

X '2 문법공부

숫자 맞추기 게임하기



두 명의 친구 중 한 명이 1부터 100까지의 수 중에서 숫자 하나를 생각한다. 나머지 한 명이 다음 알고리즘대로 숫자를 불러 친구가 생각한 수를 맞추는 게임을 해 보고 다음 물음에 답해 보자.

알고리즘 1 차례차례 맞춰 볼까?



알고리즘 2 반으로 나누면서 맞춰 볼까?



※ 생각한 숫자가 친구가 말한 숫자보다 크면 '업(up)', 작으면 '다운(down)', 같으면 '정답입니다.'라고 말한다.

1

위 게임을 짝지어 해 보고 **알고리즘 1** 과 **알고리즘 2** 를 사용했을 때 몇 번 만에 숫자를 맞힐 수 있는지 적어 보자.

내가 생각한 숫자 _____ **알고리즘 1** _____ 번 **알고리즘 2** _____ 번
친구가 생각한 숫자 _____ **알고리즘 1** _____ 번 **알고리즘 2** _____ 번

2

알고리즘 1 과 **알고리즘 2** 중 더 효율적이라고 생각하는 알고리즘을 고르고, 그 이유를 적어 보자.

→ 더 효율적인 알고리즘은? _____

→ 그 이유는? _____

활동 평가하기

항목	평가
• 알고리즘별로 비교 횟수를 찾았는가?	😊 😐 😞
• 더 효율적인 알고리즘이 무엇인지 선택하고 그 이유를 설명하였는가?	😊 😐 😞

03

알고리즘 설계

학습 목표

- 문제 해결에 적합한 전략을 선택하여 알고리즘을 설계할 수 있다.

미리 보는 학습 흐름도

줄서기 문제의
알고리즘 설계

↓
신발 찾기 문제의
알고리즘 설계



🔍 줄을 서지 않아도 먼저 온 사람에게 우선 순위를 주는 방법은?



① 줄서기 문제의 알고리즘 설계

알고리즘 설계는 문제를 해결하는 과정을 계획하는 것이다. 물건을 고른 후 계산하기 위해 계산대 앞에 줄을 선 상황을 생각해 보고, 물건을 빠르게 계산하는 알고리즘을 설계해 보자.

알고리즘 설계는 복잡한 문제를 해결할 때 꼭 필요한 과정이야.



문제 이해

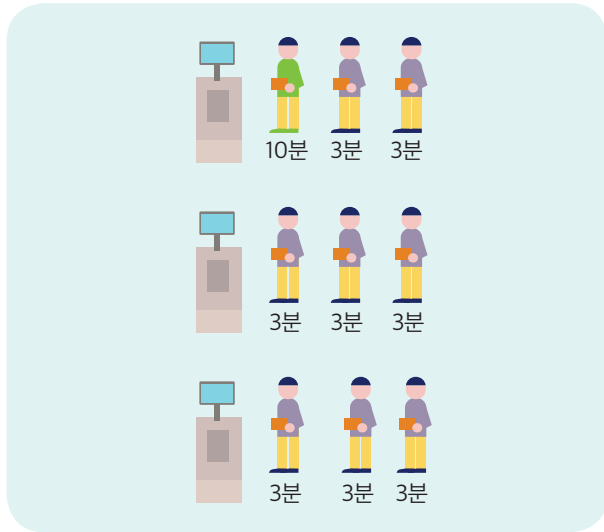
- 문제 정의
- 초기 상태: 계산대 앞에서 계산을 기다리는 상태
- 목표 상태: 모든 손님의 계산이 완료된 상태

- 마트에서 물건을 고른 후 계산하려고 한다.
- 손님은 9명이고 계산대는 3대가 있다.
- 가장 먼저 온 손님 1명은 계산해야 할 물건이 많아 시간이 10분 정도 소요되며, 나머지 8명의 손님은 3분 정도 소요된다.
- 빠른 시간에 모든 손님이 계산을 마칠 수 있는 방법을 찾아보자.

다음 **알고리즘 1**, **알고리즘 2** 를 적용하여 모든 손님이 계산을 마치는 데 걸리는 시간을 알아보자.

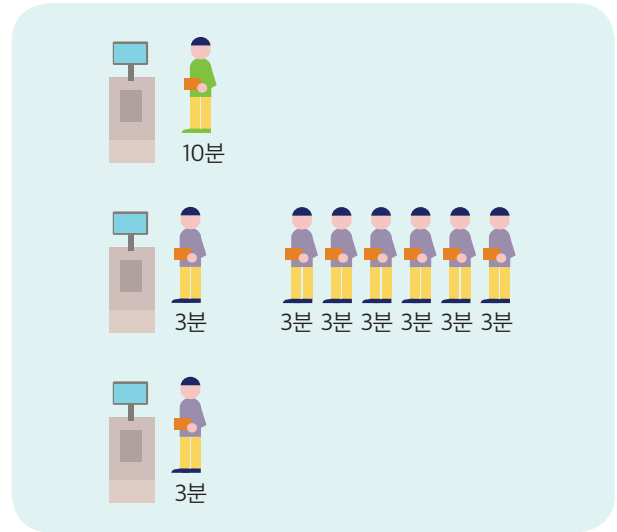
알고리즘 1 여러 줄 서기

계산대마다 3명씩 줄을 서서 계산한다.



알고리즘 2 한 줄 서기

한 줄로 서 있다가 빈 계산대가 생기면 이동하여 계산한다.



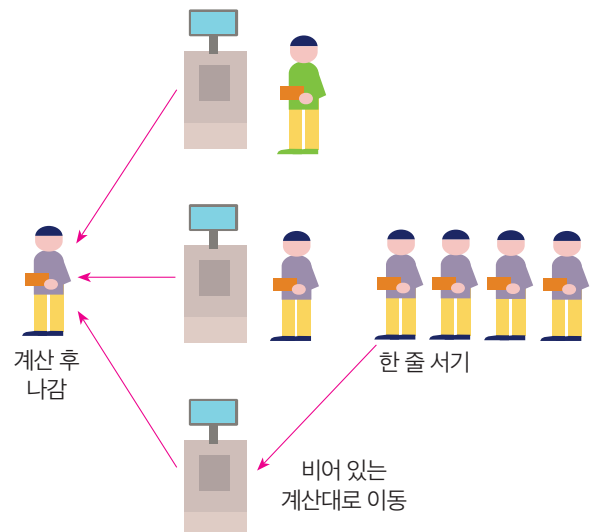
알고리즘 1 은 계산대마다 3명의 손님이 줄을 서서 계산하므로 첫 번째 손님이 있는 계산대의 소요 시간이 가장 오래 걸린다. 그러므로 모든 손님이 계산을 마치는 데 최대 16분이 걸린다. 그러나 **알고리즘 2** 는 한 줄로 서 있다가 앞의 손님이 계산을 마치면 다음 손님이 빈 계산대에서 계산할 수 있으므로 최대 12분이 걸린다. **알고리즘 2** 를 이용하면 시간이 적게 걸려 더 효율적임을 알 수 있다.

알고리즘 2 를 이용하여 문제를 해결하는 과정을 표현하면 다음과 같다.

개념을 잇다 85쪽

알고리즘을 표현하는 방법을 생각해 보자.

- ① 모두 계산을 마칠 때까지 ②~⑤ 반복
- ② 한 줄로 손님 대기
- ③ 만약 비어 있는 계산대가 있다면
- ④ 계산대로 이동
- ⑤ 계산 후 나감



② 신발 찾기 문제의 알고리즘 설계

신발 가게에 신발을 사기 위해 두 명의 손님이 왔다. 한 쪽은 정리가 되지 않아 뒤죽박죽이고 다른 한쪽은 크기순으로 정리되어 있다. 어느 쪽에서 어떻게 찾아야 원하는 크기의 신발을 빨리 찾을 수 있을지 생각해 보자.



● 문제 정의

- 초기 상태: 사이즈에 맞는 신발을 찾으려는 상태
- 목표 상태: 사이즈에 맞는 신발을 찾은 상태

문제 이해

- 손님은 각각 230mm와 250mm의 신발을 찾고 있다.
- 신발 가게의 신발이 한 쪽은 정리되지 않은 상태이고 다른 쪽은 정리되어 있다.

알고리즘 1 정리 안 된 곳에서 하나씩 비교하며 찾기

정리되지 않은 신발 더미에서 한 켤레씩 꺼내 확인하면서 230mm와 250mm의 신발을 찾는다.



알고리즘 2 정리된 곳에서 비교하며 찾기

크기순으로 정리된 곳에서 230mm와 250mm의 신발을 찾는다.



① 앞에서 하나씩 비교하며 230mm의 신발을 찾는다.



② 신발을 하나씩 비교하면서 250mm를 찾는다. 이때 이전 단계인 230mm 신발을 찾은 이후부터 하나씩 비교하면서 찾는다.



Q 알고리즘에서 '정리하기'는 무엇을 의미할까?

A 데이터를 크기나 규칙순으로 나열하는 것으로 '정렬 알고리즘'이라고 한다.

Q 알고리즘에서 '찾기'는 무엇을 의미할까?

A 데이터에서 원하는 값을 찾는 것으로 '탐색 알고리즘'이라고 한다.

문제를 해결할 때 필요한 데이터를 빨리 찾기 위해서는 데이터를 순서대로 정렬해 놓는 것이 필요할 때가 있어.



알고리즘 1 은 신발을 한 켤레씩 비교해야 하므로 신발이 많으면 시간이 오래 걸린다. 만일 신발이 100켤레 있다면 최악의 경우 197번 비교해야 두 켤레를 찾을 수 있다. **알고리즘 2** 는 신발을 정리해 놓았기 때문에 두 번째 신발을 찾을 때는 첫 번째 신발을 찾은 다음 위치부터 찾으면 되므로 찾는 시간을 줄일 수 있다.



1 알고리즘 설계는 문제를 해결하는 과정을 만드는 것이다.

(○, X)

2 수행 시간이 많이 걸리는 알고리즘이 효율적인 알고리즘이다.

(○, X)

X Z O L

역량 준비하기

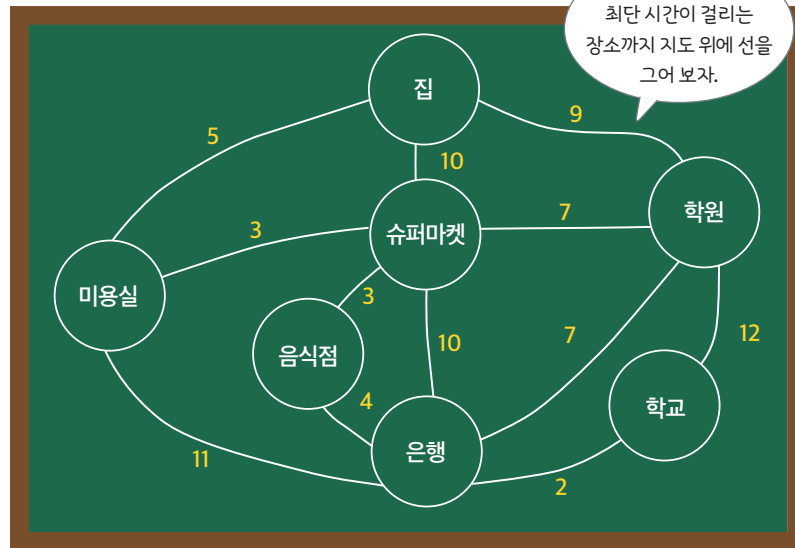
활동 1

집에서 학교까지 가는 빠른 길 찾기

다음 알고리즘을 이용해 집에서 학교까지 가는 가장 빠른 경로를 찾아 지도 위에 표시해 보자.



추상화



매 단계에서
최단 시간이 걸리는
장소까지 지도 위에 선을
그어 보자.

지도를 추상화하기
위해 건물은 동그라미로,
건물과 건물 사이의 길은 선으로
연결한 다이어그램으로
표현하였어.



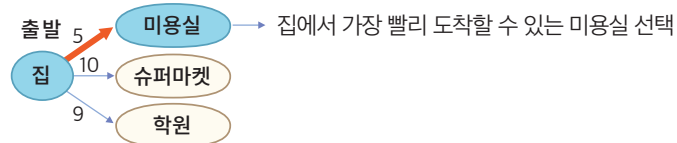
알고리즘

- ❶ 집에서 이동 가능한 장소까지의 시간을 탐색한다.
- ❷ ❸~❹를 반복한다.
- ❸ 탐색된 장소 중 집에서부터 최단 시간이 걸리는 장소를 선택한다.(단, 한번 선택했던 장소는 제외한다.)
- ❹ 선택한 곳에서 갈 수 있는 장소와 시간을 탐색한다.

1

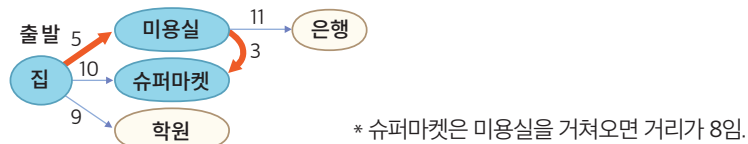
단계별로 이동 경로와 이동 시간을 표시해 보자.

- **1단계** 집에서 갈 수 있는 곳은 미용실, 슈퍼마켓, 학원이다. 집에서부터의 시간을 계산하여 최단 시간 경로를 진하게 표시한다.



- 탐색된 경로 중(미용실: 5, 슈퍼마켓: 10, 학원: 9) 최단 시간 경로는 () (으)로 가는 길이다.
- 현재까지의 경로: 집 → ()

- **2단계** 미용실에서 갈 수 있는 곳은 슈퍼마켓과 은행이다. 집에서부터의 시간을 계산하여 최단 시간 경로를 화살표로 표시한다.



- 탐색된 경로 중(슈퍼마켓: 8, 학원: 9, 은행: 16) 최단 시간 경로는 () (으)로 가는 길이다.
- 현재까지의 경로: 집 → () → ()

2

위와 같은 방법으로 집에서 학교까지 가는 최단 시간 경로를 찾고 걸린 시간을 적어 보자.

- (1) 경로: ()
- (2) 시간: ()분

활동 평가하기

항목	평가
• 집에서 학교까지의 최단 시간 경로를 찾을 수 있는가?	☺ ☹ ☹
• 집에서 학교까지 이동하는 데 몇 분이 걸리는지 계산할 수 있는가?	☺ ☹ ☹

강 건너기 게임 알고리즘 설계하기

농부가 개, 염소, 배추를 가지고 강을 건너려고 한다. 다음 조건을 고려하여 배 한 척으로 모두 무사히 건널 수 있는 방법을 찾아보자.  부록 249~250쪽 강 건너기 게임

조건

- ❶ 농부는 개, 염소, 배추 중에서 하나만 배에 태울 수 있다.
- ❷ 개와 염소만 있으면 개가 염소를 물어 버린다.
- ❸ 염소와 배추만 있으면 염소는 배추를 먹어버린다.



1

초기 상태에서 알 수 있는 정보나 사실 중 필요한 것만 선택해 보자. 그리고 추가로 알게 된 사실도 적어 보자.

예 하나만 배에 태울 수 있다.

2

부록의 강 건너기 보드 게임을 하며 조건에 맞는 강 건너기 규칙을 정리해 보자.

3

강을 건너는 횟수를 최소한으로 했을 때의 알고리즘을 설계해 보자.

※ 강을 건너가는 것과 건너오는 것은 각각 1회씩으로 계산한다.

횟수	강 건너가기	횟수	강 건너오기
1	예 (농부, 염소)가 건너간다.	2	()가 건너온다.
3		4	
5		6	
7		8	
⋮		⋮	

활동 평가하기

항목	평가
• 문제 해결에 필요한 핵심 요소를 추출할 수 있는가?	😊 😐 😞
• 문제 해결 과정을 순서대로 적고 문제를 해결하였는가?	😊 😐 😞

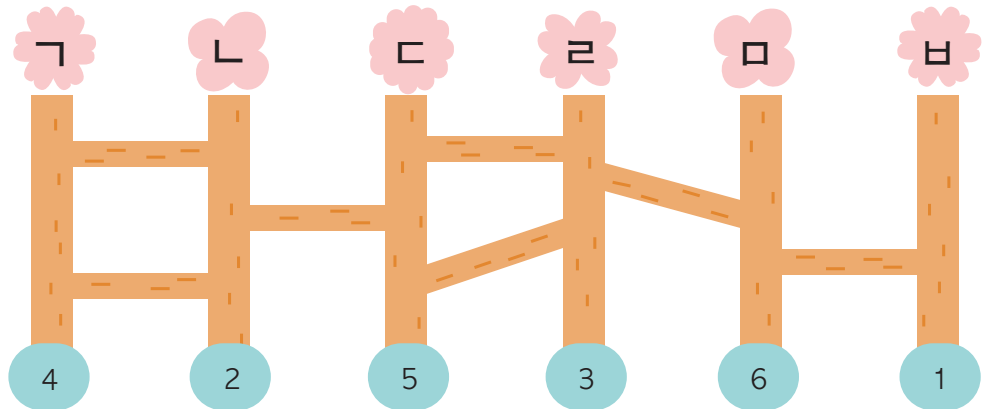
스스로 정리하기

컴퓨팅 시스템의 문제 해결 과정을 설명한 글이다. 빈칸에 들어갈 말을 <보기>에서 찾아 적고 정답이 맞는지 사다리 타기로 알아보자.

보기

① 알고리즘 ② 추상화 ③ 핵심 요소 추출 ④ 현재 상태 ⑤ 목표 상태 ⑥ 효율적인 알고리즘

문제를 해결하기 위해서는 문제 해결에 필요한 것은 남기고 불필요한 내용은 제거하는 ① 과정을 거친다. 이 과정을 거쳐 문제를 단순화시키는 과정을 ② (이)라고 한다. 문제를 해결하기 위한 순서와 절차를 표현해 놓은 것을 ③ (이)라고 하며, 이때 문제가 발생한 초기 상태에서 문제를 풀어 가는 ④, 문제가 해결된 ⑤ 을/를 구분하여 정의하는 것이 중요하다. 알고리즘을 설계할 때는 시간이 적게 걸리고, 저장 공간을 적게 사용하는 ⑥ 을/를 선택해야 한다.



알고리즘은 문제의 초기 상태에서 목표 상태로 변화하는 과정을 정리한 것이다. 나의 일상을 돌아보며 80쪽의 질문 Q를 다시 한번 생각해 볼까?



스스로 평가하기

범주	내용 요소	예	아니요
지식·이해	문제를 추상화할 수 있는가?	☐	81~84쪽
	알고리즘 표현 방법을 이해할 수 있는가?	☐	85쪽
과정·기능	문제를 초기 상태, 현재 상태, 목표 상태로 정의하고 해결 가능한 형태로 구조화할 수 있는가?	☐	81~84쪽
	문제 해결을 위한 다양한 알고리즘을 설계하고 적용할 수 있는가?	☐	92~98쪽
가치·태도	문제 분석으로 추상화의 중요성을 이해하고 실생활 문제를 해결할 수 있는가?	☐	81~85쪽
	문제 해결을 위한 다양한 해법을 탐색하고 알고리즘으로 표현할 수 있는가?	☐	86~98쪽

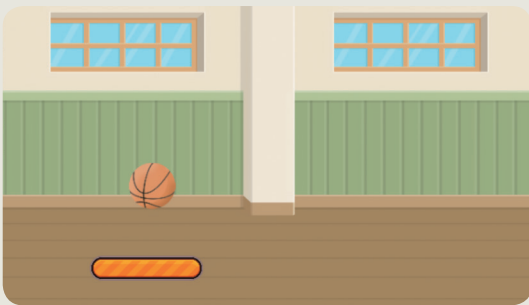
2

프로그래밍



컴퓨팅 시스템에 명령을 내리는 작업을 프로그래밍이라고 한다. 엔트리로 공 튀기기 게임을 만들며 프로그래밍의 순차, 반복, 선택 구조의 특성을 생각해 보자.

| 실행 화면 |



| 오브젝트 |



진행 상태 바



농구공



실내 체육관



마우스로 움직인다.

진행 상태 바



시작하기 버튼을 클릭했을 때

계속 반복하기

x: 마우스 x 좌표 위치로 이동하기



진행 상태 바에 닿으면 튕기고
바닥에 닿으면 게임이 멈춘다.

농구공



시작하기 버튼을 클릭했을 때

이동 방향을 0° (으)로 정하기

x: -200 부터 200 사이의 무작위 수 y: 115 위치로 이동하기

계속 반복하기

화면 끝에 닿으면 튕기기

이동 방향으로 5 만큼 움직이기

만일 진행 상태 바에 닿았는가? (이)라면

방향을 -60 부터 60 사이의 무작위 수 (으)로 정하기

만일 아래쪽 벽에 닿았는가? (이)라면

모든 코드 멈추기



게임에 어떤 기능을 추가할 수 있을까?

01

데이터의 저장과 사용

학습 목표

- 데이터를 순차적으로 저장하는 구조를 설명할 수 있다.
- 리스트를 활용하여 문제 해결 프로그램을 작성할 수 있다.

미리 보는 학습 흐름도

데이터의 입출력

리스트

리스트를 활용한 프로그래밍

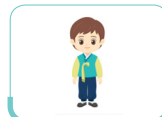
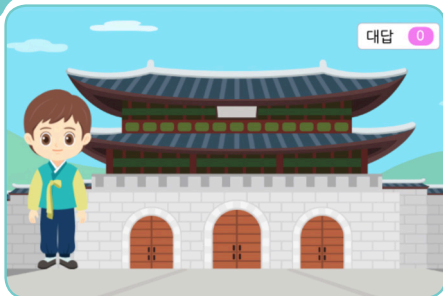


스포츠 경기를 할 때 점수판이 없으면 어떻게 점수를 알 수 있을까?

1 데이터의 입출력

프로그램을 작성할 때는 데이터의 입출력이 기본이 된다. 이름을 키보드로 입력받으면 그 내용을 출력하는 프로그램을 작성해 보자.

실행 화면



한복 입은 사람(2)



▲ 그림 III-2 입력과 출력

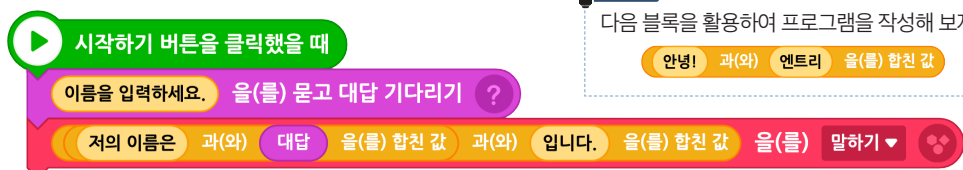
- **입력**
값을 받아들이는 것으로 프로그램 내에서 계산하거나 처리하기 위해 사용한다.
- **출력**
프로그램에서 처리한 결과를 외부로 전달하는 것으로 모니터에 표시하거나 스피커로 내보내는 것 등을 말한다.

역량 다지기 활동

이름을 입력받아 출력하기

★★★★★

- 이름을 입력하면 “저의 이름은 ○○○입니다.”를 출력하는 프로그램을 작성해 보자.



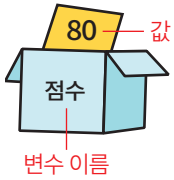
Tip! 다음 블록을 활용하여 프로그램을 작성해 보자.

안녕! 과(와) 엔트리 을(를) 합친 값

저의 이름은 과(와) 대답 을(를) 합친 값 과(와) 입니다. 을(를) 합친 값 을(를) 말하기 ▼

● 변수

데이터를 저장하는 공간으로 이름을 만들어 사용한다.



② 데이터의 순차적 저장

프로그래밍에서 숫자, 그림, 문자 등의 데이터를 저장할 때 변수, 리스트 등을 사용한다. 변수는 하나의 데이터를 저장하며, 리스트는 여러 개의 데이터를 순차적으로 저장한다.

리스트의 생성과 추가 | 명령어 탭에서 [자료]-[리스트 만들기]를 누르고 리스트의 이름을 입력한다. [리스트 추가]를 누르면 리스트가 생성된다. ‘숫자’라는 이름의 리스트를 만들고 데이터 10, 20, 30을 저장해 보자.



▲ 그림 III-3 리스트 생성

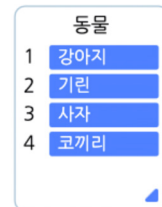


리스트를 만들고 데이터 추가하기

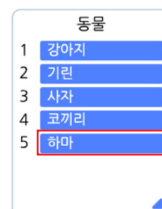
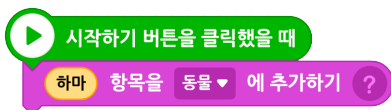
★★★★★

- 리스트를 만들고 데이터를 추가하는 프로그램을 작성해 보자.

1. ‘동물’ 리스트를 만들고 강아지, 기린, 사자, 코끼리 데이터를 기본값으로 저장해 보자.



2. **10** 항목을 동물 ▼ 에 추가하기 ? 블록을 이용해 ‘하마’ 데이터를 추가해 보자.



리스트의 데이터 수정 | 리스트의 첫 번째 항목을 5로 수정해 보자. 첫 번째 항목의 값만 5로 바뀌고 나머지 항목에는 변화가 없다.



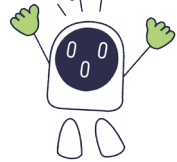
엔트리봇

숫자	
1	10
2	20
3	30



숫자	
1	5
2	20
3	30

‘자료’ 블록 꾸러미에 리스트와 관련된 블록이 모두 들어 있어.



▶ 시작하기 버튼을 클릭했을 때

숫자 ▼ 1 번째 항목을 5 (으)로 바꾸기 ?

▲ 그림 III-4 리스트의 데이터 수정

리스트의 데이터 삭제 | 리스트의 두 번째 항목을 삭제해 보자. 20을 삭제하면 두 번째 항목이 삭제되면서 세 번째 항목이 두 번째로 이동한다. 리스트에서 해당 항목을 삭제하면 이후 항목들의 위치가 한 칸씩 앞으로 이동한다.



엔트리봇

숫자	
1	10
2	20
3	30



숫자	
1	10
2	30

▶ 시작하기 버튼을 클릭했을 때

2 번째 항목을 숫자 ▼ 에서 삭제하기 ?

▲ 그림 III-5 리스트의 데이터 삭제



활동

리스트에서 무작위 수 뽑기

★★★★★

- 우주여행 지원자 100명 중 10명을 뽑아 우주여행을 보내려고 한다. 10명을 무작위로 뽑아 리스트에 저장하는 프로그램을 작성해 보자.

① ‘우주여행’ 리스트를 만든다.

② 다음 블록을 활용하여 프로그램을 작성한다.

▶ 시작하기 버튼을 클릭했을 때

0 부터 100 사이의 무작위 수

10 번 반복하기

10 항목을 우주여행 ▼ 에 추가하기 ?

③ [시작하기] 버튼을 눌러 결과를 확인한다.

우주여행	
1	18
2	14
3	3
4	9
5	92
6	22
7	72
8	66
9	91
10	65

무작위 수는 실행할 때마다 값이 변해.



온라인 실습



태블릿 활용

③ 소망 목록 만들기

방학 때 하고 싶은 일을 생각해 보고 리스트를 활용한 프로그램을 작성해 보자. ‘소망 목록’ 리스트를 만들고 초깃값으로 여행 가기, 책 읽기, 자전거 타기를 입력한다.

소망 목록 리스트 생성하기

실행 화면



오브젝트

- 자전거 타는 사람
- 추가(글상자)
- 수정(글상자)
- 삭제(글상자)
- 확인(글상자)
- 공원(3)

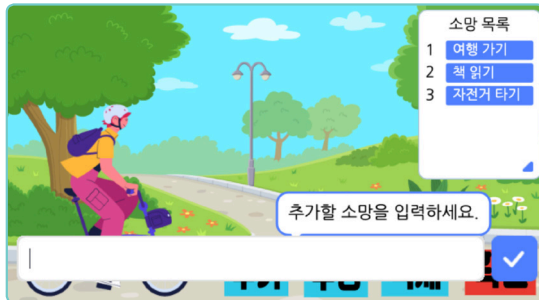
리스트

- 소망 목록

[추가] 오브젝트를 클릭했을 때

- ① “추가할 소망을 입력하세요.”라고 묻고 대답을 기다린다.
- ② 입력 창에 추가할 소망을 입력한다.
- ③ 입력한 내용이 항목에 추가되는지 확인한다.

| 실행 화면 |



추가

추가

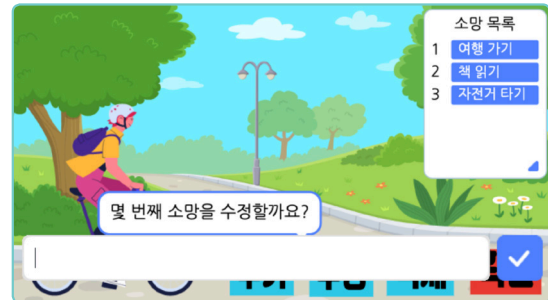
오브젝트를 클릭했을 때

- 추가할 소망을 입력하세요. 을(를) 묻고 대답 기다리기 ?
- 대답 항목을 소망 목록 ▼ 에 추가하기 ?

[수정] 오브젝트를 클릭했을 때

- ① “몇 번째 소망을 수정할까요?”라고 묻고 대답을 기다린다.
- ② 수정할 소망의 번호를 입력한다.
- ③ 대답한 번호의 항목을 입력한 소망으로 바꾼다.
- ④ 리스트의 해당 항목이 변경되는지 확인한다.

| 실행 화면 |



수정

수정

오브젝트를 클릭했을 때

- 몇 번째 소망을 수정할까요? 을(를) 묻고 대답 기다리기 ?
- 소망 목록 ▼ 대답 번째 항목을 종이접기 (으)로 바꾸기 ?

■ [삭제] 오브젝트를 클릭했을 때

- ① “몇 번째 소망을 삭제할까요?”라고 묻고 대답을 기다린다.
- ② 리스트의 해당 항목이 삭제되는지 확인한다.

| 실행 화면 |



삭제

삭제

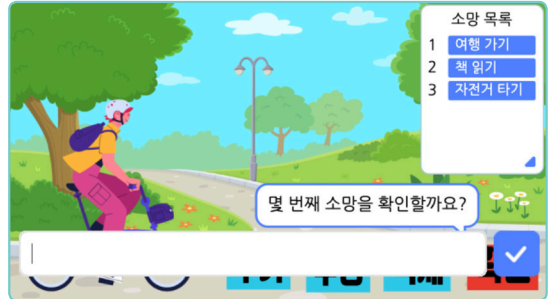
○ 오브젝트를 클릭했을 때

몇 번째 소망을 삭제할까요? 을(를) 묻고 대답 기다리기 ?
 대답 번째 항목을 소망 목록 에서 삭제하기 ?

■ [확인] 오브젝트를 클릭했을 때

- ① “몇 번째 소망을 확인할까요?”라고 묻고 대답을 기다린다.
- ② 리스트의 해당 항목이 출력되는지 확인한다.

| 실행 화면 |



확인

확인

○ 오브젝트를 클릭했을 때

몇 번째 소망을 확인할까요? 을(를) 묻고 대답 기다리기 ?
 소망 목록 의 대답 번째 항목 을(를) 말하기

역량 다지기
활동

리스트의 마지막 항목 지우기

★★★★★

- 위 프로그램에서 '삭제' 오브젝트를 클릭하면 리스트의 마지막 항목을 지우도록 프로그램을 수정해 보자.

○ 오브젝트를 클릭했을 때

마지막 소망을 삭제합니다. 을(를) 2 초 동안 말하기
 소망 목록 항목 수 번째 항목을 소망 목록 에서 삭제하기 ?

‘삭제’ 오브젝트를
클릭했을 때
리스트 전체를 지우도록
만들 수 있을까?



스스로
확인하기

- 1 변수는 한 번에 개의 값을 저장할 수 있는 저장 공간이다.
- 2 은/는 프로그래밍 언어에서 하나의 이름으로 여러 개의 값을 저장하는 데이터 구조를 말한다.

프로그래밍 2 (4월) 1

끝말잇기 게임 만들기

친구들과 끝말잇기 게임 프로그램을 작성해 보자.

1

다음 실행 화면을 참고하여 끝말잇기 게임 프로그램을 완성해 보자.

끝말잇기는 단어를 말하고 그 단어의 마지막 글자로 시작하는 새로운 단어를 이어서 말하는 게임이야.



끝말잇기
단어를 저장하는
리스트



오브젝트	변수
- 얼굴(남) [묵음]꽃과 턱 권 모습 - 끝말잇기 게임(글상자) - 단색 배경(녹색 배경.1)	단어(초깃값: 엔트리)
리스트	신호
끝말잇기	- 시작 - 엔트리 말해 - 친구 말해 - 엔트리 승 - 친구 승



꽃과 턱 권 모습

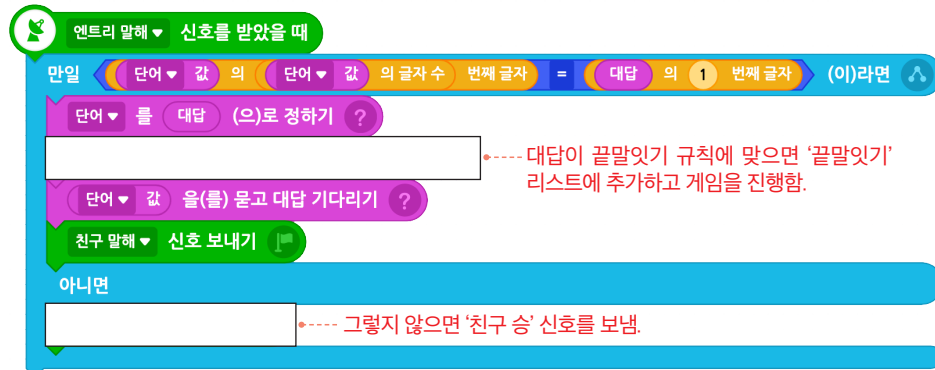


친구와 대화하면서 게임을 안내하며 첫 번째 단어를 말하고 '친구 말해' 신호를 보냄.



Tip!

신호 보내기와 신호 받기를 활용하면 프로그램의 실행 순서를 제어할 수 있다.

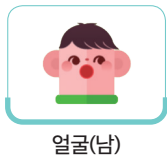


대답이 끝말잇기 규칙에 맞으면 '끝말잇기' 리스트에 추가하고 게임을 진행함.

그렇지 않으면 '친구 승' 신호를 보냄.



끝말잇기 규칙에 맞지 않음을 안내하고 승리했음을 알함.



시작하기 버튼을 클릭했을 때

- 2 초 기다리기
- 좋아요!! 을(를) 2 초 동안 말하기

----- 엔트리와 대화하며 게임을 시작함.

친구 말해 신호를 받았을 때

만일 [단어 ▼ 값 의 [단어 ▼ 값 의 글자 수 번째 글자 = [대답 의 1 번째 글자 (이)라면

[단어 ▼ 를 [대답 (으)로 정하기 ?

[단어 ▼ 값 을(를) 묻고 대답 기다리기 ?

----- 대답이 끝말잇기 규칙에 맞으면 '끝말잇기' 리스트에 추가하고 게임을 진행함.

엔트리 말해 신호 보내기

아니면

----- 그렇지 않으면 '엔트리 승' 신호를 보냄.

친구 승 신호를 받았을 때

- 끝 글자로 시작해야 해요. 을(를) 2 초 동안 말하기
- 제가 이겼어요. 을(를) 말하기

----- 끝말잇기 규칙에 맞지 않음을 안내하고 승리했음을 말함.

2 역량

위 프로그램에서 한 번 사용한 단어는 다시 사용하지 않도록 다음을 참고하여 프로그램을 수정해 보자.

- 이미 사용한 단어인지를 체크하는 단어 체크(엔트리), 단어 체크(친구) 신호로 확인한다.

예 '꽃과 턱 끈 모습' 오브젝트의 단어 체크(친구) 부분

만일 [끝말잇기 ▼ 에 [대답 이 포함되어 있는가? (이)라면

[단어 체크(친구) ▼ 신호 보내기

아니면

예 '얼굴(남)' 오브젝트의 의 단어 체크(친구) 신호 부분

단어 체크(친구) ▼ 신호를 받았을 때

- 이미 사용한 단어입니다. 을(를) 1 초 동안 말하기
- 제가 이겼어요. 을(를) 말하기

활동 평가하기

항목	평가
• 끝말잇기 게임 프로그램에서 리스트의 활용 방법을 이해할 수 있는가?	😊 😐 😞
• 끝말잇기 게임 프로그램을 완성할 수 있는가?	😊 😐 😞

02

다양한 연산

학습 목표

- 연산의 종류를 구분하고 산술, 비교, 논리 연산을 설명할 수 있다.
- 논리 연산을 활용하여 문제를 해결하는 프로그램을 작성할 수 있다.

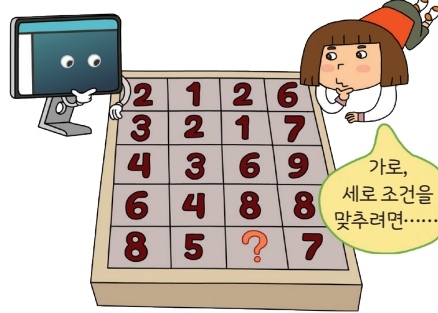
미리 보는 학습 흐름도

산술 연산

비교 연산

논리 연산

연산을 활용한 프로그래밍



컴퓨팅 시스템은 어떻게 논리적인 문제를 풀 수 있을까?

1 연산의 종류

산술 연산 | 덧셈, 뺄셈, 곱셈, 나눗셈 등 숫자를 이용하는 연산이다.

선택 활동 227쪽

산술 연산자	기능	엔트리 연산자	산술 연산자	기능	엔트리 연산자
+	더하기	10 + 10	*	곱하기	10 x 10
-	빼기	10 - 10	/	나누기	10 / 10

▲ 표 III-1 산술 연산

비교 연산 | 두 값의 크기를 비교하여 참(true) 또는 거짓(false)의 형태로 결과를 나타내는 연산이다.

연산자

연산을 하기 위한 기호를 말한다.

비교 연산자	기능	엔트리 연산자	연산 결과
=	같다	10 = 10	두 수가 같으면 참
!=	같지 않다	10 != 10	두 수가 같지 않으면 참
>	크다	10 > 10	앞의 수가 크면 참
<	작다	10 < 10	앞의 수가 작으면 참
>=	크거나 같다	10 ≥ 10	앞의 수가 크거나 같으면 참
<=	작거나 같다	10 ≤ 10	앞의 수가 작거나 같으면 참

▲ 표 III-2 비교 연산

논리 연산 | 조건을 논리적으로 판단하여 참(true)과 거짓(false)의 형태로 결과를 나타내는 연산이다.

논리 연산자	엔트리 연산자	연산 결과
그리고(AND)	참 그리고 참	두 조건을 모두 만족하면 참
또는(OR)	참 또는 거짓	두 조건을 하나라도 만족하면 참
~(이)가 아니다(NOT)	참 (이)가 아니다	조건 결과가 참이면 거짓이 되고, 거짓이면 참이 됨.

▲ 표 III-3 논리 연산



두 수의 크기 비교하기

★★★★★

- 두 개의 양수를 입력받아 두 수의 크기를 비교해 주는 프로그램을 작성해 보자.

실행 화면



오브젝트

- 어린이(1)
- 글상자
- 교실(2)

변수

- 첫 번째 수
- 두 번째 수

알고리즘

- 1 프로그램이 시작되면 “숫자 두 개를 입력하세요.”라고 말하고, 두 개의 숫자를 입력받는다.
- 2 두 개의 숫자 중 하나라도 양수가 아니면 “양수를 입력하세요.”를 말한다.
- 3 그렇지 않으면 비교 결과를 말한다.



어린이(1)



② 놀이 기구 탑승 안내 프로그램 만들기

놀이공원에서 놀이 기구를 타려고 하는데, 키에 따라 탑승에 제한이 있다고 한다. 탑승자의 키를 입력하면 놀이 기구를 탈 수 있는지 자동으로 안내하는 프로그램을 작성해 보자.

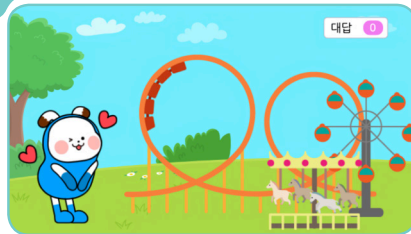
- ① 프로그램이 시작되면 “키를 입력하세요.”를 말한다.
- ② 키를 입력받는다.
- ③ 100cm 미만은 “탑승할 수 없습니다.”,
100cm 이상~110cm 미만은 “보호자와 함께 탑승해야 합니다.”,
110cm 이상은 “혼자 탑승할 수 있습니다.”를 말한다.

• 온라인 실습



태블릿 활용

실행 화면



오브젝트

- 엔트리봇 이모티콘(1)_1
- 놀이동산(1)
- 잔디 언덕(1)



엔트리봇
이모티콘(1)_1



스스로
확인하기

- 1 일정한 규칙에 따라 계산할 때 사용하는 기호를 [] (이)라고 한다.
- 2 조건을 논리적으로 판단할 때 사용하는 연산은 [] (이)라고 하며 결과는 참, 거짓의 형태로 나타난다.

자료 입력 2 자료 출력 1

별무늬 만들기

물감에 닿거나 키보드에서 'r'(빨강), 'y'(노랑), 'b'(파랑)을 누르면 지정 색의 별을 화면에 찍어 주는 별무늬 만들기 프로그램을 작성해 보자.

실행 화면



오브젝트

- 속이 빈 사각형
- [묵음] 별
- 물감(빨강/노랑/파랑)
- 별무늬 만들기(글상자)

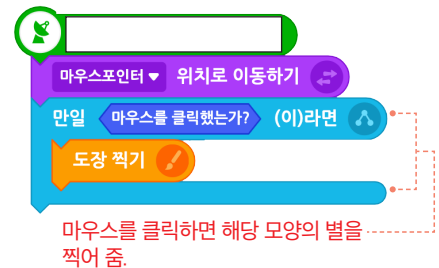
신호

- 그리기



[묵음] 별

- 물감 오브젝트에 닿거나 'r', 'y', 'b'를 누르면 해당 색의 별 모양으로 바뀐다.
- 'a'와 's' 키를 같이 누르면 '별_초록' 모양으로 바뀐다.
- 그리기 신호를 받고 마우스를 클릭하면 마우스 포인터 위치에 지정된 별 모양을 찍는다.



Tip! [묵음] 별 오브젝트의 모양 이름을 [모양] 탭에서 다음과 같이 바꾼다.

1	별_1 → 별_노랑	x
2	별_2 → 별_빨강	x
3	별_3 → 별_파랑	x
4	별_4	x
5	별_5 → 별_초록	x
6	별_6	x

활동 평가하기

항목	평가
• 연산의 결과를 예측하면서 프로그램을 작성할 수 있는가?	😊 😐 😞
• 연산을 활용하여 프로그램을 완성할 수 있는가?	😊 😐 😞

03

중첩 제어 구조

학습 목표

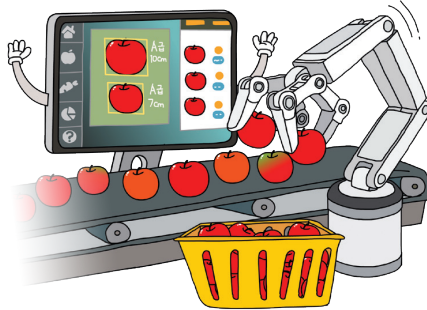
- 중첩 제어 구조를 활용하여 문제를 해결하는 프로그램을 작성할 수 있다.

미리 보는 학습 흐름도

제어 구조
(순차, 반복, 선택)

중첩 제어 구조

중첩 제어 구조를
활용한 프로그래밍



프로그램 실행 순서를 제어하는 구조에는 무엇이 있을까?

반복 구조에서
반복을 중단하기 위해서는
반복 횟수를 미리 정해
놓거나 중단 조건을
추가해야 해.



개념을 잇다 85쪽

문제 해결 과정의 표현 방법을 생각해 보자.

1 제어 구조

제어 구조란 문제 해결 과정에서 주어진 조건에 따라 프로그램의 실행 순서를 제어하는 구조로 순차, 반복, 선택이 있다. 선택 활동 228쪽

순차 구조	명령어를 순서대로 실행하는 구조
반복 구조	어떤 일을 정해진 횟수만큼 반복하거나 무한히 반복하여 실행하는 구조
선택 구조	어떤 조건이나 상황에 맞는 명령어를 선택하여 실행하는 구조

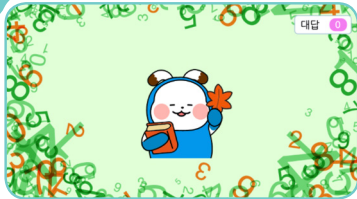
활동

홀짝 맞추기

★★★★★

- 숫자를 입력하여 홀수와 짝수를 구분하는 프로그램을 작성해 보자.

실행 화면



나뭇잎 앞모습



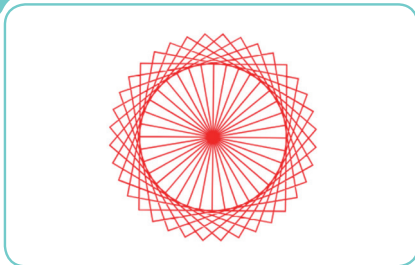
2 중첩 제어 구조

중첩 제어 구조는 제어 구조 안에 한 개 이상의 제어 구조가 포함된 것을 말한다. 같은 형태의 제어 구조가 중첩될 수도 있고 다른 형태의 제어 구조가 중첩될 수도 있다. 예를 들어 반복 구조 속의 반복 구조, 반복 구조 속의 선택 구조, 선택 구조 속의 선택 구조, 선택 구조 속의 반복 구조 등 필요에 따라 다양하게 표현할 수 있다.

반복 속의 반복 구조 | 사각형 그리기 프로그램을 활용하여 꽃 모양의 그림을 그리는 프로그램을 작성해 보자.

- 1 연필 모양이 보이지 않게 숨긴다.
- 2 3~4를 36번 반복한다.
- 3 사각형을 그린다.
- 4 방향을 10도만큼 회전한다.

실행 화면



오브젝트

• 연필(1)

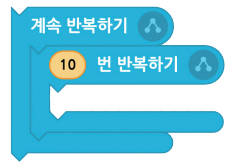


연필(1)

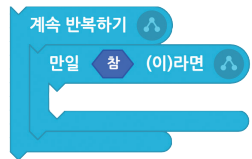


중첩 제어 구조의 예

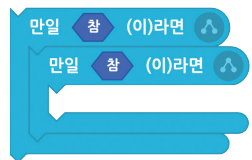
• 반복 구조 속의 반복 구조



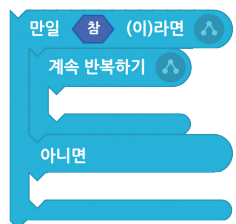
• 반복 구조 속의 선택 구조



• 선택 구조 속의 선택 구조



• 선택 구조 속의 반복 구조



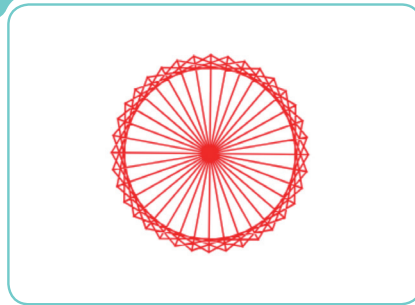
• 사각형 그리기



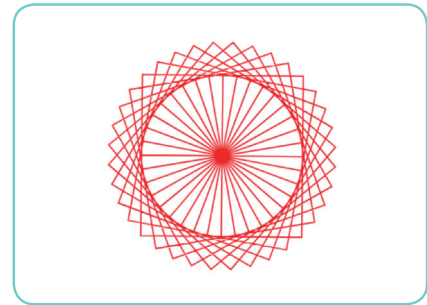
반복 속의 선택 구조 | 조건에 따라 삼각형 또는 사각형을 반복하여 무늬를 만드는 프로그램을 작성해 보자.

- ① 연필의 모양을 숨긴다.
- ② 어떤 도형을 그릴지 묻고 대답을 기다린다.
- ③ ④~⑥을 36번 반복한다.
 - ④ “삼각형”이라고 대답하면 삼각형을 그린다.
 - ⑤ “사각형”이라고 대답하면 사각형을 그린다.
 - ⑥ 방향을 10도만큼 회전한다.

실행 화면



“삼각형”을 입력했을 때

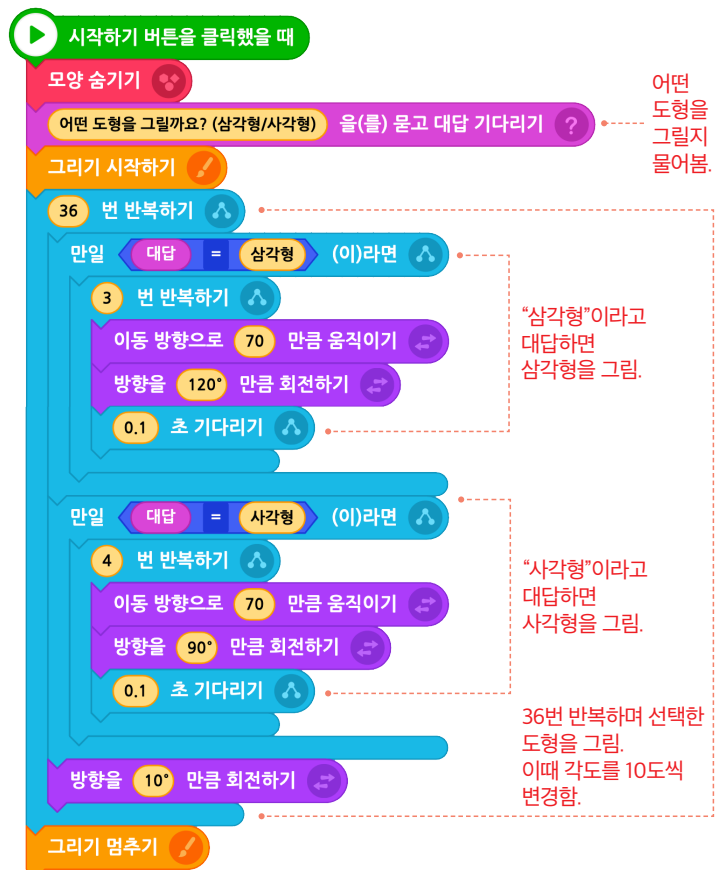


“사각형”을 입력했을 때



연필(1)

삼각형 그리기



③ 달리기 프로그램 만들기

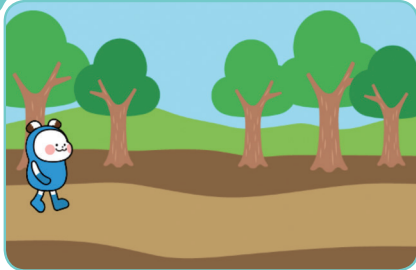
엔트리봇이 달리고 점프하는 프로그램을 작성해 보자.

- ① 왼쪽, 오른쪽 화살표 키를 누르면 화살표 키 방향으로 이동한다.
- ② ③~⑤를 반복한다.
- ③ 스페이스키와 오른쪽 화살표 키를 같이 누르면 점프하면서 오른쪽으로 이동한다.
- ④ 스페이스키와 왼쪽 화살표 키를 같이 누르면 점프하면서 왼쪽으로 이동한다.
- ⑤ 스페이스키를 누르면 제자리에서 점프한다.



블록의 포함 관계를
확인하며
프로그래밍해야 해.

실행 화면

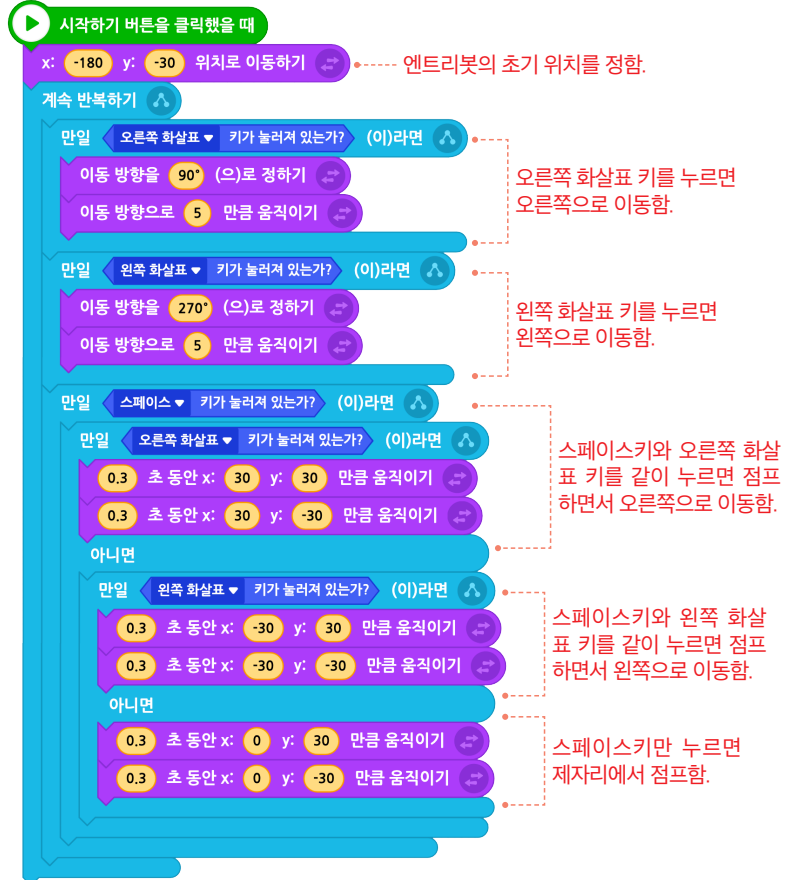


오브젝트

- 엔트리봇
- 오솔길



엔트리봇



 스스로
확인하기

- 1 프로그램의 실행 순서를 제어하는 구조에는 구조, 구조, 구조가 있다.
- 2 제어 구조 안에 또 다른 제어 구조가 한 개 이상 포함된 것을 구조라고 한다.

1. 순차, 반복, 선택 2. 조건 제어

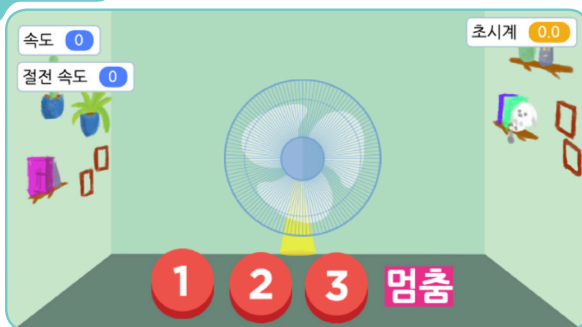
절전 선풍기 프로그램 만들기

버튼으로 속도를 조절하고 시간이 지나면 자동으로 멈추는 절전 선풍기 프로그램을 작성해 보자.

알고리즘

- 1 숫자 버튼 1, 2, 3을 누르면 '속도'가 10, 20, 30이 되도록 하고 초시계를 시작한다.
- 2 선풍기는 1, 2, 3 버튼을 누르면 설정된 속도로 회전한다.
- 3 멈춤을 누르면 선풍기가 멈추고 초시계가 정지한다.
- 4 선풍기가 켜진 후 10초가 지나면 '절전 속도'($\text{속도}/2$)로 바꾸고, 20초가 지나면 멈춘다.

실행 화면



오브젝트

- 초록 방
- 선풍기(날개, 뚜껑)
- 숫자 버튼(1, 2, 3)
- 빛
- 멈춤(글상자)

변수

- 속도
- 절전 속도

1

숫자 버튼 1

- 오브젝트를 클릭하면 '속도'를 10으로 정하고 '절전 속도'에 '속도' 값을 저장한다.
- 시간을 재기 위해 초시계를 시작한다.



Tip!

왼쪽의 블록을 엔트리에서는 다음 블록으로 표현할 수 있다.



2

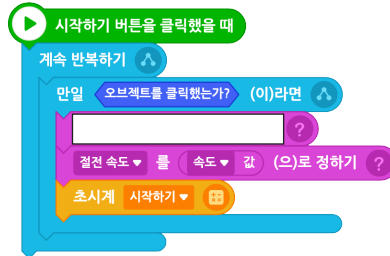
숫자 버튼 2

- 오브젝트를 클릭하면 '속도'를 20으로 정하고 '절전 속도'에 '속도' 값을 저장한다.
- 시간을 재기 위해 초시계를 시작한다.



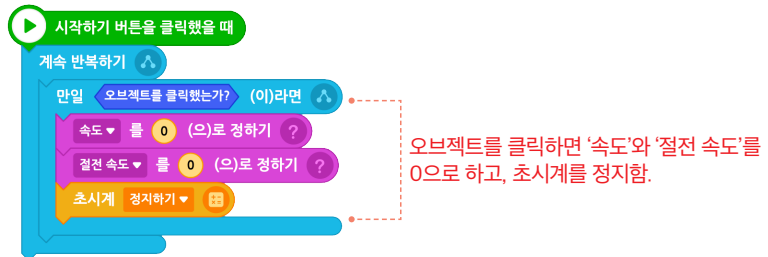
3 숫자 버튼 3

- 오브젝트를 클릭하면 '속도'를 30으로 정하고 '절전 속도'에 '속도' 값을 저장한다.
- 시간을 재기 위해 초시계를 시작한다.



멈춤

- 오브젝트를 클릭하면 '속도'와 '절전 속도'를 0으로 정한다.
- 초시계를 정지한다.



선풍기 날개

- 프로그램을 실행한 후 1, 2, 3 버튼을 누르면 지정된 '속도' 값으로 회전한다.
- 10초가 지나면 '절전 속도'(속도/2)로 바꾸고, 20초가 지나면 선풍기가 멈춘다.



활동 평가하기

항목	평가
• 중첩 제어 구조를 활용하여 프로그램을 완성할 수 있는가?	😊 😐 😞
• 중첩 제어 구조의 결과를 예측하면서 프로그램을 작성할 수 있는가?	😊 😐 😞

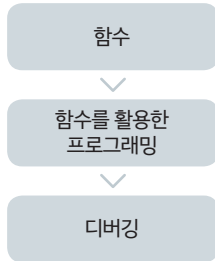
04

함수와 디버깅

학습 목표

- 프로그램 작성에 함수를 활용할 수 있다.
- 프로그램 수행 결과를 디버거로 분석하여 오류를 수정할 수 있다.

미리 보는 학습 흐름도



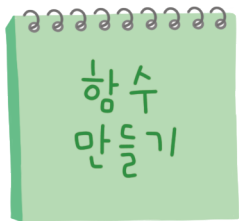
🔍 프로그래밍에서 여러 번 반복하는 명령어를 간단하게 표현하는 방법은 없을까?



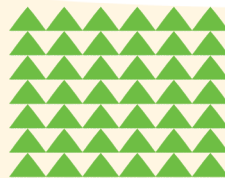
- 함수의 장점
 - 프로그램에서 중복되는 코드를 제거할 수 있다.
 - 복잡한 프로그램을 간단한 작업으로 분해해서 처리할 수 있다.
 - 한 번 만들어진 함수를 다른 프로그램에서도 재사용할 수 있다.
 - 프로그램을 수정하기 쉬워진다.

① 함수

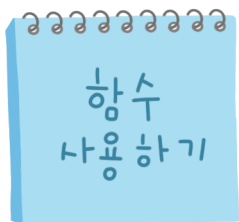
함수는 어떤 특별한 작업을 수행하도록 여러 개의 명령어를 하나로 묶어서 이름을 부여한 것이다. 반복적으로 사용되는 일을 함수로 만들어 놓으면 필요할 때마다 함수를 호출하여 편리하게 사용할 수 있다.



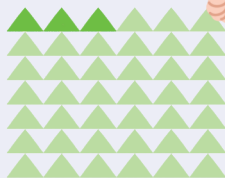
이렇게 생긴 그림을 그리려면 어떻게 해야 할까?



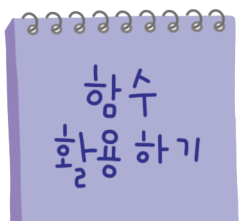
먼저 삼각형들을 만들면 어때?



만든 삼각형을 필요한 만큼 가져다 쓰면 되겠다.



그래! 도장을 찍듯이!



크기와 색도 바꾸면서 활용할 수 있겠다.



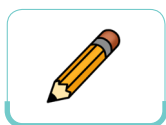
함수 만들기 | 편지봉투 모양이 여러 개 반복되는 프로그램을 만들고자 할 때, 반복되는 부분을 함수로 만들어 사용하면 편리하다. 편지봉투 모양을 함수로 만들어 여러 번 사용해 보자.

실행 화면



오브젝트

• 연필(1)



연필(1)



Tip!

함수는 프로그램을 작성한 후 반복되는 패턴을 찾아 만들거나, 미리 반복되는 부분을 알 수 있으면 처음부터 함수로 만들어 활용할 수 있다.

연필 오브젝트는 중심점을 연필 끝으로 이동해야 연필 끝에서 그림이 그려져.



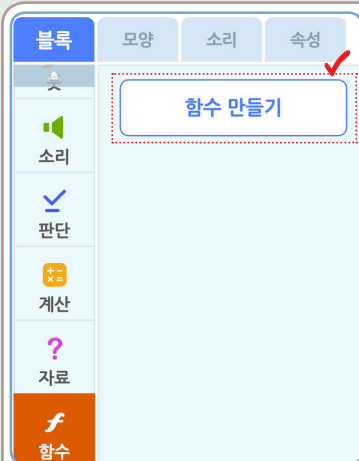
오브젝트 조절

잠금 해제 상태에서 조절 가능



- (선택 박스): 크기 조절
 - (중심점): 중심의 위치 조절
 - (핸들): 오브젝트가 바라보는 방향 조절
 - (이동 방향 표시): 오브젝트가 움직이는 이동 방향 조절
- ※ 방향과 이동 방향은 다름.

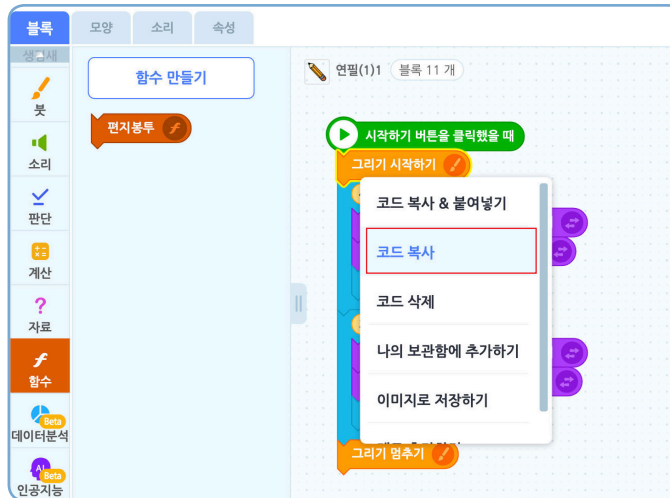
'편지봉투' 함수 만드는 과정



① [함수]-[함수 만들기]를 클릭한다.



② 함수 정의하기 대화 상자에 함수 이름을 '편지봉투'로 입력하고 저장한다.



③ 앞에서 작성한 프로그램에서 함수로 사용할 범위를 복사한다.



④ '편지봉투' 함수를 더블 클릭하여 함수 정의기에 복사한 코드를 붙여 넣고 저장한다.

- 함수 블록
함수를 만들면 생성된다.
- 함수 이름 블록
이름
- 함수값 블록
문자/숫자값
- 함수 판단 블록
판단값

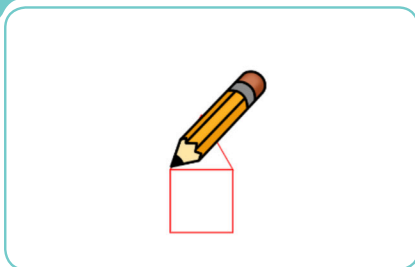


집 모양 함수 만들기

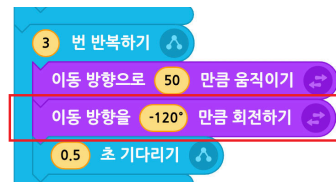
★★★★★

- '편지봉투' 함수를 수정하여 집 모양을 그리는 함수로 만들어 보자.

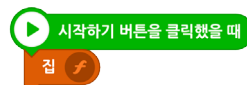
실행 화면



- 1 함수 이름을 '집'으로 수정한다.
- 2 함수 안에 삼각형 그리는 부분의 각도를 수정한다.

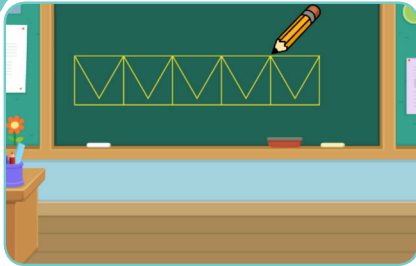


- 3 실행하여 결과를 확인한다.



함수 사용하기 | 이미 만들어 놓은 함수는 재사용이 가능하다. 만들어 놓은 함수를 사용하여 화면에 노란색 편지봉투를 5개 그리는 프로그램을 작성해 보자.

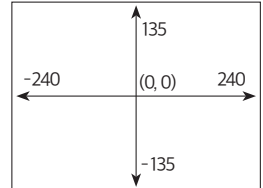
실행 화면



오브젝트
• 연필(1)
• 칠판(2)
함수
• 편지봉투

※ 처음 연필 위치는 x: -150, y: 80으로 정한다.

- 실행 화면의 좌표
- 실행 화면 중심의 좌표는 (0, 0)이다.
- x 좌표 범위는 -240~240이다. 오른쪽 방향은 양수(+)이고, 왼쪽 방향은 음수(-)이다.
- y 좌표 범위는 -135~135이다. 위쪽 방향은 양수(+)이고, 아래 방향은 음수(-)이다.



연필(1)

시작하기 버튼을 클릭했을 때

- 그리기 색을 (으)로 정하기 •----- 붓의 색을 노란색으로 지정함.
- x: -150 y: 80 위치로 이동하기 •----- 연필의 처음 위치는 x: -150, y: 80로 함.
- 편지봉투 •-----
- x 좌표를 50 만큼 바꾸기 •-----
- 편지봉투 •-----
- x 좌표를 50 만큼 바꾸기 •-----
- 편지봉투 •-----
- x 좌표를 50 만큼 바꾸기 •-----
- 편지봉투 •-----
- x 좌표를 50 만큼 바꾸기 •-----
- 편지봉투 •-----

'편지봉투' 함수를 사용함.



역량 다지기 활동

반복문으로 프로그램 수정하기

★★★★★

- 위 함수 사용하기에서 순차 구조로 작성된 프로그램을 반복문으로 수정해 보자.

- 1 노란색을 지정하고 연필(1)의 처음 그릴 위치 (-150, 80)를 지정한다.
- 2 연필(1) 위치로 이동해서 편지봉투를 그리고 연필(1)의 'x 좌표를 50만큼 바꾸기'를 5번 반복한다.

x: -150 y: 80 위치로 이동하기

5 번 반복하기

편지 봉투

x 좌표를 50 만큼 바꾸기



Tip!

x 좌표의 이동은 변수를 만들어 활용할 수도 있다.

- 3 실행하여 확인한다.

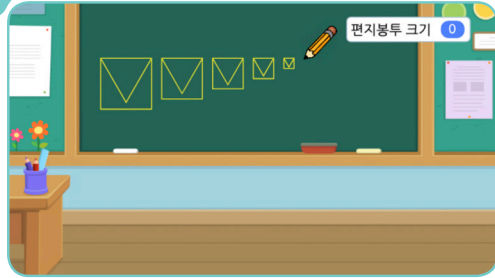
● 함수의 매개변수 사용

함수를 사용할 때 특정 값을 받아서 전달하는 경우가 있다. 이때 전달하는 값을 받는 변수를 '매개변수'라고 한다. 엔트리에서는 함수 탭의 **문자/숫자값** 을 사용한다.

함수 활용하기 | 함수는 여러 가지 형태로 다양하게 응용할 수 있다.

‘편지봉투’ 함수를 활용해 크기가 10씩 작아지는 편지봉투 프로그램을 작성해 보자.

실행 화면



오브젝트	
• 연필(1)	
• 칠판(2)	
변수	
• 편지봉투 크기	
함수	
• 편지봉투	

함수: 편지봉투(문자/숫자값 1)

함수 속성 보기

이름: 편지봉투

문자/숫자값: 10

판단값: 1

함수 정의하기

그리기 시작하기

4 번 반복하기

이동 방향으로 문자/숫자값 1 만큼 움직이기

이동 방향을 90° 만큼 회전하기

0.5 초 기다리기

3 번 반복하기

이동 방향으로 문자/숫자값 1 만큼 움직이기

이동 방향을 120° 만큼 회전하기

0.5 초 기다리기

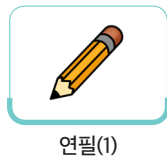
그리기 멈추기

취소 저장

① '편지봉투' 함수에 문자/숫자값 1을 추가함.

② '편지봉투' 함수의 달라진 부분 확인함.

③ 움직이는 크기를 전달 받을 블록을 드래그하여 넣음.



시작하기 버튼을 클릭했을 때

그리기 색을 (으)로 정하기

x: -150 y: 80 위치로 이동하기

편지봉투 크기 를 50 (으)로 정하기

5 번 반복하기

편지봉투 편지봉투 크기 값

x 좌표를 편지봉투 크기 + 10 만큼 바꾸기

편지봉투 크기 에 -10 만큼 더하기

'편지봉투 크기'를 50으로 초기화함.

'편지봉투' 함수에 '편지봉투 크기' 값을 전달하여 함수를 사용함.

편지봉투의 간격을 10씩 이동함.

'편지봉투 크기' 값을 10만큼 작게 함.

② 와이파이 신호 세기 프로그램 만들기

와이파이 신호 세기에 따라 결과가 달라지는 프로그램을 작성하기 위해서는 반환값을 사용해야 한다. 함수의 사용이 완료되었을 때의 결과값을 반환값이라고 한다. 반환값은 함수에 따라 있을 수도 있고 없을 수도 있다.

• 온라인 실습



태블릿 활용

실행 화면



반환값이 '좋음'일 때



반환값이 '나쁨'일 때

오브젝트

- 와이파이
- 종이 접기 프레임
- 신호 크기 안내(글상자)

함수 변수

- 신호 크기
- 신호 크기



신호 크기 함수



와이파이



③ 디버깅

작성한 프로그램이 설계한 대로 잘 작동하는지 확인하는 것도 프로그램을 작성하는 것만큼 중요해.



디버깅은 프로그램 개발 단계 중에 발생하는 논리적인 오류나 비정상적인 연산 등을 찾아내어 그 원인을 밝히고 수정하는 과정을 뜻한다. 다음 프로그램에서 오류를 찾아 수정해 보자.

• 온라인 실습



태블릿 활용

- 1 프로그램이 시작되면 화면 중앙에 우유 오브젝트가 보인다.
- 2 우유 오브젝트를 클릭하면 가로 10칸, 세로 3칸에 우유 오브젝트가 채워진다.
※ 우유와 우유 사이의 가로 간격은 45, 세로 간격은 60이다. 우유 오브젝트의 크기는 20으로 정한다.

실행 화면



▲ 초기 화면



디버깅

▲ 실행 완료 화면



▲ 목표 화면



우유 200ml

오브젝트

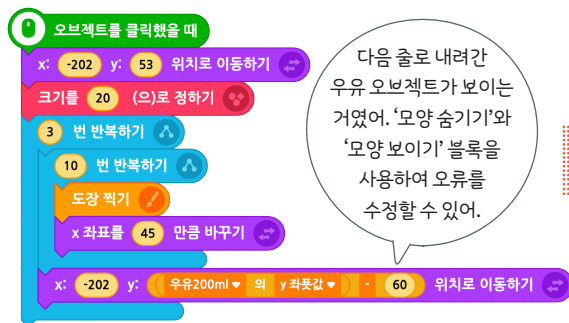
- 우유 급식 확인판
- 우유 200ml

*디버거

- 디버깅을 도와주고 프로그램을 테스트해 준다.
- 코드에 멈추는 지점을 지정하여 프로그램 실행을 정지하기도 하고, 메모리에 저장된 값을 살펴보기도 하고, 다시 실행시키거나 코드를 단계적으로 실행하는 등의 동작을 하면서 오류를 발견하여 수정할 수 있게 해 준다.



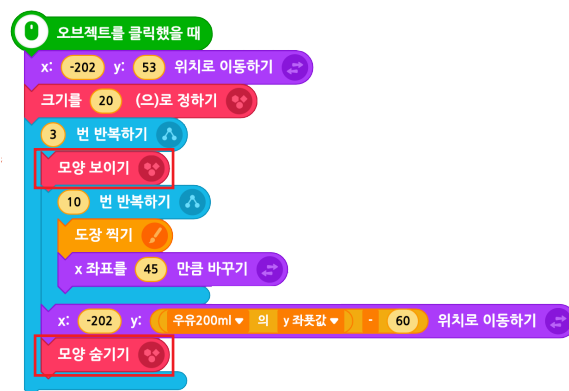
| 기존 프로그램 |



다음 줄로 내려간 우유 오브젝트가 보이는 거였어. '모양 숨기기'와 '모양 보이기' 블록을 사용하여 오류를 수정할 수 있어.



| 수정한 프로그램 |



스스로 확인하기

- 1 프로그래밍에서 어떤 특별한 작업을 수행하는 명령을 하나로 묶어서 이름을 붙여 사용하는 것을 (이)라고 한다.
- 2 프로그램의 오류를 수정하는 작업을 (이)라고 한다.

김민기 2 수록

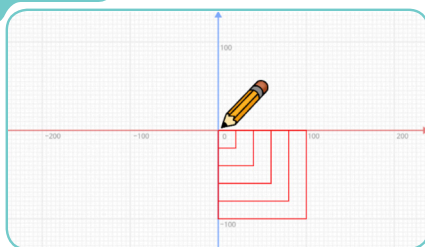
사각형을 이용한 패턴 그리기

‘정사각형 그리기’를 함수로 만들고 100부터 시작하여 20만큼씩 작아지는 정사각형 5개를 그리는 프로그램을 작성해 보자.

1

실행 화면을 참고하여 프로그램을 작성해 보자.

실행 화면



오브젝트

- 연필(1)
- 모눈종이

함수

- 정사각형 그리기

함수에서 다시 함수 자신을 호출하여 사용하는 것을 ‘재귀 함수’라고 해.



① ‘정사각형 그리기’ 함수를 만든다.

정사각형 그리기 5

정사각형 그리기 함수

함수 정의하기 정사각형 그리기 문자/숫자값 1

그리기 시작하기

4 번 반복하기

이동 방향을 90° 만큼 회전하기

0.5 초 기다리기

그리기 멈추기

정사각형 그리기 문자/숫자값 1

정사각형 개수로 지정한 값에 20을 곱한 값을 한 번의 길이로 함.

‘정사각형 그리기’ 함수에서 다시 ‘정사각형 그리기’ 함수를 불러와 사용함.

② 함수를 불러와 사용한다.

연필(1)

시작하기 버튼을 클릭했을 때

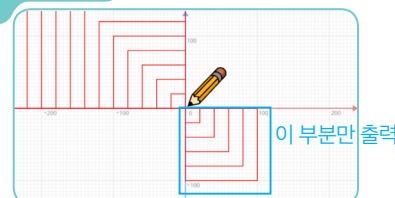
‘정사각형 그리기’ 함수를 불러와 사용함.(매개변수 5)

개수가 1씩 감소하면서 이 값을 ‘정사각형 그리기’ 함수에 전달하여 한 번의 길이를 20씩 감소시킴.

2

1을 실행하면 왼쪽 윗부분으로 사각형이 계속 그려진다. 오른쪽 아래 부분에 사각형 5개를 그린 후 실행이 종료되도록 프로그램을 수정해 보자.

실행 화면



활동 평가하기

항목	평가
함수를 활용하여 프로그램을 완성할 수 있는가?	😊 😐 😞
오류를 확인하고 디버깅할 수 있는가?	😊 😐 😞

미리 보는
학습 흐름도

문제 이해

추상화

알고리즘 설계

프로그래밍

투표 프로그램 만들기

학습 목표

- 실생활 문제에서 컴퓨팅 시스템으로 해결할 수 있는 문제를 발견할 수 있다.
- 다양한 학문 분야의 문제를 프로그래밍으로 협력하여 해결할 수 있다.



🔍 프로그래밍으로 좀 더 쉽게 해결할 수 있는
과제에는 어떤 것이 있을까?



① 문제 이해

협력 프로젝트

학생들이 스스로 계획하고 구상하며 문제를 해결하는 방식으로, 모둠원은 각자 역할을 정하고 프로젝트 수행을 위해 책임감 있는 자세로 자신이 맡은 역할을 수행해야 한다.

석준이네 반은 학급 회장 선거를 하려고 한다. 투표 용지에 직접 손으로 투표를 진행하면 준비 과정이 복잡하고 개표에도 시간이 오래 걸린다. 전자 투표 프로그램을 만들어 효율적인 선거를 진행해 보자.



문제 정의

후보자 중에 한 명을 선택하여 투표하는 프로그램 만들기

- 현재(초기) 상태: 직접 손으로 투표하고 개표를 해야 하는 상태
- 목표 상태: 전자 투표 프로그램으로 투표하고 개표가 완료된 상태

② 추상화

문제의 핵심 요소를 찾아 정리해 보자.

과정	핵심 요소	핵심 요소 설명
투표 준비	후보자	• 후보자 이름을 '후보' 리스트에 저장한다.
	투표자	• 투표 인원을 '학급 인원' 변수에 저장한다.
투표 과정	원하는 후보에 투표	• 투표자들은 원하는 후보의 번호를 누른다. • 후보들의 득표수를 각각 '1번 후보', '2번 후보', '3번 후보' 변수에 저장한다. • 무효표 수를 '무효표' 변수에 저장한다.
	투표수 안내	• 현재까지 투표수를 안내한다. • 학급 인원수만큼 투표하면 투표의 종료를 안내한다.
결과 안내	당선자 안내	• 최고 득표자를 당선자로 안내한다.
	재투표 안내	• 최고 득표자가 동점일 때 재투표를 안내한다.
	다시하기	• 다시하기 버튼을 누르면 투표를 처음부터 다시 시작한다.



③ 알고리즘 설계

투표 프로그램의 작동 원리를 이해하고 알고리즘을 설계해 보자.

투표 준비

- ① 후보를 '후보' 리스트에 입력
- ② '학급 인원'을 입력

투표 과정

- ① '학급 인원'만큼 ②~⑤ 반복
- ② 원하는 후보의 번호를 누름
- ③ 후보들의 득표수를 각 변수에 저장
- ④ 무효표는 '무효표' 변수에 저장
- ⑤ 현재까지 투표한 인원 안내

결과 안내

- ① '투표 종료' 신호를 받으면
투표 종료 안내
- ② 최고 득표수를 받은 후보의 수 ≥ 20 이면
재투표 안내
- ③ 최고 득표수를 받은 후보의 수 = 10이면
그 후보를 당선자로 안내
- ④ 다시하기 버튼을 누르면
처음부터 다시 시작

개념을 잇다

92쪽

알고리즘 설계 방법을 생각해 보자.



태블릿 활용

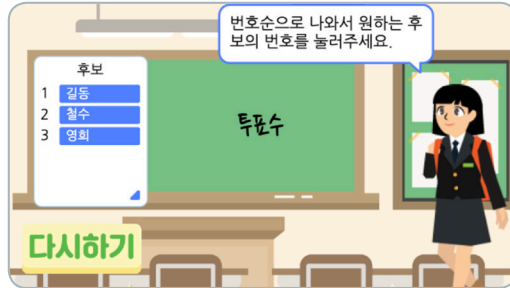
4 프로그래밍

설계한 알고리즘을 바탕으로 화면을 구성하고 프로그램을 작성한다.

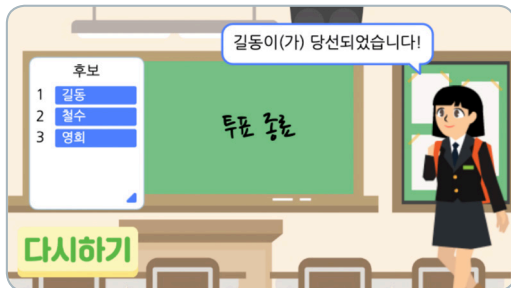
화면 구성하기



투표 준비



투표 과정



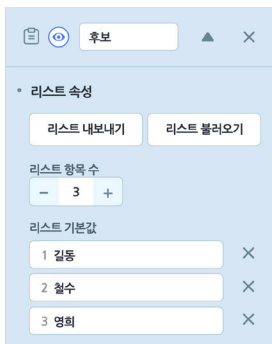
결과 안내

오브젝트	신호
• 학생(2) • 교실(배경) • 투표수(글상자) • 다시하기 버튼	• 투표 시작 • 투표 종료
변수	
• 학급 인원 • 투표 인원 • 무효표	• 1번 후보 • 2번 후보 • 3번 후보
함수	리스트
• 최다 득표자 동점	• 후보

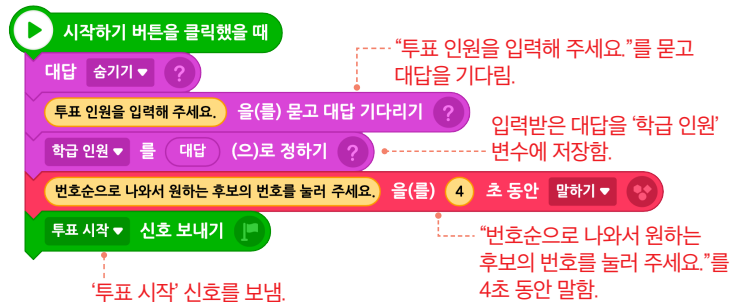
프로그래밍하기

투표 준비

- 선거 후보를 '후보' 리스트에 입력하고, 학급 인원을 입력한다.



학생(2)



투표 과정

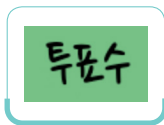
- 원하는 후보에 투표하는 과정을 '학급 인원'만큼 반복한다.



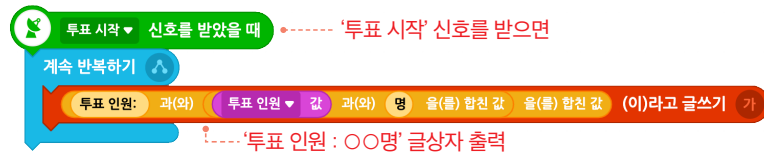
학생(2)



- 투표수 글상자는 현재까지의 투표 인원을 출력한다.



투표수



글상자 블록 꾸러미

글상자 오브젝트를 만들면 글상자 블록 꾸러미가 추가된다. 글상자 명령어 블록에는 글상자 오브젝트에 적합한 텍스트 내용, 글씨체, 글씨색, 배경색 등을 변경할 수 있는 블록이 있다.



결과 안내

- 득표수에 따라 나올 수 있는 상황과 상황에 따른 처리 방안을 생각해 본다.

득표수에 따라 나올 수 있는 상황	처리 방안
예) 한 후보에게 제일 많은 득표수가 나온다.	제일 많은 득표수를 받은 후보가 당선된다.
예) 제일 많은 득표를 받은 후보가 두 명 이상이다.	제일 많은 득표를 받은 후보들을 대상으로 다시 투표를 진행한다.

- 최다 득표를 받은 후보가 둘 이상이면 '최다 득표자 동점' 함수를 호출한다.

최다 득표자 동점

함수 정의하기 최다 득표자 동점

만일 1번 후보 > 값 = 2번 후보 > 값 (이)라면

1, 2번 후보의 득표수가 같습니다. 을(를) 4 초 동안 말하기

최소 득표자를 후보에서 삭제하고 을(를) 4 초 동안 말하기

프로그램 코드 수정 후 다시 투표해 주세요! 을(를) 4 초 동안 말하기

아니면

만일 1번 후보 > 값 = 3번 후보 > 값 (이)라면

1, 3번 후보의 득표수가 같습니다. 을(를) 4 초 동안 말하기

최소 득표자를 후보에서 삭제하고 을(를) 4 초 동안 말하기

프로그램 코드 수정 후 다시 투표해 주세요! 을(를) 4 초 동안 말하기

아니면

2, 3번 후보의 득표수가 같습니다. 을(를) 4 초 동안 말하기

최소 득표자를 후보에서 삭제하고 을(를) 4 초 동안 말하기

프로그램 코드 수정 후 다시 투표해 주세요! 을(를) 4 초 동안 말하기

모든 코드 멈추기

1번 후보' 값과 '2번 후보' 값이 같으면 최소 득표한 후보를 삭제 후 재투표하도록 안내함.

1번 후보' 값과 '3번 후보' 값이 같으면 최소 득표한 후보를 삭제 후 재투표하도록 안내함.

2번 후보' 값과 '3번 후보' 값이 같으면 최소 득표한 후보를 삭제 후 재투표하도록 안내함.

- '투표 종료' 신호를 받으면 득표수에 따라 결과를 안내하는 코드를 작성한다.

학생(2)

투표 종료 신호를 받았을 때

투표가 종료되었습니다. 을(를) 2 초 동안 말하기

투표 종료 신호를 받으면

만일 1번 후보 > 값 = 2번 후보 > 값 그리고 1번 후보 > 값 > 3번 후보 > 값 (이)라면

최다 득표자 동점

만일 1번 후보 > 값 = 3번 후보 > 값 그리고 1번 후보 > 값 > 2번 후보 > 값 (이)라면

최다 득표자 동점

만일 2번 후보 > 값 = 3번 후보 > 값 그리고 2번 후보 > 값 > 1번 후보 > 값 (이)라면

최다 득표자 동점

만일 1번 후보 > 값 > 2번 후보 > 값 그리고 1번 후보 > 값 > 3번 후보 > 값 (이)라면

후보 의 1 번째 항목 과(와) 이(가) 당선되었습니다! 을(를) 합친 값 을(를) 말하기

아니면

만일 2번 후보 > 값 > 1번 후보 > 값 그리고 2번 후보 > 값 > 3번 후보 > 값 (이)라면

후보 의 2 번째 항목 과(와) 이(가) 당선되었습니다! 을(를) 합친 값 을(를) 말하기

아니면

후보 의 3 번째 항목 과(와) 이(가) 당선되었습니다! 을(를) 합친 값 을(를) 말하기

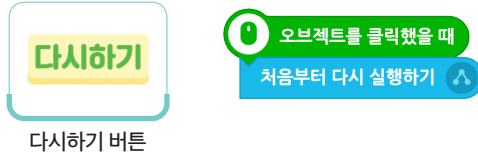
두 개의 번호가 동일한 득표수로 최다 득표일 때 '최다 득표자 동점' 함수를 호출함.

후보 중 최다 득표인 후보를 당선자로 안내함.

- '투표 종료' 신호를 받으면 글상자 내용을 '투표 종료'로 수정하여 투표가 끝났음을 안내한다.



- 다시하기 버튼을 클릭하면 투표를 처음부터 다시 시작한다.



5 프로그램 실행 확인 및 평가하기

완성된 프로그램을 실행시켜 보고 오류가 있는지 확인한다. 오류가 있다면 그 원인과 해결 방법을 찾아 오류를 해결하는 디버깅 과정을 거친다. 그리고 다른 모둠과 완성된 프로그램을 공유한다.

항목	확인
• 투표 인원수만큼 투표하는가?	☐
• 최다 득표자가 바르게 출력되는가?	☐
• 득표수가 동점일 때 바르게 해결되는가?	☐

프로그램 공유 과정에서 아이디어가 떠오르거나 친구에게 피드백 받은 것을 토대로 프로그램을 수정·보완할 수 있어.



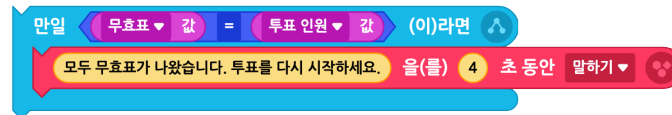
프로그램 수정 · 보완하기

★★★★★

- 투표할 때 일어날 수 있는 다양한 상황에 대처하도록 프로그램을 수정·보완해 보자.

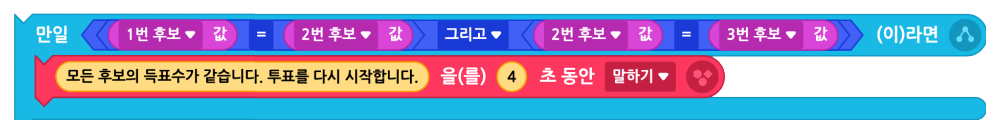
예 모두 무효표가 나오면 재투표를 실시한다.

[학생 (2)] 오브젝트의  블록 맨 밑에 다음 블록 추가하기



예 모든 후보의 득표수가 동일할 때 재투표를 실시한다.

[학생 (2)] 오브젝트의  블록 맨 밑에 다음 블록 추가하기



미리 보는
학습 흐름도

문제 이해

추상화

알고리즘 설계

프로그래밍

우주 청소 로켓 만들기

학습 목표

- 피지컬 컴퓨팅 시스템으로 동작하는 소프트웨어를 활용해 문제를 해결할 수 있다.
- 협력 프로젝트를 통해 다양한 학문 분야의 문제를 프로그래밍으로 해결할 수 있다.



🔍 프로그래밍으로 해결할 수 있는 사회 문제에는 무엇이 있을까?



① 문제 이해

소연이는 과학 과제 자료를 검색하던 중 우주에 인간이 만든 수많은 쓰레기가 떠다니고 있다는 뉴스 기사와 영상을 보고 우주 쓰레기에 대한 심각성을 느꼈다.

NEWS ○○ 일보

202X년 ○월 ○○일

● 시뮬레이션(simulation)

실제로 실행하기 어려운 과정을 실제와 비슷한 모형으로 만들어 모의실험하여 그 특성을 파악하는 것을 말한다.

지금 이 시각에도 하늘에는 연료가 바닥나거나 고장 나서 정지된 인공위성, 위성을 쏘아 올려 준 뒤 지구로 내려가지 못하고 궤도를 떠도는 로켓, 위성끼리 충돌이나 자체 폭발, 미사일 요격 실험으로 생긴 위성 부품과 파편들이 지구 주위를 떠돌고 있다. 일부 전문가들은 이미 쓰레기양이 너무 많아서 서로 충돌하면서 더욱 늘어갈 것이라고 경고하였다.

(출처: 테크월드뉴스, 2021. 5. 12.)



우주 쓰레기를 청소하는 방법 중 하나는 로켓에 레이저 장치를 달아 우주 쓰레기를 요격하는 것이라고 한다. 소연이는 우주 쓰레기 문제의 심각성을 알리는 프로그램을 만들기로 했다. 버튼과 소리 센서를 이용해 우주 쓰레기를 청소하는 시뮬레이션 로켓을 만들어 보자.

문제 정의

우주 쓰레기를 청소하는 로켓 만들기

- 현재(초기) 상태: 우주 쓰레기가 많은 상태
- 목표 상태: 우주 청소 로켓을 이용하여 우주 쓰레기를 제거한 상태

② 추상화

문제의 핵심 요소를 찾아 정리해 보자.

과정	핵심 요소	핵심 요소 설명
우주 로켓 제어	로켓	• 화면 하단에서 좌우로 움직인다. (버튼 이용) • 쓰레기 청소를 위한 레이저를 발사한다. (소리 센서 이용)
	레이저	• 위쪽 벽이나 쓰레기에 닿을 때까지 위쪽으로 이동한다.
우주 쓰레기 청소	쓰레기	• 쓰레기는 0.5초마다 새로운 복제본을 무작위 위치에 생성한다. • 쓰레기의 복제본이 레이저에 닿으면 삭제되며 점수가 10점씩 증가한다.

센서 보드의 버튼을 이용해 좌우로 움직이고, 소리 센서에 바람을 불면 레이저가 발사되는 로켓을 만들어 보자!



③ 알고리즘 설계

우주 로켓의 작동 방법을 이해하고 알고리즘을 설계해 보자.

우주 로켓 제어

- 1 초시계가 30초가 될 때까지 반복
- 2 로켓은 화면 하단에서 버튼으로 움직임
- 3 소리 센서값 ≥ 500 이면 레이저 발사
- 4 레이저가 위쪽 벽이나 쓰레기에 닿을 때까지 5 반복
- 5 레이저 화면 위쪽으로 이동
- 6 '종료' 신호를 받으면 로켓과 레이저 모양 숨기기

우주 쓰레기 청소

- 1 계속 반복
- 2 0.5초마다 쓰레기 복제본 무작위 위치에 생성
- 3 쓰레기 복제본이 레이저에 닿으면 점수 10점 증가, 복제본 삭제
- 4 획득한 점수 표시
- 5 '종료' 신호를 받으면 모든 쓰레기 복제본 삭제
- 6 점수 ≥ 300 이면 "우주 쓰레기 청소 성공!" 말하기
- 7 그렇지 않으면 "조금 더 우주 쓰레기를 청소해 주세요!" 말하기

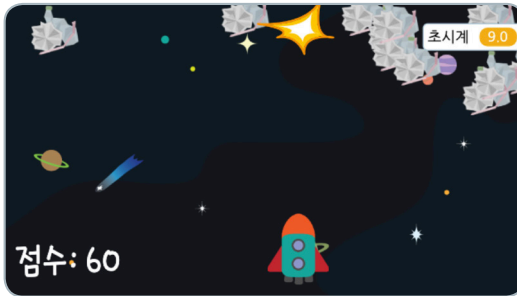
우주 청소 로켓을 만들기 위해서는 우주 로켓을 움직이는 부분과 쓰레기를 청소하는 부분으로 나누어 알고리즘을 설계해야 해.



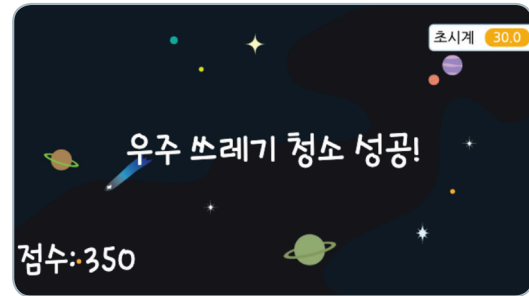
4 프로그래밍

설계한 알고리즘을 바탕으로 화면을 구성하고 프로그램을 작성한다.

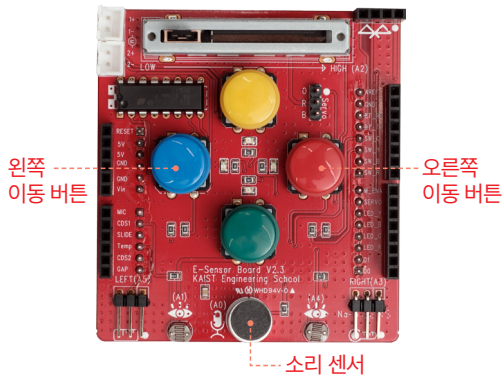
화면 구성하기



우주 쓰레기 청소

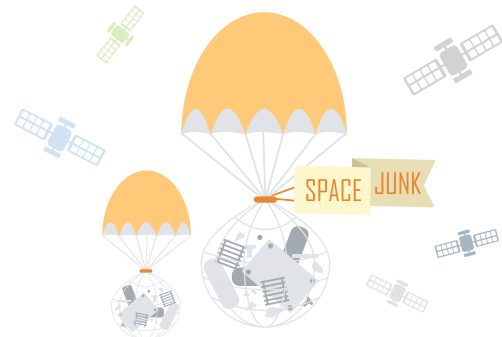


결과 화면



센서 보드

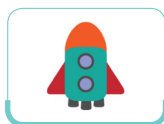
오브젝트	신호
- 로켓(3) [묵음] 폭발 효과 - 쓰레기 - 점수(글상자) - 결과 발표(글상자) - 우주(2)(배경)	- 발사 - 종료
	변수
	점수



프로그래밍하기

우주 로켓 제어

- 초시계 값이 30 이상이 되면 '종료' 신호를 보낸다.

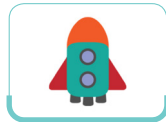


로켓(3)

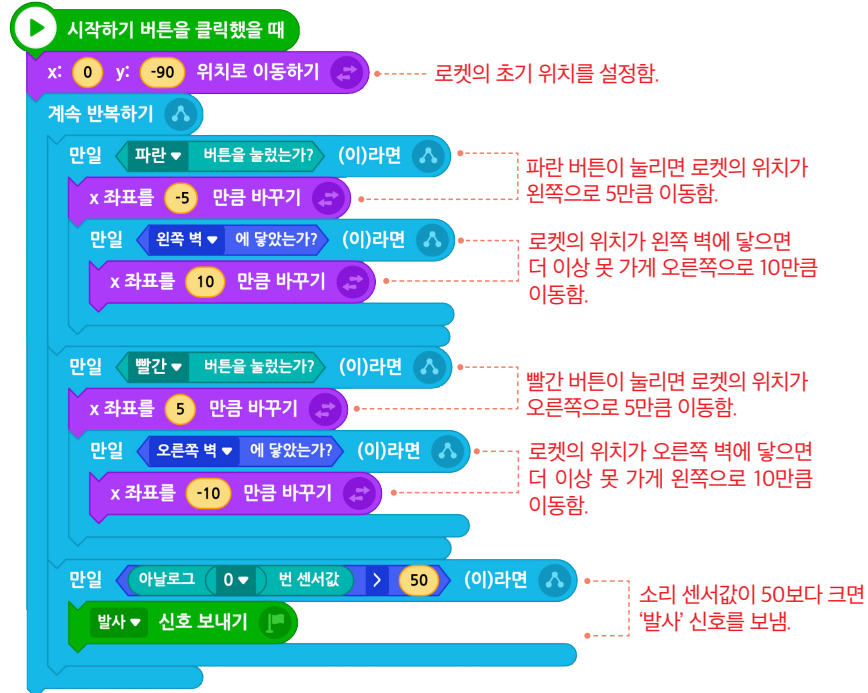


초시계 값이 30 이상이 되면 초시계를 멈추고 종료 신호를 보낸다.

- 센서 보드의 버튼으로 로켓을 좌우로 움직인다.
- 소리 센서값이 50을 초과하면 레이저를 발사한다.



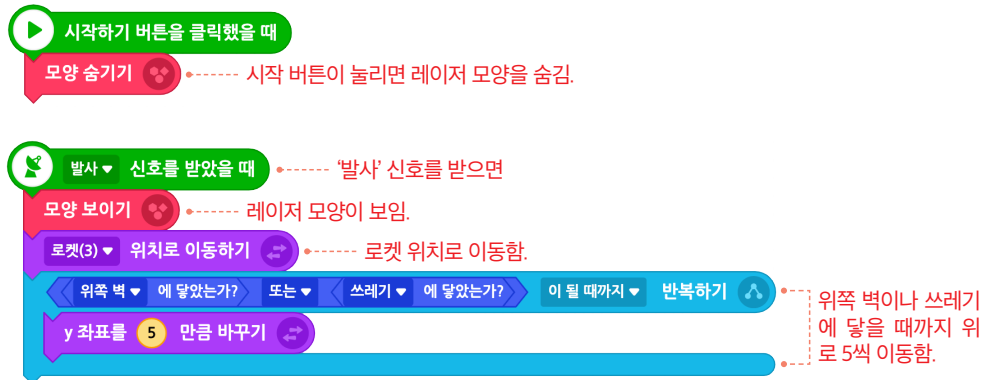
로켓(3)



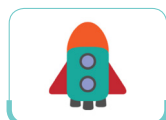
- 소리 센서값이 50 이상이면 레이저가 발사된다. 위쪽 벽이나 쓰레기에 닿을 때까지 화면 위쪽으로 이동한다.



[무음] 폭발 효과



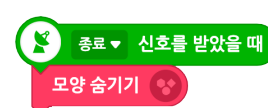
- 로켓과 레이저는 '종료' 신호를 받으면 모양을 숨긴다.



로켓(3)



폭발 효과



우주 쓰레기 청소

- 쓰레기 오브젝트의 위치는 무작위로 생성되며 0.5초마다 복제본을 만든다.



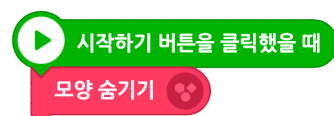
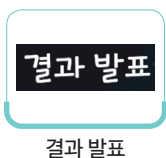
- 쓰레기 복제본이 레이저에 닿으면 점수가 10점 증가하고, 복제본은 삭제된다. 종료 신호를 받으면 모든 복제본이 삭제된다.



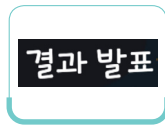
- 현재 획득한 점수를 표시한다.



- 결과 발표 글상자는 시작할 때 숨긴다.



- 30초가 지나고 점수가 300점 이상이면 “우주 쓰레기 청소 성공!”, 아니면 “조금 더 우주 쓰레기를 청소해 주세요!”를 출력한다.



결과 발표



5 프로그램 실행 확인 및 평가하기

완성된 프로그램을 실행시켜 보고 오류가 있는지 확인한다. 오류가 있다면 그 원인과 해결 방법을 찾아 오류를 해결하는 디버깅 과정을 거친다. 그리고 다른 모듈과 완성된 프로그램을 공유한다.



항목	확인
프로그램 시작 시 로켓이 정해진 곳에서 시작되는가?	☐
버튼을 누르면 우주 로켓이 움직이는가?	☐
소리 센서에 바람을 불면 레이저가 발사되는가?	☐
레이저가 쓰레기를 맞으면 쓰레기가 제거되고 점수가 10점씩 증가하는가?	☐
0.5초마다 쓰레기 오브젝트가 복제되는가?	☐
30초가 지나면 프로그램이 종료되는가?	☐
프로그램이 종료되면 로켓, 쓰레기, 레이저 오브젝트의 모양이 사라지는가?	☐
프로그램이 종료되면 결과를 알려 주는 글상자 오브젝트가 출력되는가?	☐



다양한 프로그래밍 언어

프로그래밍 언어의 종류는 여러 가지 방법으로 구분할 수 있지만, 일반적으로 블록 기반 프로그래밍 언어와 텍스트 기반 프로그래밍 언어로 구분한다. 블록 기반 프로그래밍 언어는 주어진 명령 블록을 드래그&드롭하여 프로그램을 제작할 수 있으며 다양한 시각적인 결과를 쉽게 만들 수 있다. 대표적인 블록 기반 프로그래밍 언어는 엔트리, 스크래치 등이 있다.



텍스트 기반 프로그래밍 언어에는 파이썬(Python), C 언어, 자바(Java), 자바스크립트(JS) 등이 있다.

파이썬은 활용도가 높은 쉬운 언어로 최근 인공지능을 포함하여 학계 등에서 광범위하게 사용된다. C 언어는 가장 오래되고 많이 사용되는 언어 중의 하나로 시스템 및 하드웨어 프로그래밍에 사용되는 언어이다. 자바는 이식성이 좋아 모든 플랫폼, 운영체제나 기기 등에서 많이 사용하는 프로그래밍 언어 중의 하나이다.



스스로 정리하기

프로그래밍 관련 단어를 초성을 참고하여 알아맞혀 보자.

하나의 이름으로
여러 개의 데이터를
저장할 수 있는 구조를
말한다.



리스트

제어 구조 안에
또 다른 제어 구조가 1개 이상
포함된 것을 말한다.



중첩구조

프로그램 작성 시
원하는 결과가 나오지 않거나
오류가 발생할 때 이것을 수정
하는 작업을 말한다.



디버깅

어떤 특별한 작업을
수행하도록 여러 개의 명령을
하나로 묶어서 이름을 부여하는
것을 말한다.



함수

조건을 논리적으로
판단하여 참, 거짓의 결과를
출력하는
연산을 말한다.



논리연산

스스로 평가하기

프로그래밍을 하기 위해서는
이 단원에서 배운 여러 가지 개념과
기능을 알아야 해. 100쪽의
질문 Q를 다시 한번 생각해 볼까?



범주	내용 요소	예	아니요
지식·이해	순차적 데이터의 저장, 논리 연산, 중첩 제어 구조, 함수, 디버깅의 개념을 이해할 수 있는가?	☐	101~125쪽
과정·기능	순차적인 데이터의 저장, 논리 연산, 중첩 제어 구조를 활용하여 프로그램을 작성할 수 있는가?	☐	101~117쪽
	함수를 활용하여 프로그램을 모듈화하고, 프로그램의 오류를 발견하여 수정할 수 있는가?	☐	118~125쪽
가치·태도	소프트웨어로 협력과 공유의 가치를 실천할 수 있는가?	☐	126~137쪽
	프로그램의 효과성을 분석하고, 프로그램의 오류를 해결할 수 있는가?	☐	126~137쪽

쿠키 클릭 게임을 만들어 볼까?

프로젝트 목표 | 문제 해결에 적합한 핵심 요소를 추출하고 추상화 과정을 거쳐 알고리즘을 설계하여 프로그램을 작성할 수 있다.

- 마우스로 쿠키를 클릭하여 많이 클릭한 사람이 우승하는 게임 프로그램을 작성해 보자.

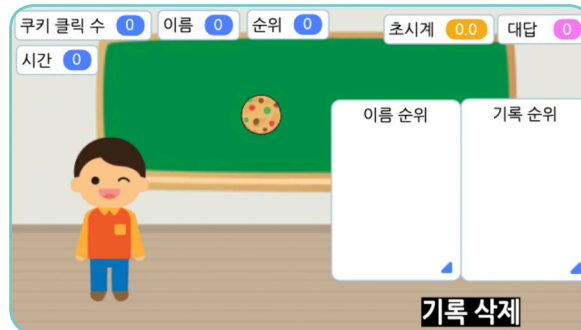


화면 구성

평가 포인트

실행 화면을 적절히 구성하였는가?

- 1 프로그램에 필요한 오브젝트, 변수, 리스트를 만들어 보자.



오브젝트

- 칠판
- 어린이(1)
- 기록 삭제(글상자)

변수

- 쿠키 클릭 수
- 시간
- 이름
- 순위

리스트

- 이름 순위
- 기록 순위

핵심 요소 추출 및 추상화

평가 포인트

프로그램 작성에 필요한 핵심 요소를 모두 찾았는가?

- 2 프로그램에 사용될 핵심 요소를 작성해 보자.

1. 시작하기(안내 메시지, 게임하는 사람 입력 받기, 변수 숨기기, 리스트 숨기기)
2. 게임하기
3. 순위 정하기

알고리즘 설계

3 핵심 요소를 바탕으로 알고리즘을 설계해 보자.

평가 포인트

알고리즘을 적절하게 설계하였는가?

알고리즘 설계

3 알고리즘을 바탕으로 프로그램을 완성해 보자.

 어린이(1)	 쿠키
--	--

평가 포인트

알고리즘에 따라 프로그램을 완성하였는가?

● 완성된 프로그램을 공유하고 나와 친구들의 프로그램을 평가해 보자.

항목	자기 평가	동료 평가
문제 해결에 필요한 핵심 요소를 추출하였는가?	☺ ☹ ☹	☺ ☹ ☹
문제를 해결하기 위한 알고리즘을 명확히 작성하였는가?	☺ ☹ ☹	☺ ☹ ☹
작성한 알고리즘을 토대로 프로그램을 작성하였는가?	☺ ☹ ☹	☺ ☹ ☹
오류가 발생하였을 경우 해결할 수 있는가?	☺ ☹ ☹	☺ ☹ ☹

스스로 풀어보기

1 문제를 해결하는 절차나 방법은?

- ① 디버깅 ② 알고리즘 ③ 문제 이해
- ④ 추상화 ⑤ 핵심 요소 추출

2 알고리즘의 효율성을 따질 때 고려할 내용을 모두 고른 것은?

보기

- ㄱ. 공간을 적게 차지하는가?
- ㄴ. 수행 시간이 적게 걸리는가?
- ㄷ. 알고리즘의 길이가 충분히 긴가?
- ㄹ. 알고리즘이 순차, 반복, 선택 구조를 모두 포함하는가?

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ ⑤ ㄷ, ㄹ

3 다음 선생님의 지시에서 밑줄 친 부분이 충족하지 못한 알고리즘의 조건은?

선생님: 여러분 책상 줄을 잘 맞추어 봅시다. 세 번째 학생 조금만 이동하세요.
학생: 조금만요?

- ① 입력 ② 출력 ③ 명확성
- ④ 유한성 ⑤ 수행 가능성

4 프로그램에서 여러 개의 데이터를 저장할 때 사용하는 것은?

- ① 함수 ② 리스트 ③ 제어 구조
- ④ 논리 연산 ⑤ 중첩 제어 구조

5 다음 프로그램을 실행한 결과 세 번째 항목은?

```

when green flag clicked
  add item to list [장화홍련전] at position 1
  add item to list [흥부전] at position 1
  remove item from list at position 3
  
```

- ① 심청전 ② 흥부전
- ③ 콩쥐팥쥐 ④ 홍길동전
- ⑤ 장화홍련전

6 다음 프로그램을 실행했을 때의 결과는?

```

when green flag clicked
  set a to 10
  set b to 20
  if a > b then say [예] for 2 sec
  else say [아니요] for 2 sec
  
```

- ① 10 ② 20 ③ 30
- ④ 예 ⑤ 아니요

- 7 다음 프로그램을 실행하고 숫자를 입력했을 때 '안녕!'이 출력되지 않는 경우는?



- ① 1 ② 3 ③ 6
④ 9 ⑤ 12

- 8 다음 중 논리 연산자가 포함된 것은?

- ① 1은 2보다 큰가?
② a와 b가 같은가?
③ 2 더하기 3은 얼마인가?
④ 2 곱하기 3은 7과 같은가?
⑤ 키가 150cm 이상이고, 나이가 14세 이상인가?

- 9 다음 프로그램을 실행했을 때의 결과는?



- ① ② ③
④ ⑤

서술형

- 10 문제 해결 과정을 세 가지 상태로 구분하고 각 상태를 서술하시오.

평가 포인트 문제 해결 과정을 세 가지 상태로 구분하고 각 상태의 의미를 정확하게 파악하고 있는가?

- 11 알고리즘의 의미와 중요성에 대해 서술하시오.

평가 포인트 알고리즘의 의미와 알고리즘의 중요성을 이해하고 있는가?

- 12 프로그램을 실행했을 때 다음과 같은 결과가 나타났다. 이러한 현상을 수정하는 과정을 무엇이라고 하는지 쓰고, 그 원인을 파악하여 해결하는 방법을 서술하시오.



경고
빨간색으로 표시된 블록을 확인해 주세요.

평가 포인트 용어의 의미를 알고 프로그램에 오류가 발생했을 때 원인을 파악하여 해결할 수 있는가?