

학습 목표

- 실생활 문제의 목표 상태를 파악하고 목표 상태 도달을 위한 문제 분석과 분해를 할 수 있다.
- 문제 상황의 패턴을 분석하고 핵심 요소를 찾아 다양한 방식으로 추상화 할 수 있다.

다양한 분야에서의 추상화 사례

일반적으로 추상화(abstraction)란 특정한 목적에 관련된 필수적인 정보만을 추출하고 기타 정보는 생략하거나 삭제하여 단순화시키는 것을 말한다. 컴퓨터에서 추상화 역시 이와 같은 의미로 복잡한 자료나 시스템 등으로부터 핵심 개념이나 기능을 간추려 표현하는 것을 말한다.

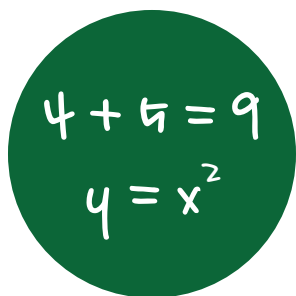
나무를 추상화하기 위해서 나무의 색채와 나뭇잎 형상을 배제하고 곡선으로 표현하다가 수직·수평선만을 이용하여 단순하게 표현할 수 있다.

사과의 개수를 4라는 숫자로 나타내고 여러 사람이 탄 사과를 합하여 '4 + 5 = 9'로 표현하는 것 역시 수와 연산자를 이용하여 추상화 한 것이다. 또한 음의 길이와 높낮이를 음표와 악보로 추상화할 수 있다.

사람들은 의사소통을 위해 상형문자에서 시작하여 오늘날과 같이 다양한 글자를 만들어 사용하며 자신의 뜻을 단순한 문자로 추상화하였는데 이는 언어의 추상화라고 볼 수 있다.



▲ 미술에서의 추상화



▲ 수학에서의 추상화



▲ 음악에서의 추상화



▲ 언어의 추상화

다음 만화를 보며 그림 암호를 풀어 빈칸에 적어 보자.

탐구 자료 부록 3(180p)

홈즈 씨, 저의 부인이 이 그림 때문에 매우 불안해 하고 있습니다. 이 내용이 무엇인지 알아봐 주십시오.

인형 그림이군요, 알겠습니다.

셜록 홈즈 (Sherlock Holmes)

홈즈는 뛰어난 두뇌와 추리로 암호들을 해독해 낸다. 슬래니(Slaney)가 줌추는 인형 암호를 써서 엘시(Elsie) 부인이 보이는 곳에 놓아둔 것이다. 원래 엘시와 슬래니는 약혼한 사이였으나 그가 갱이라는 사실을 알고 엘시는 그로부터 도망을 친 후, 현재의 남편을 만나 결혼한 것이었다.

슬래니 (Slaney)

엘시(Elsie)

이것은 슬래니와 엘시 부인이 예전에 알고 사용하던 암호문이었다. 홈즈는 인형 그림 암호를 풀어서 "am here slaney. come Elsie"라는 메시지라는 것을 알아냈다.

그리고 이미 사랑하는 남편을 만나 잘 살고 있는 엘시 부인이 보낸 답장의 내용은 "N [] [] []"였다.

이렇게 글을 그림으로 추상화하여 뜻을 전달할 수도 있군요.

이에 상처받은 슬래니는 "Elsie prepare to meet the god"이라는 암호문을 보내 겁을 주었던 것이다.

* 정답 Never



생·각·해·보·기

복잡한 의미를 간단한 그림이나 기호로 나타낸 사례를 찾아보자.



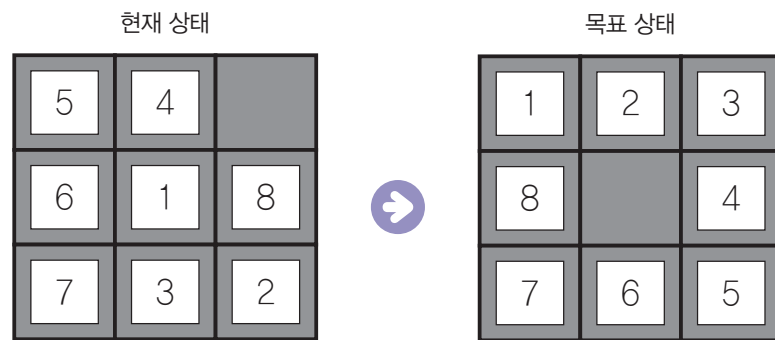
1-1

문제의 이해

현실 세계에 사는 우리는 직업, 교통, 환경, 안전, 의사소통, 금융 등과 관련된 다양한 문제 상황에 처할 수 있다. 컴퓨팅 시스템을 이용하여 이러한 현실 세계의 문제를 해결하기 위해서는 문제 상황을 어떻게 이해하고 분석해야 할까?

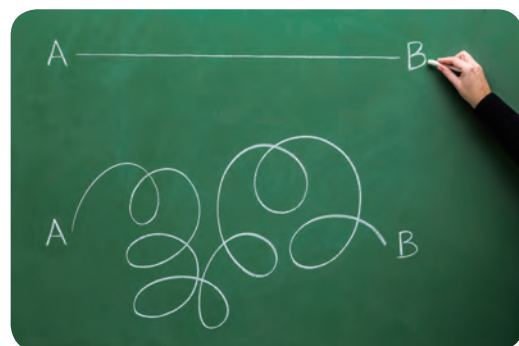
1 문제 상황 파악

문제란 해결해야 하는 일이나 과제를 말한다. 이러한 문제는 주어진 ‘현재 상태’와 원하는 ‘목표 상태’ 사이에 차이가 있기 때문에 발생한다. 이와 같이 ‘현재 상태’를 ‘목표 상태’로 도달시키는 과정을 문제 해결 과정이라고 한다. 따라서 문제 해결을 위해서는 ‘현재 상태’와 ‘목표 상태’를 잘 파악해야 한다. 다음 퍼즐 문제를 생각해 보자. 왼쪽 퍼즐과 오른쪽 퍼즐의 차이는 무엇일까?



▲ 퍼즐 문제

이 퍼즐 문제를 해결하기 위해 무작정 조각을 옮기기보다는 현재 상태와 목표 상태의 차이를 분석하는 것이 필요하다. 문제를 어떻게 파악하는지에 따라서 문제 해결 과정이 다르게 나올 수 있기 때문이다.



▲ 문제 해결 과정의 다양성

● **문제**
현재 상태와 목표 상태의 차이가 존재하는 상황

● **현재 상태**
문제가 발생한 상황

● **목표 상태**
문제가 해결된 상황

● **문제 해결 과정**
현재 상태와 목표 상태의 차이를 줄이며 해결안을 찾는 과정

● **문제 해결**
목표 상태에 도달한 상황

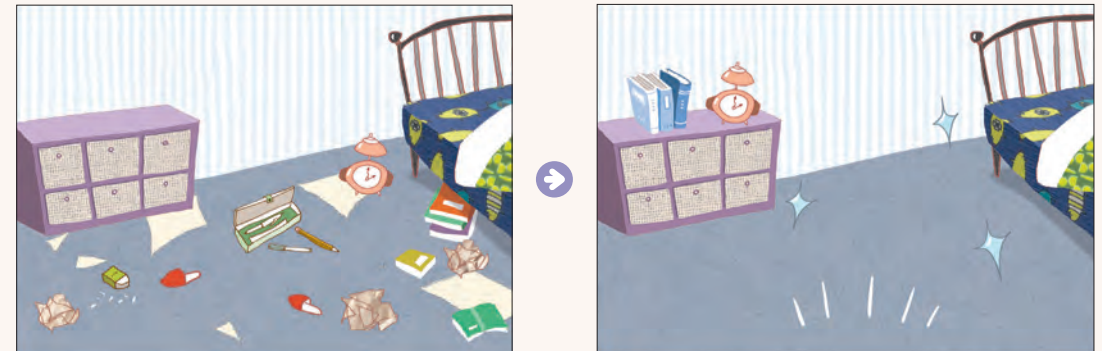


문제 상황 파악하기

1 은아의 방 청소

다음 상황을 읽고 질문에 답해 보자.

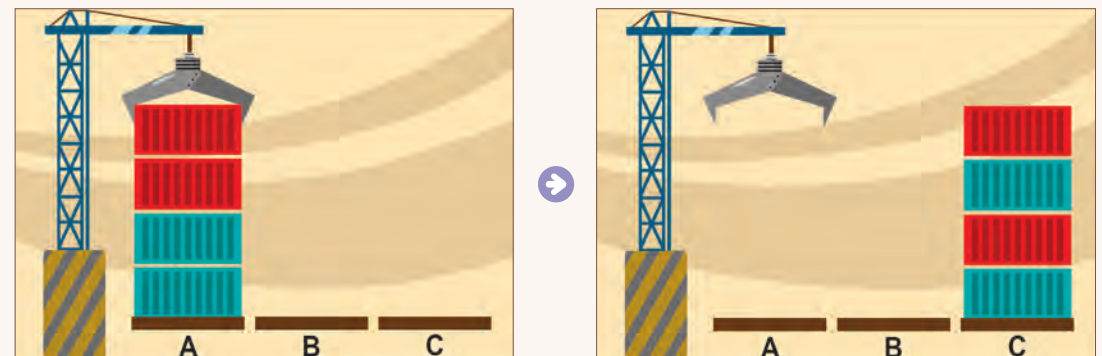
은아는 자기 방에 누워 있다. 방으로 들어온 엄마는 지저분한 방을 보고 은아에게 청소를 하지 않으면 일요일 친구들과의 외출을 금하겠다고 하신다. 하지만 막상 지저분한 방을 치우려고 하니 은아는 고민이 된다.



- 현재 은아의 방 상태는 어떠한가?
- 엄마가 원하는 상황은 어떤 것일까?

2 컨테이너의 블록 옮기기

아래와 같이 왼쪽 그림의 컨테이너를 오른쪽 그림처럼 바꾸고자 한다. 어떻게 옮기면 좋을지 방법을 알아보자. 단, 한 번에 컨테이너 하나씩만 움직일 수 있다.



- 현재 상태
- 목표 상태
- 현재 상태와 목표 상태의 차이점

2 문제 분석과 분해

1 문제 탐색 및 조사하기

문제 해결 과정에서 문제를 정확하게 이해하고 분석하는 것은 매우 중요하다. 이를 위해서는 세밀한 관찰이 필요하며, 관찰을 통해 주어진 문제에서 알아야 할 사실이나 자료가 무엇인지 발견할 수 있다.

로봇 청소기의 문제 분석

은아는 거실을 청소하기 위해 로봇 청소기를 사용하기로 하였다. 한참 후 거실로 돌아와 보니 청소가 제대로 되어 있지 않았다. 은아는 로봇 청소기에 어떤 문제가 있는지 곰곰이 생각해 보았다.

- 거실 그림을 보면서 로봇 청소기가 작동하는 데 어떤 문제가 발생하였는지 추측해 보자.



▲ 은아네 거실

2 문제 분해하기

문제를 이해하고 나면 해결을 위해 문제의 분해 과정을 거친다. 문제의 분해란 주어진 문제를 간단한 작은 문제로 나누는 과정이다. 이렇게 작게 나눈 문제를 하나씩 해결하여 전체 문제를 해결한다.

또한 문제 해결에 사용되는 자료나 절차를 작게 분해하여 컴퓨터가 처리 가능하도록 하는 과정이 필요하다.

예를 들어 스마트폰의 애플리케이션을 구현할 때 통화, 인터넷, 음악 재생기, 카메라 등의 여러 가지 기능을 나누어 생각하고 이를 각각의 개발자에게 분업시킬 수 있다. 또한 카메라의 세부 기능도 더 작게 나누어 여러 프로그래머가 나누어 작업하도록 할 수 있다.



▲ 문제 분해와 분업

- 문제의 탐색과 조사
- 1 무엇을 자동화할지 생각한다.
- 2 자동화를 위한 원리와 방법을 탐색한다.
- 3 자동화에 필요한 자료를 조사한다.
- 4 목표를 이룰 수 있는 조건을 알아본다.



문제의 분해와 해결하기

1 청소 문제 분해

은아와 엄마, 아빠가 집을 효율적으로 청소하기 위해 다음의 기준에 따라 문제를 분해하여 해결해 보자.



가. 청소해야 하는 구역

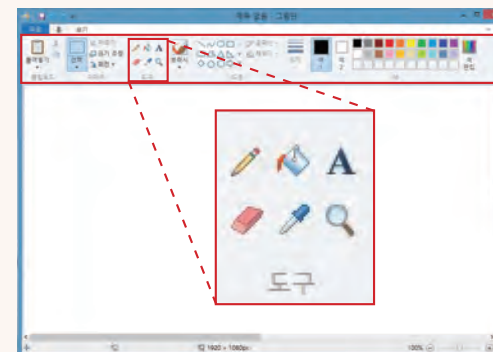
나. 청소할 수 있는 구성원

다. 어떤 청소 방식이 있는가?

- 걸레질을 한다.

2 소프트웨어 모듈(기능) 분해

그림을 그리는 그림판 소프트웨어를 만들려고 할 때 어떤 기능으로 나눌 수 있을지 생각해 보자.



가. 그림판에서 사용되는 기능

나. 그림을 그리는 도구의 유형

3 자동차 번호판 인식을 위한 분해

오른쪽 사진과 같이 자동차 번호판을 자동으로 인식하기 위해서는 자동차 본체와 번호판을 분해하여야 한다. 자동차 번호판을 컴퓨터가 인식하기 위해서는 어떤 식으로 분해해야 할까? 어떻게 나누어 인식하면 좋을지 선을 그어 보자.

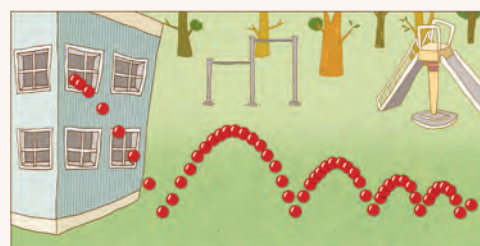
34 가 8750





핵심 요소 찾기

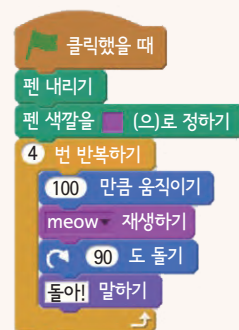
- 1 그림과 같이 공이 튕기는 궤적의 패턴을 찾아 시뮬레이션을 만들고자 한다. 이 궤적의 패턴을 표현하는데 필요한 요소는 어떤 것인지 보기에서 찾아보자.



보기

공의 크기	공의 색깔
공이 튕기는 높이	공이 나아간 거리
공의 구입 가격	공이 멈춘 곳까지의 거리

- 2 로봇에게 정사각형을 그리게 하기 위해 다음과 같은 명령어를 작성하였다. 정사각형을 그리기 위해 필요한 명령어와 불필요한 명령어로 나누시오.



필요한 명령어	불필요한 명령어

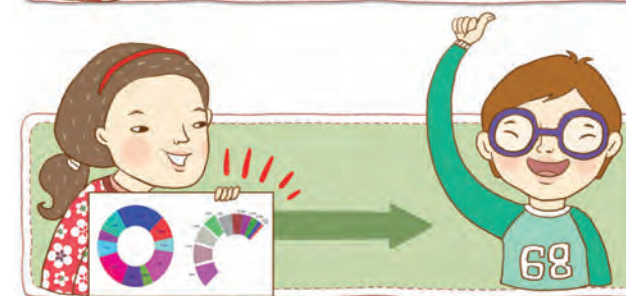
- 3 1~100 사이에서 하나의 수를 정하고 스무고개 놀이처럼 상대방이 그 수를 맞히게 하려고 한다. 수를 정한 사람은 상대방이 말한 수와 비교하여 큰 수인지 작은 수인지만 알려줄 수 있다. 가장 적은 횟수로 그 수를 맞히기 위한 규칙을 찾을 때의 필요한 핵심 요소를 찾아보자.



2 추상화

1 추상화의 개념과 이해

발견한 패턴을 바탕으로 필요한 것에 집중할 수 있도록 중요한 것은 남기고 무시할 것은 걸러 내어 단순하게 표현하는 과정을 추상화라고 한다. 문제 상황이나 전달하고자 하는 내용을 정리하지 않고 상대방에게 모든 세부 내용을 설명하다 보면 이해하기 쉽지 않고 핵심 내용이 제대로 전달되지 않을 수 있다. 반면 문제의 핵심을 간단한 도식, 표, 공식 등으로 정리하여 전달하면 쉽게 이해시킬 수 있다.



컴퓨터에서의 추상화(abstraction)는 주어진 문제에서 필요 없는 내용은 빼고 핵심적인 개념 또는 기능을 간추려 그 특징과 패턴을 알기 쉽게 표현하는 것이다.

추상화는 문제를 알고리즘으로 표현하고 프로그래밍하기 위해서도 반드시 필요한 과정이다. 추상화는 생각의 단위를 법칙이나 공식으로 표현할 수도 있고 동시에 그것이 무엇을 의미하는지 알게 해 준다.

추상화의 목적은 다음과 같다.

- 문제가 명확해지고 규칙적이 되어 이해하기가 쉽다.
- 프로그램의 개발과 유지 보수가 쉽다.
- 프로그램의 확장성이 쉽다.

추상화는 작은 수준의 의미를 함수로 표현하는 것이므로 작은 수준으로 문제를 분해하고 패턴을 인식하는 과정이 필요한 것이다. 즉 추상화는 단순하게 만드는 것이 핵심이며, 하나의 함수는 하나의 일을 하도록 해야 한다.

● 함수
어떤 재료를 넣어 새로운 값을 만드는 틀



● 추상화 표현 방법

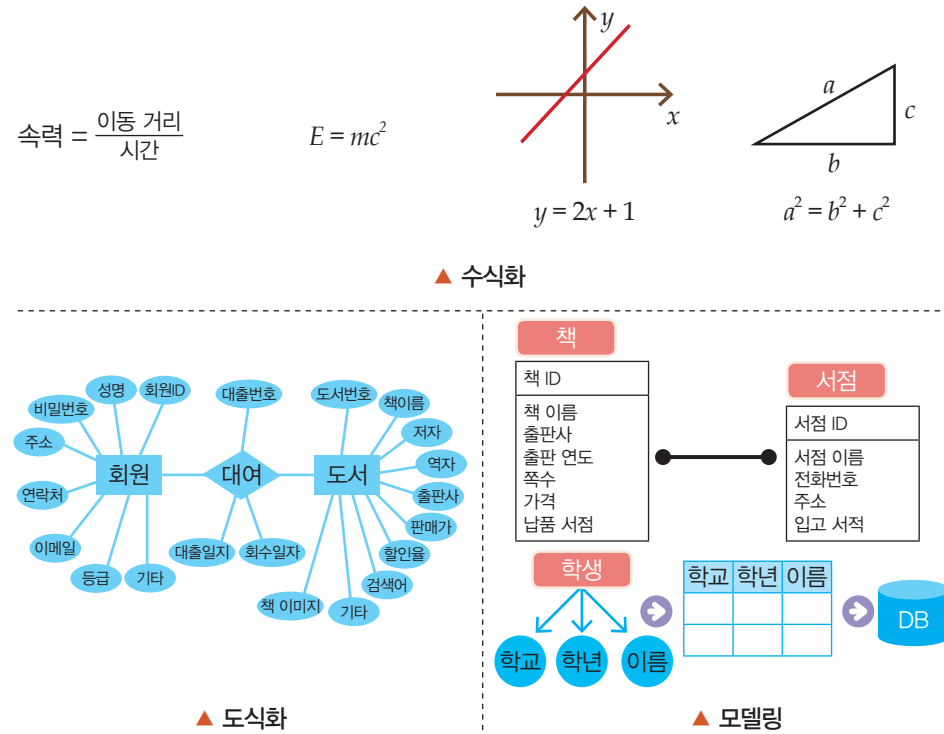
- 명명화
- 수식화
- 모델링
- 도식화
- 기호화

● 모델링(modeling)

현실 세계에 존재하는 자료(데이터)를 컴퓨터 세계의 데이터베이스로 옮기는 변환 과정

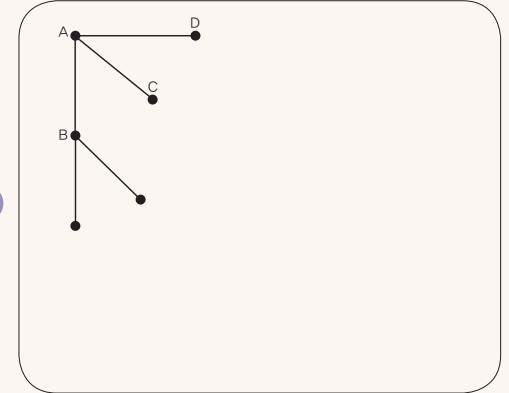
2 추상화의 표현

추상화의 표현 방법에는 다음과 같은 다양한 방법이 있다.



추상화하기

1 아래 그림은 마을 지도이다. 건물은 알파벳과 점으로, 도로는 선으로 표기하자. 마을 지도를 간단하게 나타내기 위해 건물과 도로를 선과 점의 도식으로 그려 보자.



2 교통 수단의 속력을 측정하기 위해 이동 거리, 시간과 속력의 관계를 관찰하고 그 패턴을 찾아 수식으로 나타내 보자.

	이동 거리(km)	시간(h)	속력(km/h)
자전거	55	5	11
자동차	100	2	50
기차	560	4	140

3 다음은 다양한 몬스터를 그리기 위한 지시문이다. 다음 빈칸에 원하는 숫자를 넣고 짝에게 지시문에 맞게 그리도록 해 보자.

몬스터 그리기 지시문	지시문을 따라 그린 몬스터
____ 모양의 얼굴 ____ 개를 그리시오. ____ 모양의 입 ____ 개를 그리시오. ____ 모양의 코 ____ 개를 그리시오. ____ 모양의 눈 ____ 개를 그리시오. ____ 모양의 귀 ____ 개를 그리시오.	

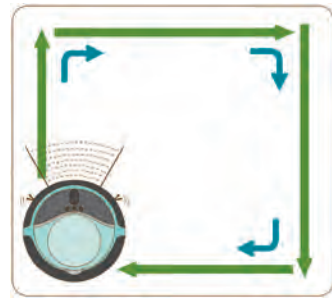
위 지시문을 아래와 같이 표로 나타내고자 할 때 행과 열의 머리글로 알맞은 요소를 적어 보자.

	세모	네모	동그라미	별	물결
	1	2	4	2	3

실 | 습 | 활 | 동

오른쪽 그림과 같이 로봇 청소기가 사각형 모양으로 이동하도록 하고자 한다. 다음 질문에 알맞은 답을 <보기>의 단어를 이용하여 쓰시오.

보기
직진하기, 돌기, 왼쪽 아래, 사각형 모양



1 문제 상황의 현재 상태와 목표 상태를 설명하시오.

현재 상태	에서 대기
목표 상태	모양으로 돌며 움직이기

2 문제 분해: 그림에서 나타나는 로봇 청소기의 세부 동작은 , , 장애물 감지하기 등이다.

3 패턴 찾기: 로봇 청소기가 사각형을 그리며 방을 도는 과정에서 반복되는 패턴은 벽에 닿을 때까지 , 90도 이다.

4 핵심 요소 찾기: 로봇 청소기가 사각형을 그리며 움직이게 하는 기능에 대한 핵심 동작 요소는 와 이다.

5 추상화: 로봇 청소기는 '직진하다 벽에 닿으면 90도 '의 패턴을 4번 반복한다.

자 | 기 | 평 | 가

평가 내용	잘함	보통	노력 요함
1. 실생활 문제 상황을 이해하여 작게 분해할 수 있다.			
2. 문제 상황에서 핵심 요소를 파악할 수 있다.			
3. 목표 상태에 도달하기 위한 패턴을 추상화할 수 있다.			

토 | 론 | 하 | 기

미술, 수학, 음악, 언어에서의 추상화와 컴퓨터 과학에서 추상화의 공통점과 차이점에 대해 토의해 보자.

공통점	차이점

배운 내용 정 | 리 | 하 | 기



평 | 가 | 하 | 기

※ 다음 문제 상황을 읽고 질문에 답해 봅시다. (1~3)

우리 가족은 주말마다 농장에 가서 감자와 고구마 등을 심어서 가꾸고 있다. 주말에만 농장에 갈 수 있어 주중에는 물을 주기 어렵다. 주중에도 땅이 마르거나 날이 더우면 자동으로 물을 주는 스프링클러가 필요하다.

① 위 문제 상황에 대한 목표 상태는 무엇인가?

② 스프링클러가 자동으로 작동하기 위한 핵심 요소는 무엇인가?

③ 스프링클러, 습도 센서, 온도 센서 등의 관계를 추상화하여 도식화해 보시오.

학습 목표

- 사람과 컴퓨터의 문제 해결 방법의 차이를 설명할 수 있다.
- 알고리즘의 개념을 이해하고 중요성을 인식할 수 있다.
- 알고리즘을 다양한 방법으로 표현하고 설계할 수 있다.

주사위 게임 규칙 만들기

은아와 지원은 주사위 게임을 직접 만들어서 놀기로 하고 게임 방식을 상의하여 다음과 같이 규칙을 정하였다.



	TURN 1	TURN 2	TURN 3	TURN 4	TURN 5	TURN 6	TOTAL
Player 1	_____	_____	_____	_____	_____	_____	_____
Player 2	_____	_____	_____	_____	_____	_____	_____

- 1 가위바위보를 해서 누가 먼저 주사위를 굴릴지 결정한다.
- 2 모든 사람의 점수는 0으로 시작한다.
- 3 한 사람이 주사위를 굴린다.
- 4 주사위에 나온 숫자를 이전 숫자에 더한다.
- 5 다음 사람이 주사위를 굴린다.
- 6 규칙 4~5 과정을 여섯 번 반복한다.
- 7 최종 숫자를 확인한다.
- 8 숫자가 큰 사람으로 승자를 결정한다.

게임 규칙을 정하는 과정에서 처음에 지원이 규칙 3~5번 과정을 총 6번 반복하여 적어 놓았다. 은아가 반복되는 규칙인 3~5번은 한 번만 적고 규칙 6번을 은아가 추가하자 규칙의 개수가 줄고 이해하기 쉬워졌다.

이렇게 문제를 해결하기 위한 일련의 절차 및 순서를 알고리즘이라고 한다. 이러한 순서와 과정이 명확히 제시되지 않는다면 주어진 일을 제대로 수행할 수 없을 것이다. 주어진 일을 처리하는 방법은 다양하지만 일을 신속하고 정확하게 처리하기 위해서는 알고리즘을 효율적으로 만들 필요가 있다.

다음 만화를 보며 정보를 효율적으로 처리하는 데 알고리즘이 왜 필요한지 알아보자.



생·각·해·보·기

위의 만화를 보고 알고리즘이 적용된 실생활의 예를 더 찾아보자.



2-1 알고리즘의 이해

유명인의 프로필을 찾고자 할 때 검색 엔진은 어떠한 절차로 적합한 검색 결과를 보여 줄까? 스마트폰으로 지문을 인식하는 절차는 어떤 방법을 사용할까? 이러한 문제를 해결하는 효율적인 절차와 방법에 대해 알아보자.

1 알고리즘의 개념 및 조건

알고리즘이란 문제를 해결하기 위한 방법을 순서대로 나열한 절차이다. 일상생활에서 해결해야 할 문제에 직면했을 때 해결 과정을 구체적으로 계획한 후에 실행해야 한다. 이때 여러 가지 해결 방법과 단계별 순서를 생각해보고, 그 중 가장 적합한 방법을 선택하여 실행해야 한다.

여러 개 나열된 숫자 카드 중에서 임의의 숫자를 어떻게 찾아낼까?

임의의 숫자 찾기

제시된 문제 상황: 10장의 뒤집힌 카드에서 임의의 숫자 87을 찾는 방법에는 어떤 것들이 있을까?

뒷면										
앞면	43	19	65	32	14	50	90	87	79	25

찾는 방법 적어 보기

- 1 아무 카드나 무작정 뒤집어 본다.
- 2 가장 앞에 있는 카드부터 뒤집어 본다.

3

4

좋은 알고리즘이 갖추어야 할 조건은 입력과 출력이 있어야 하며, 유한성, 수행 가능성 그리고 명확성이다.

2 알고리즘의 효율성

주어진 문제에 대해 여러 가지의 해결 방법이 있는 것처럼 알고리즘도 여러 종류의 알고리즘이 있을 수 있으나 보다 효율적인 알고리즘을 고안하는 것이 중요하다. 똑같이 주어진 일이라도 어떠한 방식을 사용하느냐에 따라 비용과 효율성에서 많은 차이가 나기 때문이다.

정렬되어 있는 숫자 카드를 찾는 두 가지 방법을 비교해 보자.

오름차순으로 정렬된 카드에서 임의의 숫자 찾기

제시된 문제 상황: 10장의 뒤집힌 카드에서 임의의 숫자 87을 찾는 방법에는 어떤 것이 있을까?

14	19	25	32	43	50	65	79	87	90
----	----	----	----	----	----	----	----	----	----

▶ 10장의 카드가 오름차순으로 정렬되어 있을 때 숫자 87을 찾아내는 방법은 어떤 것이 더 효율적일까?

방법 1 순서대로 찾기

▶ 10장의 카드를 왼쪽부터 순서대로 읽어가며 탐색

1	2	3	4	5	6	7	8	9	
14	19	25	32	43	50	65	79	87	90

14 < 87 19 < 87 25 < 87 32 < 87 43 < 87 50 < 87 65 < 87 79 < 87 87 = 87

▶ **방법 1 순차 탐색(sequential search)**

정렬되어 있는 자료를 순서대로 탐색하는 방법

▶ **방법 2 이진 탐색(binary search)**

오름차순으로 정렬된 자료를 반으로 나누고, 나누어진 반을 다시 반으로 나누는 과정을 반복하여 원하는 결과를 찾는 탐색 방법

방법 2 반으로 나누어 찾기

- ▶ 중간에 있는 카드의 숫자인 43이나 50과 비교
- ▶ 43은 87보다 작으므로 43보다 오른쪽에 있는 자료들을 탐색
- ▶ 오른쪽에 있는 값을 다시 반으로 나누어 65나 79를 87과 비교해 가면서 87을 탐색

14	19	25	32	43	50	65	79	87	90
				43 < 87					
					65 < 87				
					50	65	79	87	90
							79 < 87		
							79	87	90
									87 = 87
							87	90	

Think

▶ **방법 1** 과 **방법 2** 는 87을 찾기 위해 각각 몇 번의 비교 횟수가 필요한가?

● 오름차순 정렬

알파벳 또는 숫자 자료를 작은 것부터 큰 것의 순으로 나열하는 방법이다. 반대의 경우는 내림차순 정렬이라고 한다.

● 최초의 알고리즘

기원전 300년에 유클리드의 최대 공약수를 찾는 알고리즘이 최초라고 할 수 있다.

입·출력

주어진 입력에 대해 올바른 결과를 출력해야 한다.

유한성

주어진 시간 내에 문제가 해결되어야 한다.

수행 가능성

반드시 실행 가능한 것이어야 한다.

명확성

각 단계는 명확하여 애매함이 없어야 한다.

▲ 알고리즘의 조건

숫자 카드 중 임의의 수를 찾아내는 사례에서 보듯이 어떤 알고리즘을 사용하느냐에 따라 탐색 횟수가 9번이 될 수도 있고 4번이 될 수도 있다. 숫자 카드가 많아질수록 탐색(비교) 횟수에 더 많은 차이가 난다. 수많은 자료를 처리하는 컴퓨터의 경우 대량의 자료를 반복 처리하는 일을 하므로 알고리즘을 효율적으로 작성해야만 합리적으로 문제를 해결할 수 있게 된다. 예를 들어 1,000만 건의 자료를 처리하는 데 있어 한 건당 0.1초씩의 시간 손해가 발생했다고 하면 무려 11일의 시간 손해가 발생하게 된다.

따라서 효율적인 알고리즘이 컴퓨터의 성능과 소프트웨어의 처리 비용에 중요한 영향을 미친다.



2-2 알고리즘의 표현

음악을 연주하기 위해 악보로 표현하고 수학 문제를 해결하기 위해서 공식으로 표현한다. 컴퓨터로 문제를 해결하기 위한 알고리즘은 어떻게 표현하는지 살펴보자.

1 알고리즘 표현 방법

알고리즘을 표현하는 방법은 다양하나 일반적으로 자연어, 의사 코드, 순서도 등의 방법을 사용한다.

1 자연어

자연어 표현 방법은 우리가 일상생활에서 사용하는 언어로 알고리즘을 표현하는 방식이다. 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 자연어로 표현한 예를 살펴보자.

자판기의 동작 원리

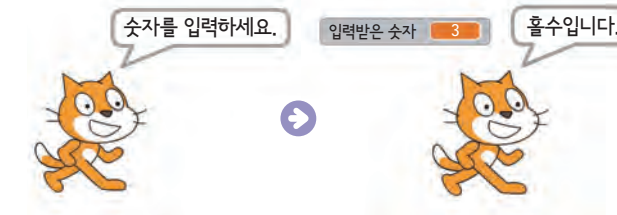
700원짜리 음료수 버튼이 눌러지면 동전이 투입되길 기다린다. 동전이 투입되었을 때 700원보다 많으면 음료수와 잔돈을 돌려준다. 700원과 같으면 음료수만 주고, 700원보다 적으면 음료수를 내보내지 않고 받은 돈을 돌려준다.



짝수·홀수의 판별

임의의 숫자를 입력받아 그 값을 2로 나눈 후 나머지가 0이면 짝수, 아니면 홀수를 출력한다.

문제 해결 순서



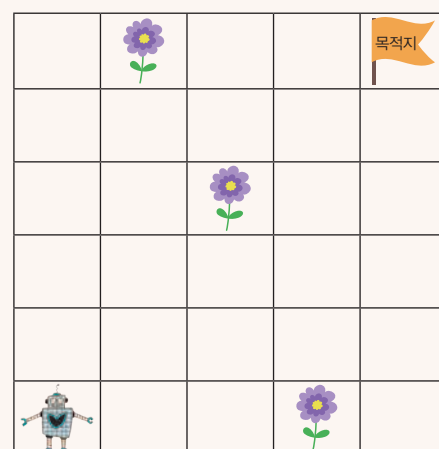
● 자연어 표현 방식

인간의 언어를 사용하기 때문에 쉽게 이해할 수 있다는 장점이 있지만, 읽는 사람에 따라 모호한 해석이 이루어질 수 있으므로 최대한 명료하게 표현해야 한다.

예 사과 크기를 대략 300g 초과하는 대, 300g~250g 사이는 중, 250g 미만은 소로 구분하자.

로봇을 목적지까지 이동시키는 알고리즘

로봇을 목적지까지 이동시키려 한다. 로봇은 한 번에 한 칸씩 이동 가능하며 은 장애물이다. 명령어를 이용하여 목적지까지 이동하도록 순서대로 알고리즘을 만들어 보자. (장애물은 피해 가야 한다.)



명령어

- ↑ 위로 1칸 이동
- ↓ 아래로 1칸 이동
- ← 왼쪽으로 1칸 이동
- 오른쪽으로 1칸 이동
- ↷ 오른쪽으로 90도 회전
- ↶ 왼쪽으로 90도 회전

이동한 순서의 명령어 또는 이동 기호

- 탐구 과제: ① 친구들이 해결한 알고리즘 중 누구의 알고리즘이 더 나은지 비교해 보자.
② 누구의 알고리즘이 좋은지 비교해 보는 기준은 무엇인가?
③ 다음과 같은 명령어가 추가된다면 로봇의 이동 알고리즘은 어떻게 될까?

×2 2회 반복 ×4 4회 반복

● **의사 코드(pseudo code)**

- 의사(pseudo)는 '모조 의', '흉내 낸'이라는 의미이고, 코드(code)는 프로그래밍 코드를 말한다.
- 특정 프로그래밍 언어에 대한 지식이 없어도 알기 쉽게 표현이 가능하고, 다른 프로그래밍 언어로 전환하기 쉽다.

● 의사 코드를 사용할 경우 들여쓰기를 하면 이해하기 쉬워진다.

● **순서도**

1965년 10월 SC7 총회에서 표준안으로 결의된 기호를 사용한다. 순서도의 기호는 총 30개이며 기본 기호, 프로그래밍 관련 기호, 시스템 관련 기호 등 3가지 종류로 나뉜다.

기호와 그림을 사용하므로 알고리즘의 구조나 특성을 한눈에 파악하기 쉽고 오류를 쉽게 찾을 수 있다.

2 의사 코드

자연어의 표현 방법을 간략하게 구조화하여 표현한 것으로 프로그래밍 언어보다 이해하기 쉽다. 음료 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 의사 코드로 살펴보자.

자판기의 작동 원리

- ① $A \leftarrow$ 받은 동전
- ② $A > 700$
예) $B \leftarrow A - 700$
'B', '음료수' 출력
아니오) $A = 700$?
예) '음료수' 출력
아니오) 'A' 출력



짝수·홀수의 판별

- ① $A \leftarrow$ 입력 값
- ② $A \div 2$ 의 나머지 = 0?
예) '짝수' 출력
아니오) '홀수' 출력

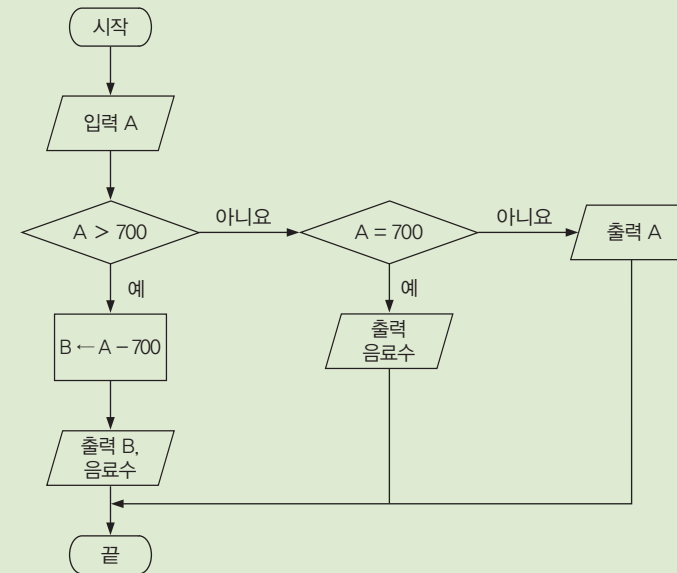
3 순서도

순서도는 미리 약속된 기호나 그림 등을 이용해 논리적 절차, 흐름, 처리 방법 등을 표현하는 방법이다. 순서도는 다음과 같은 기호를 사용한다.

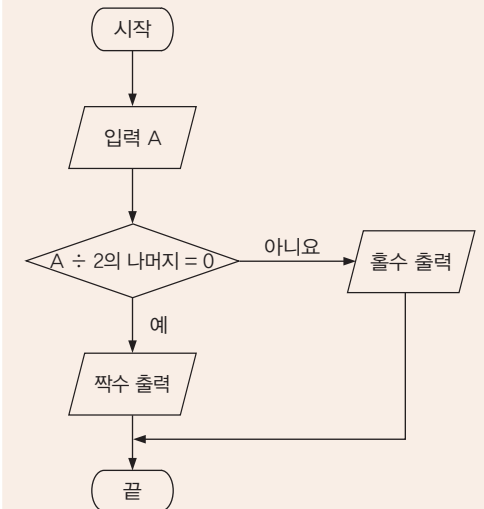
기호	명칭	의미
	단말 기호	알고리즘의 시작과 끝
	처리 기호	각종 연산이나 자료의 이동, 편집 등의 처리
	판단 기호	조건의 참, 거짓 등을 판단
	자료 입출력	자료의 입력 및 출력
	준비 기호	변수의 선언 및 초기값을 설정
	흐름선	실행 순서

음료 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 순서도로 살펴보자.

자판기의 작동 원리



짝수·홀수의 판별



알고리즘을 표현하는 설계는 그림을 그리기 전에 밑그림을 그리는 과정과 비슷하다. 알고리즘을 표현할 때는 전체의 흐름을 쉽게 이해할 수 있도록 흐름을 규칙적이고 단순하게 작성하는 것이 좋다.

알고리즘을 나타낼 때 실제로 해 보면 어느 하나의 표현 방법만 사용하는 것보다 세 가지를 함께 사용하여 알고리즘을 작성하는 것이 더 좋은 경우가 많다.

2 알고리즘의 설계

문제를 해결하기 위한 방법과 절차를 구체적으로 표현하는 것을 알고리즘 설계라고 한다. 알고리즘을 표현하는 데 사용되는 설계 구조에는 순차 구조, 선택 구조, 반복 구조가 있다.

1 순차 구조

순차 구조는 가장 간단하게 표현되는 구조로, 특별한 분기 또는 반복 없이 처음부터 끝까지 순서대로 처리하는 구조이다.

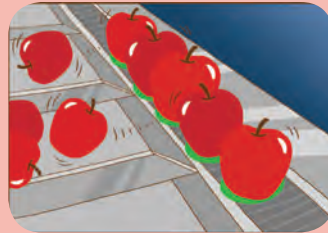
다음은 입력받은 두 수의 합을 순차 구조로 표현한 예이다.





문제 해결 과정을 자연어·의사 코드·순서도로 표현하기

문제 상황: 컨베이어 벨트로 이동하는 사과를 무게에 따라 구분하여 3개의 상자로 나누는 장치의 작동 알고리즘을 만들어야 한다. 컨베이어 벨트로 사과가 이동하여 300g 초과는 '대' 등급 상자로 이동, 300~250g은 '중' 등급 상자로 이동, 나머지는 '소' 등급 상자로 이동시킨다.



1 자연어로 표현하기

컨베이어 벨트로 사과 A가 이동한다.

2 의사 코드로 표현하기

의사 코드 표현

사과 A 입력

3 순서도로 표현하기

순서도 표현



입력받은 두 수의 합을 출력하는 알고리즘

자연어	의사 코드	순서도
첫 번째 수를 입력받아서 A에 저장한다. 두 번째 수를 입력받아서 B에 저장한다. A의 값과 B의 값을 더하여 C에 저장한다. C의 값을 출력하고 끝낸다.	① A 입력 ② B 입력 ③ $C \leftarrow A + B$ ④ C 출력	<pre> graph TD Start([시작]) --> Input[/INPUT A
INPUT B/] Input --> Process[C ← A + B] Process --> Output[/PRINT C/] Output --> End([끝]) </pre>

● **순서도 속 연산자의 의미**
 \leftarrow : 오른쪽의 값을 왼쪽의 값에 넣는다.
 $=$: 오른쪽의 값과 왼쪽의 값이 같다.

2 선택 구조

주어진 조건에 따라 처리하는 내용이나 순서를 다르게 진행하는 구조이다. 주어진 조건의 만족 여부에 따라 결과 값이 달라지며, 조건을 만족하면 “예(Yes)”의 흐름선을 따라 수행하고, 조건을 만족하지 않으면 “아니요(No)”의 흐름선을 따라 수행한다.

다음은 주민 등록 번호에서 성별을 판별하는 방법을 선택 구조로 표현한 예이다.

주민 등록 번호에서 성별 판별 알고리즘

의사 코드	순서도
① 주민 등록 번호 입력 ② $A \leftarrow$ 주민 등록 번호 7번째 자리 숫자 ③ $A = 1$ or 3? 예) '남성' 출력 아니요) $A = 2$ or 4? 예) '여성' 출력 아니요) ①번으로 다시 입력 받기	<pre> graph TD Start([시작]) --> Input[/INPUT 주민 등록 번호/] Input --> Process[A ← 주민 등록 번호
7번째 자리 숫자] Process --> Dec1{A = 1 or 3?} Dec1 -- 예 --> Output1[/남성/] Dec1 -- 아니요 --> Dec2{A = 2 or 4?} Dec2 -- 예 --> Output2[/여성/] Dec2 -- 아니요 --> Input Output1 --> End([끝]) Output2 --> End </pre>

3 반복 구조

주어진 조건을 만족할 때까지 특정 작업을 반복하여 수행하는 구조를 말한다. 다음 예제와 같이 동일한 명령을 반복할 때에는 반복 구조를 사용하여 표현하면 간편하고 효율적이다.

아래와 같은 프로그램을 작성하여 실행시키면 오른쪽과 같은 글자가 출력된다.

프로그램	화면 출력
화면 표시 "반갑다"	반갑다
화면 표시 "친구야"	친구야
화면 표시 "반갑다"	반갑다
화면 표시 "친구야"	친구야

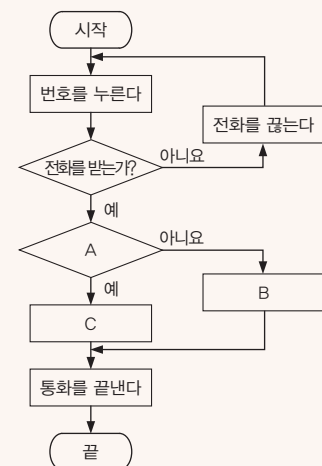
그러나 이렇게 두 가지 이상의 명령이 반복될 경우 다음과 같이 표현하면 좀 더 효율적으로 표현할 수 있다.

프로그램	화면 출력
반복하기 2번 { 화면 표시 "반갑다" 화면 표시 "친구야" }	반갑다 친구야 반갑다 친구야

● 예제 프로그램에서 '반복하기 2번 { }'은 { }안의 내용을 2번 반복해서 화면에 표시한다는 의미이다.

문제 해결 과정 순서도 표현하기

승훈이 지원에게 집으로 전화를 걸어 용건을 말하는 순서는 아래와 같다. 문제 해결 과정을 순서도로 표현할 때 빈칸 A, B, C에 들어갈 적합한 말을 적으시오.



순서

- 가. 전화번호를 누른다.
 나. 상대방이 전화를 받는지 확인한다.
 다. 통화가 되면 받은 사람이 지원이 맞는지를 확인한다.
 라. 지원이 맞으면 통화 내용을 말한다.
 마. 지원이 집에 없으면 받은 사람에게 전할 내용을 말한다.
 바. 상대방이 전화를 받지 않으면 전화를 끊는다.
 사. 통화를 끝낸다.

A

B

C

반복 알고리즘 작성하기

1 화면에 아래와 같은 글자가 출력되도록 반복 구조로 표현하기 위해 _____을 채워 보자.

화면출력

교통봉사
 양로원봉사
 교실청소
 교통봉사
 양로원봉사
 교실청소
 교통봉사
 양로원봉사
 교실청소

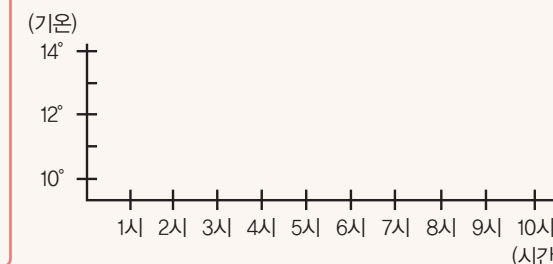
프로그램

반복하기 _____번
 {
 화면 표시 "_____"
 화면 표시 "_____"
 화면 표시 "_____"
 }

2 기상 관측소에서 1시간마다 기온을 측정하는 알고리즘이다. 알고리즘을 보고 실행한 결과를 그래프로 나타내 보자.

시작
 시작 시각 ← 1
 반복하기 10번
 {
 매 한 시간마다 기온 측정하기
 시간과 기온 data에 저장하기
 그래프에 동그라미로 기온 표시하기
 }
 끝

data (1, 12), (2, 11.5), (3, 11), (4, 11), (5, 12),
 (6, 12.5), (7, 13), (8, 12), (9, 11), (10, 10)



3 알고리즘의 활용

컴퓨터에서 자료를 저장하고 처리하기 위해서는 정렬의 과정이 필요하다. 정렬과 탐색은 자료를 구조화하고 결과를 처리하기 위해 매우 중요한 알고리즘이다.

정렬은 여러 개의 자료를 일정한 기준에 따라 순서대로 나열하는 것이다. 컴퓨터로 숫자를 정렬하기 위해 다양한 정렬 알고리즘을 사용한다. 그 가운데 버블 정렬(bubble sort)은 인접한 두 자료를 비교하여 가장 큰 데이터를 가장 뒤로 보내며 자리를 교환하는 방식이다.

탐색은 원하는 자료를 찾는 방법으로 순차 탐색과 이진 탐색이 있다.

● 정렬 알고리즘의 중요성

탐색한 자료가 정렬되어 있지 않다면 순서대로 탐색하는 방법밖에 없다. 하지만 정렬되면 보다 다양한 탐색 방법이 사용될 수 있다.

다음은 버블 정렬을 이용하여 5개의 숫자를 오름차순으로 나타내는 방법이다.

5장의 숫자 카드를 버블 정렬 방법으로 오름차순으로 나타내기

단계	진행 과정	진행 내용
정렬되지 않은 상태	45 20 60 35 10	인접한 두 숫자를 비교하여 왼쪽의 숫자가 크면 자리 교환이 일어난다.
1단계	45 20 60 35 10	45, 20 중 45가 더 크므로 교환함.
	20 45 60 35 10	45, 60 중 60이 더 크므로 교환 안 함.
	20 45 60 35 10	60, 35 중 60이 더 크므로 교환함.
	20 45 35 60 10	60, 10 중 60이 더 크므로 교환함.
	20 45 35 10 60	60은 가장 큰 숫자로 고정됨.
2단계	20 45 35 10 60	20, 45 중 45가 더 크므로 교환 안 함.
	20 45 35 10 60	45, 35 중 45가 더 크므로 교환함.
	20 35 45 10 60	45, 10 중 45가 더 크므로 교환함.
	20 35 10 45 60	45는 두 번째 큰 숫자로 고정됨.
3단계	20 35 10 45 60	20, 35 중 35가 더 크므로 교환 안 함.
	20 35 10 45 60	35, 10 중 35가 더 크므로 교환함.
	20 10 35 45 60	35는 세 번째 큰 숫자로 고정됨.
4단계	20 10 35 45 60	20, 10 중 20이 더 크므로 교환함.
	10 20 35 45 60	오름차순 정렬이 완료됨.
정렬이 완료된 상태	10 20 35 45 60	오름차순 정렬이 완료됨.

정렬 알고리즘은 데이터베이스, 인공지능 등의 다양한 분야에 적용되어 우리의 실생활 문제를 해결하며 세상을 변화시키는 데 중요한 역할을 하고 있다. 정렬 알고리즘 이외에 정보 통신 기술의 발달을 가져온 대표적인 알고리즘은 다음과 같다.



▲ 대표적인 알고리즘의 종류

다양한 자료 정렬해 보기

다음과 같은 다양한 카드를 버블 정렬을 이용해서 정렬해 보자. 몇 번의 교환 과정이 필요한가?

1 영어 단어 정렬(알파벳순)

create earth apple think computer binary

▲ 영어 카드

2 여러 가지 음표 정렬(음표 길이순)

♪ ♪ ♫ ♯ ♪ ♫

▲ 음표 카드

거스름돈의 동전 수 최소화하기

물건을 사고 거스름돈을 받을 때 동전의 수를 최소화하기 위해서는 어떤 방법이 있을까?

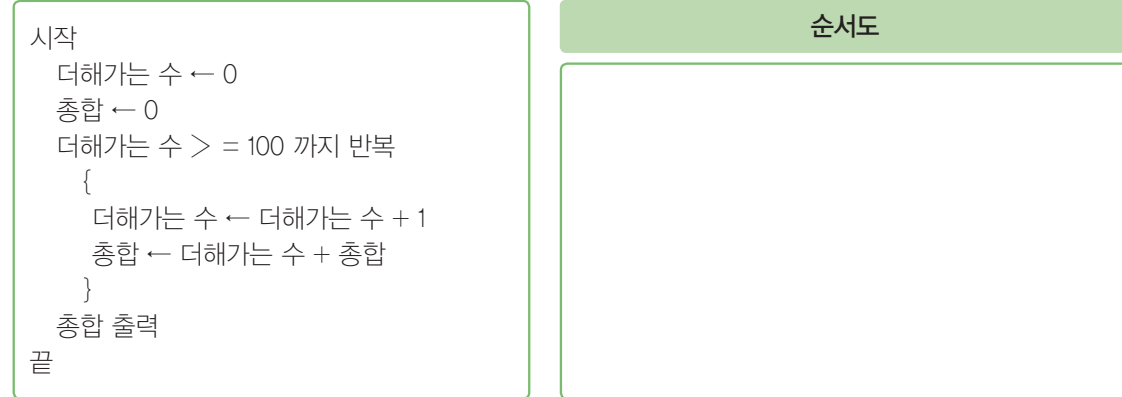


1 해결 아이디어: 만약 거스름돈이 640원이라면 500원짜리 동전 1개, 100원짜리 동전 1개, 10원짜리 동전 4개를 받으면 거스름돈 640원에 대한 최소 동전의 수이다.

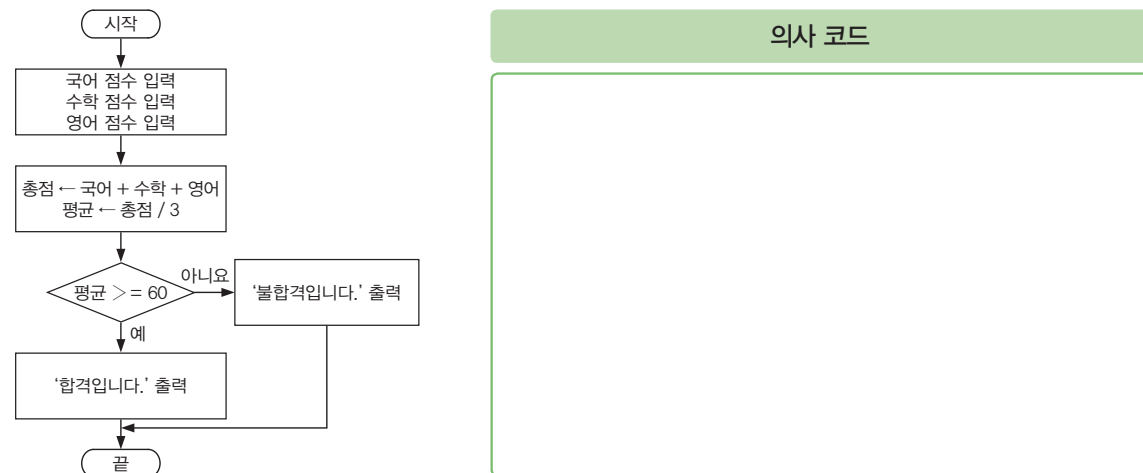
2 문제 해결 방법 및 절차: 거스름돈의 액수와 상관없이 최소의 거스름돈을 받기 위한 문제 해결 방법과 절차를 자연어, 의사 코드, 순서도 중 한 가지 방법을 선택하여 표현해 보자.



1 다음은 1부터 100까지 합을 구하는 의사코드이다. 이것을 순서도로 표현해 보자.



2 다음 그림은 세 과목의 점수를 입력받아 총점, 평균을 구한 뒤 합격 여부를 출력하는 순서도이다. 의사 코드로 표현해 보자.



3 우리 학교 도서관은 아래와 같은 도서 대출 규칙이 있다. 도서 대출 규칙에 따라 순서도를 표현할 때 (가), (나), (다), (라)에 들어갈 내용은 무엇인가?

도서 대출 규칙

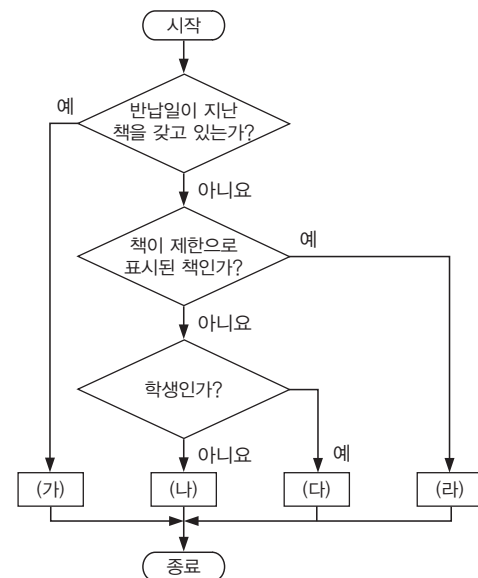
- ▶ '제한'으로 표시된 책은 누구나 대출 기간이 1일이다.
- ▶ '제한'이 표시되지 않은 책은 대출 기간이 선생님 14일, 학생 7일이다.
- ▶ 반납일이 지난 책을 가지고 있는 사람은 대출이 안 된다.

(가) _____

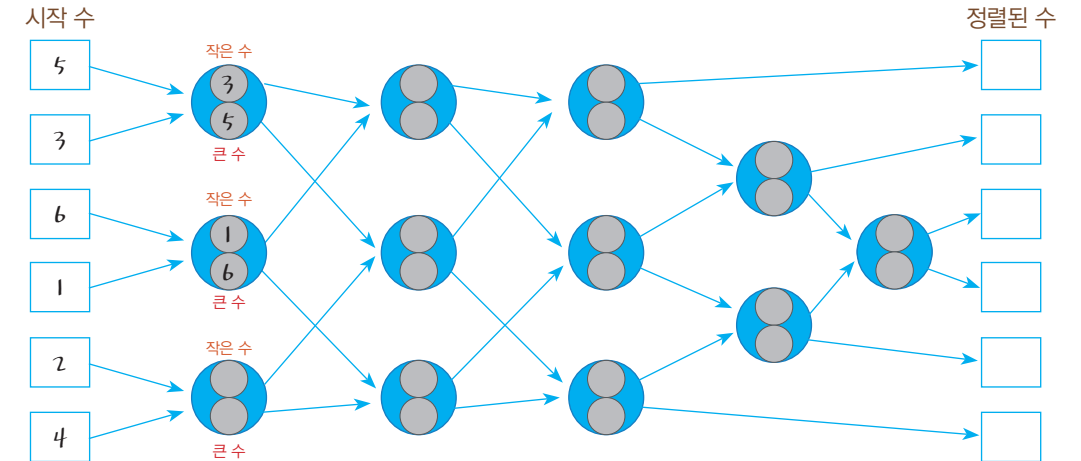
(나) _____

(다) _____

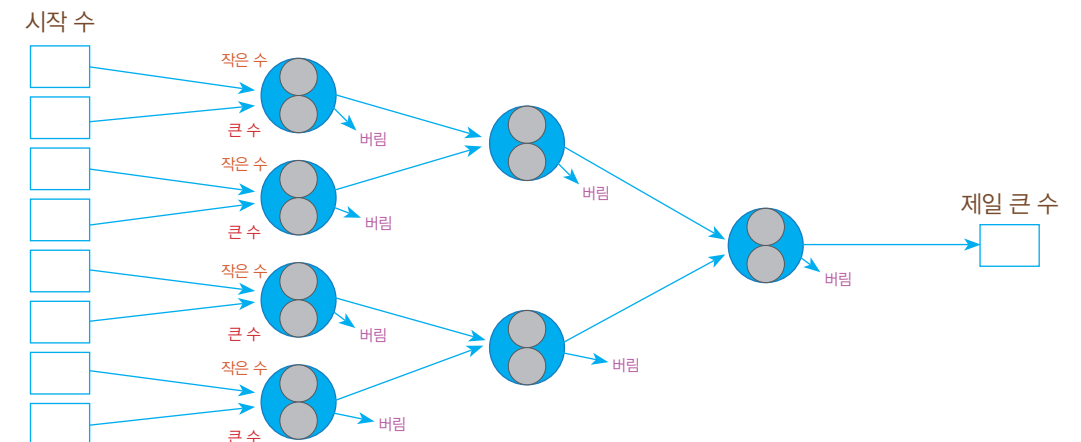
(라) _____



4 아래의 그림과 같이 숫자를 정렬하는 망(정렬망)을 이용하여 숫자를 정렬해 보자. 이웃한 두 수를 비교하여 더 큰 수는 위쪽 화살표를 따라가고 작은 수는 아래쪽 화살표를 따라가며, 원 안에는 작은 수를 위로, 큰 수를 아래 기재한다.



5 아래 그림은 제일 큰 수를 찾기 위한 정렬망이다. 4번과 같은 규칙으로 큰 수가 위쪽 화살표를, 작은 수가 아래쪽 화살표를 따라가지만 원 안에는 큰 수를 위로 기재하여 여러분이 시작 숫자를 넣어서 제일 큰 수를 찾아보자.



자 기 평 가

평가 내용	잘함	보통	노력 요함
1. 자연어, 순서도, 의사 코드를 활용하여 알고리즘을 표현할 수 있다.			
2. 순차, 선택, 반복 구조를 활용하여 알고리즘을 설계할 수 있다.			
3. 다양한 정렬 알고리즘과 다양한 분야에 활용되는 알고리즘을 이해할 수 있다.			

검색 엔진과 색인

검색 엔진은 수많은 자료 중에 찾고자 하는 단어를 쉽게 검색하기 위해 색인은 사용한다. 검색 엔진에서 검색을 빠르게 하는 아이디어는 색인이 페이지 번호뿐만 아니라 페이지 안의 위치도 저장해야 한다는 생각에서 출발한다. 아래 그림의 검색 단어 번호는 세 페이지짜리 자료 모음에 단어 위치를 추가적으로 작성한 예시이다. 여기서 위치는 페이지 내에 단어의 위치를 말한다. 예를 들면 2번째 단어의 위치는 2, 19번째 단어의 위치는 19로 표시한다.

그림의 색인 표는 페이지 번호와 단어 위치를 모두 저장한 곳에서 나온 색인 번호들이다. 이러한 방법으로 검색하려는 단어를 입력하면 검색 엔진은 색인 표에 입력된 수치를 이용하여 정확하고 빠르게 검색 작업을 처리하게 된다.

검색 단어 번호					색인 표	
1 페이지	the	cat	sat	on	a	3-5
	1	2	3	4	cat	1-2, 3-2
		the	mat		dog	2-2, 3-6
		5	6		mat	1-6, 2-6
2 페이지	the	dog	stood		on	1-4, 2-4
	1	2	3		sat	1-3, 3-7
	on	the	mat		stood	2-3, 3-3
	4	5	6		the	1-1, 1-5, 2-1, 2-5, 3-1
3 페이지		the	cat	stood	while	3-4
		1	2	3		
	while	a	dog	sat		
	4	5	6	7		

색인의 첫 줄에 'a'란 단어가 데이터 모음에서 정확히 한 번 등장하고 3페이지의 다섯 번째 단어란 뜻이다. 이러한 방법은 문장에 대해 질문한 것을 효율적으로 처리하기 위해 사용한다.

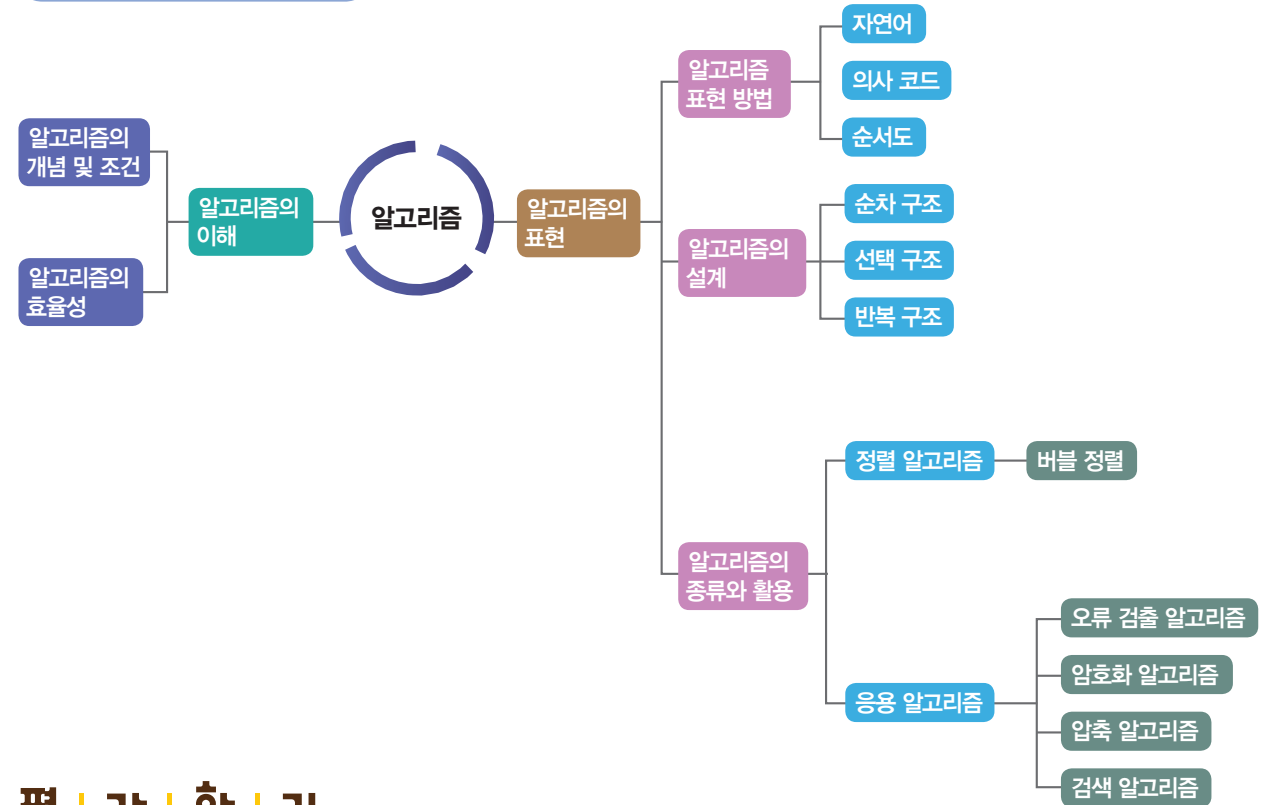
예를 들어 'cat sat'을 찾고자 할 때 먼저 개별 단어의 위치를 찾는 색인에서 추출한다. 그러면 'cat'에 대해서는 1-2와 3-2를, 'sat'에 대해서는 1-3과 3-7을 얻는다. 두 단어의 위치를 추출했지만 두 단어가 정확한 순서로 붙어 있지 않을 수 있다. 다행히도 위치 정보로부터 이를 확인할 수 있다. 1페이지를 살펴보면 'cat'이 위치 2에 있고, 'sat'이 위치 3에 있으므로 'sat'이 'cat' 바로 다음에 있다는 사실을 알 수 있다. 따라서 우리가 찾고자 하는 'cat sat'은 1페이지의 위치 2에서 시작함에 틀림없으며, 이렇게 간단한 방법이 검색 엔진을 작동하는 열쇠가 되었다.

출처: 『미래를 바꾼 9가지 알고리즘』 존 매코믹

토론하기

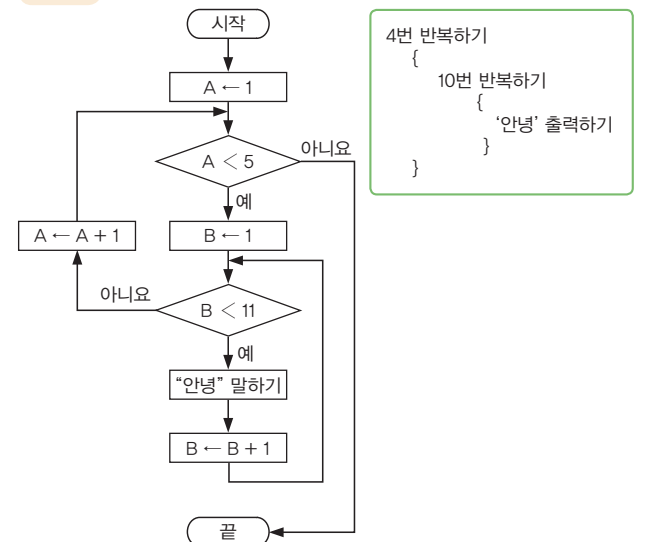
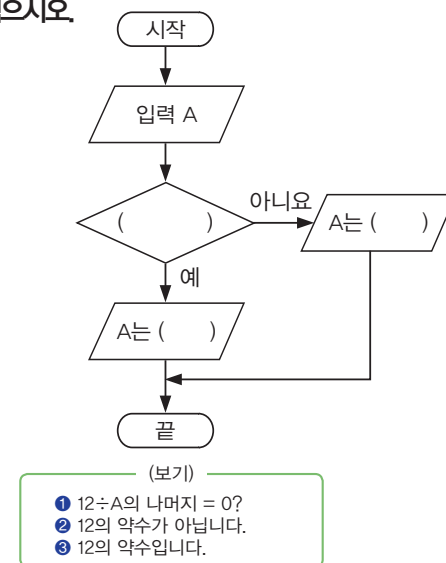
내비게이션과 같이 실생활에 활용된 다양한 알고리즘을 찾아보고 우리 생활에 미친 영향에 대해 논해 보자.

배운 내용 정리하기



평가하기

- 문제를 해결하기 위한 방법을 순서대로 나열한 절차를 무엇이라고 할까?
- 다음은 입력 A가 12의 약수인지 판별하는 알고리즘을 구현한 순서도이다. 빈칸에 알맞은 내용을 보기에서 찾아 번호를 적으시오.
- 다음 알고리즘을 실행하면 '안녕'을 몇 번 말할까?



학습 목표

- 프로그래밍 언어를 이해하고 자료의 입력과 출력을 할 수 있다.
- 변수를 이해하고, 변수와 연산자를 사용하여 프로그램을 작성할 수 있다.
- 순차, 선택, 반복 구조의 명령을 사용하여 구조적인 프로그램을 작성할 수 있다.
- 실생활의 문제를 해결하기 위한 프로젝트를 설계하고 프로그램으로 구현할 수 있다.

프로그래머(programmer)는 어떤 직업?

선희는 스마트폰 앱을 만드는 개발자가 되고 싶고, 재영은 온라인 가상 게임을 만들거나 영화에 들어가는 세밀한 3D 영상 프로그래머가 되길 원한다. 그래서 직업 정보 사이트를 통해 프로그래머라는 직업에 대해 조사하였다.

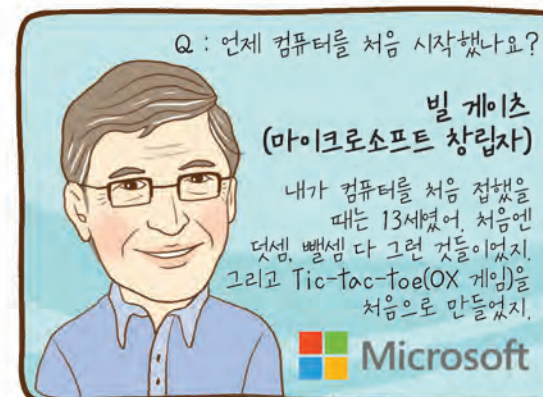
“컴퓨터 프로그래머는 컴퓨터가 인식할 수 있는 언어를 사용해서 필요한 명령 체계나 시스템을 만드는 일을 하는 사람을 말한다.”

프로그래머는 다양한 언어로 다양한 환경에서 프로그래밍을 하는 사람으로 세분화되어 있다.

프로그래머들은 모바일 앱이나 게임 프로그래머, 운영 체제나 네트워크 프로그램을 다루는 시스템 엔지니어 외에도 빅 데이터 분석가, IT 컨설턴트 등으로 직업이 다양화되는 경우도 있으며, 프로그래머들의 정책을 결정하는 최고 기술 경영자(CTO: Chief Technical Officer)가 되어 기술적으로 중요한 회사의 방향을 결정하는 책임자가 될 수도 있다.



유명한 IT 기업 수장들과의 프로그래밍에 대한 인터뷰 내용을 살펴보자.



생각해보기

위의 만화를 보고 프로그래머가 되려면 어떤 일부터 시작하면 좋을지 생각해 보자.



3-1

프로그래밍!

사람들은 대화를 하거나 글을 쓸 때 각 나라의 언어를 사용한다. 내가 컴퓨터와 대화하거나 소셜 네트워크 서비스(SNS) 애플리케이션, 문서 작성 프로그램 등을 만들려면 프로그래밍 언어를 사용한다. 프로그래밍 언어를 살펴 보면서 각 언어의 특징을 알아보자.

1 프로그래밍의 이해



우리가 일상생활에서 사용하는 인터넷 웹 브라우저, 스프레드시트, 스마트 앱을 프로그램이라 한다. 프로그램(program)이란 프로그래밍 언어를 통해 컴퓨터가 사용자의 명령을 알아들을 수 있도록 작업을 지시하고, 동작시키는 명령어의 모임이다. 프로그래밍 언어(programming language)는 컴퓨터와 소통할 수 있도록 문법을 갖춘 기호 체계이다. 프로그래밍 언어를 이용하면 문제의 흐름이나 동작 방법을 지시하고 전달할 수 있다.

이 프로그램을 작성하는 일을 프로그래밍(programming)이라고 하고, 이 프로그래밍을 통해 소프트웨어를 만드는 사람을 프로그래머(programmer)라고 한다.

2 프로그래밍 언어의 유형

1 프로그래밍 언어

프로그래밍 언어는 문자와 숫자, 기호로 이루어져 있다. 이때 프로그램을 표현하는 언어에 대해서 인간이 이해하기 쉽고 인간의 사고에 적합한 개념과 구조를 갖느냐에 따라 저급 언어와 고급 언어로 구분한다.

저급 언어	고급 언어
사람보다는 기계가 이해하기 쉽게 작성된 컴퓨터 중심의 프로그래밍 언어이다. 프로그램의 처리 속도는 빠르지만 사용자가 프로그램을 작성하는 것은 어렵다.	사용자가 이해하기 쉽도록 사람들이 사용하는 자연어와 비슷한 구조를 가지고 있다. 프로그램 작성은 쉽지만 처리 속도가 저급 언어보다 느리다.
기계어와 어셈블리 언어	C언어, 자바(Java), PHP 등

● **기계어**
컴퓨터가 직접 이해할 수 있는 0과 1로 구성된 컴퓨터 프로그래밍 언어이다.

● **어셈블리 언어**
기계어와 일대일로 대응시켜 기호화한 언어로, 기계어보다 쉽게 프로그래밍할 수 있다.

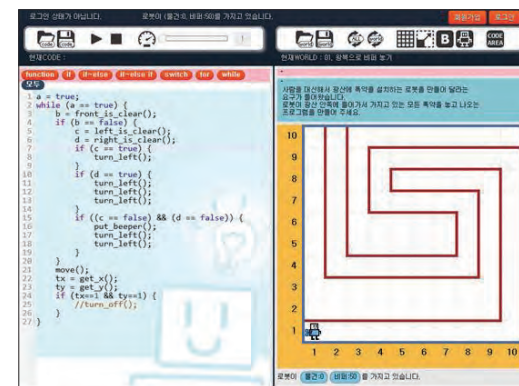
언어 번역 프로그램은 어셈블리 언어나 고급 언어와 같이 사람 중심의 언어로 작성된 프로그램을 컴퓨터가 직접 이해할 수 있는 기계어로 번역해 주는 프로그램이다. 어셈블러, 인터프리터, 컴파일러 등이 있다.



2 교육용 프로그래밍 언어

교육용 프로그래밍 언어(EPL: Educational Programming Language)는 사용자들이 쉽게 프로그래밍을 배울 수 있도록 손쉬운 인터페이스와 사용법을 갖추고 있다.

교육용 프로그래밍 언어에는 문자로 명령어를 직접 입력하는 로고, 러플, 플레이봇, 파이선과 같은 텍스트 기반의 프로그래밍 언어와 명령 블록을 이용하는 스크래치, 엔트리, 앨리스, 스냅 등과 같은 블록 기반 프로그래밍 언어로 구분된다. 교육용 프로그래밍 언어는 실제적인 문제를 직접 구현하여 해결하려는 것이 아니라 컴퓨팅 사고력, 문제 해결력을 신장시키기 위한 것이다.



▲ 텍스트 기반의 교육용 프로그래밍 언어(플레이봇)

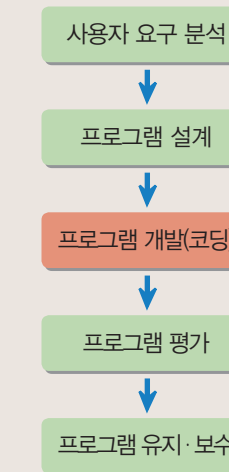


▲ 블록 기반의 교육용 프로그래밍 언어(스크래치)

● **어셈블러(Assembler)**
어셈블리 언어로 작성된 프로그램을 실행 가능한 기계어 형태로 바꾸어 주는 프로그램이다.

● **인터프리터(interpreter), 컴파일러(compiler)**
고급 언어로 작성된 프로그램을 즉시 실행 가능한 기계어로 바꿔 주는 프로그램이다.

● 프로그램 개발 과정



알고리즘이 소프트웨어로 작성되기까지의 과정

1 앞 단원에서 배운 알고리즘부터 소프트웨어가 작성되기까지의 과정이다. 보기의 용어 정리를 참고하여 빈칸에 들어갈 프로그래밍 과정 단계를 입력해 보자.

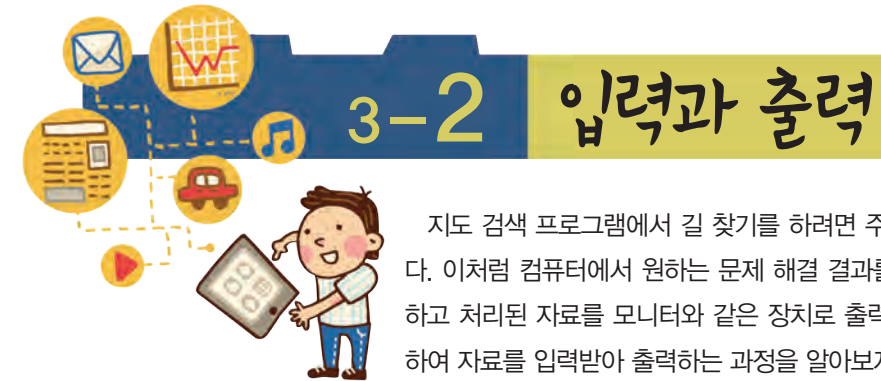


보기 코딩, 컴파일, 링크, 원시 프로그램, 목적 프로그램

- 코딩: 프로그램을 작성하기 위해 명령문을 입력하는 작업
- 컴파일: 원시 프로그램을 번역하여 목적 프로그램을 생성하는 작업
- 링크: 목적 프로그램을 이용하여 실행 프로그램을 만드는 작업
- 원시 프로그램: 저급·고급 언어 등 프로그래밍 언어를 이용하여 작성한 프로그램
- 목적 프로그램: 언어 번역 프로그램에 의해 기계어로 번역된 프로그램

2 인터넷을 이용하여 다양한 프로그래밍 언어를 검색하고 그 특징과 용도를 조사해 보자.

프로그래밍 언어	특징 및 용도



지도 검색 프로그램에서 길 찾기를 하려면 주소를 입력하고 검색 후 결과를 확인하게 된다. 이처럼 컴퓨터에서 원하는 문제 해결 결과를 얻기 위해서는 자료를 컴퓨터에 입력해야 하고 처리된 자료를 모니터와 같은 장치로 출력하도록 해야 한다. 프로그래밍 언어를 활용하여 자료를 입력받아 출력하는 과정을 알아보자.

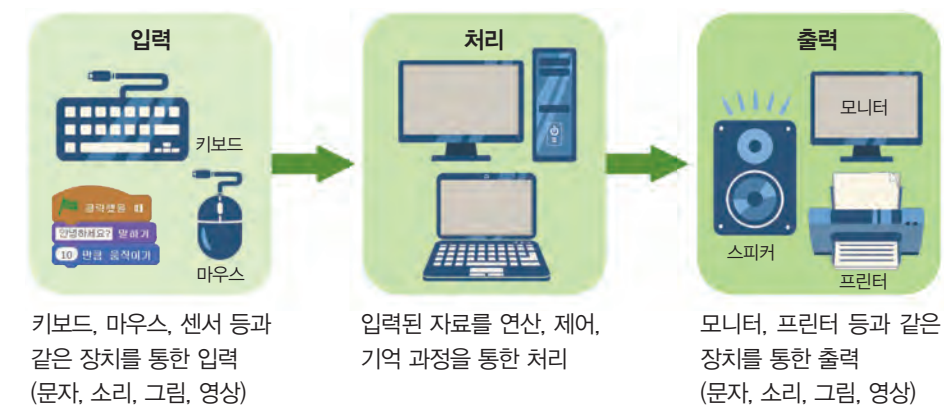
1 프로그래밍 언어의 입력과 출력

자동판매기나 교통 카드의 동작과 같이 일상생활 속에서도 다양한 입력과 출력이 존재한다. 컴퓨터에서도 문제 해결을 위해서는 처리할 자료를 입력받아 처리 과정을 거친 후 해결된 실행 결과를 모니터나 파일 등으로 출력하게 된다.

입력과 출력 자료 형태는 숫자, 문자, 문자열뿐만 아니라 소리, 그림, 동영상 등 다양한 형태로 존재한다. 그러므로 프로그램의 목적에 따라 적합한 자료를 적절히 이용할 수 있도록 해야 한다.

모니터 화면에 나타난 고양이가 ‘안녕하세요?’ 하는 인사를 출력하게 하려면 어떤 자료를 입력하여야 할까?

교육용 프로그래밍 언어를 활용하여 스프라이트를 동작시키기 위해서는 다양한 명령 블록을 입력하고 실행하여 처리 결과를 모니터와 같은 장치로 출력해야 한다.



▲ 프로그래밍 언어의 입·출력 과정



스크래치 프로그램

1 스크래치(Scratch)란?

MIT 미디어랩이 개발한 공개 소프트웨어로 블록 형태의 명령어를 조합하여 프로그래밍을 할 수 있는 대표적인 교육용 프로그래밍 언어이다. 블록을 조합할 때는 드래그 앤 드롭 방식을 사용한다.

스크래치 프로그래밍은 애니메이션, 게임, 음악 등으로 쉽게 표현할 수 있고 웹을 통해서 다른 사람들과 공유할 수 있다.



2 스크래치 실행

- 1 온라인과 오프라인에서 모두 사용할 수 있다. 온라인에서 사용할 수 있는 웹 버전은 웹 브라우저 주소창에 'https://scratch.mit.edu/'를 입력하여 접속하며, 오프라인 버전은 웹사이트 하단의 [지원]-[오프라인 에디터]에서 설치한다.
- 2 [만들기] 메뉴를 클릭하면 새로운 프로젝트 창이 실행된다.
- 3 [탐색하기] 메뉴로는 각 주제별로 전 세계에서 만들어져 공유된 프로젝트를 확인할 수 있다.

3 스크래치 화면

스크래치에서 꼭 알아 두어야 할 개념은 스프라이트, 무대, 스크립트이다. 이 세 가지 개념이 어우러져 스크래치 프로그램이 완성된다.



2 자료의 입력과 출력

1 키보드를 통한 입력과 출력

키보드를 이용하여 문자 자료를 입력받은 컴퓨터가 프로그래밍 스크립트에 따라 처리하고, 그 처리 결과를 말풍선으로 출력하는 프로그램을 작성해 보자.

핵심 명령 블록 문자의 입력과 출력

카테고리	명령 블록	기능
이벤트	클릭했을 때	스크래치 프로그램을 시작하는 명령어
형태	Hello! 말하기	입력한 내용을 말풍선으로 나타냄
동작	10 만큼 움직이기	설정된 값만큼 이동
관찰	What's your name? 묻고 기다리기	질문을 하고 입력이 있을 때까지 기다림
	대답	[묻고 기다리기] 명령이 실행되고 키보드로 문자 또는 숫자가 입력되면 저장

따라 하기 고양이 인사하면서 앞으로 움직이는 프로그램

- 1 고양이 스프라이트 선택하기
- 2 이벤트 카테고리에서 클릭했을 때 블록 스크립트 영역으로 가져오기
- 3 형태 카테고리의 Hello! 말하기 블록을 가져와 '안녕하세요?'라고 바꾸기
- 4 동작 카테고리의 10 만큼 움직이기 가져오기
- 5 클릭했을 때 클릭하여 실행하기

핵심 명령 블록 방향키 입력과 출력

카테고리	명령 블록	기능
이벤트	스페이스 키를 눌렀을 때	키보드의 특정 키를 눌러 스크립트가 시작
동작	90 도 방향 보기	방향을 나타내는 명령어

스크립트 탭의 10개 카테고리

스크립트	모양	소리
동작	이벤트	제어
형태	제어	관찰
소리	관찰	연산
펜	연산	추가블록
데이터	추가블록	

도전하기

What's your name? 묻고 기다리기

대답 블록을 이용하여 자기를 소개하는 프로그램을 작성해 보자.

실행 결과



도전하기

방향키와 A S D W 키를 활용하여 두 마리의 펭귄이 무대에서 움직이도록 프로그램을 작성해 보자.



따라 하기 고양이 키보드 오른쪽 화살표 키로 이동시키기

오른쪽 화살표 키를 눌렀을 때

90도 방향 보기

10만큼 움직이기

1 스페이스 키를 눌렀을 때 블록을 스크립트 창으로 드래그하여

오른쪽 화살표 선택.

2 90도 방향 보기 블록 가져오기

3 10만큼 움직이기 블록 가져오기

2 마우스를 통한 입력과 출력

마우스를 클릭하면 스프라이트가 실행되는 프로그램을 작성해 본다.

핵심 명령 블록 마우스를 이용한 입력과 출력

카테고리	명령 블록	기능
이벤트	이 스프라이트가 클릭될 때	마우스로 클릭할 때 아래 블록 실행
형태	다음 모양으로 바꾸기	스프라이트의 모양을 모양 탭의 목록 순으로 바꾸기
동작	무당벌레 위치로 이동하기	스프라이트가 선택한 스프라이트 또는 마우스 포인터의 위치로 이동
펜	펜 내리기	스크립트가 선을 그릴 수 있도록 함.

따라 하기 마우스가 클릭하면 모양 바꾸기

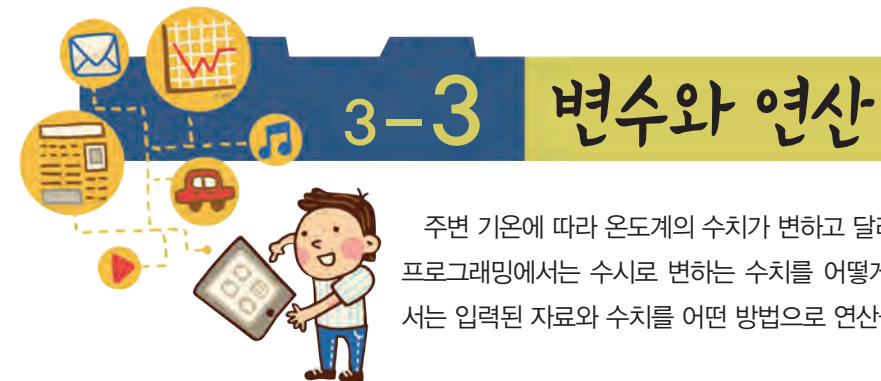
스프라이트	스크립트	실행 결과
	이 스프라이트가 클릭될 때 다음 모양으로 바꾸기	

따라 하기 마우스를 따라 움직이기 · 마우스로 그림 그리기

스프라이트	스크립트	실행 결과
	클릭했을 때 무한 반복하기 마우스 포인터 위치로 이동하기	
	클릭했을 때 펜 내리기 무한 반복하기 마우스 포인터 위치로 이동하기	

도전하기

스프라이트를 마우스로 클릭할 때 소리를 재생하면서 모양이 바뀌고 이동하는 프로그램을 작성해 보자.



주변 기온에 따라 온도계의 수치가 변하고 달리는 자동차의 속도계도 상황에 따라 변한다. 프로그래밍에서는 수시로 변하는 수치를 어떻게 처리하는지 알아보자. 그리고 프로그램에서는 입력된 자료와 수치를 어떤 방법으로 연산을 하는지도 살펴보자.

1 변수

1 변수의 이해

입력된 자료에 따라 변하는 값을 변수라 한다. 하나의 상자에 다양한 물건을 담을 수 있는 것처럼 프로그래밍에서도 변하는 값을 담는 공간을 변수라고 한다. 변수의 유형에는 정수나 실수와 같은 숫자가 저장되는 숫자형 변수, 문자를 저장하는 문자형 변수, 참과 거짓의 자료가 저장되는 논리형 변수가 있다.

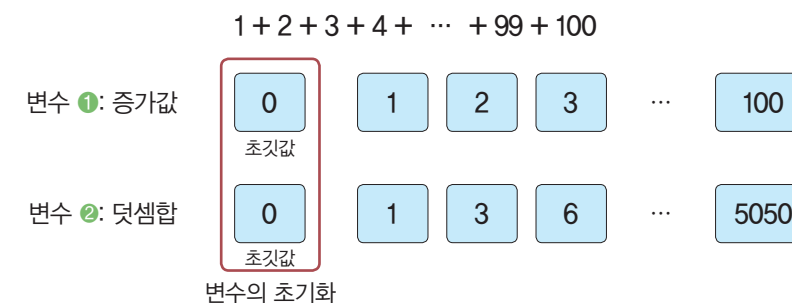


▲ 변수의 유형

2 변수 만들기

1부터 100까지 합을 구하는 프로그램을 구해 보자.

1 + 2 + 3 + ... + 100까지 1씩 증가되는 값과 덧셈 값을 보관하는 변수가 필요하다. 변수를 선언하고 처음으로 값을 저장하는 것을 '변수의 초기화'라고 한다.



▲ 1부터 100까지 합 구하기

● 상수
변수와는 달리 고정된 값을 나타낸다.

● 변수는 자유롭게 이름을 붙일 수 있으나, 가능하면 결과 값의 의미를 명확히 알 수 있는 변수명으로 정하고, 초깃값을 부여한다.

● 변수의 초기화

클릭했을 때
증가값을 (를) 0로 정하기

변수 만들기

1 데이터 카테고리의 변수 만들기

스크립트	모양	소리
동작	이벤트	제어
형태	제어	관찰
소리	관찰	연산
펜	연산	추가블록
데이터	추가블록	

2 변수 이름 만들기

새로운 변수
변수 이름 점수
모든 스프라이트에서 사용 ○ 이 스프라이트에서만 사용
확인 취소

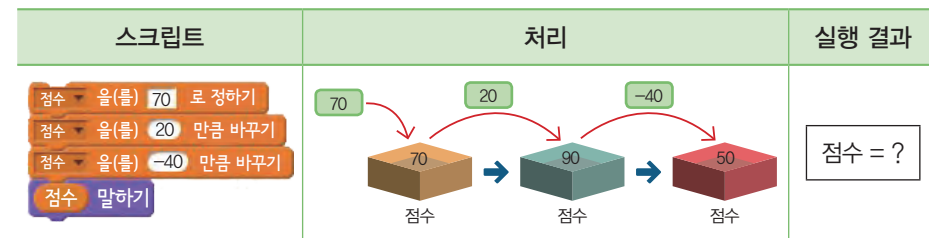
변수 명령 블록 비교

점수 = 을(를) 70 로 정하기 와
점수 = 을(를) 20 만큼 바꾸기 의
차이점이 무엇인지 알아
보자.

도전하기

'점수' 변수 값이 변화하는
과정을 실행 창에서 확인
할 수 있도록 블록을 추가
해 보자.

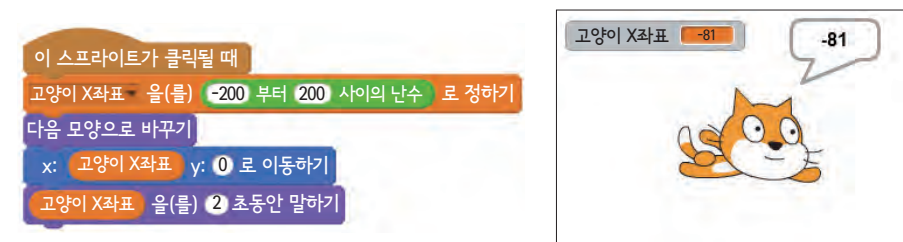
'점수' 변수를 만든 후 다음과 같이 세 개의 명령 블록이 실행되면 '점수' 변수의 값은 어떻게 변할까? 다음 '점수' 변수에 최종적으로 저장되는 값은 얼마일까?



핵심 명령 블록 움직이는 고양이

카테고리	명령 블록	기능
이벤트	이 스프라이트가 클릭될 때	스프라이트를 클릭할 때 아래 블록 실행
연산	1 부터 10 사이의 난수	입력한 두 수 사이에서 선택된 무작위 수의 값
동작	x: 0 y: 0 로 이동하기	스프라이트를 입력한 X좌표, Y좌표로 이동
형태	색깔 효과를 25 만큼 바꾸기	스프라이트에 선택한 효과를 입력한 값만큼 주고 색깔, 밝기, 투명도 등을 선택
데이터	고양이 변수 보이기	추가된 변수가 실행 화면에 나타나도록 함

따라 하기 스프라이트를 클릭할 때마다 모양을 바꾸면서 X좌표 위치를 변수로 입력받고 이동하기



도전하기

20초가 지나면 야옹 소리가
나는 타이머를 만들어 보자.

작품 공유하기

스크래치 홈페이지에 있는 프로젝트를 살펴보거나 특별한 타입의 프로젝트를 찾기 위해 [탐험하기] 메뉴를 클릭해 보자.



스프라이트를 클릭할 때마다 점수 1씩 증가시키기

사과, 오렌지, 바나나 스프라이트를 클릭할 때마다 점수를 1씩 증가시키고, 결과 확인 버튼을 클릭하면 결과 점수를 보여 주도록 하자.

스프라이트	스크립트	실행 결과
	클릭했을 때 이 스프라이트가 클릭될 때 사과 = 을(를) 0 로 정하기 사과 = 을(를) 1 만큼 바꾸기 사과 깨우는 소리 재생하기	사과 0 오렌지 0 바나나 1
	이 스프라이트가 클릭될 때 사과 변수 보이기	

2 연산자

1 연산자의 종류

연산자는 프로그래밍 과정에서 계산 처리를 위해 사용되는 중요한 요소이다. 산술, 논리, 비교 연산자를 사용한다. 산술 연산자는 사칙 연산을 처리할 때, 논리 연산자는 참인지 거짓인지의 논리적인 판단을 할 때, 비교 연산자는 크기를 비교할 때 사용한다.

핵심 명령 블록 연산 블록

구분	카테고리	명령 블록	기능
	산술 연산		덧셈, 뺄셈, 곱셈, 나눗셈
	비교 연산		~ 크다, ~ 같다, ~ 작다
연산	논리 연산	그리고	논리곱 양쪽 조건이 맞을 때의 논리 값
		또는	논리합 두 조건 중 하나 이상 맞을 때의 논리 값
		가(이) 아니다	부정 조건이 거짓일 때 참이 되는 논리 값

● 프로그래밍 언어에 따라 연산자를 표시하는 방법이나 사용법이 조금씩 다르다.

● 논리곱

A	그리고	B	→	C
참		참		참
참		거짓		거짓
거짓		참		거짓
거짓		거짓		거짓

● 논리합

A	또는	B	→	C
참		참		참
참		거짓		참
거짓		참		참
거짓		거짓		거짓

● 부정

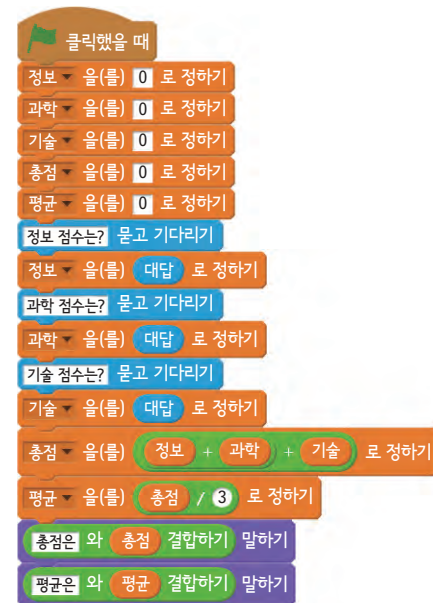
A	~A
참	거짓
거짓	참

2 연산자의 활용

계산 블록을 통해 숫자 값과 논리 값을 이용하고, 스프라이트의 움직임을 정하거나 조건을 줄 때 사용한다.

따라 하기

총점과 평균 구하기



- 1 '정보', '과학', '기술', '총점', '평균' 변수를 만들어 0을 초깃값으로 정한다.
- 2 What's your name? 묻고 기다리기 과 대답 을 이용하여 정보, 과학, 기술 점수를 입력받는다.
- 3 블록 과 을 이용하여 총점과 평균을 구하는 식으로 만들어 총점과 평균 변수에 입력한다.
- 4 Hello! 말하기 블록과 hello 와 world 결합하기 블록을 이용하여 총점과 평균을 출력하도록 한다.

큰 수, 작은 수

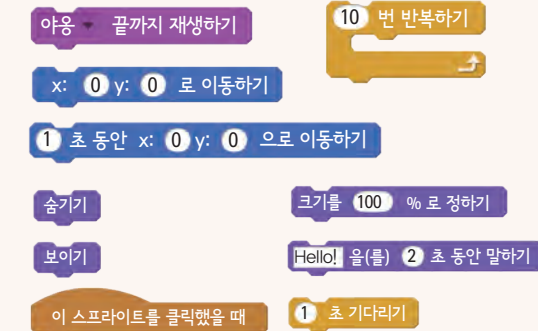
1에서 100 사이 임의의 숫자를 입력받은 뒤에 입력받은 수가 50보다 작으면 “작은 수입니다”, 50과 같거나 크면 “같거나 큰수입니다”를 출력해 보자.

명령 블록	실행 결과
<div>1~100 사이의 숫자 묻고 기다리기</div> <div>50보다 작은 수입니다 말하기</div> <div>만약 라면 대답</div> <div>아니면 50</div>	<div>1~100 사이의 숫자</div> <div></div> <div>50보다 작은 수입니다.</div> <div></div>
프로그램 심화시키기	50보다 작거나 큰 수를 비교하는 스크립트를 수정하여 짝수와 홀수를 판단하는 프로그램을 작성해 보자.

10개의 블록



10개의 명령 블록으로 어떤 프로그램을 만들 수 있을까? 다음 명령 블록을 이용하여 프로젝트를 만들어 보자. 각 블록을 복사하여 한 번 이상 사용해도 된다. 단, 각각의 블록을 적어도 한 번은 꼭 사용해야 한다.



1 각각의 블록을 적용하면서 자신의 아이디어를 표현해 보자.

- 다른 스프라이트, 모양, 배경으로 꾸며 보자.
- 10개의 블록으로 만들 수 있는 프로젝트가 얼마나 많은지 알아보자.
- 친구들과 아이디어를 브레인스토밍하여 각각의 창작물을 공유해 보자.

2 평가하기 다음 물음에 대한 여러분의 생각을 빈칸에 써 보자.

- 명령어 전체를 사용할 때보다 10개의 블록만 사용할 때 어떤 점이 쉬웠는가?

- 명령어 전체를 사용할 때보다 10개의 블록만 사용할 때 어떤 점이 어려웠는가?

- 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

3 심화하기 변수, 소리, 형태, 연산 블록 중 10개를 더 추가하여 나만의 작품을 만들어 보자.



3-4 제어 구조

우리는 일상생활에서 일을 할 때 순서대로 처리하거나 반복 또는 조건에 따라 선택하여 처리한다. 주어진 문제를 해결하기 위해 프로그램으로 구현할 때도 같은 내용을 여러 번 반복하거나 조건에 따라 일련의 실행 순서가 정해지는 절차 구조로 처리한다. 이 단원에서는 순차, 반복, 선택 구조를 이해하고, 효율적인 프로그램 작성을 위해 어떤 명령을 사용하는지 살펴보자.

1 순차 구조

종이접기를 할 때는 도안을 보고 순서대로 접는다. 이처럼 주어진 문제를 해결하기 위해 처음부터 끝까지 정해진 순서에 따라 차례대로 명령이 나열된 것을 순차 구조라 한다.

프로그래밍 언어로 정사각형을 어떻게 그릴 수 있을까? 그림을 그리기 위해 먼저 펜 카테고리의 명령 블록을 살펴보자.

핵심 명령 블록 그림 그리기 블록

카테고리	명령 블록	기능 설명
펜	펜 내리기	스크립트가 선을 그리도록 함
	펜 색깔을 (으)로 정하기	선의 색깔을 바꿈
	펜 굵기를 1 만큼 바꾸기	선의 굵기를 바꿈
	지우기	그린 선을 모두 지움

따라 하기 정삼각형 그리기

클릭했을 때

펜 내리기

100 만큼 움직이기

120 도 돌기

100 만큼 움직이기

120 도 돌기

100 만큼 움직이기

120 도 돌기

- 1 펜 카테고리 펜 내리기 블록을 드래그하여 스크립트 영역에 둔다.
- 2 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.
- 3 한 번 더 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.
- 4 또 한 번 더 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.

2 반복 구조

프로그램을 작성할 때 중복되는 부분을 효율적으로 처리하기 위해 반복 구조가 필요하다. 반복 구조는 조건이 만족하는 동안 정해진 구간을 반복 명령어를 이용하여 수행한다. 반복을 실행하기 위한 '반복 횟수'와 조건이 만족됐을 때 실행할 '실행 영역'으로 구분된다.

핵심 명령 블록 반복 명령 블록

구분	명령 블록	기능
제어	10 번 반복하기	블록 안의 스크립트를 정해진 횟수만큼 반복
	무한 반복하기	블록 안의 스크립트를 무한 반복
	까지 반복하기	조건이 참이 될 때까지 실행 영역 안의 스크립트를 반복 실행

따라 하기 삼각형 그리기

3 키를 눌렀을 때

펜 내리기

3 번 반복하기

100 만큼 움직이기

1 초 기다리기

360 / 3 도 돌기

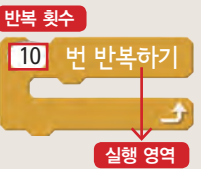
1 초 기다리기

펜 올리기

- 1 펜 카테고리 펜 내리기 블록을 드래그하여 스크립트 영역에 둔다.
- 2 제어 카테고리의 10 번 반복하기 블록을 드래그한 후 3 번 반복하기 를 입력한다.
- 3 100 만큼 움직이기 반시계 방향으로 120도 회전하도록 360 / 3 도 돌기 연산을 입력한다.
- 4 펜 올리기 블록을 드래그한다.
- 5 삼각형을 그리는 모습을 차근차근 보여 주기 위해 움직이기와 돌기 사이에 1 초 기다리기 를 넣는다

3 선택 구조

조건에 따른 참과 거짓 여부에 따라 다른 처리를 하기 위해 흐름을 변경해야 하는 것을 선택 구조라 한다. 조건 명령의 조건식에 따라 다음에 실행할 명령을 선택하여 실행한다.



각에 의한 꼭짓점을 표현하기 위한 '돌기' 값을 추상화하면 '360도/선분의 수' 또는 '180 - 내각의 크기'가 된다.

도전하기

정사각형 그리기와 삼각형 그리기 스크립트를 이용하여 다양한 도형을 그려 보자.

정다각형을 그리기 위한 돌기 값을 정하기 위해 여러 정다각형의 특징을 조사하고 패턴을 찾아보자.

정다각형의 핵심 요소
정삼각형, 정사각형, 정오각형, 정육각형 등 정다각형의 공통점은 무엇일까? 이러한 도형들을 그리기 위한 핵심 요소는 선분(움직이기)과 각에 의한 꼭짓점(돌기)이다.

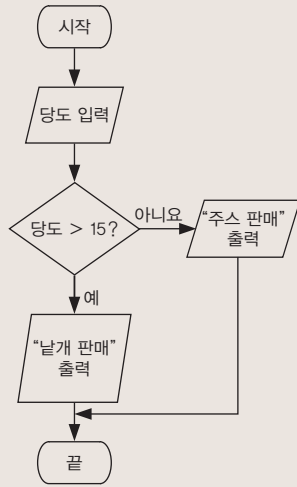
제어 구조

사람이 컴퓨터를 사용하는 이유는 복잡한 계산을 빨리 처리하거나 단순 반복 작업을 컴퓨터가 대신해 주기 때문이다. 이처럼 일련의 처리 과정을 조건에 따른 명령 실행이나 반복 동작을 통해 프로그램 실행의 흐름을 제어하는 것을 제어 구조라 한다.

프로그램 개선

정삼각형을 그리는 프로그램을 작성할 때 어떤 점이 불편했는가? 그 개선 방법을 토론했다.

- **핵심 요소:** 당도
- **알고리즘**



도전하기

1~4 사이에 난수 값을 받아 10이면 '동쪽', 20이면 '서쪽', 30이면 '남쪽', 40이면 '북쪽'으로 스프라이트를 이동시키는 프로그램을 만들어 보자.

핵심 명령 블록 선택 블록

카테고리	명령 블록	활용
제어	만약 ~라면	'만약 ~라면' 조건이 맞을 경우 블록 안의 스크립트를 실행
	만약 ~라면 아니면	'만약 ~라면' 조건이 맞을 경우 첫 번째 빈 칸 안의 스크립트를 실행, 조건이 맞지 않으면 두 번째 빈 칸 안의 스크립트를 실행

따라 하기 사과 분류하기

사과의 당도를 입력받아 15브릭스를 초과하면 '날개 판매', 15브릭스 이하면 '주스 판매'로 구분하도록 출력해 보자.

클릭했을 때

사과의 당도가 얼마인가요? 묻고 기다리기

만약 대답 > 15 라면

날개 판매 말하기

아니면

주스 판매 말하기

- 1 사과 당도를 입력받기 위해 묻고 기다린다.
- 2 제어에서 조건 명령 블록을 선택한다.
- 3 조건식에서 사용될 비교 연산자를 선택한다.
- 4 비교 연산자 > 에 대답 변수를 넣고 15와 비교한다.
- 5 조건 명령에 말하기 명령을 추가한다.

3-5

프로그래밍 응용

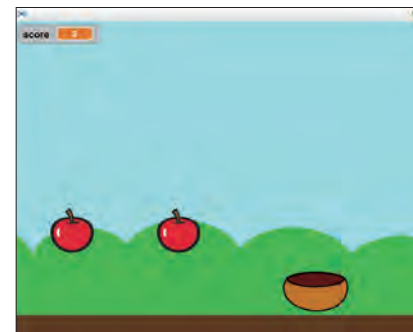
선풍기, 로봇 청소기, 자동판매기와 같은 생활 속 제품에는 프로그램이 포함되어 있다. 실생활의 다양한 문제 해결을 위해 프로그램을 구현해 보고, 효율적인 프로그램을 완성할 수 있도록 개발 과정을 공유·비교·분석하는 활동을 통해 협력적 프로젝트를 수행해 보자.

프로젝트 1

'사과 받기' 게임 재구성



'사과 받기' 게임을 실행해 보자.



'사과 받기' 게임을 해 보고 게임의 내용을 적어 보자.

'사과 받기' 게임은 바구니를 화살표 키(←, →)를 이용하여 좌우로 움직여 가며 떨어지는 사과를 받아 내는 게임이다. 사과를 받을 때 마다 1점씩 올라가며 사과가 바구니에 닿는 순간 소리도 함께 나도록 하였다.

놀이하기 '사과 받기' 게임을 해 보고 어떤 기능이 있는지 관찰하여 적어 보자.

사과	위에서 아래로 떨어진다. X좌표 값은 무작위로, Y좌표 값은 일정하게 하여 무작위의 위치에서 떨어진다.
바구니	
변수	
음향 효과	사과가 바구니에 닿을 때 '뽕' 소리 내기

스크래치 카드 놀이

4명이 한 모둠을 이루어 스크래치 카드를 이용하여 함께 활동해 보자.

- 1 Step** 선택한 '스크래치 카드'의 스크립트를 모둠원 모두가 만들면 카드 획득
- 2 Step** 획득한 여러 장의 '스크래치 카드'를 이용하여 새로운 작품 만들기
- 3 Step** 다른 모둠과 카드를 교환하여 작품을 추가하거나 수정하기



수정하기

명령어 수정

- 1 사과가 떨어지는 속도를 변경해 보자.
- 2 바구니의 방향 조절을 키보드 대신 마우스를 따라다니게 해 보자.
- 3 매개 변수에서 쌓이는 점수를 바꾸어 보자.
- 4 음향 효과를 바꾸어 보자.
- 5 ()

기능을 바꾸어 볼까?



명령어 복제

- 1 떨어지는 사과의 수를 늘린다.
- 2 ()

명령어 추가

- 1 타이머 변수를 넣어 시간 제한을 주자.
- 2 해당 점수가 되면 다음 단계의 장면으로 이동해 보자.
- 3 ()

스프라이트를 추가해 볼까?



새로운 스프라이트 ()의 기능

- 1
- 2
- 3



구현하기

위의 수정 계획을 바탕으로 사과 받기 게임을 완성해 보자.

되돌아보기

- 재구성한 게임을 실행시키고 오류가 있는 부분을 수정해 보자.
- 똑같은 프로젝트를 어떻게 다양하게 변화시켰는지 서로 발표하며 공유해 보자.

- 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

프로젝트 2

친구와 이야기 속으로



작은 별에 사는 어린 왕자는 나무를 뽑고 화산을 청소하며 하루하루를 보내고 있었다. 어느 날 피어난 장미를 사랑하게 된 어린 왕자는 장미가 원하는 것은 무엇이든 들어주었다. 하지만 변덕스러운 장미 때문에 어린 왕자는 별을 떠나기로 결심하였다. 여러 별을 여행하던 어린 왕자는 많은 사람을 만나며 다양한 경험을 한다. 지구에 도착한 어린 왕자는 불시착한 조종사를 만나며 이야기를 나누게 된다.

준비하기

짝과 함께 어린 왕자의 이야기 중에서 구현하고 싶은 장면을 선택하여 간단히 그려 보자.

스토리 :

장면 1	장면 2
장면 3	장면 4

수정하기

1 사용된 스프라이트의 주요 기능을 설계해 보자.

스프라이트	주요 기능(알고리즘)

2 각 장면에 사용되는 여러 가지 기능과 요소를 적어 보자.

장면	1	2	3	4
캐릭터				
애니메이션 멀티미디어				
스프라이트 간의 상호작용				

3 어린 왕자 스토리를 친구들과 공유하고 내용을 추가하거나 수정해 보자.

협력하여 개발하기 친구들과 함께 스토리를 구현할 때 다음 조건을 지키도록 하자.

짹 프로그래밍 조건



- 1 한 명당 한 번에 5개의 블록만 사용할 수 있다.
- 2 한 번의 기회에 3분을 초과할 수 없다.
- 3 다른 친구가 구현한 블록을 수정하거나 삭제할 수 있다.

피드백하기

우리 팀이 구현한 어린 왕자 스토리를 발표하고 친구들의 피드백을 받아 보자.

잘된 점	보완해야 할 점

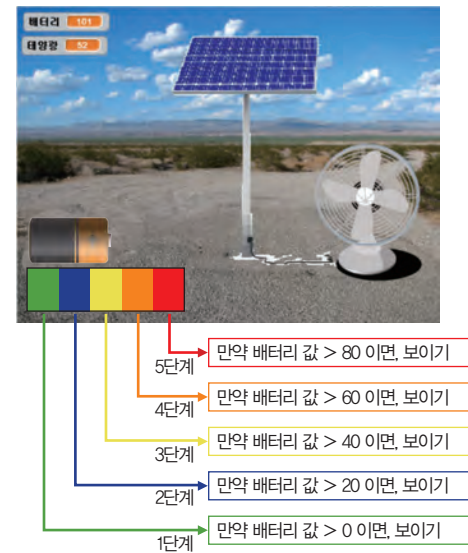


프로젝트 3

스마트 선풍기



배터리의 충전 표시



탐구하기

1 문제 상황의 목표 상태를 적어 보자.

현재 상태	시동이 꺼지면 전기를 사용할 수 없어 선풍기를 사용할 수 없다.
목표 상태	

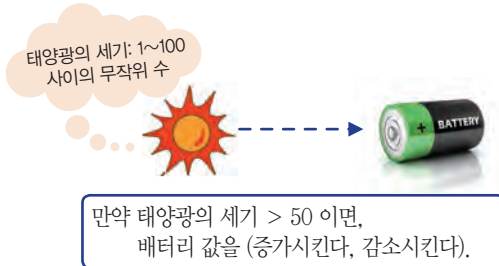


2 태양열을 이용한 스마트 선풍기를 작동시키기 위한 기능들을 분해하여 보았다. 빈칸에 알맞은 단 어들을 보기에서 골라 적어 보자.

보기 전기, 배터리, 세기, 선풍기	태양광이 일정한 <input type="text"/> 가 되면 <input type="text"/> 에 <input type="text"/> 가 충전된다.
	<input type="text"/> 에 충전된 <input type="text"/> 로 <input type="text"/> 를 돌린다.
	<input type="text"/> 가 작동되면 충전된 <input type="text"/> 가 사용된다.

설계하기

3 각 핵심 요소의 작동 조건들을 괄호 안 단어 중 하나를 선택하여 완성시켜 보자.

4 스마트 선풍기를 구현하기 위한 알고리즘을 완성해 보자.

핵심 요소	작동 조건	알고리즘
1 배터리 충전	 <p>만약 태양광의 세기 > 50 이면, 배터리 값을 (증가시킨다, 감소시킨다).</p>	<p>태양광의 세기: 1~100 사이 난수 값 태양광의 세기 <input type="text"/> 50 ? (예) 배터리 값 ← 배터리 값 + 2</p>
2 배터리의 상태	 <p>만약 배터리 값 < 5 이라면, “(방전, 충전 중)”이라고 표시한다. 아니면, “(방전, 충전 중)”이라고 표시한다.</p>	<p>배터리 값 <input type="text"/> 5 ? (예) “<input type="text"/> 중입니다.” 메시지 보이기 (아니요) “<input type="text"/> 입니다.” 메시지 보이기</p>
3 선풍기의 회전	 <p>만약 배터리 값 > 30 이라면, 선풍기를 돌린다. 배터리 값을 (감소시킨다, 증가시킨다).</p>	<p>배터리 값 <input type="text"/> 30 ? (예) 선풍기 돌기 속도 ← 선풍기 돌기 속도 + 30 배터리 값 ← 배터리 값 - 1</p>

구현하기

5 3가지 핵심 요소에 대한 알고리즘을 이용하여 프로그램을 개발해 보자.

사용 스프라이트

태양광 패널, 선풍기, 배터리

핵심 요소	명령 블록
1 배터리 충전	<p>무한 반복하기</p> <p>만약 <input type="text"/> 라면</p> <p>태양광 을(를) <input type="text"/> 로 정하기</p> <p>태양광 > 50</p> <p>배터리 을(를) 1 만큼 바꾸기</p> <p>1 부터 100 사이의 난수</p>
2 배터리의 상태	<p>무한 반복하기</p> <p>만약 <input type="text"/> 라면</p> <p>아니면</p> <p>배터리 > 5</p> <p><input type="text"/> 말하기</p>
3 선풍기의 회전	<p>무한 반복하기</p> <p>만약 <input type="text"/> 라면</p> <p>아니면</p> <p>배터리 > 30</p> <p>선풍기속도 을(를) 1 만큼 바꾸기</p> <p>선풍기속도 도 돌기</p> <p>배터리 을(를) -1 만큼 바꾸기</p> <p>선풍기속도 을(를) -1 만큼 바꾸기</p>

되돌아보기

- 만든 프로그램을 다른 친구들과 공유하여 개선 방법을 논의해 보자.
- 프로그램이 구동은 되나 자신의 생각대로 작동되지 않는 경우 디버깅 과정을 통해 오류를 수정해 보자.

• 선풍기가 제대로 작동하였는가? 그렇지 않다면 이유는 무엇인가?

• 선풍기 날개의 회전 방향을 반대로 작동시키려면 어떻게 해야 할까?

• 충전 상태를 그림으로 나타내는 스크립트를 작성해 보자. 그리고 태양광 스마트 선풍기의 기능을 조금 더 보완하기 위하여 여러 가지 기능을 추가시켜 수정하여 보자.

• 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.



프로젝트 4

자동 문자 응답 시뮬레이터



수업 시간이나 공부하는 시간에 올리는 문자 때문에 여간 신경 쓰이는 게 아닙니다. 상대방에게 응답을 해야 할 것 같지만 응답을 하려다가 집중하던 공부에 지장을 줄 수 있기 때문입니다. 하지만 상대방이 급하게 연락한 경우라면 내 상황에 대한 문자 응답을 자동으로 보내려 한다.

요구 분석하기

1 문제 상황과 목표 상태를 적어 보자.

현재 상태	
목표 상태	

2 아래와 같이 내게 수신된 문자들을 수집해 보자. 그리고 어떤 문자에 답장을 보내야 하는지 그 번호들의 패턴을 찾아 보자.

친구, 가족들의 휴대전화 번호 010-***-****문자	연락처에 저장되어 있지 않지만 아는 사람 휴대전화 번호 010-***-****	지역 전화번호로 시작되는 학교 등의 연락문자	지역 전화 번호로 시작하는 홍보 문자
명진아 학교 끝났니? 연락 줘~	선영아, ** 초등학교 동문회 있다~ 00월 00일 00시 **초등학교 앞 분식점	안녕하세요? 기말 시험이 25일이라서 이번 주말 학원 특강 있습니다.	*** 도서관 00일 00일 저녁 7시 반 <산화가 발견한 예술 인문학> 관심 부탁 드립니다
인터넷 전화 070-***-****로 시작하는 스팸 번호 문자	15**-****로 시작하는 광고성 문자	114, 112 등의 공공 번호 문자	
△△△ 클리닉 명절 이벤트 선착순 00명 20% 할인	[**은행 상품 안내] 주거례 적금 특별금리우대 이벤트 진행중! (00월 00일까지)	119에서 긴급 구조를 위해 귀하의 위치를 확인하였습니다	

3 응답 문자를 보내야 하는 번호와 그렇지 않은 번호를 구분하는 기준은 무엇일까?

설계하기

4 자동 문자 응답 프로그램을 구현하기 위해 다음 사항을 생각해 보자.

핵심 요소	<input type="text"/> 문자 번호, <input type="text"/> 에 저장된 번호	
패턴 찾기	응답 문자를 보내야 하는 번호의 패턴: 번호 앞 3자리가 <input type="text"/> 으로 시작함	
추상화	리스트가 필요한 항목	<input type="text"/> , <input type="text"/>
	변수가 필요한 항목	<input type="text"/>
	① 수신 문자의 수신 번호 = <input type="text"/> : 누구인지 되묻는 문자 ② 수신 문자의 수신 번호 = <input type="text"/> : 지인에게 부재중 문자	

5 자동 문자 전송 앱의 작동 순서를 세부적으로 나누어 알고리즘을 적어 보자.

- ① 문자 수신 → 수신 번호의 () = 1 and () = 0 ?
- ② (yes) → ()
- ③ (no) → 스팸 문자로 인식
- ④ 수신 번호 = () ?
- ⑤ (yes) → ()
- ⑥ (no) → 무응답 → ①행 반복 시작

구현하기

6 실행 화면과 주요 스크립트를 참고하여 자동 문자 응답 프로그램의 시뮬레이터를 제작해 보자.

1 번째 응답문자 목록 항목 와 전술 결합하기 울(를) 3 초동안 말하기

만약 연락처 목록 리스트에 대담 포함되었는가? 라면

아니면 문자 수신번호는? 묻고 기다리기

2 번째 글자 (대담) = 1 그리고 3 번째 글자 (대담) = 0 가(이) 아니다

만약 라면 스팸문자입니다. 울(를) 2 초동안 말하기

아니면 2 번째 응답문자 목록 항목 와 전술 결합하기 울(를) 3 초동안 말하기

모두 번째 항목을 연락처 목록 에서 삭제하기

01011112222 항목을 연락처 목록 에 추가하기

01033334444 항목을 연락처 목록 에 추가하기

공유하기

자신이 만든 프로그램을 다른 친구들과 공유하여 개선 방법을 논의해 보자.

잘된 점	보완해야 할 점

되돌아보기

- 프로그램이 구동은 되나 자신의 생각대로 작동되지 않는 경우 디버깅 과정을 통해 오류를 수정해 보자.
- 리스트와 변수의 차이는 무엇일까? 전화번호의 값을 리스트 대신 변수를 사용하여 구현이 가능할까?
- 실행해서 오류가 생기는 경우는 언제인가? 어떻게 해결해야 하는가?
- 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

컴퓨터 과학자처럼 **생각하기**

컴퓨터 과학자들은 프로그램을 개발할 때 아래의 사고 과정을 중요하게 생각한다.
여러분도 앞의 4가지 프로젝트를 개발하면서 컴퓨터 과학자처럼 생각했는지 확인해 보자.

실험하기와 반복하기

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트를 작은 부분에서 시작하여 점차 확대하여 만들었는가?			
프로젝트의 일부분이 제대로 실행되는지 확인하였는가?			
프로젝트 실행 결과에 따라 다양하게 수정하여 보았는가?			
프로젝트를 다른 방법으로 시도하거나 새로운 방법으로 시도하였는가?			

시험하기와 수정하기

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트를 실행하였을 때 결과를 다양하게 시험하였는가?			
원하는 실행 결과와 다른 점을 발견하였는가?			
문제의 원인을 찾기 위해 스크립트를 분석하였는가?			
오류가 발생한 결과를 확인하고 수정하였는가?			

재사용과 재구성

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
다른 사람들이 만든 프로젝트를 실행하여 아이디어를 찾았는가?			
다른 프로젝트에서 좋은 부분을 선택하여 내 프로젝트에 적용하였는가?			
프로젝트를 개선하기 위해 기존의 프로젝트를 수정하였는가?			
프로젝트에 참고한 작품이나 아이디어를 얻은 작품의 출처를 밝혔는가?			

추상화와 모듈화

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트에 필요한 스프라이트를 어디에 사용할지 결정하였는가?			
프로젝트에 필요한 핵심 스크립트를 구현하였는가?			
다른 사람들이 이해하기 쉽게 스크립트를 개발하였는가?			



평가하기

1 빈칸에 들어갈 용어를 쓰시오.

소프트웨어는 컴퓨터 프로그램이라고도 한다. 따라서 소프트웨어는 인간의 언어가 아닌 컴퓨터가 알아들을 수 있는 컴퓨터 언어로 만들어져야 한다. 컴퓨터 언어로 소프트웨어를 만드는 사람을 (), 소프트웨어를 만드는 작업을 ()이라 한다. 그래서 이 프로그래밍에 사용되는 컴퓨터 언어는 ()라고 한다.

2 빈칸에 들어갈 용어를 쓰시오.

프로그램의 명령 실행 구조는 순차, 반복, 선택 구조이다. () 구조는 하나의 방향으로 진행되는 것을 말하고, () 구조는 주어진 조건에 따라 실행 결과가 달라진다. () 구조는 주어진 조건에 따라 특정 명령어를 반복해서 실행한다.

3 스크래치 프로그램에서 사용되는 연산 블록을 유형에 맞게 연결하시오.

산술 연산 •



비교 연산 •



논리 연산 •



1 다음 문제 상황을 읽고 다음 문제를 해결해 보자.

• 스마트 스프링클러를 이용한 땅의 수분 측정하기

A 학생 가족은 주말마다 농장에 가서 감자와 고구마 등을 심어서 가꾸고 있다. 주말밖에 갈 시간이 없어서 주중에 물을 주기 어렵다. 주중에도 땅이 마르거나 날이 더우면 자동으로 물을 주는 스프링클러가 필요하다.

① 위 문제 상황의 목표 상태는 무엇인가?

② 위 문제를 해결하기 위하여 다음과 같이 땅의 수분 정도에 따른 센서 저항을 측정해 보았다.

- 6개의 종이컵에 마른 흙을 담고 물을 조금씩 다르게 넣는다.
- 저항 양쪽을 흙에 꽂아 흙의 수분에 대한 저항 값을 측정한다.

손으로 만졌을 경우 흙의 수분 정도	0	1	2	3	4	5
센서의 저항 값	98	87	65	35	28	12

스프링클러가 흙의 수분 정도가 0과 1일 때 작동하기 위한 조건식을 부등호를 사용하여 완성해 보시오,

수분 센서의 저항 값 () 80이면 스프링클러 돌기

2 다음과 같이 이진 탐색을 이용하여 1~1,000의 숫자 중에 875를 찾는 과정에서 탐색 기준 값에 해당하는 숫자를 적어 보시오.

탐색 횟수	0회	1회	2회	3회
탐색기준	1000			

3 아래 동전을 이용하여 2,980원을 지불하기 위해서 필요한 가장 적은 동전의 개수는 몇 개인가?

예를 들어 아래의 동전들로 1,050원을 지불할 때를 생각해 보자. 1,050원을 지불하는 가장 적은 동전 수는 500원짜리 2개와 50원짜리 1개를 사용하여 총 3개의 동전을 이용하는 것이다.



빈칸에 알맞은 숫자를 적어보시오.

	500원	100원	50원	10원	계
동전 필요 개수					

단원 평가

단원 되돌아보기

- 4 은아, 찬민, 지민이가 학교를 마치고 어디로 놀러갈지를 이야기하고 있다. 주사위 3개를 순서대로 던져 아래와 같은 규칙으로 놀러 가려고 한다.

- 1 만약 첫 번째 주사위 눈이 두 번째 주사위 눈보다 작은가?
- 2 **그러면:** 강으로 놀러 간다.
- 3 **아니면:** 만약 세 번째 주사위 눈이 첫 번째 보다 작은가?
- 4 **그러면:** 산으로 놀러 간다.
- 5 **아니면:** 놀이터로 놀러 간다.



놀이터로 놀러가게 되는 주사위의 조건을 부등호로 나타내 보고 예시를 하나 적으시오.

첫 번째 주사위 눈 () 두 번째 주사위 눈
세 번째 주사위 눈 () 첫 번째 주사위 눈
주사위 눈의 예: () - () - ()

- 5 다음 연산을 수식으로 표현하고, 그 결과 값을 계산해 보자. 연산의 결과 값이 다르게 나타난다면 왜 그런지 이유를 살펴 보자.

연산 스크립트	수식	결과값
$20 + 5 * 3 - 8 / 2$		
$20 + 5 * 3 - 8 / 2$		
$20 + 5 * 3 - 8 / 2$		

결과 값이 다르게 나타나는 이유 :

- 6 다음 보기처럼 연필이 중심에서 네 개의 같은 정사각형을 그릴 수 있도록 스크립트의 빈칸을 완성하시오.

보기	스크립트
	<pre> 스페이스 키를 눌렀을 때 펜 내리기 번 반복하기 번 반복하기 100 만큼 움직이기 도 돌기 도 돌기 </pre>

- 7 미로에서 고양이가 녹색 사각형에 부딪치지 않고 연두색 사각형까지 가야 한다. 고양이가 녹색 벽을 통과하지 못하도록 프로그램 명령을 추가해 보자.

	<pre> 클릭했을 때 크기를 50 %로 정하기 x: -220 y: 150 로 이동하기 색에 닿았는가? 까지 기다리기 You win 을(를) 2 초동안 말하기 아래쪽 화살표 키를 눌렀을 때 180 도 방향 보기 10 만큼 움직이기 위쪽 화살표 키를 눌렀을 때 0 도 방향 보기 10 만큼 움직이기 오른쪽 화살표 키를 눌렀을 때 90 도 방향 보기 10 만큼 움직이기 왼쪽 화살표 키를 눌렀을 때 -90 도 방향 보기 10 만큼 움직이기 </pre>
--	--

추가해야 할 명령