교사용

초등학교 정보교육

학습 보완 자료



교육부 Ministry of Education

한국과학창의재단 Korea Foundation for Science and Creativity

서울교육대학교 Seoul National University of Education

초등학교 정보교육 학습 보완 자료

교육부 Ministry of Education

한국과학창의재단 Korea Foundation for Science and Creativity

서울교육대학교 Seoul National University of Education

초등학교 정보교육

학습 보완 자료

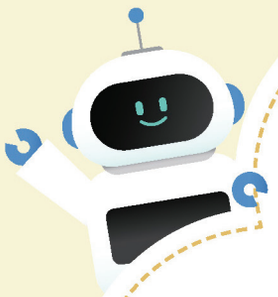
교사용

교육부 Ministry of Education

한국과학창의재단 Korea Foundation for Science and Creativity

서울교육대학교 Seoul National University of Education

비매품/무료
63370
9 791163 698807
ISBN 979-11-6369-880-7



초등학교 정보교육

학습 보완 자료

교사용

CONTENTS

Ⅰ. 학습 보완 자료 안내

1. 학습 보완 자료의 필요성 ····· 2

2. 교육과정 개정에 따른 초등학교 정보교육 성취기준 비교 ····· 3

3. 2022 개정 교육과정 초등 정보교육과정 성취기준 및 성취기준 해설 ····· 4

4. 정보교육과정의 학습 공백 ····· 7

5. 보완 대상 성취기준 및 차시 운영 계획 ····· 8

6. 학습 보완 자료 활용 방향 ····· 9

7. 학교 교육과정 편성 및 운영 방향 (6학년 2학기) ····· 10



Ⅱ. 학습 보완 자료

1. 생활 속 컴퓨터와 문제 해결 ····· 14
2. 컴퓨터에게 명령하기 ····· 51
3. 함께 만드는 생활 속 프로그램 ····· 84

Ⅲ. 부록

1. 정보 교과 교수·학습 모형 ····· 144
2. 수업 배경지식 ····· 154
3. 정보 용어 사전 ····· 165
4. 저작권 및 AI 윤리 자료 ····· 167
5. 참고 문헌 및 사이트 ····· 170
6. 생성형 AI 활용 내역 ····· 172

학습 보완 자료 안내



1. 학습 보완 자료의 필요성 ····· 2

2. 교육과정 개정에 따른 초등학교 정보교육 성취기준 비교 ····· 3

3. 2022 개정 교육과정 초등 정보교육과정 성취기준 및 성취기준 해설 ····· 4

4. 정보교육과정의 학습 공백 ····· 7

5. 보완 대상 성취기준 및 차시 운영 계획 ····· 8

6. 학습 보완 자료 활용 방향 ····· 9

7. 학교 교육과정 편성 및 운영 방향 (6학년 2학기) ····· 10

1 학습 보완 자료의 필요성

2022 개정 교육과정은 2024년에 1, 2학년 적용을 시작으로 2025년에 3, 4학년이 적용되고, 2026년에 5, 6학년에 적용되어 초등학교 전체 학년에 최신의 교육과정이 도입된다. 새로운 교육과정의 적용은 학년이 아니라 학년군을 기준으로 적용됨에 따라 교과별로 교육과정이 개정되어 학년별 내용요소 및 성취기준의 이동이 발생하거나 새로운 내용이 추가되는 경우 학습 공백이 발생하는 특정 학년 및 시기가 존재할 수 있다.

초등학교 정보교육은 실과 내 정보교육내용으로 존재하며 프로그래밍, 데이터, 인공지능, 디지털 소양 등의 내용을 다루고 있기 때문에 교육과정이 개정되면서 내용체계와 학년별 편성이 크게 변화하고 있다. 2015 개정 교육과정에서는 소프트웨어교육이 강조되어 프로그래밍과 컴퓨팅사고력이 강조되었고 5~6학년에 17시간으로 편제되었으나 검인정교과서에서 모두 6학년에 해당 내용을 다루고 있어 6학년에서 프로그래밍 중심의 활동으로 학교교육과정이 대부분 편성 및 운영되었다. 2022 개정 교육과정에서는 인공지능시대를 대비하기 위하여 프로그래밍에 데이터와 인공지능의 내용을 더하여 34시간으로 편제되었고, 학교자율시간 등을 활용하여 운영하는 17시간 이상의 시수를 확보하기 위한 내용체계가 반영되었다. 이에 따라, 2015 개정 교육과정의 내용이 2022 개정 교육과정에서는 5학년 교과서에 반영되었고, 새롭게 반영된 데이터와 인공지능에 대한 내용은 6학년 교과서에 반영되었다. 따라서, 2026년에 5학년인 초등학교 학생들은 초등학교 정보교육이 처음 시작되는 시기에 프로그래밍으로부터 시작하여 내년도 6학년으로 진급하면 데이터와 인공지능에 대한 내용을 위계를 고려하여 배울 수 있는 반면, 2026년에 6학년인 초등학교 학생들은 5학년 과정에서 프로그래밍에 대한 내용을 배웠다는 가정하에 데이터와 인공지능의 내용을 배워야 하는 학습 공백이 발생하였다. 이에 따라 교육부와 한국과학창의재단에서는 2026학년도 6학년 학생에게 발생하는 5학년에 편성되어 있는 프로그래밍에 대한 학습 공백을 지원하기 위해 본 학습 보완 자료를 개발하게 되었다. 본 학습 보완 자료는 2026년 현재 6학년 학생들을 지원하기 위해 개발되었으며 새로운 교육과정의 변화에 따라 학교 현장에서 참고하여 활용할 수 있는 교육과정 편성과 운영 및 학습 공백이 발생되는 내용 요소와 성취 기준에 대한 내용을 활동과 교수·학습 과정안을 통해 지원하고자 한다.

2026년도에 2022 개정 교육과정 적용 시 발생할 수 있는 6학년에서의 문제점은 다음과 같다.

(1) 2015 개정 교육과정의 6학년 학습 내용의 일부가 2022 개정 교육과정에 따라 5학년으로 이동하여 학습하지 못한 경우
(2) 2022 개정 교육과정에 따라 5학년에 새롭게 내용이 신설되어 학습하지 못한 경우

이에 따라, 2026학년도에 6학년은 초등학교 정보교육에 대한 지도서에 제시된 학기 지도 계획을 수정해 적용할 필요가 있다. 이 자료는 학기 지도 계획을 수립하거나 일부 단원의 내용을 재구성해 수업하는 데 활용할 수 있도록 한다.

2 교육과정 개정에 따른 초등학교 정보교육 성취기준 비교

2015 개정 교육과정에서 초등학교 정보교육은 실과에 편성된 5개 영역 중 4번째 영역인 '기술시스템' 영역의 '소통'이라는 핵심개념에 3가지 내용 요소(소프트웨어의 이해, 절차적 문제해결, 프로그래밍 요소와 구조)로 아래 표와 같이 5개의 성취기준이 반영되어 있었다.

구분	성취기준
[6실04-07]	소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.
[6실04-08]	절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.
[6실04-09]	프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.
[6실04-10]	자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.
[6실04-11]	문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.

2022 개정 교육과정에서 초등학교 정보교육은 2015 개정 교육과정과 마찬가지로 실과 내에 존재하나 별도의 독립된 영역으로 제시되어 중학교 정보교과와 연계되는 체계를 갖추기 시작했다는 점에서 의미를 갖는다. 다만, 2022 개정 교육과정에서 초등학교 정보교육은 34차시 이상을 편성 및 운영해야 하나, 추가하여 17시간 이상을 편성 및 운영하기 위해 학교자율시간 등을 활용하거나 교과(군) 및 창의적 체험활동의 기준시수를 20%범위에서 증배하여 편성 및 운영해야 한다. 따라서, 2022 개정 교육과정에서 초등학교 정보교육은 내용 체계가 실과와 부록으로 나뉘어 반영되었으며, 상호 연계하여 위계가 고려된 성취기준임을 고려할 때 아래 표를 함께 연계하여 학교교육과정 편성이 요구된다.

구분	성취기준
[6실05-01]	컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.
[6실05-02]	컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.
[6실05-03]	실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.
[6실05-04]	디지털 데이터와 아날로그 데이터의 특징을 이해하고, 인공지능에 활용할 수 있는 데이터의 유형이나 형태를 탐색한다.
[6실05-05]	인공지능이 만들어지는 과정을 체험하고, 인공지능이 사회에 미치는 영향을 탐색한다.
2022 개정 교육과정 실과내 편제 정보교육 성취기준	

구분	성취기준
[06자율-1]	생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다.
[06자율-2]	놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.
[06자율-3]	프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.
[06자율-4]	공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.
[06자율-5]	생활 속에서 접하는 다양한 데이터가 서로 다른 의미를 갖고 있음을 파악하는 활동을 경험한다.
[06자율-6]	인공지능이 데이터의 공통되는 부분을 찾아 분류하는 과정을 체험한다.
[06자율-7]	사례를 중심으로 인공지능을 올바르게 사용하는 방법을 토론하고 실천하는 활동을 수행한다.
2022 개정 교육과정 부록 편제 정보교육 성취기준	

3 2022 개정 교육과정 초등 정보교육과정 성취기준 및 성취기준 해설[1)]

내용 요소	구분	성취기준	성취기준 해설	비고
컴퓨터의 개념	[06자율-1]	생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다.	컴퓨터에 사용된 소프트웨어 이외에도 휴대폰, 가전제품, 사물인터넷 제품까지 여러 상황에서 사용된 소프트웨어를 탐색해 보고 우리 생활에 미치는 영향을 이해한다.	2015 개정 교육과정 [6실04-07]참조
	[6실05-01]	컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.	이 성취기준은 컴퓨터의 개념을 이해하고 다양한 문제 상황에서 컴퓨터가 활용되는 사례 탐색을 통해 그 가치와 활용 방법을 알 수 있어야 한다. 일상생활에서 컴퓨터를 활용해 해결할 수 있는 문제를 탐색하고, 문제 상황에 제시된 요소들을 분석해 문제의 현재 상태와 목표 상태를 정의한다. 언플러그드 활동 등을 통한 문제 해결 과정을 알고리즘으로 작성하기 위해 자연어, 의사코드(pseudocode), 순서도 등으로 문제 해결 과정을 표현하도록 한다.	
문제 찾기와 문제 해결 절차	[06자율-2]	놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.	[6실05-01]과 연계하여, 생활 속 문제를 해결하는 알고리즘의 다양한 사례를 이야기로 제시하거나 신체를 움직이고, 다양한 도구 및 학습지를 활용하여 알고리즘을 이해하며 필요성을 설명할 수 있어야 한다.	
	[6실05-02]	컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	블록 기반의 교육용 프로그래밍 도구를 활용하여 기초적인 프로그래밍 과정을 체험하고 자신만의 간단한 프로그램을 만들어 본다.	2015 개정 교육과정 [6실04-09]참조
컴퓨터에게 명령하는 방법	[06자율-3]	프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	[6실05-02]와 연계하여, 우리 주변의 가까운 곳에서 프로그래밍으로 해결된 문제를 찾아 간단한 프로그램을 작성하며 자료의 입·출력, 변수를 이용하는 경험을 하고 순차, 반복, 선택 과정을 이해할 수 있어야 한다.	
	[6실05-03]	실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.	수치 값을 입력하여 덧셈이나 뺄셈의 결과를 출력하거나, 복수의 문자열을 입력 하여 두 문자열을 서로 연결한 결과를 출력하는 프로그램을 만들어 봄으로써, 소프트웨어의 입력, 처리, 출력 과정을 이해한다.	2015 개정 교육과정 [6실04-10]참조
	[06자율-4]	공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.	프로그램 개발 과정을 공유·비교·분석하는 활동을 통해 프로그램을 지속적으로 수정·보완하여 효율적인 프로그램을 완성할 수 있도록 지도한다.	2015 개정 교육과정 중학교 정보교과교육과정 교수·학습 방법 및 유의사항 참조

1) 신승기(2025). 2022 개정 교육과정에 따른 초등학교 정보교육의 학습공백 분석 연구. 창의정보문화연구. 11권 4호. pp.469~483의 내용을 발췌하여 재구성함

내용 요소	구분	성취기준	성취기준 해설	비고
데이터의 종류와 표현	[6실05-04]	디지털 데이터와 아날로그 데이터의 특징을 이해하고, 인공지능에 활용할 수 있는 데이터의 유형이나 형태를 탐색한다.	이 성취기준은 놀이·체험을 중심으로 생활 주변에서 접할 수 있는 디지털 및 아날로그 데이터를 찾아 각각의 특징을 비교하는 과정을 통해 데이터의 의미를 이해하고, 인공지능에 활용할 수 있는 숫자, 글자, 소리, 이미지 데이터의 유형이나 형태를 탐색할 수 있도록 한다.	
	[06자율-5]	생활 속에서 접하는 다양한 데이터가 서로 다른 의미를 갖고 있음을 파악하는 활동을 경험한다.	[6실05-04]와 연계하여, 생활 속에는 많은 데이터가 있으며, 그 데이터들은 각각 사용되는 목적이 다르고 목적에 따라 다양한 의미를 갖고 있다는 것을 활동을 통해 이해할 수 있어야 한다.	
생활 속 인공 지능	[06자율-6]	인공지능이 데이터의 공통되는 부분을 찾아 분류하는 과정을 체험한다.	[6실05-04]와 연계하여, 인공지능이 우리에게 추천하는 다양한 내용, 즉, 인터넷동영상 플랫폼에서 추천, 서점에서 내가 좋아하는 책 종류 추천 등을 할 때, 인공지능은 어떤 기준을 갖고 내가 선호하는 것을 알아서 추천하는 지 등의 과정을 체험해서 인공지능의 분류 과정을 내면화 할 수 있어야 한다.	
	[6실05-05]	인공지능이 만들어지는 과정을 체험하고, 인공지능이 사회에 미치는 영향을 탐색한다.	이 성취기준은 기계학습이 적용된 간단한 인공지능 도구의 체험을 통해 기계학습의 기본 원리를 이해하고 인공지능으로 인한 사회의 발전과 직업의 변화를 이해하여 인공지능이 사회에 미치는 영향을 탐색할 수 있어야 한다.	
	[06자율-7]	사례를 중심으로 인공지능을 올바르게 사용하는 방법을 토론하고 실천하는 활동을 수행한다.	인공지능 사회의 구성원으로서 인공지능 윤리의 중요성을 인식하고 사회적 책임감을 갖고 공정성을 추구할 수 있는 방안을 제시한다.	2015 개정 교육과정 [12인기04-04]참조

※ Note. 음영표기는 성취기준 해설이 교육과정에 제시되지 않은 부분임.

2022 개정 교육과정은 2015 개정 교육과정과 달리 모든 성취기준에 대한 성취기준 해설이 제시되어 있지 않기 때문에 2015 개정 교육과정에 함께 제시된 성취기준의 내용은 2015 개정 교육과정의 성취기준 해석을 통해 세부 내용을 확인해볼 필요가 있다. 2022 개정 교육과정의 초등학교 정보교육 중에서 실과에 편제된 성취기준에 대한 해설을 살펴보면 다음과 같다.

[6실05-01]에 제시된 내용을 통해 문제를 탐색하고, 문제 상황에 제시된 요소들을 분석하여 "현재 상태"와 "목표 상태"를 정의하도록 한다. 또한, 알고리즘을 표현하기 위해, 자연어, 의사코드, 순서도 등으로 문제해결과정을 구성하도록 한다.

[6실05-04]는 데이터과학에 대한 내용이며 디지털 데이터와 아날로그 데이터를 찾아 놀이와 체험을 통해 의미를 이해하도록 한다. 특히, 숫자, 글자, 소리, 이미지 데이터를 찾아보고 데이터의 유형과 형태를 탐색한다.

[6실05-05]에서는 2022 개정 교육과정에 처음으로 반영된 내용으로 기계학습이 적용된 간단한 인공지능 도구의 체험을 통해 기계학습의 기본 원리를 이해하며 사회적 영향을 고려하도록 한다.

2015 개정 교육과정에서 "교수·학습방법 및 유의사항"으로 표기된 부분이 2022 개정 교육과정에서는 "성취기준 적용시 고려사항"으로 나타나 있으며, 특정 프로그래밍 언어의 기능을 익히는 것을 지양하고 알고리즘을 기반으로 실제로 프로그래밍을 수행하며 기본 원리를 익히고, 프로그래밍의 과정을 통해

컴퓨팅사고력의 향상 정도를 측정하도록 한다. 또한, 인공지능에 대해서 인공지능의 기계 학습 과정을 이해하도록 지도하며 인공지능 기술을 단지 체험해보는 것을 넘어서 인공지능 소양을 함양하도록 제시되어 있다.

부록에 제시된 초등학교 정보교육의 성취기준에 대해서도 성취기준 해설이 제시되어 있으며, 실과의 정보교육 내용과 연계하여 구성되어 있다.

[06자율-2]는 [6실05-01]과 연계하여 알고리즘의 다양한 사례를 이야기로 제시하거나 신체를 움직이는 등을 통해 다양한 도구를 활용하여 알고리즘을 이해하고 필요성을 설명하도록 하였다.

[06자율-3]은 [6실05-02]와 연계하여 생활 속의 문제를 찾아 간단한 프로그램을 작성하고, 자료의 입력과 출력 및 변수를 이용하고 순차, 반복, 선택의 과정을 이해하도록 제시되어 있다. 이 부분은 2015개정 교육과정과 연계되는 부분이나, 2022 개정 교육과정에서 "변수"의 개념이 추가되었다는 특징이 있다. 또한, 2015 개정 교육과정에서는 "순차, 선택, 반복"으로 제시된 구조가 2022 개정 교육과정에서는 "순차, 반복, 선택"으로 순서가 변경되었다는 점이 변화된 모습이다.

[06자율-5]는 [6실05-04]와 연계하여 생활 속의 데이터를 찾고 목적에 따른 의미를 찾는 활동으로 구성되어 있다. [06자율-6]은 [6실05-04]와 연계되며 인공지능의 분류 과정을 추천 알고리즘을 통해 이해하도록 제시되어 있으며, 두 가지 성취기준은 각각 데이터와 인공지능에 해당하는 내용으로 2022 개정 교육과정에서 새롭게 나타난 부분이라고 할 수 있다.

초등학교 정보교육의 성취기준이 실과내 편제된 내용과 부록에 편제된 내용에서 성취기준 해설이 표현되지 않은 내용은 2015 개정 교육과정을 통해 세부 내용을 살펴볼 수 있으며, 순서를 고려한 초등학교내 성취기준의 위계를 재구성하는데 살펴볼 수 있다.

2022 개정 교육과정의 [6실05-02]와 [6실05-03]의 내용은 2015 개정 교육과정의 [6실04-10]의 내용을 참고하여 살펴볼 수 있으며, "수치 값을 입력하여 덧셈이나 뺄셈의 결과를 출력하거나, 복수의 문자열을 입력 하여 두 문자열을 서로 연결한 결과를 출력하는 프로그램을 만들어 봄으로써, 소프트웨어의 입력, 처리, 출력 과정을 이해한다."의 내용을 토대로 입력-처리-출력으로 이어지는 과정을 경험하는 과정으로 연계할 수 있다.

2022 개정 교육과정의 [06자율-1]의 내용은 2015 개정 교육과정의 [6실04-07]과 연계하여 "컴퓨터에 사용된 소프트웨어 이외에도 휴대폰, 가전제품, 사물인터넷 제품까지 여러 상황에서 사용된 소프트웨어를 탐색해 보고 우리 생활에 미치는 영향을 이해한다."의 내용을 토대로 생활 속 컴퓨터의 범위에 대해서 참고할 수 있다.

2022 개정 교육과정의 [06자율-4]는 프로그래밍 결과에 대한 공유와 탐색을 강조하는 것으로서 2015 개정 교육과정의 중학교 정보과 교육과정에 제시된 교수·학습 방법 및 유의사항에 제시된 "프로그램 개발 과정을 공유·비교·분석하는 활동을 통해 프로그램을 지속적으로 수정·보완하여 효율적인 프로그램을 완성할 수 있도록 지도한다."의 내용을 연계하여 살펴볼 수 있다.

2022 개정 교육과정의 [06자율-7] 내용은 인공지능에 대한 내용으로서 인공지능의 가치를 파악하고 올바르게 사용하며 실천적 활용에 대한 토론을 제시하고 있는 부분으로, 2022 개정 교육과정에서 정보과 교육과정의 영역 중 하나로 새롭게 도입된 부분이라고 할 수 있으나, 2015 개정 교육과정의 수시 개정을 통해 2020년에 정보과의 인공지능 기초 과목에서 제시된 [12인기04-04]에서 설명하는 인공지능 윤리의 중요성과 공정성 추구의 방안을 제시하는 활동과 연계된다고 할 수 있다.

4 정보교육과정의 학습 공백[2)]

2026년에 초등학교 6학년인 학생들은 2022 개정 교육과정에 따른 정보교육과정이 적용된다. 그런데, 2022 개정 교육과정의 정보교육에 대한 교과서를 살펴보면 5학년에서 프로그래밍의 내용을 다루고 있으므로 6학년이 되었을 때 해당 내용은 배우지 못한 채로 인공지능과 데이터의 내용을 배우게 되며, 중학교와 고등학교에 진학하였을 때에도 초등학교 5학년에 편성되어 있는 프로그래밍 내용에 대한 학습 공백으로 인해 프로그래밍의 구조에 대한 이해와 처리 절차 등의 기본적인 요소에 대한 이해가 부족할 수 있다. 또한, 2026년 현재 중학교 1학년 학생들은 2022 개정 교육과정에서 새롭게 도입된 인공지능과 데이터에 대한 내용을 배우지 못했으나 배웠다고 가정하고 중학교의 교육과정이 적용된다. 특히 변수의 개념은 2022 개정 교육과정에서 초등학교로 내려오게 되었는데, 해당 학생들은 이미 알고 있다고 가정하고 중학교 정보교육과정이 적용된다는 특징이 있다.

초등학교 뿐만 아니라 중학교와 고등학교도 교육과정이 전환되는 과정에서 누락되는 교육과정의 시기를 경험하게 된다. 아래 표는 2026년 현재 편성된 학년에 따라 2022 개정 교육과정의 초등학교 정보교육과정의 이수 여부를 내용요소 및 성취기준에 따라 정리한 것이다. 2026년 기준으로 초등학교 5학년 이하는 초등학교 정보교육의 모든 내용요소를 이수할 수 있지만 초등학교 6학년은 컴퓨터의 개념, 문제찾기와 문제해결절차, 컴퓨터에게 명령하는 방법(프로그래밍)의 내용을 이수하지 못하는 학습공백이 나타난다. 반면 중학교 1학년 이상의 경우에는 초등학교에서 데이터와 인공지능에 대한 내용을 배우지 못하고 상급학교로 진학하는 모습도 살펴볼 수 있다.

초등학교 정보교육		2026년 편성에 따른 학습여부		
내용요소	성취기준	초5 이하	초6	중1 이상
컴퓨터의 개념	[06자율-1] [6실05-01]	O	X	O
문제찾기와 문제해결절차	[06자율-2] [6실05-02]	O	X	O
컴퓨터에게 명령하는 방법	[06자율-3] [6실05-03] [06자율-4]	O	X	O
데이터의 종류와 표현	[6실05-04] [06자율-5]	O	O	X
생활 속 인공지능	[06자율-6] [6실05-05] [06자율-7]	O	O	X

2) 신승기(2025). 2022 개정 교육과정에 따른 초등학교 정보교육의 학습공백 분석 연구. 창의정보문화연구. 11권 4호. pp.469~483의 내용을 발췌하여 재구성함

5 보완 대상 성취기준 및 차시 운영 계획

5-1 2022 개정 교육과정 초등학교 정보 교육과정 중 5학년의 성취 기준 도출

내용 요소 (지식·이해 기준)	실과 내 '디지털 사회와 인공지능 영역' 성취기준	학교자율시간 등에 활용하는 성취기준
컴퓨터의 개념		[06자율-01] 생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다 → ①
문제 찾기와 문제 해결 절차	[6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다. → ②	
		[06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다. → ③
컴퓨터에게 명령하는 방법	[6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다. → ④	
		[06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다. → ⑤
	[6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다 → ⑥	
		[06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다. → ⑦
데이터의 종류와 표현	6학년 교과서에 편성됨	
생활 속 인공지능	6학년 교과서에 편성됨	

5-2 단원 구성 및 성취기준 기반 차시 구성(안)

단원명	성취기준	차시 수
1. 생활 속 컴퓨터와 문제 해결	① [06자율-1] 생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다.	1차시
	② [6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.	2차시
	③ [06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.	2차시
2. 컴퓨터에게 명령하기	④ [6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	4차시
3. 함께 만드는 생활 속 프로그램	⑤ [06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	4차시
	⑥ [6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.	2차시
	⑦ [06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.	2차시

6 학습 보완 자료 활용 방향

본 자료는 2022 개정 교육과정의 단계적 적용 과정에서 발생한 6학년 정보교육 학습 결손을 한시적으로 보완하기 위한 자료이다. 정규 교과서를 대체하는 자료가 아니라, 학교의 여건에 따라 필요한 부분을 선택하여 활용하는 보완용 자료로서의 성격을 가진다. 따라서 교사는 자료에 제시된 차시와 활동을 그대로 적용할 수도 있고, 학생 수준과 수업 시수에 맞추어 일부를 선택·조정하여 운영할 수도 있다.

본 자료는 교사가 별도의 추가 준비 없이 수업을 진행할 수 있도록, 차시별 교수·학습 과정안과 학생용 활동지를 중심으로 구성되었다. 각 구성 요소는 수업의 흐름에 따라 유기적으로 연계되어 있으며, 교사는 다음과 같이 각 요소를 활용할 수 있다.

구성 요소	활용 방안
교수·학습 과정안	차시별 수업의 주요 내용이 담겨 있는 교수·학습 과정안이며, 도입–전개–정리의 흐름에 따라 수업을 운영하는 기본 자료로 활용한다. 본시 학습 내용과 함께 제시된 자료 및 유의점을 참고하여 수업을 설계한다.
수업 활동지	수업 전개의 활동 자료로 활용한다. 핵심 개념을 학습하는 내용과 함께 학생들의 확산적 사고를 촉진하는 활동으로 구성되어 있다.
수업 평가지	학습 목표 도달 여부를 확인하는 자료로 활용한다. 제시된 예시 답안과 잘함·보통·노력 요함의 채점 기준을 참고하여 학생의 수준을 파악하고, 후속 차시 수업 설계에 이를 활용한다.
맞춤형 피드백	수업 중 나타날 수 있는 학생의 반응 유형에 따라 교사가 제공할 수 있는 피드백 예시이다. 예시를 참고하여 학생의 반응에 따라 피드백을 제공한다.
수업 운영 팁	수업 운영 중 발생할 수 있는 어려움에 대한 대응 방안이다. 컴퓨터실 부족, 개념 이해 부족 등 수업 운영 중 문제가 발생할 경우 이를 참고한다.
디지털 도구 활용 가이드	수업 중 활용할 수 있는 디지털 도구의 예시이다. 해당 차시에서 활용할 수 있는 디지털 도구의 예시와 활용 방법 및 유의점을 확인하는 자료로 활용한다.

7 학교 교육과정 편성 및 운영 방향 (6학년 2학기)

월	주	단원	과목	차시	차시명
8	4	1. 생활 속 컴퓨터와 문제 해결	학교자율 시간	1	• 생활 속 컴퓨터가 활용되는 모습을 찾아볼까요?
9	1		실과	2	• 컴퓨터로 해결할 수 있는 문제를 찾아볼까요?
				3	• 문제 해결 과정을 차례대로 나타내 볼까요?
	2		학교자율시간	4	• 놀이를 통해 알고리즘을 체험해볼까요?
				5	• 알고리즘을 바꿔볼까요?
	3	2. 컴퓨터에게 명령하기	실과	6	• 컴퓨터에게 명령하는 방법을 체험해 볼까요?
				7	• 명령어 블록을 조립하여 프로그램을 만들어 볼까요?
	4			8	• 자료를 입력받아 처리하는 프로그램을 만들어 볼까요?
				9	• 문제를 해결하는 프로그램을 만들어 볼까요?
10	1	3. 함께 만드는 생활 속 프로그램	학교자율시간	10	• 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾아볼까요?
				11	• 생활 속 문제를 프로그래밍으로 해결하기 위한 알고리즘을 표현해 볼까요?
	2			12	• 프로그래밍으로 나에게 필요한 프로그램을 만들어 볼까요?
				13	• 나에게 필요한 프로그램을 발전시켜 볼까요?
	3		실과	14	• 생활 속 문제를 해결하는 방법을 알고리즘으로 표현해 볼까요?
				15	• 알고리즘에 맞게 프로그램을 완성해 볼까요?
	4		학교자율시간	16	• 친구의 프로그램을 탐색하고 의견을 나눠볼까요?
				17	• 공유된 프로그램을 더 유용하게 수정해 볼까요?
11	1	4. 데이터의 이해	학교자율시간	18	• 우리 생활 속에는 어떤 데이터가 있을까요?
				19	• 데이터는 어떤 종류와 형태로 나타날까요?
	2			20	• 데이터에 담긴 의미를 살펴볼까요?
				21	• 생활 속 데이터의 의미를 놀이로 알아볼까요?
	3		실과	22	• 아날로그 데이터와 디지털 데이터는 어떻게 다를까요?
				23	• 컴퓨터는 데이터를 어떤 형태로 처리할까요?
	4			24	• 인공지능에는 어떤 데이터가 활용될까요?
		5. 인공지능의 이해와 올바른 활용	학교자율시간	25	• 인공지능은 데이터에서 무엇을 찾아낼까요?
12	1			26	• 인공지능은 비슷한 데이터를 어떻게 모을까요?
				27	• 인공지능이 데이터를 분류하는 과정을 체험해 볼까요?
	2		실과	28	• 인공지능은 어떤 과정을 거쳐 만들어질까요?
				29	• 인공지능을 만드는 과정을 직접 체험해 볼까요?
	3			30	• 인공지능은 우리 사회를 어떻게 바꾸고 있을까요?
				31	• 인공지능은 우리의 직업과 생활을 어떻게 바꿀까요?
	4		학교자율시간	32	• 인공지능을 사용할 때 무엇을 생각해야 할까요?
				33	• 인공지능을 올바르게 사용하는 방법을 함께 이야기해 볼까요?
				34	• 인공지능을 바르게 사용하기 위해 무엇을 실천할 수 있을까요?

※ Note. 본 학습 보완 자료는 1차시부터 17차시까지의 학습 내용을 대상으로 한다. 18차시부터 34차시까지의 학습 내용은 학교별 실과 교과서의 내용 체계와 차시 구성에 따라 재구성하여 운영한다. 본 자료의 차시는 학교 여건에 따라 선택적으로 운영할 수 있다. 현 6학년 학생은 인공지능 선도학교 운영, 창의적 체험활동을 통한 소프트웨어 교육 등 다양한 형태의 정보교육을 이수하였을 수 있다. 따라서 학교는 학생의 사전 학습 경험을 고려하여 이미 학습한 내용과 중복되는 차시를 제외하고, 학습 공백이 확인되는 내용을 중심으로 교육과정을 압축적으로 편성·운영할 수 있다.


초등학교
정보교육
학습 보완 자료

II

학습 보완 자료



1. 생활 속 컴퓨터와 문제 해결 ····· 14

2. 컴퓨터에게 명령하기 ····· 51

3. 함께 만드는 생활 속 프로그램 ····· 84

1 단원

생활 속 컴퓨터와 문제 해결



1 단원 개관

이 단원에서 학생들은 생활 속에서 컴퓨터가 활용되는 다양한 모습을 살펴보고, 컴퓨터가 일상생활의 문제를 해결하는데 어떻게 도움을 주는지 탐색한다. 이를 통해 컴퓨터를 데스크톱이나 노트북에 한정하지 않고, 스마트폰, 태블릿 PC, 키오스크, 디지털 학습 도구 등 생활 속 여러 기기와 시스템으로 확장하여 이해할 수 있도록 한다.

또한, 컴퓨터를 활용해 해결할 수 있는 생활 속 문제를 탐색하고, 문제 상황에 제시된 요소를 살펴 현재 상태와 목표 상태를 구분한다. 이후 문제를 해결하기 위한 방법과 순서를 알고리즘으로 구상하고, 이를 자연어, 의사 코드, 순서도 등 다양한 방법으로 표현한다. 놀이와 언플러그드 활동에서는 알고리즘에 따라 명령을 실행하고 그 결과를 점검하며, 원하는 결과가 나오지 않거나 더 간단한 방법이 있는 경우 알고리즘을 바꿔보는 경험을 한다.

2 단원 학습 연계

이전에 배운 내용	이번에 배울 내용	다음에 배울 내용
–	• 컴퓨터의 개념 • 문제 찾기와 문제 해결 절차 • 일상생활의 문제를 해결하기 위한 알고리즘 구상하기	• 컴퓨팅 시스템의 동작 원리 • 문제 추상화 • 알고리즘 표현 방법

3 단원 구성

차시	관련 성취기준	차시명
1	[06자율-1] 생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다.	• 생활 속 컴퓨터가 활용되는 모습을 찾아볼까요?
2	[6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.	• 컴퓨터로 해결할 수 있는 문제를 찾아볼까요?
3	[6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.	• 문제 해결 과정을 차례대로 나타내 볼까요?
4	[06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.	• 놀이를 통해 알고리즘을 체험해볼까요?
5	[06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.	• 알고리즘을 바꿔볼까요?

4 단원 학습목표

지식·이해	• 생활 속에서 컴퓨터가 활용되는 다양한 사례를 설명할 수 있다. • 문제 상황에서 현재 상태와 목표 상태의 의미를 이해할 수 있다. • 알고리즘이 문제를 해결하기 위한 명확한 절차이며, 정해진 순서대로 실행됨을 이해할 수 있다. • 알고리즘의 필요성을 설명할 수 있다.
과정·기능	• 생활 속에서 컴퓨터를 활용해 해결할 수 있는 문제를 탐색할 수 있다. • 문제 상황에 제시된 요소를 살펴 현재 상태와 목표 상태를 구분할 수 있다. • 일상생활 속 문제를 해결하기 위한 알고리즘을 자연어, 의사 코드, 순서도 등 다양한 방법으로 표현할 수 있다. • 놀이와 언플러그드 활동에서 알고리즘을 순서대로 실행하고, 실행 결과를 바탕으로 알고리즘을 수정·개선할 수 있다.
가치·태도	• 생활 속 문제를 컴퓨터를 활용하여 해결할 수 있는지 탐색하려는 태도를 가진다. • 알고리즘을 실행한 결과를 살펴보고, 더 나은 해결 방법으로 개선하려는 태도를 가진다. • 친구와 함께 문제 해결 절차를 표현하고, 서로의 생각을 존중하며 문제 해결 절차를 개선하려는 태도를 가진다.

1. 생활 속 컴퓨터가 활용되는 모습을 찾아볼까요?

- **관련 성취 기준 :** [06자율-1] 생활 속에서 컴퓨터가 활용되는 사례를 찾아보는 활동을 경험한다.
- **학습 목표 :** 생활 속에서 컴퓨터가 활용되는 다양한 사례를 찾아보고, 컴퓨터가 어떤 생활 문제를 해결하는지 설명할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 평소 컴퓨터를 사용해 본 자신의 경험 이야기 하기 • 컴퓨터가 무엇인지 생각해보기	• 컴퓨터의 의미 알아보기 • 생활 속에서 활용되는 다양한 컴퓨터 찾아보기 • 컴퓨터가 해결하는 생활 속 문제 연결하기 • 자신이 찾은 다양한 컴퓨터의 활용 모습 발표하기	• 생활 속 컴퓨터 활용 사례 정리하기 • 컴퓨터가 우리 생활에 주는 도움 이야기하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 학생의 일상 경험에서 출발하여 컴퓨터의 활용 사례를 탐색하도록 한다. • 컴퓨터를 데스크톱이나 노트북에 한정하지 않고 스마트폰, 태블릿 PC, 키오스크 등으로 확장하여 이해하도록 한다. • 단순히 “어디에 쓰이는가?”를 찾는 데 그치지 않고, “어떤 문제를 해결하는가?”를 생각하도록 한다.	• 학생 간 디지털 기기 사용 경험의 차이를 고려하여 발표 기회가 일부 학생에게만 집중되지 않도록 한다. • 스마트폰, 태블릿 PC 등의 기기도 정보를 입력하고 처리하며 결과를 보여 주는 컴퓨터의 한 종류임을 자연스럽게 안내한다. • 컴퓨터로 해결할 수 있는 문제와 사람이 직접 판단하고 소통해야 하는 문제를 지나치게 단순화하지 않도록 지도한다.	• 교사: 생활 속 컴퓨터 활용 사진 및 영상 자료, 학생용 활동지, 평가지 • 학생: 학생용 활동지, 태블릿 PC 등 인터넷 검색이 가능한 디지털 기기

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 평소 컴퓨터를 사용해 본 경험 이야기하기 T : 생활 속에서 컴퓨터를 사용해 본 경험이 있나요? S : 스마트폰을 사용하여 길을 찾아본 경험이 있습니다. S : 태블릿 PC를 활용하여 공부를 한 경험이 있습니다. ◆ 학습문제 확인하기 생활 속에서 컴퓨터가 어떻게 활용되는지 찾아보고, 컴퓨터가 어떤 문제를 해결해 주는지 알아봅시다.	• 컴퓨터를 사용해 본 경험이 데스크톱이나 노트북을 사용한 경험에만 머물지 않도록 다양한 예시를 안내한다.
전개	◆ 활동1. 컴퓨터의 의미 알아보기 T : 컴퓨터의 의미를 알아봅시다. S : 컴퓨터는 정보를 입력받아 정해진 과정에 따라 처리하고, 그 결과를 보여 주거나 저장하는 기기임을 확인한다. S : 컴퓨터는 데스크톱, 노트북뿐만 아니라 스마트폰, 태블릿 PC, 키오스크 등 다양한 모습으로 생활 속에 활용됨을 확인한다. ◆ 활동2. 생활 속 컴퓨터 활용 사례 찾아보기 T : 자신의 학생용 디바이스를 사용하여 생활 속에서 컴퓨터가 활용되는 사례를 찾아 활동지에 정리해봅시다. S : 학생용 디바이스를 사용하여 생활 속에서 활용되고 있는 컴퓨터의 종류와 활용 상황을 검색한다. S : 검색한 컴퓨터의 종류 및 활용 상황 및 해당 컴퓨터가 해결하는 문제를 활동지에 작성한다. 예) 키오스크 주문 – 음식점, 실시간 버스 정보 – 버스 정류장 등 ◆ 활동3. 컴퓨터가 해결한 문제 연결하기 T : 컴퓨터가 생활 속의 어떤 문제를 해결하는지 찾아보고 이를 활동지에 작성해 봅시다. T : 작성한 내용을 친구들에게 발표해 봅시다. S : 지도 앱은 목적지까지 가는 길을 모를 때, 가장 빠른 길을 알려줍니다.	• 준비물: 학생용 디바이스, 활동지 • 컴퓨터의 의미는 지나치게 기술적으로 설명하지 않는다. • 검색 활동이 어려운 학생에게는 교사가 예시 검색어를 제공하거나, 친구와 함께 사례를 확인하도록 한다. ※ 교과서 7페이지 정답 ① 컴퓨터 ② 하드웨어, 소프트웨어
정리	◆ 학습 내용 정리하기 T : 오늘 무엇을 배웠나요? S : 컴퓨터는 정보를 입력받아 이를 처리하고, 결과를 보여 주거나 저장하는 기기임을 알게 되었습니다. S : 컴퓨터는 생활 속 다양한 곳에서 활용되며, 사람들의 생활 속 문제 해결을 돕는다는 것을 알게 되었습니다.	• 컴퓨터는 문제 해결을 돕는 도구이며, 어떤 문제를 어떻게 해결할지 생각하는 것은 사람의 역할임을 안내한다.

수업 활동지

생활 속 컴퓨터가 활용되는 모습을 찾아볼까요?

1. 컴퓨터의 의미를 작성해 봅시다.

컴퓨터란 _____ 입니다.

2. 생활 속 컴퓨터 활용 사례를 작성해 봅시다.

내가 활용해 본 컴퓨터	어떤 상황에서 활용해 보았나요?	해결한 문제는 무엇인가요?
예) 키오스크	*예) 음식점*	*예) 사람이 부족해도 음식 주문을 할 수 있게 도와줍니다.*

3. 내가 정리한 사례를 친구들에게 소개해 봅시다.

내가 찾은 생활 속 컴퓨터는 _____ 입니다.

이 컴퓨터는 _____ 문제를 해결하는 데 도움을 줍니다.

왜냐하면 _____ 때문입니다.

수업 평가지

1. 다음 중 생활 속에서 컴퓨터가 활용되는 사례에 해당하는 것을 모두 골라 봅시다.

① 스마트폰으로 길 찾기
② 키오스크로 음식 주문하기
③ 종이 공책에 연필로 일기 쓰기
④ 태블릿 PC로 온라인 학습하기
⑤ 종이 달력에서 오늘의 날짜 확인하기

정답 : _____

2. 다음 문장의 빈칸에 알맞은 말을 써 봅시다.

컴퓨터는 정보를 입력받아 정해진 과정에 따라 처리하고, 그 결과를 보여 주거나 저장하여 생활 속 _____ 해결을 돕는다.

정답 : _____

3. 생활 속 컴퓨터 활용 사례를 두 가지 쓰고, 어떤 문제를 해결하는지 설명해 봅시다.

생활 속 컴퓨터	활용되는 장소	해결하는 문제

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ①, ②, ④
2. 문제
3. 예) 생활 속 컴퓨터 : 스마트폰
   해결하는 문제 : 스마트폰의 날씨 앱을 통해 오늘의 날씨를 확인하여 날씨에 적합한 옷차림을 선택할 수 있도록 도와준다.
   예) 생활 속 컴퓨터 : 지도 앱
   해결하는 문제 : 길을 모를 때 목적지까지 가는 방법을 알려 주어 길 찾기의 어려움을 해결해 준다.

평가 목표

생활 속에서 컴퓨터가 활용되는 사례를 찾고, 컴퓨터가 어떤 생활 속 문제를 해결하는지 설명할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	생활 속에서 컴퓨터가 활용되는 사례를 두 가지 이상 찾고, 각 사례가 해결하는 생활 속 문제를 구체적으로 설명할 수 있다.
보통	생활 속에서 컴퓨터가 활용되는 사례를 찾고, 컴퓨터가 해결하는 문제를 간단히 설명할 수 있다.
노력 요함	생활 속에서 컴퓨터가 활용되는 사례를 주변의 도움을 받아 찾을 수 있다.

맞춤형 피드백 예시

학생 반응	피드백 예시
컴퓨터를 데스크톱이나 노트북으로만 생각하는 경우	스마트폰이나 키오스크도 정보를 처리하고 결과를 보여 주는 컴퓨터의 한 종류로 볼 수 있어요. 우리 주변에서 비슷한 기기를 더 찾아볼까요?
사례는 찾았지만 해결하는 문제를 설명하지 못하는 경우	그 기기를 사용할 때 무엇이 더 편리해졌는지 생각해 봅시다. 예를 들어 지도 앱은 어떤 어려움을 해결해 줄까요?

수업 운영 팁

Q1. 학생들이 컴퓨터를 데스크톱이나 노트북으로만 생각하면 어떻게 지도하나요?

A1. 스마트폰, 태블릿 PC, 키오스크, 스마트 워치 등도 정보를 입력받고 처리하여 결과를 보여 주는 컴퓨터의 한 종류임을 사례 중심으로 안내해주세요. 단, 모든 전자제품을 무조건 컴퓨터로 설명하기보다는 "정보를 처리하고 정해진 명령에 따라 작동하는가?"를 기준으로 생각하게 해주세요.

Q2. 학생 발표가 단순한 컴퓨터 사용 경험 나열로 끝나지 않게 하려면 어떻게 하나요?

A2. 발표할 때 "무엇을 사용했나요?", "어떤 상황에서 사용했나요?", "어떤 문제가 해결되었나요?", "왜 편리했나요?"의 네 가지 질문을 순차적으로 활용해보세요. 이를 통해 학생들의 컴퓨터 사용 경험을 문제 해결 관점으로 확장할 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
검색 도구	태블릿 PC, 데스크톱	생활 속 컴퓨터 활용 사례 검색
	구글, 유튜브	키오스크, 내비게이션 등 사례 검색

활용 시 유의점

- 학생 개인 계정 로그인이 필요한 디지털 도구는 되도록 사용하지 않는다.
- 검색 활동을 할 경우, 교사가 미리 안전한 검색어와 참고 사이트를 안내한다.
- 학생의 이름, 사진, 위치 정보 등 개인정보가 포함된 자료를 공유하지 않도록 지도한다.
- 인터넷 사용이 어려운 환경에서는 교사가 준비한 사진 및 영상 자료로 대체한다.

2. 컴퓨터로 해결할 수 있는 문제를 찾아볼까요?

- **관련 성취 기준** : [6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.
- **학습 목표 :** 컴퓨터나 앱을 활용해 생활 속 문제를 해결하는 사례를 살펴보고, 문제의 현재 상태와 목표 상태를 구분할 수 있다.
- **수업 로드맵**

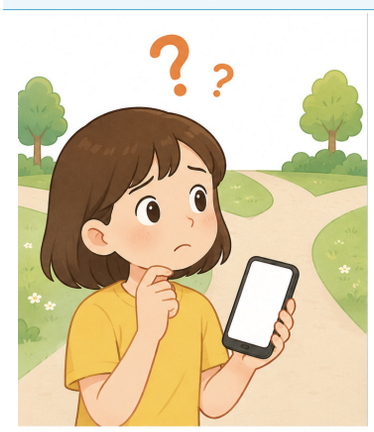
도입 (5분)	전개 (30분)	정리 (5분)
• 가는 길과 걸리는 시간을 몰라 곤란했던 경험 떠올리기 • 학습 문제 확인하기	• 문제 상황에 제시된 요소 찾기 • 현재 상태와 목표 상태 구분하기 • 다른 생활 속 문제에서 현재 상태와 목표 상태 찾기	• 문제를 현재 상태·목표 상태로 나눌 수 있음을 정리하기 • 다음 시간 예고하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 컴퓨터·앱이 '우리가 입력한 정보를 이용해 문제를 해결한다'는 점을 생활 사례로 이해하도록 한다. • '앱이 길을 찾아 준다'처럼 학생이 바로 공감하는 사례에서 출발한다. • 현재 상태와 목표 상태는 '구분해 보는' 수준으로 다룬다.	• 초점을 '사람이 앱을 켜고 누르는 순서'가 아니라 '앱'이 정보를 이용해 문제를 해결하는 과정'에 둔다. • 현재 상태는 '불편함'이 아니라 '문제가 아직 해결되지 않은 지금의 상태'임을 분명히 안내한다.	• 교사: 지도 앱 화면 사진(길 표시 전·후), 문제 상황 자료, 학생용 활동지· 평가지 • 학생: 학생용 활동지, 필기도구

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 생활 속 경험 떠올리기
T : 새로 생긴 장소나 약속 장소에 갈 때, 어느 길로 가야 할지 또는 몇 분쯤 걸릴지 몰랐던 적이 있나요?
S : 네, 처음 가는 곳이라 길을 잘 몰랐습니다.
S : 약속 장소까지 몇 분 걸리는지 찾아본 적이 있습니다.
S : 지도 앱으로 가는 길을 확인한 적이 있습니다.

◆ 학습문제 확인하기
컴퓨터로 해결하는 생활 속 문제의 현재 상태와 목표 상태를 파악해 봅시다.</td><td></td></tr>
<tr><td>전개</td><td>◆ 활동1. 문제 상황에 제시된 요소 찾기
T : 다음 문제 상황을 살펴봅시다.
다른 지역의 도서관에서 열리는 체험 행사에 참여하려고 합니다. 처음 가는 곳이라 도서관의 위치와 가는 길, 걸리는 시간을 알지 못합니다. 이때 지도 앱을 사용하면 도서관의 위치와 가는 길, 걸리는 시간을 확인할 수 있습니다.
T : 문제 해결에 필요한 것을 찾아봅시다.
〈보기〉
도서관 / 지도앱 / 가는 길 / 걸리는 시간 / 좋아하는 색 / 내 키
T : 문제 해결에 필요한 것은 무엇인가요?

◆ 활동2. 현재 상태와 목표 상태 구분하기
T : 두 장면을 비교해 봅시다.
현재 상태 / 목표 상태

</td><td>• 지도 앱 화면 사진, 활동지

• ‘앱을 누르는 순서’가 아니라 ‘앱이 정보를 이용해 길을 찾아 주는 일’에 초점을 둔다.

• 현재 상태는 ‘불편함’이 아니라 ‘아직 해결되지 않은 지금 상태’임을 분명히 한다.</td></tr>
</table>

T : 현재 상태와 목표 상태를 정리해 봅시다.

현재 상태 : 참여하려는 도서관의 위치와 가는 길, 걸리는 시간을 모르는 상태
목표 상태 : 참여하려는 도서관의 위치와 가는 길, 걸리는 시간을 알게 된 상태

◆ **활동3. 다른 생활 속 문제에서 현재 상태와 목표 상태 찾아보기**

T : 다른 생활 속 문제도 살펴봅시다.

[문제 상황]
아침에 하늘이 흐립니다. 비가 올지 몰라 우산을 챙길지 고민됩니다.
날씨 앱을 보면 오늘 비 예보를 알 수 있습니다.

T : 이 문제의 현재 상태와 목표 상태를 찾아 말해 봅시다.

현재 상태	목표 상태
	

〈보기〉 모르는 / 알게 된

현재 상태	목표 상태
비가 올지 안 올지 (　　) 상태	비가 올지 안 올지 (　　) 상태

T : 생활 속 문제도 현재 상태와 목표 상태로 나누어 볼 수 있습니다.

※ 교과서 11페이지 정답
① 현재
② 목표

정리

◆ **학습 내용 정리하기**

T : 생활 속 문제는 현재 상태와 목표 상태로 나누어 볼 수 있습니다.

T : 여러 가지 생활 속 문제도 같은 방법으로 현재 상태와 목표 상태로 나누어 볼 수 있습니다.

T : 다음 시간에는 현재 상태를 목표 상태로 바꾸는 해결 과정을 순서대로 나타내 봅니다.

수업 활동지

컴퓨터로 해결할 수 있는 문제를 찾아볼까요?

1. 문제 상황을 읽고, 문제 해결에 필요한 것에 모두 ○표 해 봅시다.

[문제 상황]
다른 지역의 도서관에서 열리는 체험 행사에 참여하려고 합니다. 처음 가는 곳이라 도서관의 위치와 가는 길, 걸리는 시간을 알지 못합니다. 이때 지도 앱을 사용하면 도서관의 위치와 가는 길, 걸리는 시간을 확인할 수 있습니다.

도서관 위치 ( )	지도 앱 ( )	가는 길 ( )	걸리는 시간 ( )	좋아하는 색 ( )	내 키 ( )

2. 두 장면을 보고 〈보기〉에서 알맞은 말을 골라 빈칸을 채워 봅시다.





〈보기〉 모르는 / 알게 된

현재 상태: 도서관의 위치와 가는 길, 걸리는 시간을 ( ) 상태	목표 상태: 도서관의 위치와 가는 길, 걸리는 시간을 ( ) 상태

3. 다른 문제도 보기에서 골라 현재 상태와 목표 상태로 나누어 봅시다.

[문제 상황]
아침에 하늘이 흐립니다. 비가 올지 몰라 우산을 챙길지 고민됩니다. 날씨 앱을 보면 오늘 비 예보를 알 수 있습니다.

〈보기〉 모르는 / 알게 된
현재 상태: 오늘 비 예보를 ( ) 상태
목표 상태: 오늘 비 예보를 ( ) 상태

수업 평가지

1. 지도 앱으로 길을 찾는 문제 해결과 관련이 가장 적은 것은 무엇인가요?

① 출발하는 곳
② 도착하는 곳
③ 가는 길
④ 걸리는 시간
⑤ 내가 좋아하는 색

정답 : _____

2. 다음 상황을 읽고, 현재 상태와 목표 상태를 써 봅시다.

[상황] 친구 집에 처음 가는데 길을 몰랐다. 지도 앱에 친구 집 주소를 입력하니 가는 길이 표시되어 잘 찾아갔다.

현재 상태: _____
목표 상태: _____

3. 컴퓨터나 앱으로 해결할 수 있는 생활 속 문제를 떠올려, 현재 상태와 활용할 기기나 앱, 목표 상태를 써 봅시다.

현재 상태	활용 기기 • 앱	목표 상태

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ⑤
2. 현재 상태: 친구 집에 가는 길을 모르는 상태
   목표 상태: 친구 집까지 가는 길을 알게 된 상태
3. 현재 상태: ① 외국어 안내문의 뜻을 모르는 상태 ② 버스가 언제 도착하는지 모르는 상태
   활용 기기·앱: ① 번역 앱 ② 버스 정보 앱
   목표 상태: ① 안내문의 뜻을 아는 상태 ② 버스 도착 시간을 아는 상태

평가 목표

컴퓨터나 앱이 문제를 해결한 사례에서 필요한 정보를 구분하고, 그 문제를 현재 상태와 목표 상태로 나누어 설명할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	문제 해결에 필요한 정보를 구분하고, 컴퓨터·앱이 해결한 문제를 현재 상태와 목표 상태로 분명하게 나누어 설명할 수 있다.
보통	컴퓨터·앱이 해결한 문제를 현재 상태와 목표 상태로 구분할 수 있다.
노력 요함	도움을 받아 현재 상태와 목표 상태를 한 가지 이상 나타낼 수 있다.

맞춤형 피드백 예시

학생 반응	피드백 예시
현재 상태를 '불편한 점'으로만 적는 경우	현재 상태는 '불편한 느낌'이 아니라 '아직 해결되지 않은 지금의 상태'를 그대로 적는 것임을 안내하세요. '길을 모르는 상태'처럼 '지금 어떤 상태인가'를 적게 합니다.
'앱을 켜고 누른다'처럼 사람이 조작하는 순서로 답하는 경우	'앱이 무엇을 보고(출발지·도착지) 무엇을 찾아 주었나요?'로 물어, '앱이 정보를 이용해 해결한 일'로 초점을 옮겨 주세요.
문제 상황에서 필요한 요소를 못 고르는 경우	'이 정보가 없으면 길을 찾을 수 있을까?'를 하나씩 물어보며 필요한 정보와 필요 없는 정보를 가려내게 하세요.

수업 운영 팁

Q1. '현재 상태'를 '불편한 일'과 같다고 생각하는 학생이 많아요.

A1. 현재 상태는 불편함 자체가 아니라 '문제가 아직 해결되지 않은 지금의 상태'입니다. '길을 모르는 상태'처럼 지금 어떤 상태에 있는지를 나타내는 것임을 안내합니다. 학생이 "길을 못 찾아서 답답해요"처럼 감정을 적으면, "그래서 지금 무엇을 모르고 있나요?"라고 되물어 '길을 모르는 상태'처럼 상태 문장으로 바꾸게 합니다.

Q2. 학생이 '앱 켜고 주소 누르고…'처럼 사람이 조작하는 순서를 말하면 어떻게 하나요?

A2. 이 차시의 초점은 '사람이 앱을 어떻게 사용하는가'가 아니라, '앱이 어떤 정보를 이용해 무엇을 해결해 주었는가'에 있습니다. 학생이 조작 순서를 말하면, "앱이 출발지와 도착지를 받아서 무엇을 알려 주었나요?"처럼 앱이 한 일을 중심으로 다시 질문합니다. 이렇게 하면 '입력 정보 → 처리 → 결과'라는 컴퓨터의 기본 동작 흐름을 자연스럽게 인식할 수 있으며, 3차시의 알고리즘 표현으로도 연결됩니다.

Q3. 지도 앱을 써 본 경험이 적은 학생은 어떻게 하나요?

A3. 앱을 직접 사용한 경험이 없어도 참여하는 데 문제가 없습니다. 길이 표시되기 전과 후의 지도 앱 화면 사진을 교사가 미리 준비해 두면 경험 없이도 상황을 충분히 이해할 수 있습니다. "차를 타고 처음 가는 곳에 갈 때 내비게이션이 안내해 준 것을 본 적 있나요?"처럼 지도 앱 대신 내비게이션 경험으로 대체해도 됩니다. 중요한 것은 '모르는 상태 → 알게 된 상태'라는 변화를 이해하는 것이므로, 경험이 비슷하다면 어떤 사례로 대체해도 무방합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
검색 도구	유튜브, 네이버, 구글	컴퓨터·앱이 생활 속 문제를 해결한 다른 사례 찾기
지도 앱	네이버지도, 구글맵	지도 앱이 길을 찾아 주는 전·후 화면을 함께 보기

활용 시 유의점

- 인터넷 사용이 어려운 환경에서는 교사가 준비한 화면 사진·영상 자료로 대체한다.
- 이 차시는 앱을 직접 조작하지 않아도 된다. 화면 사진 한두 장이면 충분하며, 기기·앱 환경 차이로 수업 흐름이 끊기지 않도록 한다.

3. 문제 해결 과정을 차례대로 나타내 볼까요?

- **관련 성취 기준 :** [6실05-01] 컴퓨터를 활용한 생활 속 문제해결 사례를 탐색하고 일상생활 속 문제를 해결하기 위한 알고리즘을 다양한 방법으로 표현한다.
- **학습 목표 :** 문제 해결 과정을 자연어로 순서대로 나타낼 수 있고, 같은 알고리즘을 자연어·의사 코드·순서도로 표현할 수 있음을 안다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 지난 시간 다른 지역의 도서관에서 열리는 체험 행사 가는 길에 관한 문제 떠올리기 • 학습 문제 확인하기	• 문제 해결 과정을 자연어로 나타내기 • 같은 과정을 자연어·의사 코드·순서도로 비교하기 • 보기를 이용해 순서도 완성하기	• 알고리즘과 세 가지 표현 방법 정리하기 • 같은 과정을 여러 방법으로 나타낼 수 있음을 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 자연어 순서 표현은 모든 학생의 공통 활동으로 한다. • 의사 코드·순서도는 예시를 비교하며 살펴보게 한다. • 같은 알고리즘을 자연어·의사 코드·순서도로 나타낼 수 있음을 강조한다.	• 자연어·의사 코드·순서도가 '같은 문제 해결 과정'을 다른 방법으로 나타낸 것임을 같은 부분끼리 짚어 준다. • '가려는 곳이 보이는가?', '비 예보가 있는가?'처럼 갈라지는 판단을 짚어 준다. • 순서도 기호(시작/끝, 처리, 판단, 흐름선)는 간단히만 소개한다.	• 교사: 자연어, 의사코드, 순서도 예시 자료, 활동지·평가지 • 학생: 학생용 활동지, 필기도구

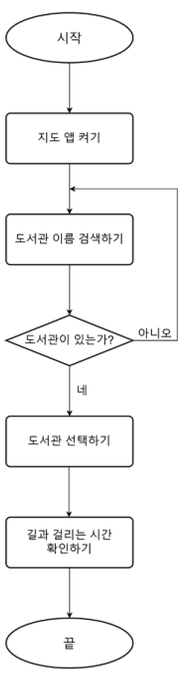
교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 지난 시간 내용 떠올리기 T : 지난 시간에는 어떤 생활 속 문제를 살펴보았나요? S : 다른 지역의 도서관에서 열리는 체험 행사까지 가는 길과 시간을 모르는 문제였습니다. T : 지난 시간에 나눈 문제 해결 방법을 다시 살펴봅시다. S : 지도 앱 켜기, 도서관 이름 검색하기, 가려는 목적지 선택하기, 길과 시간 확인하기로 해결했습니다. ◆ 학습문제 확인하기 문제 해결 과정을 차례대로 나타내고, 여러 가지 방법으로 표현해 봅시다.	• 지난 차시의 문제·해결 방법을 그대로 이어서 사용한다.
전개	◆ 활동1. 문제 해결 과정을 자연어로 나타내기 • 개념 설명하기 [자연어 개념 정의] 문제 해결 방법을 순서대로 나열한 것을 알고리즘이라고 합니다. 알고리즘은 여러 가지 방법으로 나타낼 수 있습니다. 평소에 쓰는 말로 나타낸 것이 자연어 표현입니다. T : 아래 문장을 알맞은 순서로 나타내려면 번호를 어떻게 써야 할까요? ① 화면에 나타난 길과 걸리는 시간을 확인한다. ② 지도앱을 켠다. ③ 지도앱에서 도서관 이름을 검색한다. ④ 가려는 도서관을 선택한다. 알맞은 순서: ( ) → ( ) → ( ) → ( ) 정답: ② → ③ → ④ → ① T : 왜 이 순서로 나타내야 하는지 이야기해 봅시다. S : 지도 앱을 먼저 켜야 도서관 이름을 검색할 수 있습니다. S : 도서관을 선택해야 길과 걸리는 시간을 확인할 수 있습니다. T : 문제 해결 과정에서 한 단계가 빠지거나 순서가 달라지면 어떻게 될까요? S : 필요한 정보를 확인하지 못할 수 있습니다. S : 문제 해결 과정이 자연스럽게 이어지지 않을 수 있습니다. T : 자연어는 일상생활에서 사용하는 말로 문제 해결 과정을 나타내는 방법입니다.	• 자연어·의사 코드·순서도 예시 자료, 활동지 • 자연어 표현은 모든 학생 공통 활동으로 한다. • 의사 코드·순서도는 '보고 비교하기·빈칸 채우기' 수준으로 운영한다. • 판단(있는가?/오는가?)에서 행동이 갈라짐을 짚어 준다.

◆ 활동2. 같은 과정을 자연어·의사 코드·순서도로 비교하기

- 같은 문제 해결 과정을 세 가지 방법으로 나타낸 예를 살펴봅시다.

문제 상황: 지도 앱으로 도서관까지 가는 길과 걸리는 시간을 확인합니다.

자연어	의사코드	순서도
	시작	
① 지도 앱을 켠다.	지도앱 켜기	
② 도서관 이름을 검색한다.	도서관 이름 검색하기	
③ 가려는 도서관이 보이면 선택한다.	만약 (가려는 도서관이 있다면)	
	도서관 선택하기	
④ 가려는 곳이 없으면 다시 검색한다.	아니면	
	도서관 이름 다시 검색하기	
⑤ 길과 걸리는 시간을 확인한다.	길과 시간 확인하기	
	끝	

T : 세 가지 표현에서 같은 내용을 찾아봅시다.
S : '지도앱 켜기'가 모두 들어 있습니다.
S : '도서관 이름 검색하기'가 모두 들어 있습니다.
S : '검색 결과에 내가 가려는 도서관이 있는가?에 따라 다음 행동이 달라집니다.
S : 마지막에는 길과 걸리는 시간을 확인합니다.

T : 세 가지 표현을 보고 무엇을 알 수 있었나요?
S : 세 표현이 모두 다른 모습입니다.
S : 해결하고자 하는 문제는 같은 문제였습니다.

T : 자연어는 일상생활에서 사용하는 말로 나타내는 방법입니다.
T : 의사 코드는 프로그래밍 언어와 비슷한 형식으로 자연어를 사용하여 나타내는 방법입니다.
T : 순서도는 약속된 기호와 화살표를 사용해 순서대로 나타내는 방법입니다.
T : 세 가지 표현의 모습은 다르지만, 해결하려는 문제와 문제를 해결하는 순서는 같습니다.

◆ 활동3. 문제 해결 과정을 자연어로 표현하여 봅시다.

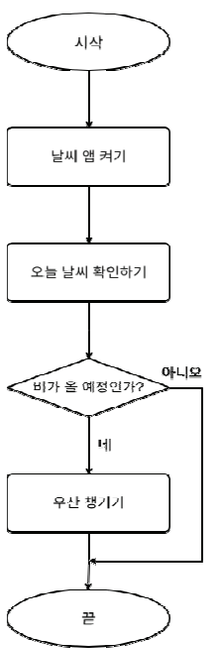
T : 다음 문제 상황의 해결 과정을 자연어로 나타낸 것입니다. 빈칸에 알맞은 내용을 〈보기〉에서 찾아 써 봅시다.

[문제 상황]
아침에 일어나 창문 밖을 확인하니 구름이 많이 껴 흐린 날씨이다. 그래서 날씨 앱을 통해 오늘 비가 올지 안 올지 알아보고자 한다.

같은 과정을 나타낸 의사코드와 순서도도 함께 살펴봅시다.

〈보기〉
㉠비 예보가 있는가? ㉡ 점심에 떡볶이가 나오는가?
㉢ 알림장에 준비물이 써져 있는가?

① 날씨앱을 켠다. ② 오늘 날짜의 날씨를 확인한다.
③ (㉠) ④ 우산을 챙긴다.
⑤ 끝

의사코드	순서도
시작 날씨 앱 켜기 오늘 날씨 확인하기 만약 비 예보라면 우산 챙기기 끝	

T : 손가락으로 문제 해결 과정을 따라가봅시다.

T : 시작부터 끝까지 행동이 어떤 순서로 이어지는지 확인합니다.

※ 교과서 15페이지 정답
① 알고리즘
② 순서(절차)

정리

◆ 학습 내용 정리하기

T : 문제 해결 방법을 순서대로 나열한 것을 알고리즘이라고 합니다.

T : 같은 알고리즘도 자연어, 의사 코드, 순서도처럼 여러 가지 방법으로 나타낼 수 있습니다.

T : 자연어·의사 코드·순서도는 모습이 다르지만, 같은 문제 해결 과정을 나타낼 수 있습니다.

• 형식보다 '순서를 분명히 나누었는가'를 강조한다.

수업 활동지

문제 해결 과정을 차례대로 나타내 볼까요?

1. 다음 문장을 알맞은 순서로 나타내려면 번호를 어떻게 써야 할까요? (지도 앱으로 도서관 찾기)

① 화면에 나타난 길과 걸리는 시간을 확인한다.　② 지도앱을 켠다.
③ 지도앱에서 도서관 이름을 검색한다.　④ 가려는 도서관을 선택한다.

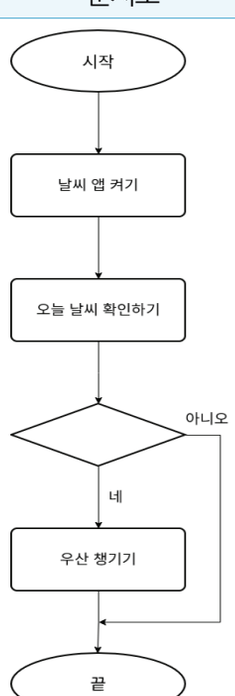
알맞은 순서 : (　　) → (　　) → (　　) → (　　)

2. 다음 문제 상황의 해결 과정을 자연어로 나타낸 것입니다. 〈보기〉에서 알맞은 내용을 찾아 빈칸에 쓰고, 같은 과정을 나타낸 의사코드와 순서도를 함께 살펴봅시다.

[문제 상황]
아침에 일어나 창문 밖을 확인하니 구름이 많이 껴 흐린 날씨이다. 그래서 날씨 앱을 통해 오늘 비가 올지 안올지 알아보고자 한다.

〈보기〉
㉠ 비 예보가 있는가?　㉡ 점심에 떡볶이가 나오는가?　㉢ 알림장에 준비물이 써져 있는가?

① 날씨 앱을 켠다.　② 오늘 날짜의 날씨를 확인한다.
③ _____　④ 우산을 챙긴다.

의사코드	순서도
시작 날씨 앱 켜기 빈칸 : _____ 만약 비 예보라면 　우산 챙기기 끝	

수업 평가지

1. 문제 해결 과정을 차례대로 나타낸 것을 무엇이라고 하나요?

① 목적지
② 출발지
③ 알고리즘
④ 목표 상태
⑤ 현재 상태

정답 : _____

2. 지도 앱으로 도서관을 찾는 과정입니다. 알맞은 순서대로 나열해보세요

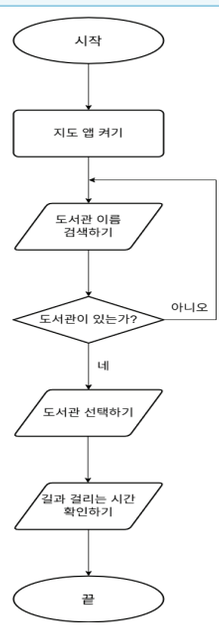
① 지도앱을 켠다.
② 화면에 나타난 길과 걸리는 시간을 확인한다.
③ 지도앱에서 도서관 이름을 검색한다.

순서: ( ) → ( ) → ( )

3. 다음은 같은 과정을 세 가지 방법으로 나타낸 것입니다. ㉠㉡㉢이 무엇인지 보기에서 골라 써 봅시다.

[보기] 자연어, 의사 코드, 순서도

문제 상황: 지도 앱으로 도서관까지 가는 길과 걸리는 시간을 확인합니다.

㉠ : _____	㉡ : _____	㉢: _____
① 지도 앱을 켠다. ② 도서관 이름을 검색한다. ③ 가려는 도서관이 보이면 선택한다. ④ 가려는 곳이 없으면 다시 검색한다. ⑤ 길과 걸리는 시간을 확인한다.	시작 지도앱 켜기 도서관 이름 검색하기 만약 (가려는 곳이 있다면) 목적지 선택하기 아니면 도서관 이름 다시 검색하기 길과 시간 확인하기 끝	

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ③ 알고리즘
2. ① → ③ → ②
3. ㉠ 자연어 ㉡ 의사 코드 ㉢ 순서도

평가 목표

문제 해결 과정을 자연어로 순서대로 나타내고, 자연어·의사 코드·순서도가 같은 알고리즘을 나타내는 서로 다른 방법임을 이해하는지 확인한다.

평가 기준

수준	평가 기준
잘함	문제 해결 과정을 순서대로 바르게 나타내고, 자연어·의사 코드·순서도가 같은 알고리즘을 나타내는 서로 다른 방법임을 이해한다.
보통	문제 해결 과정을 순서대로 나타내고, 세 가지 표현이 같은 과정을 나타낸 것임을 안다.
노력 요함	도움을 받아 문제 해결 과정의 순서를 나타낼 수 있다.

맞춤형 피드백 예시

학생 반응	피드백 예시
순서가 뒤섞이거나 단계가 빠지는 경우	'가장 먼저 무엇을 해야 하나요?'부터 한 단계씩 짚어 보게 하고, 순서를 바꾸면 결과가 어떻게 달라지는지 생각하게 해 주세요.
세 가지 표현이 같은 것임을 모르는 경우	'도서관 이름 검색하기' 부분을 자연어·의사 코드·순서도에서 차례로 짚어 주세요. 같은 내용이 방법만 달라진다는 것을 눈으로 확인하게 합니다.
순서도의 판단(㉠)을 어려워하는 경우	'무엇에 따라 우산을 챙길지 갈라지나요?'라고 물어, 판단은 '예/아니오로 답할 수 있는 질문'임을 알려 주세요.

수업 운영 팁

Q1. 의사 코드·순서도를 모든 학생이 직접 만들어야 하나요?

A1. 아닙니다. 자연어로 나타내는 것만 공통 활동이고, 의사 코드·순서도는 '보고 비교하기·빈칸 채우기' 수준이면 충분합니다. 빠르게 끝낸 학생에게는 순서도의 판단(마름모)을 다른 상황(예: '비 예보가 있는가?')으로 바꿔 보게 하고, 어려워하는 학생에게는 세 표현의 '같은 줄'을 손으로 짚어 가며 내용이 같음을 눈으로 확인하게 하세요. 형식의 완성도보다 '순서를 분명히 나누었는가'에 초점을 두는 까닭은, 이 차시의 목표가 표현 방법에 대한 숙달이 아니라 같은 알고리즘을 여러 방법으로 나타낼 수 있음을 아는 데 있기 때문입니다.

Q2. 순서도의 판단(마름모) 기호를 어떻게 설명해야 학생들이 이해하나요?

A2. '예 또는 아니오로만 대답할 수 있는 질문'이 들어가는 자리라고 설명합니다. '도서관이 있는가?', '비 예보가 있는가?'처럼 두 가지 중 하나로만 답할 수 있는 질문을 마름모 안에 쓰고, 예와 아니오 방향으로 화살표가 갈라진다는 것을 짚어 주면 됩니다. '좋아하는 음식은?'처럼 답이 여러 개인 질문은 판단 기호에 들어갈 수 없다는 점도 함께 안내하면 개념이 더 명확해집니다. 순서도 기호 전체를 다 외우도록 요구하지 않습니다.

Q3. 자연어·의사 코드·순서도가 서로 완전히 다른 것이라고 생각하는 학생은 어떻게 하나요?

A3. 세 가지는 '같은 알고리즘'을 다른 방법으로 표현한 것입니다. 교사가 자연어·의사 코드·순서도를 가로로 나란히 제시하고, 각 표현에서 '지도 앱 켜기', '도서관 이름 검색하기' 등 동일한 내용이 각각 어디에 있는지 함께 찾아봅니다. 내용은 같고 모습만 다르다는 것을 직접 손으로 짚어 확인하면 세 가지가 별개라는 오해를 자연스럽게 해소할 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
지도 앱	네이버지도, 구글맵	지도 앱이 길을 찾아 주는 전·후 화면을 함께 보기
날씨 앱	날씨 기본 앱	날짜별 기온, 강수량, 일출, 일몰 시간 등 확인하기

활용 시 유의점

- 학생이 직접 순서도를 그리는 도구 사용은 권하지 않으며, 보고 비교하는 수준으로 운영한다.
- 비교 자료를 띄울 때 자연어·의사 코드·순서도를 가로로 나란히 두어, '같은 줄 = 같은 내용, 방법만 다름'을 손으로 짚어 줄 수 있게 한다.
- 인터넷 사용이 어려운 환경에서는 교사가 미리 준비한 화면 사진 자료로 대체한다.

4. 놀이를 통해 알고리즘을 체험해볼까요?

- **관련 성취 기준 :** [06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.
- **학습 목표 :** 언플러그드 활동을 통해, 알고리즘이 정해진 순서대로 실행됨을 체험할 수 있다.
- **수업 로드맵**

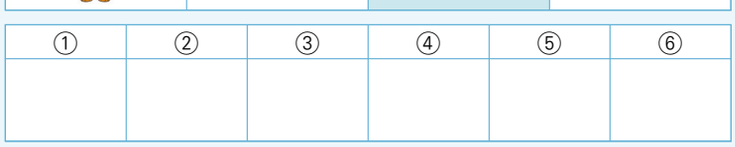
도입 (5분)	전개 (30분)	정리 (5분)
• 게임 캐릭터를 움직인 경험 떠올리기 • 명령을 순서대로 따르면 어떻게 되는지 이야기하기 • 학습 문제 확인하기	• 말판과 명령(↑ ↓ ⟷), ★ 꼭 지나기 칸 알아보기 • 고양이를 ★을 지나 집까지 보내는 명령 만들기 • 짝과 바꾸어 명령 그대로 실행하기	• 명령은 정해진 순서대로 실행됨을 정리하기 • 활동지를 보관하고 다음 시간 예고하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 컴퓨터·코딩 도구 없이 누구나 참여할 수 있는 말판(언플러그드) 놀이로 운영한다. • 정답을 빨리 맞히기보다 명령을 순서대로 만들어 보는 과정에 의미를 둔다. • '명령을 순서대로 늘어놓은 것'이 알고리즘임을 놀이로 자연스럽게 체험하게 한다.	• 명령 하나는 한 칸만 이동하며, 왼쪽에서 오른쪽으로 차례대로 실행함을 분명히 안내한다. • 도착하는 길이 한 가지가 아닐 수 있음을 허용한다. • 짝의 명령을 바꾸지 않고 적힌 순서대로만 실행하게 하여 '순서대로 실행'을 체험하게 한다. • 이번 활동지는 5차시에 다시 사용하므로 보관하도록 안내한다.	• 교사: 말판(바닥·칠판용), 명령 카드(↑ ↓ ⟷), 학생 활동지·평가지 • 학생: 학생용 활동지, 필기도구

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 생활 속 알고리즘 경험 떠올리기
T : 게임에서 캐릭터를 움직이려면 어떻게 해야 하나요?
S : 위쪽, 아래쪽, 왼쪽, 오른쪽으로 움직이게 합니다.
S : 버튼을 순서대로 눌러야 합니다.
T : 명령을 순서대로 따라가면 어떤 일이 생길까요?
S : 정해진 곳으로 움직입니다.

◆ 학습문제 확인하기
순서에 따라 알고리즘을 설계하여 주어진 문제를 해결해 봅시다.</td><td>• 컴퓨터 없이도 할 수 있는 놀이임을 안내한다.</td></tr>
<tr><td>전개</td><td>◆ 활동1. 말판과 명령 알아보기
• 말판에서 고양이와 집의 위치를 확인해 봅시다.
[설명]

고양이가 출발하여 집으로 가야 합니다.
사용할 수 있는 명령을 알아봅시다.
↑ 위로 한 칸
↓ 아래로 한 칸
← 왼쪽으로 한 칸
→ 오른쪽으로 한 칸
① 명령 하나를 읽으면 한 칸만 움직입니다.
② 명령은 왼쪽에서 오른쪽으로 차례대로 실행합니다.
③ 필요한 만큼만 명령을 쓰고, 남는 칸은 비워 둡니다.
④ 명령대로 움직인 뒤 집에 도착했는지 확인합니다.</td><td>• 말판, 명령 카드, 활동지

• 명령 하나는 한 칸만 이동함을 분명히 한다.

• 짝의 명령을 바꾸지 않고 그대로 실행하게 한다.</td></tr>
</table>

◆ 활동2. 집까지 가는 명령 만들기

T : 고양이가 집까지 가도록 명령을 만들어 봅시다.

T : 명령 칸에 화살표를 차례대로 그려 봅시다.

〈예시〉

① ↑ ② ↑ ③ ↑ ④ ↑ ⑤ → ⑥ →

T : 내가 만든 명령대로 고양이를 움직여 봅시다.

①	②	③	④	⑤	⑥
→	→	↑	↑	↑	→

T : 결과를 표시해 봅시다.

■ 집에 도착했다.
□ 다른 곳에 도착했다.

◆ 활동3. 짝과 바꾸어 실행하기

T : 이번에는 짝이 만든 명령을 그대로 따라가 봅시다.

T : 짝의 명령을 내 생각대로 바꾸지 않고, 적힌 순서대로만 실행합니다.

①	②	③	④	⑤	⑥
→	→	↑	→	↑	↑

T : 짝의 명령대로 움직인 결과를 표시해 봅시다.

■ 집에 도착했다.
□ 다른 곳에 도착했다.

T : 명령을 순서대로 실행했나요?

S : 네, 적힌 순서대로 하나씩 실행했어요.

T : 짝의 명령과 내 명령의 결과가 같았나요, 달랐나요?

S : 같았다 / 달랐다

T : 명령이 빠지거나 순서가 달라지면 결과가 어떻게 될까요?

S : 생각한 곳이 아닌 다른 곳에 도착해요. 명령 하나가 빠지면 한 칸 덜 가요.

※ 교과서 19페이지 정답
① 알고리즘
② 다른

정리

◆ 학습 내용 정리하기

T : 명령은 정해진 순서대로 실행됩니다.

T : 명령을 정확한 순서로 나타내면 원하는 곳에 도착할 수 있습니다.

T : 명령이 빠지거나 순서가 맞지 않으면 생각한 곳과 다른 곳에 도착할 수 있습니다.

T : 오늘 만든 명령은 다음 시간에 다시 사용하므로 활동지를 보관합니다.

• 활동지 보관을 꼭 안내한다.

수업 활동지

놀이를 통해 알고리즘을 만들어볼까요?

1. 말판에서 고양이(출발), 집(도착), ★ 꼭 지나기 칸을 확인하고, 사용할 명령을 알아봅시다.

			
		★ 꼭 지나기	

명령 : ↑ 위로 한 칸 ↓ 아래로 한 칸 ← 왼쪽으로 한 칸 → 오른쪽으로 한 칸

※ 대각선으로 가는 명령은 없어요. ↑와 →를 함께 써서 가야 해요.

※ 고양이는 ★(꼭 지나야 하는 칸)을 반드시 지나 집까지 가야 해요. 명령 하나는 한 칸만 움직이고, 남는 칸은 비워 둬요.

1-1. 고양이가 ★을 지나 집까지 가도록 명령을 차례대로 써 봅시다. (한 칸에 화살표 하나씩, 예: ① → ② ↑)

①	②	③	④	⑤	⑥

내 명령대로 움직인 결과 : ☐ 집에 도착했다 ☐ 다른 곳에 도착했다

2. 짝과 활동지를 바꾸어, 짝의 명령을 적힌 순서대로 실행해 봅시다.

짝이 만든 명령 순서대로 적기

①	②	③	④	⑤	⑥

짝의 명령대로 움직인 결과 : ☐ 집에 도착했다 ☐ 다른 곳에 도착했다

이웃한 두 명령의 자리를 바꾸어 실행하면 결과가 어떻게 달라질까요?

_____

수업 평가지

1. 명령을 순서대로 늘어놓아 문제를 해결하는 것을 무엇이라고 하나요?

① 말판
② 자석
③ 그림
④ 장애물
⑤ 알고리즘

정답 : _____

2. 고양이가 ★(꼭 지나야 하는 칸)을 지나 집까지 가도록 명령을 차례대로 써 봅시다.

		
	★ 꼭 지나기	

①	②	③	④	⑤

3. 명령이 빠지거나 순서가 바뀌면 결과는 어떻게 될까요?

① 항상 집에 도착한다
② 아무 일도 일어나지 않는다
③ 명령이 저절로 고쳐진다
④ 생각한 곳과 다른 곳에 도착할 수 있다

정답 : _____

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ⑤
2. →→↑↑↑ (★을 지나 집까지) — ★을 지나 집에 도착하는 다른 순서도 정답
3. ④

평가 목표

명령을 순서대로 만들어 목표 지점까지 이동시키고, 명령의 순서가 결과에 영향을 줌을 이해하는지 확인한다.

평가 기준

수준	평가 기준
잘함	고양이를 집까지 보내는 명령을 순서대로 바르게 만들고, 순서가 결과에 영향을 줌을 설명할 수 있다.
보통	고양이를 집까지 보내는 명령을 순서대로 만들 수 있다.
노력 요함	도움을 받아 한두 개의 명령을 순서대로 나타낼 수 있다.

맞춤형 피드백 예시

학생 반응	피드백 예시
방향(↑↓←→)을 헷갈리는 경우	손가락으로 말판 위를 한 칸씩 짚으며 명령을 하나씩 따라가게 해 주세요. 한 명령에 한 칸만 움직임을 함께 확인합니다.
명령을 한꺼번에 여러 칸으로 생각하는 경우	'명령 하나는 한 칸' 규칙을 다시 짚고, 필요한 칸 수만큼 화살표를 쓰게 하세요.
짝의 명령을 자기 생각대로 바꾸는 경우	'적힌 순서 그대로 실행하기'가 규칙임을 안내하고, 결과가 달라지면 명령을 함께 살펴보게 하세요.

수업 운영 팁

Q1. 컴퓨터가 없어도 수업을 진행할 수 있나요?

A1. 이 차시는 컴퓨터 없이 진행하는 언플러그드 활동이므로 컴퓨터실이 없어도 전혀 문제가 없습니다. 말판은 칠판에 격자를 그리거나 A4 용지에 인쇄하여 사용할 수 있고, 교실 바닥에 테이프로 격자 칸을 만들어 학생이 직접 '고양이' 역할을 하며 이동하는 방식으로도 운영할 수 있습니다. 명령 카드는 화살표를 적은 종이 카드로 간단히 대체할 수 있으며, 준비물 없이 칠판만 있어도 수업 진행이 가능합니다.

Q2. 학생마다 주어진 문제를 해결하는 방법이 다릅니다. 어떻게 지도할까요?

A2. ★을 지나 집에 도착하면 모두 정답입니다. 빨리 끝낸 학생은 ★이나 장애물을 하나 더 두고 새 명령을 만들게 합니다. 서로 다른 길을 비교하되, '★을 지났는가', '한 명령은 한 칸인가'와 같은 규칙은 지키게 합니다.

Q3. 활동지를 보관하는 이유는 무엇인가요?

A3. 오늘 만든 명령은 5차시 수업에서 그대로 사용합니다. 5차시에서는 말판에 장애물이 추가된 상황에서 오늘의 명령을 실행해 보고, 막히는 부분을 찾아 고치는 활동을 진행합니다. 오늘 만든 명령이 출발점이 되기 때문에 활동지를 반드시 보관해야 하며, 수업 전 안내와 함께 가방 안에 바로 넣도록 지도하는 것이 좋습니다. 분실한 학생은 5차시 시작 전에 간단한 명령을 새로 만들면 되므로, 분실 자체가 활동 참여를 막지는 않습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
영상 자료	유튜브	명령으로 움직이는 로봇 영상을 보며, 제대로 된 명령이 프로그래밍에서 왜 중요한지 공감하기

활용 시 유의점

- 이 차시는 디지털 도구 없이 말판 활동만으로 충분히 운영할 수 있으며, 도구는 선택적으로만 활용한다.
- 영상은 명령대로 움직이는 로봇·캐릭터처럼 '순서대로 실행' 개념에 맞는 것만 짧게 보여 주고, 놀이 시간을 침범하지 않게 한다.

5. 알고리즘을 바꿔볼까요?

- **관련 성취 기준 :** [06자율-2] 놀이를 통해 알고리즘의 다양한 사례를 체험하는 활동을 수행한다.
- **학습 목표 :** 실행 결과를 보고 명령(알고리즘)을 바꿔 더 정확하고 짧은 알고리즘으로 만들 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 지난 시간에 만든 명령 떠올리기 • 길에 장애물이 생긴 상황 확인하기 • 학습 문제 확인하기	• 지난 시간 명령을 장애물 말판에서 실행하기 • 결과를 보고 명령 고치기 • 고친 명령으로 다시 실행하기	• 알고리즘은 실행해 보고 고칠 수 있음을 정리하기 • 더 짧고 알기 쉬운 알고리즘이 더 나음을 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 알고리즘은 한 번 만들고 끝이 아니라, 실행해 보고 고치며 더 낫게 만들 수 있음을 체험하게 한다. • '왜 결과가 달라졌는지' 까닭을 찾는 과정에 의미를 둔다. • 같은 도착이라도 더 짧고 알기 쉬운 명령이 더 나은 알고리즘임을 안내한다.	• 장애물 칸은 지나갈 수 없음을 분명히 안내한다. • 막힌 명령을 찾아 그 부분만 고치도록 안내하여, 전체를 새로 만들지 않아도 됨을 알게 한다. • 도착에 성공한 학생은 '더 짧게 필요 없는 명령 줄이기'로 나아가게 한다. • 4차시 활동지를 준비하게 하고, 없으면 간단한 명령을 새로 만들게 한다.	• 교사: 장애물이 있는 말판, 명령 카드, 학생용 활동지·평가지 • 학생: 4차시 활동지, 학생용 활동지, 필기도구

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 생활 속 명령 경험 떠올리기
T : 지난 시간에는 무엇을 했나요?
S : 고양이를 집까지 보내는 명령을 만들었습니다.
S : 짝의 명령을 순서대로 따라가 보았습니다.
T : 오늘은 지난 시간에 만든 명령을 다시 사용해 봅니다.
T : 활동지를 잃어버렸다면, 고양이가 집까지 가는 명령을 새로 간단히 만들어 봅니다.
<table><tr><td colspan="6">[지난 시간 명령]</td></tr><tr><td>①</td><td>②</td><td>③</td><td>④</td><td>⑤</td><td>⑥</td></tr><tr><td>→</td><td>→</td><td>→</td><td>↑</td><td>↑</td><td>↑</td></tr><tr><td colspan="6">[새로 만들기]</td></tr><tr><td>①</td><td>②</td><td>③</td><td>④</td><td>⑤</td><td>⑥</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>◆ 학습문제 확인하기
명령을 바꿔 고양이를 다시 집까지 보내 봅시다.</td><td>• 장애물은 지나갈 수 없음을 분명히 한다.</td></tr>
<tr><td>전개</td><td>◆ 활동1. 장애물 말판에서 실행해 보기
T : 지난 시간 명령을 장애물 말판에서 그대로 실행해 봅시다.

T : 장애물 칸은 지나갈 수 없습니다.
T : 결과를 표시해 봅시다.
■ 장애물에 막혔다.
□ 집에 도착했다.
□ 다른 곳에 도착했다.</td><td>• 장애물 말판, 명령 카드, 4차시 활동지
• 막힌 부분만 찾아 고치도록 안내한다.
• 도착한 학생은 '더 짧게'로 나아가게 한다.</td></tr>
</table>

T : 명령은 그대로인데 결과가 달라진 까닭은 무엇일까요?
S : 가는 길이 막혔습니다.
S : 장애물이 있습니다.
T : 어느 명령을 바꾸면 장애물을 피할 수 있을까요?

◆ 활동2. 명령 고치기

T : 실행 결과를 보고 명령을 바꿔봅시다.

집에 도착하지 못했다면
• 장애물에 막히거나 다른 곳으로 간 까닭을 찾습니다.
• 고양이가 집까지 갈 수 있도록 명령을 고칩니다.

집에 도착했다면
• 더 짧게 갈 수 있는지 살펴봅니다.
• 필요 없는 명령이 있으면 줄여 봅니다.

다른 곳에 도착했다면
• 명령을 한 칸씩 다시 실행하며 잘못된 명령을 찾아 고칩니다.

[고친 명령]

①	②	③	④	⑤	⑥
→	↑	→	→	↑	↑

◆ 활동3. 고친 명령 다시 실행하기

T : 고친 명령대로 다시 움직여 봅시다.
T : 결과를 표시해 봅시다.

□ 집에 도착했다.
□ 아직 도착하지 못했다.
□ 처음보다 명령이 짧아졌다.
□ 처음보다 알기 쉬워졌다.

T : 만약 고친 명령으로 아직 집에 도착하지 못했다면, 도착하도록 바꿔봅시다.

※ 교과서 23페이지 정답
① 바꿀(고칠)
② 짧은

정리

◆ 학습 내용 정리하기

T : 알고리즘은 실행해 보고 고칠 수 있습니다.
T : 원하는 결과가 나오지 않으면 잘못된 명령을 찾아 바꾸어야 합니다.
T : 같은 결과에 도착하더라도 더 짧고 알기 쉬운 명령이 더 나은 알고리즘이 될 수 있습니다.
T : 알고리즘은 한 번 만들고 끝나는 것이 아니라, 실행해 보고 고치며 더 낫게 만들 수 있습니다.

• '고치며 더 낫게'를 강조한다.

수업 활동지

내 알고리즘이 작동하지 않아요! 바꿔볼까요?

1. 지난 시간(4차시)에 만든 명령을 그대로 옮겨 적어 봅시다.

(활동지를 잃어버렸다면 고양이가 집까지 가는 명령을 새로 간단히 만들어요.)

①	②	③	④	⑤	⑥

2. 장애물 말판에서 지난 시간 명령을 그대로 실행해 봅시다. (장애물 칸은 지나갈 수 없어요)

실행 결과 : ☐ 장애물에 막혔다 ☐ 집에 도착했다 ☐ 다른 곳에 도착했다

명령은 그대로인데 결과가 달라진 까닭은 무엇일까요?

_____

3. 실행 결과를 보고 명령을 바꿔 봅시다.

- 집에 도착하지 못했다면: 장애물에 막히거나 다른 곳으로 간 까닭을 찾아, 명령을 바꿔요.
- 집에 도착했다면: 더 짧게 갈 수 있는지 살펴보고, 필요 없는 명령을 줄여요.

[고친 명령]

①	②	③	④	⑤	⑥

4. 고친 명령대로 다시 움직여 보고, 결과를 표시해 봅시다.

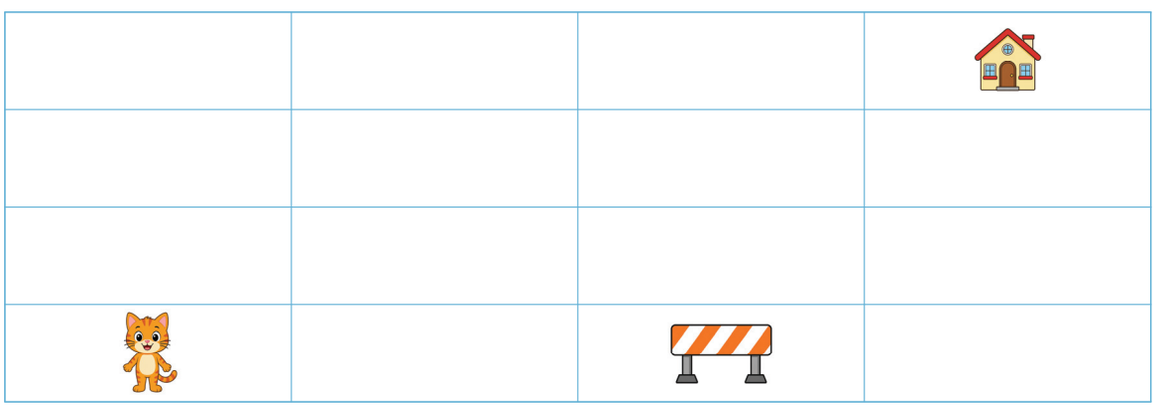
☐ 집에 도착했다
☐ 처음보다 명령이 짧아졌다
☐ 아직 도착하지 못했다
☐ 처음보다 알기 쉬워졌다

수업 평가지

1. 알고리즘에 대한 설명으로 알맞은 것은 무엇인가요?

① 순서는 중요하지 않다
② 길이가 길수록 항상 더 좋다
③ 한 번 만들면 절대 고칠 수 없다
④ 실행해 보고 바꿔서 더 낫게 만들 수 있다
정답 : _____

2. 말판에서 명령어대로 움직였지만 집에 도착하지 못했습니다. 그 까닭으로 알맞은 것은?



[명령]					
①	②	③	④	⑤	⑥
→	→	→	↑	↑	↑

① 말판이 작아서
② 고양이가 사라져서
③ 명령이 스스로 바뀌어서
④ 장애물 칸을 지나갈 수 없어서
정답 : _____

3. 두 명령 모두 집에 도착했습니다. 더 나은 알고리즘은 어느 것이며, 그 까닭은 무엇인가요?

㉮ → → ↑ ↑ ← →　　㉯ → → ↑ ↑
더 나은 것 : (　　)
까닭: _____

_____

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ④
2. ④
3. ㉯ / 까닭: 같은 집에 도착하지만 명령이 더 짧고 알기 쉽기 때문이다.

평가 목표

실행 결과를 보고 명령(알고리즘)을 바꿔 목표에 도착하게 만들고, 더 짧고 알기 쉬운 알고리즘이 더 나음을 이해하는지 확인한다.

평가 기준

수준	평가 기준
잘함	실행 결과를 보고 막힌 명령을 찾아 바꿔 집에 도착하게 만들고, 더 짧고 알기 쉬운 알고리즘이 더 나음을 설명할 수 있다.
보통	실행 결과를 보고 명령을 바꿔 집에 도착하게 만들 수 있다.
노력 요함	도움을 받아 막힌 명령을 찾아 한 부분을 고칠 수 있다.

맞춤형 피드백 예시

학생 반응	피드백 예시
막힌 까닭을 찾지 못하는 경우	명령을 한 칸씩 따라가며 어디에서 장애물에 막히는지 함께 짚어 주세요. 막히는 그 명령이 바꿔야 할 부분입니다.
전체 명령을 처음부터 새로 만들려는 경우	막힌 부분 앞까지는 그대로 두고, 막힌 명령만 바꾸어 장애물을 돌아가게 하면 됨을 안내하세요.
도착에 성공해 더 나아가지 못하는 경우	'같은 집에 더 적은 명령으로 갈 수 있을까?'를 물어 필요 없는 명령을 줄여 보게 하세요.

수업 운영 팁

Q1. 명령을 전부 새로 만들려는 학생이 많아요. 어떻게 지도하나요?

A1. 이 차시의 핵심은 '잘못된 한 부분을 찾아 그 부분만 고치는 것'입니다. 처음부터 전부 다시 만들면 오류를 찾아 수정하는 경험 없이 단순히 명령을 다시 작성하는 것에 그치게 됩니다. "어느 칸에서 장애물에 막혔나요?"라고 먼저 물어 막힌 위치를 특정하게 한 뒤, "그 앞 명령은 맞고, 그 명령 하나만 바꾸면 어떻게 될까요?"라고 이어 물어 부분 수정으로 유도합니다. 잘못된 명령을 찾아내는 과정 자체가 이 차시에서 가장 중요한 사고 활동임을 학생에게도 미리 안내해 두면 좋습니다.

Q2. 4차시 활동지를 잃어버린 학생은 어떻게 하나요?

A2. 활동지가 없어도 수업 참여에 문제가 없습니다. 5차시 시작 전 3~5분을 활용하여 고양이를 집까지 보내는 간단한 명령을 새로 만들게 하면 됩니다. 이때 명령 칸을 6칸으로 제한하지 않고 자유롭게 적게 해도 됩니다. 활동의 핵심은 새 명령을 만드는 것이 아니라 '실행해 보고 막힌 부분을 고치는 경험'이므로, 명령의 내용보다 고치는 과정에 집중하도록 안내합니다.

Q3. '더 짧은 알고리즘'이 왜 더 나은지 학생이 납득하지 못하면 어떻게 설명하나요?

A3. 긴 명령과 짧은 명령을 칠판에 나란히 적어 두고, "어느 쪽이 친구에게 말로 전달하기 쉬운가?", "어느 쪽이 실수할 가능성이 적은가?"를 직접 물어봅니다. 명령이 짧을수록 읽기·전달하기·고치기가 쉽다는 것을 체감하게 하는 것이 목표입니다. 다만 '무조건 짧을수록 좋다'는 오해가 생기지 않도록, ★을 빠뜨리거나 장애물을 그대로 지나가는 짧은 명령은 규칙을 어긴 것이므로 더 나은 알고리즘이 아님을 함께 짚습니다. '올바른 결과를 내면서 동시에 더 짧고 읽기 쉬운 것'이 더 나은 알고리즘의 기준임을 정리해 줍니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
영상 자료	유튜브	명령으로 움직이는 로봇·캐릭터 사례 보기

활용 시 유의점

- 이 차시는 디지털 도구 없이 말판 활동만으로 충분히 운영할 수 있으며, 도구는 선택적으로만 활용한다.
- '잘못된 부분만 찾아 고치는' 이 차시의 핵심이 흐려지지 않게 한다.

2단원 컴퓨터에게 명령하기



1 단원 개관

이 단원에서 학생들은 컴퓨터에게 일을 시키기 위해서는 컴퓨터가 이해할 수 있는 방식으로 명령을 전달해야 함을 이해한다. 학생들은 프로그램과 프로그래밍의 의미를 알고, 블록형 교육용 프로그래밍 언어인 엔트리의 화면 구성과 사용 방법을 익힌다.

또한 학생들은 명령어 블록을 순서대로 조립하여 컴퓨터에게 명령하는 과정을 직접 체험한다. 순차, 반복, 선택의 기본 구조를 이해하고, 변수를 활용하여 자료를 입력받아 처리하고 결과를 출력하는 과정을 경험한다. 이를 바탕으로 문제를 이해하고 분석한 뒤 알고리즘을 설계하고, 주어진 문제를 해결하는 프로그램을 직접 작성하며 실행 결과를 점검하고 수정하는 능력을 기른다.

2 단원 학습 연계

이전에 배운 내용	이번에 배울 내용	다음에 배울 내용
–	• 컴퓨터에게 명령하는 방법	• 문제 추상화 • 알고리즘 표현 방법

3 단원 구성

차시	관련 성취기준	차시명
1	[6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	• 컴퓨터에게 명령하는 방법을 체험해 볼까요?
2	[6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	• 명령어 블록을 조립하여 프로그램을 만들어 볼까요?
3	[6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	• 자료를 입력받아 처리하는 프로그램을 만들어 볼까요?
4	[6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.	• 문제를 해결하는 프로그램을 만들어 볼까요?

4 단원 학습목표

지식·이해	• 프로그램과 프로그래밍의 의미를 이해할 수 있다. • 프로그래밍 언어와 엔트리의 화면 구성을 이해할 수 있다. • 변수와 순차·반복·선택의 의미를 이해할 수 있다.
과정·기능	• 명령어 블록을 조립하여 컴퓨터에게 명령할 수 있다. • 자료를 입력받아 처리하고 결과를 출력하는 프로그램을 작성할 수 있다. • 문제를 해결하기 위한 알고리즘을 설계하고 문제를 해결하는 프로그램을 작성할 수 있다.
가치·태도	• 프로그래밍으로 문제를 해결하려는 적극적인 태도를 가진다. • 프로그램의 실행 결과를 확인하고 끈기 있게 수정·보완하려는 태도를 가진다.

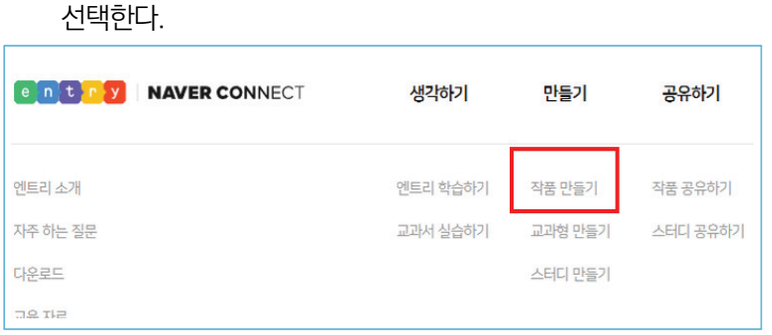
1. 컴퓨터에게 명령하는 방법을 체험해 볼까요?

- **관련 성취 기준 :** [6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.
- **학습 목표 :** 프로그램과 프로그래밍의 의미를 이해하고, 블록 프로그래밍 언어인 엔트리의 구성과 기능을 설명할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 컴퓨터에게 명령을 내릴 수 있다면 무엇을 시키고 싶은지 이야기해 보기 • 사람의 말을 컴퓨터가 바로 알아듣지 못하는 까닭 생각해 보기 • 학습 문제 확인하기	• 프로그램과 프로그래밍의 의미 알아보기 • 프로그래밍 언어가 필요한 까닭 이해하기 • 엔트리에 접속하여 화면 구성 살펴보기 • 각 화면 요소의 이름과 기능 정리하기 • 블록을 끌어와 블록 조립소에서 조립해 보기	• 프로그램과 프로그래밍의 의미 정리하기 • 오늘 배운 개념과 컴퓨터에게 명령할 때 중요한 점 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 학생의 일상 경험에서 출발하여 명령의 정확성에 대한 필요를 느끼도록 한다. • 컴퓨터는 정확한 순서에 의해 일을 한다는 점을 직접 체험으로 익히게 한다. • 본 차시에서는 직접 프로그램을 만들기보다 도구의 구조를 탐색하는 데 중점을 둔다.	• 특정 프로그래밍 도구의 세부 기능 익히기에 치중하지 않도록 한다. • 프로그래밍 언어의 종류나 문법을 자세히 다루지 않고, 블록을 조립하는 수준의 방식임을 안내한다. • 모든 명령어 블록의 기능을 한 번에 익히려 하지 않도록 한다.	• 교사: 교사용 컴퓨터 • 학생: 디지털 디바이스, 활동지

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 컴퓨터와 사람에게 일을 시킬 때 차이 생각해보기 T : 컴퓨터에게 무엇이든 명령할 수 있다면 어떤 일을 시키고 싶나요? S : 어려운 계산을 대신 해 달라고 하고 싶습니다. S : 정해진 시간에 알림을 보내 달라고 하고 싶습니다. T : 사람의 말을 컴퓨터가 바로 알아듣지 못하는 까닭은 무엇일까요? S : 컴퓨터는 사람의 말을 그대로 이해하지 못하기 때문입니다. ◆ 학습문제 확인하기 프로그램과 프로그래밍의 의미를 이해하고, 컴퓨터에게 명령하는 언어인 엔트리의 구성과 기능을 알아봅시다.	• 학생이 AI는 말만 해도 알아서 대답한다는 식의 질문을 할 경우 AI도 사람이 설계한 프로그램 안에서 미리 만들어진 수학적 모델이 작동한 결과임을 간단히 안내한다.
전개	◆ 활동1. 프로그램과 프로그래밍의 의미 알아보기 T : 프로그램과 프로그래밍이 무엇인지 알아봅시다. S : 프로그램은 컴퓨터가 일을 처리하도록 그 방법과 순서를 명령으로 모아 놓은 것입니다. S : 프로그램을 만드는 과정을 프로그래밍이라고 합니다. T : 프로그래밍을 할 때 프로그래밍 언어가 필요한 까닭을 알아봅시다. S : 컴퓨터가 이해할 수 있는 언어로 명령을 바꿔야 합니다. T : 컴퓨터가 이해할 수 있는 언어를 프로그래밍 언어라고 합니다. S : 명령어를 블록처럼 조립하는 교육용 프로그래밍 언어로 엔트리가 있음을 이해한다. ◆ 활동2. 엔트리 화면 구성 살펴보기 T : '엔트리'에 접속하여 화면을 살펴봅시다. S : 엔트리(playentry.org)에 접속한 뒤 [만들기]에서 작품 만들기를 선택한다.  T : 엔트리 화면에서 각 요소가 어떤 역할을 하는지 활동지에 정리해 봅시다.	• 준비물: 엔트리 사이트, 활동지 • 프로그램과 프로그래밍 언어의 의미를 지나치게 기술적으로 설명하지 않는다. • 로그인이 필요한 기능은 사용하지 않고 명령어 조립을 위한 핵심 요소를 중심으로 화면 구성을 살펴보는 데 집중한다.

S : 블록 꾸러미는 각 카테고리의 모든 블록을 모아놓은 곳 입니다.
S : 블록 꾸러미, 블록 조립소, 실행 화면, 오브젝트 목록 등의 이름과 역할을 살펴본다.



T : 블록 꾸러미에 있는 블록을 블록 조립소로 끌어서 가져옵니다. 블록끼리 조립해 봅니다.
S : 블록끼리 가까워지면 '딸깍' 소리가 나고 블록이 조립됩니다.
T : 조립된 블록을 분리해서 휴지통에 버리고 다른 블록을 끌어서 가져와 조립해 봅시다.
S : 블록끼리 가까워지면 '딸깍' 소리가 나고 블록이 조립됩니다.

※ 교과서 29페이지 정답
① 프로그램
② 프로그래밍 언어

정리

◆ 학습 내용 정리하기

T : 오늘 무엇을 배웠나요?
S : 프로그램은 컴퓨터가 할 일을 순서대로 모아 놓은 명령 묶음이라는 것을 알게 되었습니다.
S : 컴퓨터가 알아들을 수 있는 언어로 명령해야 한다는 것을 알게 되었습니다.
S : 엔트리의 화면이 어떻게 구성되었는지 알게 되었습니다.
T : 다음 시간에는 명령어 블록을 조립해 간단한 프로그램을 만들어 보겠습니다.

수업 활동지

컴퓨터에게 명령하는 방법을 체험해 볼까요?

1. 프로그램과 프로그래밍의 뜻을 정리해 봅시다.

프로그램이란 _____ 입니다.

프로그래밍이란 _____ 입니다.

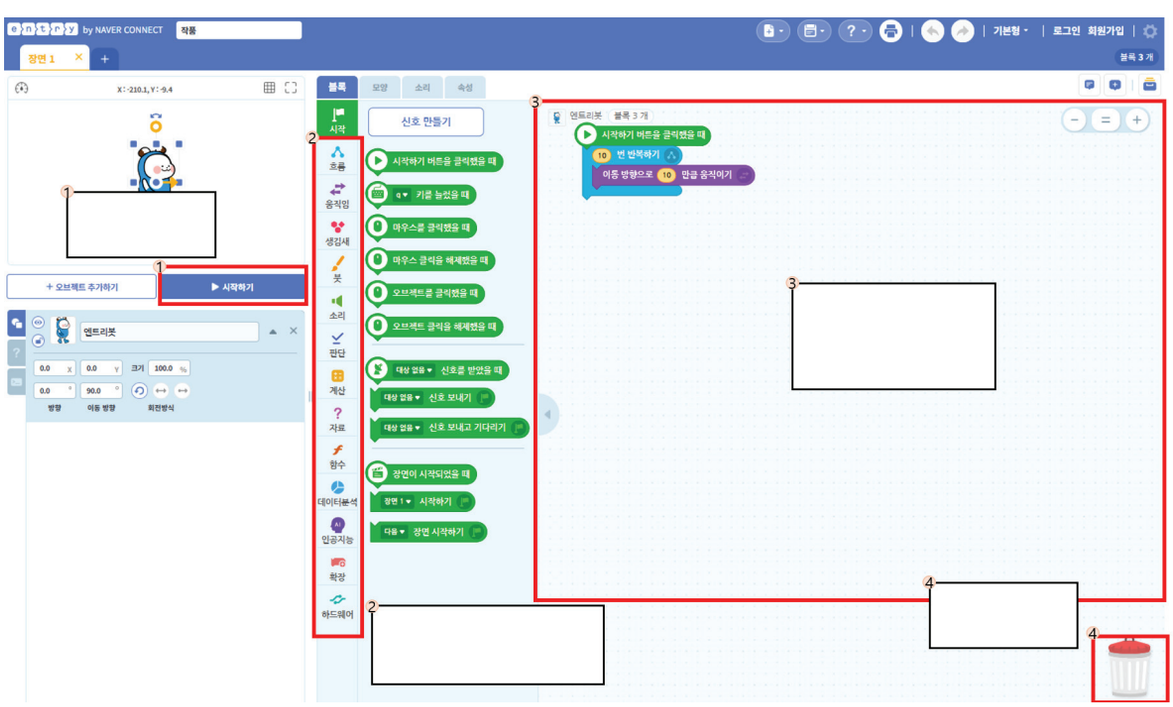
2. 다음 특징을 보고 사람에게 일을 시킬 때와 컴퓨터에게 일을 시킬 때의 차이점을 한 문장으로 써 봅시다.

사람에게 일을 시킬 때	컴퓨터에게 일을 시킬 때
• 말이 조금 달라도 이해할 수 있다. • 상황에 따라 판단할 수 있다.	• 명령을 정확하게 입력해야 한다. • 입력한 명령대로만 실행한다.

컴퓨터에게 일을 시킬 때는

_____ 해야 합니다.

3. 엔트리 화면에서 각 요소가 하는 일을 찾아 적어 봅시다.



수업 평가지

1. 배운 내용을 떠올리며 빈칸에 알맞은 말을 써 봅시다.

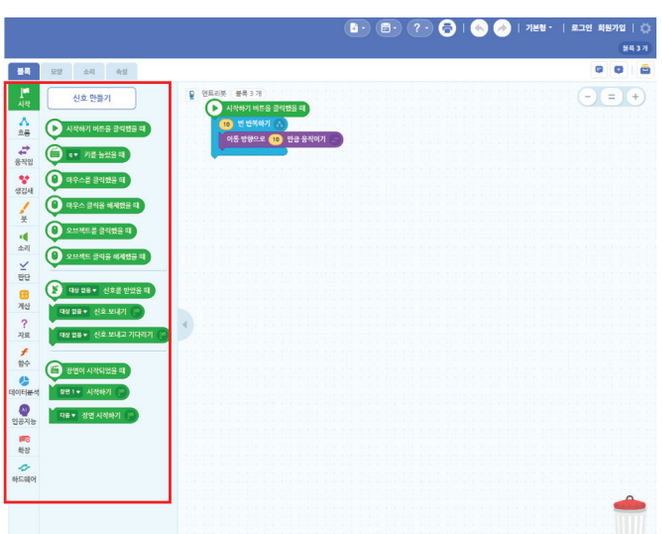
컴퓨터가 해야 할 일의 순서와 방법을 명령어로 모아 놓은 것을 _____(이)라고 하고,
이것을 만드는 과정을 _____(이)라고 합니다.

2. 컴퓨터가 명령을 이해하려면 무엇이 필요할까요?

① 사람의 말
② 프로그래밍 언어
③ 큰 목소리
④ 손짓과 몸짓

정답 : _____

3. 다음 엔트리 화면에서 빨간색으로 표시된 부분의 이름과 하는 일을 써 봅시다.



- 이름: ( )
- 하는 일: ( )

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. 프로그램/프로그래밍
2. ②
3. 이름: 블록 꾸러미, 하는 일: 각 카테고리와 관련된 블록을 모아놓은 곳

평가 목표

프로그램과 프로그래밍의 의미를 이해하고, 컴퓨터가 이해하는 프로그래밍 언어의 필요성을 설명하며, 엔트리 화면의 주요 구성 요소와 역할을 이해한다.

평가 기준

수준	평가 기준
잘함	프로그램과 프로그래밍의 의미를 정확히 이해하고, 컴퓨터가 프로그래밍 언어를 사용하는 이유를 설명할 수 있으며, 엔트리 화면의 주요 구성 요소의 이름과 하는 일을 정확하게 말할 수 있다.
보통	프로그램과 프로그래밍의 의미를 알고, 컴퓨터가 프로그래밍 언어를 사용한다는 것을 이해하며, 엔트리 화면의 주요 구성 요소의 이름이나 하는 일을 대부분 설명할 수 있다.
노력 요함	프로그램과 프로그래밍의 의미를 구별하는 데 어려움이 있거나, 프로그래밍 언어의 필요성과 엔트리 화면의 주요 구성 요소를 이해하는 데 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
프로그램과 프로그래밍을 혼동하는 경우	프로그램은 컴퓨터가 할 일을 순서대로 모아 놓은 것이고, 프로그래밍은 그 프로그램을 만드는 과정입니다. 두 단어의 뜻을 다시 비교해 봅시다.
엔트리 화면의 이름이나 역할을 잘 모르는 경우	엔트리 화면을 다시 살펴보며 블록 꾸러미, 블록 조립소, 실행 화면, 오브젝트 목록이 각각 무엇을 하는 곳인지 확인해 봅시다.

수업 운영 팁

Q1. 인터넷 환경이 불안정하여 엔트리 접속이 어려울 때는 어떻게 하나요?

A1. 엔트리는 설치형(오프라인) 버전도 제공되므로 미리 설치해 두면 인터넷 없이도 화면 구성을 살펴볼 수 있습니다. 접속이 모두 어려운 경우에는 교사용 컴퓨터의 화면을 함께 보며 요소를 짚어 주는 방식으로 진행할 수 있습니다.

Q2. 학생마다 디지털 기기 사용 능력의 차이가 클 때는 어떻게 지도하나요?

A2. 처음에는 교사가 한 단계씩 시범을 보이며 따라오게 하고, 익숙한 학생이 어려워하는 학생을 돕는 짝 활동을 활용하면 좋습니다.

Q3. 명령어 블록을 미리 자세히 가르쳐야 하나요?

A3. 이 차시에서는 모든 블록을 익히기보다 '어떤 종류의 블록이 있는지' 둘러보는 수준이면 충분합니다. 구체적인 블록 사용은 다음 차시부터 단계적으로 다룹니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치	화면의 주요 구성 요소 살펴보고 블록 꾸러미의 종류와 기능 확인하기

활용 시 유의점

- 학생 개인 계정 로그인이 필요한 디지털 도구는 되도록 사용하지 않는다.
- 엔트리 접속 시 교사가 미리 안전한 접속 경로를 안내한다.
- 인터넷 사용이 어려운 환경에서는 교사가 준비한 사진 및 영상 자료로 대체한다.

2. 명령어 블록을 조립하여 프로그램을 만들어 볼까요?

- **관련 성취 기준 :** [6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.
- **학습 목표 :** 명령어 블록을 순서대로 조립하여 순차 구조와 반복 구조를 가진 간단한 프로그램을 작성할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 같은 동작을 여러 번 시킬 때 어떤 블록을 사용하면 편할지 생각해 보기 • 학습 문제 확인하기	• 거북이가 자기소개를 하고 순서대로 헤엄치게 하기(순차 구조) • 정해진 횟수만큼 반복하기 (횟수 반복) • 상어를 만나면 도망가도록 어떤 조건이 될 때까지 반복하기 (조건 반복) • 프로그램을 실행하고 결과 확인하기	• 순차 구조와 반복 구조의 의미 정리하기 • 내가 만든 프로그램을 짝과 함께 실행해 보기

교수·학습 방향	지도상의 유의점	수업 준비물
• 블록을 하나씩 조립하고 바로 실행해 결과를 확인하는 습관을 기르도록 한다. • 횟수 반복과 조건 반복이 어떻게 다른지 직접 비교하며 느끼도록 한다.	• 블록의 기능을 모두 외우게 하지 않고, 필요한 블록을 찾아 쓰도록 안내한다. • 한 번에 완성하기보다 단계별로 만들고 실행하며 오류를 확인하도록 지도한다.	• 교사: 교사용 컴퓨터, 엔트리 예시 작품 • 학생: 디지털 디바이스, 활동지

교수·학습 과정

<table>
<tr><th>학습 단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 양치를 할 때 순서를 생각해봅시다.
T : 양치할 때 어떤 순서로 해야 하나요?
S : 칫솔에 치약을 묻힙니다.
S : 이를 닦습니다.
S : 입을 헹굽니다.
S : 칫솔을 앞뒤로 반복해서 움직입니다.
T : 윗니와 아랫니를 닦을 때는 어떤 동작이 반복되나요?
S : 윗니를 닦을 때처럼 칫솔을 앞뒤로 반복해서 움직이고 물로 입을 헹굽니다.
T : 순서가 바뀌면 어떻게 될지 생각해 봅시다.

◆ 학습문제 확인하기
명령어 블록을 순서대로 조립하여 반복 구조를 가진 간단한 프로그램을 만들어 봅시다.</td><td>• 일상생활 속 순서대로 일을 처리하는 사례를 활용해 학생들이 순차 구조와 반복 구조의 필요성을 자연스럽게 떠올리도록 한다.</td></tr>
<tr><td>전개</td><td>◆ 활동1. 거북이 헤엄치게 하기(순차구조)
T : 오늘 사용할 오브젝트를 추가해봅시다.
<table><tr><th>바닷속(3)</th><th>상어(2)</th><th>거북이</th></tr><tr><td></td><td></td><td></td></tr></table>S : 거북이와 바닷속 오브젝트를 추가합니다.
T : 거북이가 자기소개를 하도록 말하기 블록을 추가해 봅시다.
T : 거북이가 헤엄치는 것처럼 보일 수 있게 이동하기 블록을 여러 개 연결해 봅시다.
S : [움직임]-[이동 방향으로 10만큼 움직이기] 블록을 가져와 20만큼 움직이기로 바꿉니다.
S : [흐름]-[2초 기다리기] 블록을 가져옵니다.
S : 움직일 때마다 2초씩 기다리면 지나치게 느리게 움직이는 것 같아 0.5초 기다리기로 숫자를 변경했습니다.
<table><tr><th>알고리즘</th><th>블록</th></tr><tr><td>이동방향으로
일정한 거리만큼
움직이기</td><td></td></tr></table></td><td>• 준비물: 활동지, 스마트기기(엔트리)
엔트리 완성 파일
https://naver.me/xtNnHitN

• 반복 블록 사용 전후를 비교하여 편리함을 느끼도록 한다.

• 반복 횟수에 따라 결과가 달라짐을 확인하게 한다.

• 조건 반복의 의미를 실제 상황과 연결하여 설명한다.</td></tr>
</table>

T : 블록의 순서를 바꾸면 어떻게 될까요?
S : 시작하면 움직인 다음 인사를 하게 됩니다.

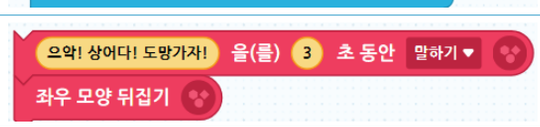
◆ 활동2. 반복 블록으로 더 쉽게 만들기(횟수 반복)

T : 이동하기 블록이 여러 개 사용된 부분을 살펴봅시다.
S : 같은 명령이 반복됩니다.
T : 이동하기 블록을 10번 반복하기 블록 안으로 옮겨 봅시다.
S : 블록 수가 줄어듭니다.

알고리즘	블록
횟수 반복으로 이동방향으로 움직이기	

T : 반복 횟수를 바꾸어 실행해 봅시다.
S : 더 멀리 이동합니다.
S : 더 적게 이동합니다.
T : 정해진 횟수만큼 반복하는 구조를 횟수 반복이라고 합니다.

◆ 활동3. 상어를 만나면 도망가기(조건 반복)

알고리즘	블록
상어에 닿을 때까지 움직이기	
상어를 만나면 반응하기	

T : 상어 오브젝트를 추가해 봅시다.
T : 거북이가 상어를 만나면 어떤 말을 하게 하면 좋을까요?
S : "으악! 도망가자!"
T : 상어에 닿으면 반대 방향으로 움직이도록 프로그램을 만들어 봅시다.
S : [흐름]-[참이 될 때까지 반복하기] 블록을 가져옵니다.
S : [판단]-[마우스 포인터에 닿았는가?] 블록을 가져옵니다.
S : 포인터를 눌러 상어에 닿았는가?로 변경하고 반복하기 블록과 합칩니다.
S : 이동하기 블록을 조건 반복 블록 안에 넣어줍니다.
T : 상어를 만나면 거북이가 반대쪽 벽 끝까지 계속 이동하게 해

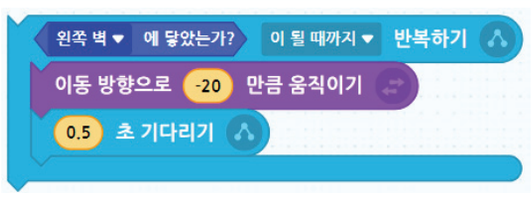
- "벽에 닿을 때까지 반복하기" 블록을 활용하도록 안내한다.
- 상어를 만났을 때 방향을 바꾸어 이동하는 과정을 충분히 시연한다.
- 오류를 발견하면 블록을 수정하고 다시 실행하도록 지도한다.

봅시다.

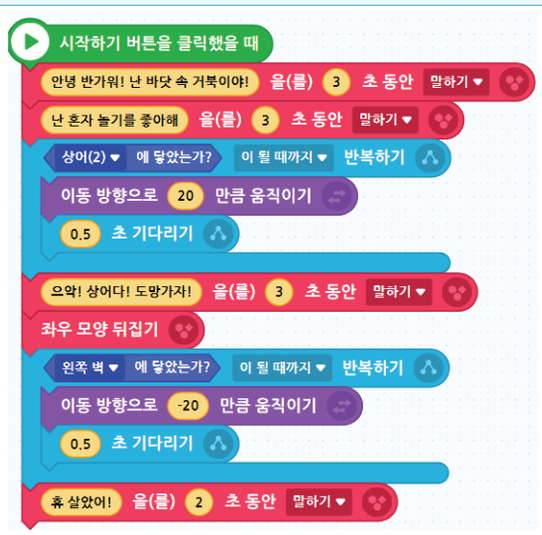
S : 반대 방향으로 이동해야 하므로 숫자를 -20으로 바꿔줍니다.

S : 반대쪽으로 헤엄치는 것처럼 보이게 좌우 모양을 뒤집어 줍니다.

S : 벽에 닿을 때까지 반복합니다.

알고리즘	블록
왼쪽 벽까지 도망가기	

T : 어떤 조건이 만족될 때까지 반복하는 구조를 조건 반복이라고 합니다.

알고리즘	블록
탈출 완료 알리기(완성)	

※ 교과서 33페이지 정답
① 순차 구조
② 반복

정리

◆ 학습 내용 정리하기

T : 오늘 무엇을 배웠나요?

S : 명령어 블록을 조립해 프로그램을 만들었습니다.

S : 순차 구조를 사용했습니다.

S : 횟수 반복을 사용해 프로그램을 간단하게 만들었습니다.

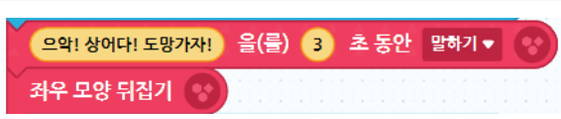
S : 조건 반복을 사용해 상어를 피하도록 만들었습니다.

T : 다음 시간에는 반복 구조를 활용하여 더욱 다양한 프로그램을 만들어 보겠습니다.

수업 활동지

명령어 블록을 조립하여 프로그램을 만들어 볼까요?

1. 완성된 프로그램을 보고 빈칸에 들어갈 말을 적어봅시다.

알고리즘	블록
시작하기 버튼을 클릭했을 때 “안녕 반가워! 난 바닷 속 거북이야. 난 혼자 놀기를 좋아해”를 말하기	
상어에 닿을 때까지 (① _____)	
"으악! 상어다! 도망가자!"를 3초 동안 말하기 좌우 모양 뒤집기	
왼쪽 벽에 닿을 때까지 (② _____) “휴 살았어!”를 2초 동안 말하기	

2. 완성한 프로그램을 실행해보고, 더 발전시키고 싶은 점을 적어봅시다.

수업 평가지

1. 명령이 위에서부터 차례대로 실행되는 구조는 무엇일까요?

① 반복 구조
② 순차 구조
③ 선택 구조
④ 조건 구조

정답 번호: _____

2. 다음 설명에 알맞은 반복 블록을 보기에서 골라 써 봅시다.

〈보기〉
여러 번 반복하기
조건을 달성할 때까지 반복하기

㉠ 정해진 횟수만큼 같은 동작을 되풀이한다. → ( )

㉡ 어떤 조건이 될 때까지 같은 동작을 되풀이한다. → ( )

3. 다음 자연어 알고리즘을 보고, 의사코드 빈칸에 들어갈 알맞은 말을 부등호를 사용하여 적어 봅시다.

[자연어]

1. 시작 버튼을 누르면 "출발!"이라고 말한다.
2. 걸음 수를 0으로 정한다.
3. 걸음 수가 10보다 작은 동안 아래를 반복한다.
   3-1. 1걸음 앞으로 움직인다.
   3-2. 걸음 수에 1을 더한다.
4. 반복이 끝나면 "도착했어요!"를 3초 동안 말한다.

[의사코드]

시작하기 버튼을 클릭했을 때
  “출발!” 을(를) 말하기
  [걸음 수]를 0으로 정하기
  (①) 인 동안 반복하기
    앞으로 1 걸음 움직이기
    [걸음 수] 에 1 만큼 더하기
  “도착했어요!” 을(를) 3 초 동안 말하기

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ②
2. ㉠ ○번 반복하기, ㉡ ○○할 때까지 반복하기
3. ① 걸음 수 < 10

평가 목표

순차 구조와 두 가지 반복 구조(횟수 반복, 조건 반복)의 의미를 이해하고, 명령어 블록을 조립하여 간단한 프로그램을 작성할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	순차 구조와 두 가지 반복 구조를 구분하여 설명하고, 상황에 알맞은 반복을 골라 그 까닭을 정확히 설명할 수 있다.
보통	순차 구조와 반복 구조를 이해하고, 상황에 알맞은 반복을 고를 수 있다.
노력 요함	반복 구조의 의미를 이해하는 데 어려움을 겪는다.

맞춤형 피드백 예시

학생 반응	피드백 예시
횟수 반복과 조건 반복을 헷갈려 하는 경우	'열 번 줄넘기하기'는 횟수가 정해져 있고, '지칠 때까지 줄넘기하기'는 멈출 조건이 정해져 있어요. 거북이가 벽에 닿는 것은 횟수일까요, 조건일까요?
조건 반복이 멈추지 않고 계속되는 경우	'언제 멈출지' 조건을 정해 주었나요? '벽에 닿았는가?'처럼 멈출 조건을 넣어야 반복이 끝나요.

수업 운영 팁

Q1. 오브젝트가 화면 밖으로 나가 버릴 때는 어떻게 지도하나요?

A1. 이동하는 값(예: 10)을 작게 하거나 반복 횟수를 줄이도록 안내합니다. 화면 끝에 닿으면 튕기게 하는 블록을 소개해 줄 수도 있습니다.

Q2. 횟수 반복과 조건 반복을 헷갈려 하는 학생은 어떻게 지도하나요?

A2. '줄넘기를 열 번 한다'는 횟수 반복, '지칠 때까지 줄넘기를 한다'는 조건 반복처럼 생활 속 예로 비교해 주면 차이를 쉽게 이해합니다. 빨리 끝낸 학생에게는 거북이가 벽에 닿을 때까지 헤엄치게 하는 등 조건 반복을 더 활용하도록 안내합니다.

Q3. 블록을 찾기 어려워하는 학생은 어떻게 돕나요?

A3. 블록 꾸러미의 색깔과 블록의 색깔이 같다는 점을 알려 주면 원하는 블록을 빠르게 찾을 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치	순차 구조와 반복 블록을 이용해 오브젝트(엔트리) 또는 스프라이트(스크래치)의 움직임을 구현할 수 있음

활용 시 유의점

- 작품을 저장할 때 개인정보가 드러나는 이름을 사용하지 않도록 한다.
- 한 단계씩 만들고 실행하여 오류를 그때그때 확인하도록 안내한다.

3. 자료를 입력받아 처리하는 프로그램을 만들어 볼까요?

- **관련 성취 기준 :** [6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.
- **학습 목표 :** 자료가 입력·처리·출력되는 과정과 변수의 의미를 이해하고, 변수를 활용하여 자료를 처리하는 프로그램을 작성할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 여러 숫자를 더해야 할 때 컴퓨터가 어떻게 도와줄 수 있을지 이야기하기	• 입력·처리·출력 과정과 변수의 의미 알아보기 • 변수를 만들고 자료를 저장하는 방법 익히기 • 입력받은 값을 더하여 결과를 출력하는 프로그램 만들기	• 입력·처리·출력 과정과 변수의 의미 정리하기 • 내가 만든 프로그램을 실행해 결과 확인하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 일상에서 여러 자료를 모아 계산하는 상황과 연결하여 변수의 필요성을 느끼게 한다. • '대답'에 저장된 값을 변수에 옮겨 저장하는 과정을 차근차근 안내한다.	• 변수의 개념을 어렵게 설명하기보다 '값을 담아 두는 상자'로 비유하여 이해를 돕는다. • 계산 결과가 예상과 다르면 어느 단계의 값이 잘못되었는지 함께 짚어 본다.	• 교사: 교사용 컴퓨터, 엔트리 예시 작품 • 학생: 디지털 디바이스, 활동지

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 계산 프로그램의 필요성 알아보기 T : 일주일 동안 읽은 책의 쪽수를 모두 더하려면 어떻게 하면 좋을까요? S : 숫자를 하나씩 적어 더하면 번거롭고 틀리기 쉽습니다. S : 컴퓨터에게 숫자를 알려 주면 대신 더해 줄 수 있을 것 같습니다. T : 오늘은 엔트리로 독서 기록 프로그램을 만들어 봅시다. ◆ 학습문제 확인하기 자료를 입력 받는 블록을 활용하여 나만의 독서 기록 프로그램을 만들어 봅시다.	• 자판기나 계산기처럼 값을 넣으면 결과가 나오는 경험을 떠올리게 한다.
전개	◆ 활동1. 자료의 입력·출력과 변수 알아보기 T : 독서 기록 프로그램에는 어떤 정보가 필요할까요? S : 책 제목을 넣어야 합니다. S : 오늘 읽은 페이지 수가 필요합니다. S : 지금까지 읽은 총 페이지 수가 필요합니다. T : 독서 기록 프로그램에는 어떤 정보를 입력해야 할까요? S : 읽고 있는 책의 제목과 오늘 읽은 페이지 수를 입력해야 합니다. T : 독서 기록 프로그램은 그 정보를 입력받으면 어떻게 해줄까요? S : 그동안 읽은 페이지 수를 다 더해서 지금까지 읽은 페이지 총수를 알려 줍니다. T : 컴퓨터가 자료를 다루는 과정을 알아봅시다. S : 입력: 사용자가 숫자나 글자 등 자료를 넣는 단계임을 확인한다. S : 처리: 입력된 자료를 정해진 방법에 따라 계산하거나 바꾸는 단계임을 확인한다. S : 출력: 처리한 결과를 화면 등으로 보여 주는 단계임을 확인한다. T : 독서 기록 프로그램이 기억해야 하는 정보는 무엇일까요? S : 책 제목입니다. 누적 페이지 수입니다. T : 프로그램이 기억해야 하는 정보를 저장하는 공간을 변수라고 합니다. 변수는 자료를 저장해 두는 기억 공간으로, 값을 담아 두는 상자에 비유할 수 있습니다.	• 준비물: 활동지, 스마트기기(엔트리) 엔트리 완성 파일 https://naver.me/xeFhhDxM • 변수는 컴퓨터가 기억해야 하는 정보를 저장하는 공간임을 설명한다. • 모든 입력값을 변수에 저장할 필요는 없음을 안내한다.

◆ 활동2. 자료를 입력 받는 프로그램 설계하기

T : 필요한 오브젝트를 추가합니다.

S : 배경과 책을 추가합니다.

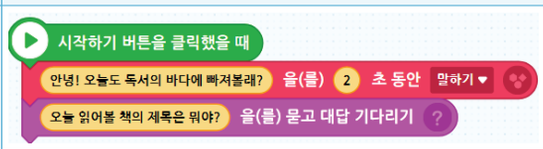
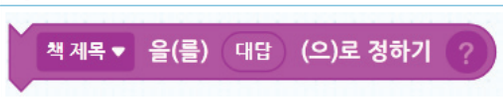
도서관	펼쳐져 있는 책(2)
	

T : 책 제목을 입력 받아 변수에 저장해 봅시다.

S : [자료]-[안녕을(를) 묻고 대답 기다리기] 블록을 가져와 "오늘 읽을 책의 제목은 뭐야?"로 수정합니다.

S : [자료]-[변수 만들기]에서 **책 제목**과 **누적 페이지** 변수를 생성합니다.

S : [자료]-[책 제목을(를) 10으로 정하기] 블록을 가져와 **책 제목**을 [대답] 으로 저장합니다.

알고리즘	블록
책 제목 입력받기	
입력한 값을 변수에 저장하기	

T : 입력한 제목을 활용하여 말풍선을 만들어 봅시다.

S : [계산]에서 더하기 연산 블록을 가져와 **책 제목**과 말풍선을 합쳐서 작성합니다.

S : "**책 제목**을(를) 읽고 있구나! 아주 좋아!를 2초 동안 말합니다.

알고리즘	블록
입력한 정보 출력하기	

◆ 활동3. 독서 기록 기능 추가하기

T : 책 오브젝트를 클릭하면 독서 기록을 시작하도록 만들어 봅시다.

S : 책 오브젝트를 선택합니다.

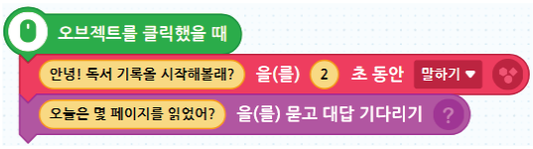
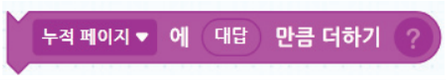
S : [시작]-[오브젝트를 클릭했을 때] 블록을 가져옵니다.

T : 오늘 읽은 페이지 수를 입력받아 봅시다.

S : [자료]-[안녕!을(를) 묻고 대답 기다리기] 블록을 가져와 "오늘은 몇 페이지를 읽었어?"로 수정합니다.

S : 프로그램을 실행하여 페이지 수를 입력해 봅니다.

- '대답'에 저장된 값은 새로 입력하면 바뀌므로, 필요한 값은 변수에 옮겨 저장하도록 안내한다.
- 묻고 대답하기 블록과 변수 저장 블록을 활용하도록 안내한다.
- 입력값이 변수에 저장되는 과정을 확인하게 한다.
- '누적 페이지에 대답만큼 더하기' 블록의 의미를 충분히 설명한다.
- 누적페이지 = 기존 누적페이지 + 새로 읽은 페이지 수라는 개념을 구체적인 숫자로 예시를 들어 설명한다.

알고리즘	블록
페이지 입력 받고 기다리기	
입력한 값을 변수에 저장하기	

T : 입력한 페이지 수를 누적 페이지 변수에 더해 봅시다.
S : [자료]-[**누적 페이지**에 10만큼 더하기] 블록을 가져옵니다.
S : 10 대신 [대답] 블록을 끼워 넣습니다.
S : 입력한 페이지 수가 누적 페이지에 더해지는지 확인합니다.
T : 지금까지 읽은 페이지 수를 말풍선으로 출력해 봅시다.
S : [계산]의 더하기 연산 블록을 이용하여 문장을 만듭니다.
S : "지금까지 ", **누적 페이지**, "페이지 읽었구나! 아주 잘하고 있어!"를 차례로 연결합니다.
S : [생김새]-[○초 동안 말하기] 블록에 끼워 넣습니다.
T : 독서 기록을 계속할 수 있도록 안내 문구를 추가해 봅시다.
S : [생김새]-[○초 동안 말하기] 블록을 추가합니다.
S : "또 기록하고 싶으면 나를 클릭해!"라고 입력합니다.

알고리즘	블록
입력한 값을 누적하고 누적된 결과 출력하기	

T : 프로그램을 여러 번 실행하여 누적 페이지가 계속 증가하는지 확인합니다.

※ 교과서 37페이지 정답
① 처리
② 변수

정리

◆ **학습 내용 정리하기**

T : 오늘 무엇을 배웠나요?
S : 변수를 사용해 정보를 저장했습니다.
S : 책 제목을 입력받았습니다.
S : 읽은 페이지 수를 누적 계산했습니다.
T : **책 제목** 변수와 **누적 페이지** 변수는 어떤 역할을 했나요?
S : 책 제목을 기억해 주었습니다.
S : 지금까지 읽은 페이지 수를 저장해 주었습니다.
T : 다음 시간에는 지금까지 배운 내용을 활용하여 문제를 해결하는 프로그램을 만들어 보도록 하겠습니다.

수업 활동지

자료를 입력받아 처리하는 프로그램을 만들어 볼까요?

1. 보기에서 알맞은 말을 골라 자료를 다루는 과정을 순서대로 써 봅시다.

〈보기〉
처리　　입력　　출력

(　　) → (　　) → (　　)

2. 내가 만든 독서 기록 프로그램에 사용한 변수를 적어 봅시다.

변수 이름	저장하는 값	사용한 곳

3. 프로그램이 실행되는 순서대로 번호를 써 봅시다.



① 누적 페이지에 대답만큼 더한다.
② “오늘은 몇 페이지를 읽었어?”라고 묻는다.
③ “독서 기록을 시작해볼래?”라고 말한다.
④ 지금까지 읽은 페이지 수를 말한다.

정답: _____

수업 평가지

1. 다음 프로그램에서 '책 제목'은 무엇일까요?



① 변수
② 반복 블록
③ 오브젝트
④ 말하기 블록

정답 : _____

2. 빈칸에 알맞은 말을 보기에서 골라 써 봅시다.

〈보기〉
입력, 처리, 출력, 저장, 반복, 조건, 변수, 명령어

컴퓨터는 자료를 (　　)받아 정해진 방법으로 (　　)한 뒤, 그 결과를 (　　)한다.

3. 독서 기록 프로그램에서 '누적 페이지에 대답만큼 더하기' 블록을 사용하는 이유는 무엇일까요?

① 읽은 페이지 수를 계속 더해서 저장하기 위해
② 책 제목을 저장하기 위해
③ 프로그램을 끝내기 위해
④ 말을 크게 하기 위해

정답 : _____

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ①
2. 입력, 처리, 출력
3. ①

평가 목표

입력·처리·출력 과정과 변수의 의미를 이해하고, 변수를 활용하여 자료를 처리하는 프로그램을 작성할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	입력·처리·출력 과정과 변수의 의미를 설명하고, 변수에 값을 누적하는 까닭을 정확히 설명할 수 있다.
보통	입력·처리·출력 과정과 변수의 의미를 이해하고, 변수를 활용하여 자료를 저장할 수 있다.
노력 요함	변수의 의미와 자료 처리 과정을 이해하는 데 어려움을 겪는다.

맞춤형 피드백 예시

학생 반응	피드백 예시
누적 페이지가 더해지지 않고 마지막 값만 남는 경우	'정하기' 블록을 썼는지, '더하기' 블록을 썼는지 살펴볼까요? 값을 쌓으려면 '누적 페이지에 ~만큼 더하기' 블록을 사용해야 해요.
변수의 개념을 어려워하는 경우	변수는 값을 담아 두는 상자예요. 상자에 이름을 붙이고 숫자를 넣어 두면 언제든 다시 꺼내 쓸 수 있답니다.

수업 운영 팁

Q1. 변수를 처음 배우는 학생이 많을 때는 어떻게 시작하나요?

A1. 이름표를 붙인 상자에 물건을 넣고 꺼내는 모습에 빗대어 설명하면 이해가 쉽습니다. 변수 하나로 시작해 익숙해진 뒤 개수를 늘려 갑니다.

Q2. 계산 결과가 예상과 다를 때는 어떻게 지도하나요?

A2. 변수 값을 화면에 보이게 설정하면 각 단계에서 값이 어떻게 바뀌는지 눈으로 확인할 수 있어 오류를 찾기 쉽습니다.

Q3. 추가 활동을 해보고 싶어 하는 학생에게는 어떤 활동을 제안할 수 있나요?

A3. 평균 구하기, 가장 큰 값 찾기처럼 입력받은 값을 다르게 처리하는 프로그램으로 확장해 보도록 안내합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치	변수를 만들어 입력한 자료를 저장하고 값을 변경해보고, 변수에 저장된 값을 이용해 결과를 출력하는 프로그램 작성

활용 시 유의점

- 변수 값을 화면에 표시해 두면 처리 과정을 확인하기 쉽다.
- 입력 단계에서 숫자가 아닌 값을 넣으면 오류가 날 수 있음을 안내한다.

4. 문제를 해결하는 프로그램을 만들어 볼까요?

- **관련 성취 기준 :** [6실05-02] 컴퓨터에게 명령하는 방법을 체험하고, 주어진 문제를 해결하는 프로그램을 작성한다.
- **학습 목표 :** 문제를 해결하기 위한 프로그램의 알고리즘을 설계하고, 주어진 문제를 해결하는 프로그램을 작성할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 하루 동안 물을 얼마나 마시는지 생각해 본 적이 있는지 이야기하기 • 물을 충분히 마셨는지 쉽게 알 수 있는 방법 생각해 보기 • 학습 문제 확인하기	• 물 마시기 목표 관리 문제를 이해하고 필요한 요소(입력·변수·판단·출력) 분석하기 • 문제 해결 방법을 순서대로 말로 표현하여 알고리즘을 설계하기 • 프로그램을 실행하고 점검하여 수정·보완하기	• 문제 해결 과정 정리하기 • 완성한 프로그램을 짝과 함께 실행하고 의견 나누기

교수·학습 방향	지도상의 유의점	수업 준비물
• 문제 이해, 알고리즘 설계, 프로그래밍, 실행·점검의 과정을 차례대로 경험하게 한다. • 조건에 따라 실행 결과가 달라지는 모습과 선택 구조를 연결 지어 이해하도록 한다.	• 프로그래밍 산출물의 완성도보다 문제 해결의 과정과 사고력에 중점을 두어 지도한다. • 목표가 달성 될 때까지 다시 입력받는 부분에서 조건 반복을 함께 다룬다. • 변수 값이 어떻게 변하는지 화면의 변수 창을 보며 확인하도록 한다.	• 교사: 교사용 컴퓨터, 엔트리 예시 작품 • 학생: 디지털 디바이스, 활동지

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 물 마시기 프로그램이 필요한 이유 생각해보기 T : 하루 동안 물을 얼마나 마시는지 생각해 본 적이 있나요? S : 물을 자주 마시는 편입니다. S : 몇 번을 마셨는지 세어 본 적은 없습니다. T : 물을 충분히 마셨는지 쉽게 알 수 있는 방법이 있을까요? S : 목표를 정하고 기록하면 됩니다. S : 마신 양을 계산해 보면 됩니다. T : 오늘은 엔트리로 물 마시기 목표를 관리해 주는 프로그램을 만들어 봅시다.	• 학생들이 건강한 생활 습관과 프로그램 활용을 연결하여 생각할 수 있도록 한다. 목표 컵 수의 기준값은 예시이며, 학생의 상황과 학급 여건에 따라 교사가 조정하여 안내할 수 있다.
	◆ 학습문제 확인하기 문제를 해결하기 위한 프로그램의 알고리즘을 설계하고 물 마시기 목표를 관리하는 프로그램을 만들어 봅시다.	
전개	◆ 활동1. 문제 이해하고 분석하기 T : 해결할 문제를 살펴봅시다. 문제 상황: 하루 동안 마실 물의 목표 컵 수를 정하고, 물을 마실 때마다 기록하여 목표를 달성했는지 알려 주는 프로그램이 필요하다. T : 문제 해결에 필요한 요소를 찾아봅시다. S : 입력받을 것: 목표 컵 수 S : 저장할 것: 목표 컵 수, 현재 컵 수 S : 판단할 것: 현재 컵 수가 목표 컵 수에 도달했는지 S : 알려 줄 것: 목표 달성 여부와 남은 컵 수 ◆ 활동2. 알고리즘 설계하기 T : 문제 해결 방법을 순서대로 말로 나타내어 봅시다. 예) ① 오늘의 물 섭취 목표를 입력받는다. ② 목표가 8컵 미만이면 다시 입력받는다. ③ 목표컵수 변수에 저장한다. ④ 물을 마실 때마다 물병을 클릭한다. ⑤ 현재컵수를 1 증가시킨다. ⑥ 현재컵수가 목표컵수와 같은지 확인한다. ⑦ 목표를 달성했으면 축하 메시지를 출력한다. ⑧ 목표를 달성하지 못했으면 남은 컵 수를 알려 준다.	• 준비물: 활동지, 스마트기기(엔트리) 엔트리 완성 파일 https://naver.me/GUTlcWTi • 문제 해결 과정을 말로 먼저 표현한 뒤 프로그램으로 옮기도록 한다.

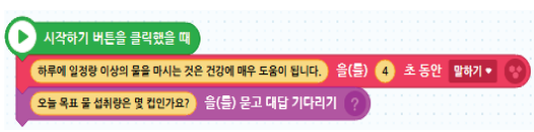
◆ 활동3. 물 마시기 목표 관리 프로그램 제작하기

파란 방	물병	의사
		

T : 프로그램에 사용할 변수를 만들어 봅시다.

- [자료]-[변수 만들기]를 선택합니다.
- 목표컵수와 현재컵수 변수를 생성합니다.
- 현재컵수는 오늘 마신 물의 양을 저장하고, 목표컵수는 오늘의 물 섭취 목표를 저장하는 데 사용합니다.

T : 의사 오브젝트에 목표를 입력받는 프로그램을 작성해 봅시다.

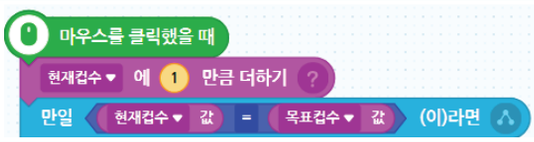
알고리즘	블록
오늘의 물 섭취 목표	
조건을 만족할 때까지 다시 입력받기	
조건을 만족하면 목표를 변수에 저장하기 (완성)	

- [자료]-[묻고 대답 기다리기] 블록을 가져와 “오늘 목표 물 섭취량은 몇 컵인가요?”로 수정합니다.
- [흐름]-[만일 ~이라면, 아니면] 블록에 조건을 넣습니다.
- 만약 입력한 목표량이 8컵 보다 적으면 “하루 섭취량이 너무 적어요! 목표를 더 늘려봐요!”를 말하도록 하고 [흐름]-[○○이 될 때까지 반복하기] 블록을 사용하여 목표가 8컵 이상이 될 때까지 다시 입력받도록 만듭니다.

- 남은 컵 수 계산에 목표컵수-현재컵수를 사용하도록 안내한다.
- 완성 여부보다 수정 과정을 통해 문제를 해결하는 경험에 중점을 둔다.

T : 입력한 목표를 변수에 저장해 봅시다.
- 목표컵수 변수에 입력한 값을 저장합니다.

T : 물병 오브젝트를 클릭하면 물을 마신 것으로 기록하도록 만들어 봅시다.

알고리즘	블록
현재 컵수랑 목표 컵수가 같은지 비교하기	

T : 목표를 달성했는지 판단하는 조건을 만들어 봅시다.
- [흐름]-[만일 ~이라면, 아니면] 블록을 가져옵니다.
- [판단]-[현재컵수 = 목표컵수] 블록을 조건 부분에 넣습니다.
- 현재컵수가 목표컵수와 같아지면 목표를 달성한 것으로 판단합니다.

T : 목표를 달성한 경우의 안내 문구를 작성해 봅시다.
- [생김새]-[○초 동안 말하기] 블록을 가져옵니다.
- "축하해요! 오늘 목표를 달성했어요!"라고 입력합니다.

T : 목표를 달성하지 못한 경우의 안내 문구를 작성해 봅시다.
- [계산]-[빼기] 연산 블록을 가져와 목표컵수에서 현재컵수를 뺍니다.
- [계산]-[더하기] 연산 블록을 사용하여 문장을 만듭니다.
- "잘했어요! 앞으로 ○컵 남았어요!" 말풍선을 띄웁니다.

알고리즘	블록
목표 달성 여부에 따라 출력하는 메시지 다르게 하기	

T : 프로그램을 반복 실행해 보면서 수정할 곳은 없는지 검토합니다.
- 5를 입력하면 다시 입력을 요구하는지 확인한다.
- 8을 입력하면 정상적으로 저장되는지 확인한다.

T : 원하는 대로 동작하지 않으면 원인을 찾아 수정해 봅시다.

※ 교과서 41페이지 정답
① 선택
② 목표

정리

◆ 학습 내용 정리하기

T : 오늘 무엇을 배웠나요?
- 문제를 분석하고 알고리즘으로 표현하는 방법을 배웠습니다.
- 선택 구조를 활용하여 프로그램을 만들었습니다.
- 조건에 따라 다른 결과를 출력하는 프로그램을 만들었습니다.

T : 이 프로그램에 어떤 기능을 더 추가하면 좋을까요?
- 하루 목표 달성 횟수를 기록하면 좋겠습니다.
- 물병 모양이 채워지게 만들면 좋겠습니다.

수업 활동지

문제를 해결하는 프로그램을 만들어 볼까요?

1. 물 마시기 목표 관리 프로그램을 만들기 위해 필요한 요소를 알맞게 골라 써 봅시다.

〈보기〉
목표 물 섭취량, 목표컵수, 현재컵수, 목표 달성 여부, 남은 컵 수, 오늘 날짜, 좋아하는 음료

입력받을 것 :
저장할 것 :
판단할 것 :
알려 줄 것 :

2. 다음 프로그램 코드를 보고 실행 결과를 예측해봅시다.



"오늘 목표 물 섭취량은 몇 컵인가요?" 묻고 대답하기에 입력한 순서입니다.

입력: 6 → 7 → 8

다음과 같이 입력했을 때 위의 프로그램이 어떻게 동작할지 쓰세요.

① "하루 섭취량이 너무 적어요!"라는 안내는 몇 번 나올까요? _____

② 마지막에 목표컵수에는 얼마가 저장될까요? _____

수업 평가지

1. 다음 중 물 마시기 프로그램에서 '목표컵수' 변수에 저장되는 것은 무엇입니까?

① 현재까지 마신 물의 양
② 사용자가 입력한 목표 물 섭취량
③ 물병을 클릭한 횟수
④ 남은 물의 양
⑤ 프로그램이 말하는 문장

정답 : _____

2. 다음 빈칸에 들어갈 알맞은 말을 쓰세요.



정답 : ① _____ ② _____

3. 다음 프로그램에서 목표 물 섭취량을 다시 입력받도록 하는 경우는 언제입니까?



① 입력한 물 마시기 목표가 8컵 이상일 때
② 입력한 물 마시기 목표가 8컵보다 작을 때
③ 물병을 클릭했을 때
④ 현재컵수가 목표컵수와 같을 때
⑤ 프로그램이 시작될 때마다

정답 : _____

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ②
2. ① 목표컵수　② 현재컵수
3. ②

평가 목표

물 마시기 목표 관리 프로그램을 분석하여 변수와 선택 구조의 역할을 이해하고, 프로그램의 실행 결과를 예측할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	프로그램에서 변수와 선택 구조의 역할을 정확하게 이해하고, 실행 결과를 올바르게 예측할 수 있다.
보통	변수와 선택 구조의 역할을 부분적으로 이해하며, 도움을 받아 프로그램의 실행 결과를 설명할 수 있다.
노력 요함	변수와 선택 구조의 역할을 이해하는 데 어려움을 보이며, 프로그램의 실행 결과를 예측하는 데 어려움을 겪는다.

맞춤형 피드백 예시

학생 반응	피드백 예시
목표컵수와 현재컵수의 역할을 혼동하는 경우	목표컵수는 사용자가 처음 입력한 목표를 저장하고, 현재컵수는 물을 마실 때마다 증가해요. 두 변수가 각각 어떤 값을 저장하는지 다시 확인해 볼까요?
목표 달성 여부 안내가 똑같이 나오는 경우	'만일 ~이라면, 아니면' 블록에서 조건이 참일 때와 거짓일 때 자리에 각각 다른 말하기 블록을 넣었는지 확인해 볼까요?

수업 운영 팁

Q1. 변수와 조건문의 역할을 어려워하는 학생은 어떻게 지도하나요?

A1. 학생들이 목표컵수와 현재컵수를 헷갈리지 않도록 실제 예를 들어 설명합니다. 예를 들어 목표를 8컵으로 정하고 물을 한 컵씩 마실 때마다 현재컵수를 칠판에 기록하며 두 변수의 값이 어떻게 달라지는지 함께 확인합니다. 또한 조건문은 '조건이 맞으면 ○○, 아니면 △△'의 흐름을 말로 먼저 설명한 뒤 블록 프로그램과 연결하여 이해하도록 지도합니다.

Q2. 컴퓨터실이 부족하거나 기기가 모자랄 때는 어떻게 운영하나요?

A2. 두세 명이 한 기기를 함께 쓰며 역할을 나누는 짝·모둠 프로그래밍으로 운영할 수 있습니다. 한 명은 블록을 조립하고 다른 한 명은 알고리즘을 확인하는 식으로 번갈아 진행합니다.

Q3. 학생마다 진도 차이가 클 때는 어떻게 하나요?

A3. 기본 과제를 먼저 완성하게 한 뒤, 빠른 학생에게는 오브젝트를 더 추가하거나 조건을 늘리는 확장 과제를 제시합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치	선택 구조와 반복 구조를 활용하여 변수의 값을 저장 또는 변경하고 목표 달성 여부를 판단하는 프로그램 작성

활용 시 유의점

- 조건에 사용하는 글자는 입력값과 정확히 일치해야 함을 안내한다.
- 작품을 공유할 경우 개인정보가 드러나지 않도록 지도한다.

3단원

함께 만드는 생활 속 프로그램



1 단원 개관

이 단원에서는 우리 생활 속에서 프로그래밍으로 해결할 수 있는 문제를 스스로 찾고, 직접 프로그램을 설계하고 완성하는 전 과정을 경험한다. 이를 위해 학생들은 먼저 실생활에서 불편함이나 필요를 느끼는 상황을 탐색하고, 이를 프로그래밍으로 해결할 수 있는 문제로 재정의하는 활동을 한다. 이후 알고리즘으로 해결 절차를 표현하고, 이를 바탕으로 프로그램을 직접 구현하며 수정·보완하는 과정을 반복한다. 또한 완성한 프로그램을 친구들과 공유하고, 서로의 프로그램을 탐색하며 좋은 점과 개선할 점을 나누는 활동을 통해 프로그램을 더 유용하게 발전시키는 경험을 한다. 이러한 일련의 과정을 통해 학생들은 단순히 주어진 문제를 푸는 것을 넘어, 스스로 문제를 발견하고 창의적으로 해결하는 능동적인 문제 해결자로 성장하게 된다.

2 단원 학습 연계

이전에 배운 내용	이번에 배울 내용	다음에 배울 내용
• 컴퓨터의 개념 • 컴퓨터에게 명령하는 방법	• 문제 찾기와 문제 해결 절차 • 문제를 해결하는 기초적인 프로그래밍하기 • 프로그래밍으로 만든 산출물을 타인과 공유하고 협력하려는 자세	• 문제 추상화 • 핵심요소를 알고리즘으로 표현

3 단원 구성

차시	관련 성취기준	차시명
1	[06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	• 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾아볼까요?
2	[06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	• 생활 속 문제를 프로그래밍으로 해결하기 위한 알고리즘을 표현해볼까요?
3	[06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	• 프로그래밍으로 나에게 필요한 프로그램을 만들어 볼까요?
4	[06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.	• 나에게 필요한 프로그램을 발전시켜 볼까요?
5	[6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.	• 생활 속 문제를 해결하는 방법을 알고리즘으로 표현해 볼까요?
6	[6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.	• 알고리즘에 맞게 프로그램을 완성해 볼까요?
7	[06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.	• 친구의 프로그램을 탐색하고 의견을 나눠볼까요?
8	[06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.	• 공유된 프로그램을 더 유용하게 수정해볼까요?

4 단원 학습목표

지식·이해	• 생활 속 문제를 찾고 문제를 해결하는 절차를 순서대로 나타낼 수 있음을 안다. • 문제를 해결하는 기초적인 프로그래밍 방법과 과정을 안다.
과정·기능	• 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾을 수 있다. • 실생활 문제를 찾고 해결을 위한 아이디어를 알고리즘으로 표현할 수 있다. • 실생활 문제 해결을 위한 프로그램을 설계할 수 있다. • 실생활 문제 해결을 위한 프로그램을 지속적으로 발전시킬 수 있다.
가치·태도	• 다양한 실생활 속 문제를 탐색하고, 프로그래밍으로 해결하려는 적극적인 자세를 기른다. • 프로그래밍 산출물을 타인과 공유하고 협력하려는 자세를 갖는다.

1. 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾아볼까요?

- **관련 성취 기준 :** [06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.
- **학습 목표 :** 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾을 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 우리 주변에 프로그램으로 문제를 해결하는 사례 나누기 • 프로그램으로 문제가 해결된 실제 사례 살펴보기	• 키워드별 문제 해결 사례 탐색하기 • 문제 해결 사례 모으기 • 나만의 문제 정하기	• 우리가 찾은 문제들 정리하기 • 다음 차시 미리보기

교수·학습 방향	지도상의 유의점	수업 준비물
• 학생의 실생활 경험에서 출발하여 사례를 자연스럽게 끌어낼 수 있도록 안내한다. • 키워드를 중심으로, 다양한 사례와 생각을 자유롭게 나누는 탐색 중심의 활동으로 운영한다. • 모둠 활동을 통해 서로 다른 관점에서 문제를 발견하고, 혼자보다 함께 생각할 때 더 다양한 아이디어가 나온다는 것을 경험하게 한다.	• 사례를 탐색할 때, 프로그램의 목적 및 도움이 되는 대상을 살펴보며 문제 해결이 많은 사람에게 도움이 됨을 알게 한다. • 스마트 기기를 활용할 경우 검색 목적과 사용 시간을 명확히 안내하여 효율적으로 활용할 수 있게 한다. • 학생이 정한 문제가 다음 차시로 자연스럽게 이어질 수 있도록, 너무 복잡하거나 프로그램으로 구현하기 어려운 문제는 방향을 조정해준다.	• 교사: 실생활 문제 해결 사례, 문제 해결 사례 탐색 키워드 • 학생: 활동지, 스마트 기기, 필기도구

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 프로그램으로 문제를 해결한 사례 나누기 T : 우리 주변에 컴퓨터 프로그램으로 문제를 해결한 사례가 무엇이 있을까요? S : 알림장 알림 앱이 있습니다. S : 실시간 길을 안내하는 내비게이션이 있습니다. ◆ 프로그램으로 문제가 해결된 실제 사례 살펴보기 T : 컴퓨터 프로그램으로 문제를 해결한 실제 사례를 함께 살펴봅시다. 예) 시간관리 앱: 하루의 루틴 만들기를 돕는 앱 예) 용돈관리 프로그램: 내가 쓴 금액을 자동으로 계산해서 앞으로의 용돈 관리 방향을 안내해주는 프로그램 등 ◆ 학습문제 확인하기 프로그래밍으로 해결할 수 있는 생활 속 문제를 찾을 수 있다.	• 다양한 예시를 학생의 경험을 통해서 나누게 한 뒤, 1-2가지 실제 사례를 공유한다.
전개	◆ 활동1. 키워드별 문제 해결 사례 탐색하기 T : 모둠 친구들과 함께 키워드별 프로그램(앱, 소프트웨어 등)으로 문제를 해결한 다양한 사례를 살펴보고, 활동지에 적어봅시다. 예) 학교 생활: 알림장 앱, 원격 화상 프로그램 예) 건강: 만보기 앱, 수면 측정 앱, 복약 알림 앱 예) 안전: 어린이 귀가 알림 앱, 지진 대비 안내 시스템 ◆ 활동2. 문제 해결 사례 모으기 T : 모둠별로 찾은 사례 중 1-2가지를 소개해봅니다. T : 친구들의 발표를 들으며 활동지에 내가 해결하고 싶은 문제 후보를 적어봅시다. ◆ 활동3. 나만의 문제 정하기 T : 나는 어떤 생활 속 문제를 해결하고 싶은가를 떠올려 활동지에 작성해봅시다. T : 가장 만들고 싶은 문제를 1개 선택하여 프로그램의 이름, 해결하고 싶은 문제, 대상 등을 작성해봅시다. T : 나만의 문제를 담은 프로그램 내용을 모둠 친구들과 함께 이야기해 봅시다.	• 준비물: 활동지 • 키워드(학교 생활, 건강, 안전 등)를 모둠별로 배분하여 사전에 탐색 범위를 나눌 수 있다. • 필요한 경우 스마트 기기를 활용하여 활동지를 작성할 수 있게 한다. • 문제 해결에 도움이 되는 대상을 항상 함께 떠올리게 하여, 더 많은 사람에게 도움이 될 수 있는 문제로 발전시키도록 안내한다. ※ 교과서 47페이지 정답 ① 문제 ② 누구에게 도움이 되는지
정리	◆ 우리가 찾은 문제들 정리하기 T : 오늘 우리가 찾은 문제에 관해 이야기 해봅시다. S : 우리 생활 속에 프로그램으로 해결할 수 있는 다양한 문제가 있다는 것을 알았습니다. ◆ 다음 차시 미리보기 T : 다음 시간에는 오늘 정한 문제를 어떻게 해결할지 다양하게 표현해 봅시다.	

수업 활동지

프로그램으로 해결할 수 있는 생활 속 문제를 찾아볼까요?

1. 우리 모둠의 탐색 키워드를 적어봅시다

( )

2. 해당 키워드와 관련된 프로그램(앱, 소프트웨어 등) 사례를 찾아 적어봅시다.

프로그램 이름	어떤 문제를 해결했나요?	누구에게 도움이 되나요?
예) 타이머 앱	*예) 시간이 얼마나 지났는지 몰라 공부나 휴식시간을 조절하기 어려운 문제를 해결*	*예) 시간 관리가 필요한 학생과 집중력 조절이 필요한 모든 사람에게 도움이 됩니다.*

3. 내가 해결하고 싶은 문제들을 적어봅시다.

①:

②:

③:

4. 내가 가장 해결하고 싶은 문제 1가지를 골라서 아래 표를 작성해봅시다.

프로그램 이름	어떤 문제를 해결하고 싶나요?	누구에게 도움이 될까요?

수업 평가지

1. 다음 중 프로그래밍으로 해결하고 있는 생활 속 문제 사례에 해당하는 것을 3가지 골라봅시다.

① 키오스크로 급식 메뉴 확인하기
② 종이 알림장에 손으로 준비물 적기
③ 만보기 앱으로 하루 걸음 수 확인하기
④ 지진 대피 안내 앱으로 대피 경로 확인하기
⑤ 부모님께 직접 여쭤보아 오늘 일정 확인하기

정답 : _____

2. 다음 문제 상황을 읽고, 아래 문제를 해결해 줄 수 있는 프로그램을 생각해봅시다.

문제 상황: "나는 아침마다 준비물을 깜빡해서 학교에 가서 당황한 적이 많아.
가방을 싸기 전에 미리 알려주는 프로그램이 있으면 좋겠어."

항목	내용
해결하려는 문제	
프로그램 이름	
누구에게 도움이 되나요?	
프로그램이 하는 일	

3. 오늘 수업에서 내가 직접 정한 나만의 프로그램을 소개해봅시다.

항목	내용
해결하려는 문제	
프로그램 이름	
누구에게 도움이 되나요?	
프로그램이 하는 일	

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ①, ③, ④
2. 예) 해결하려는 문제: 아침마다 준비물을 깜빡하는 문제
   프로그램 이름: 준비물 알림 앱
   누구에게 도움이 되나요?: 준비물을 자주 잊어버리는 학생
   프로그램이 하는 일: 전날 밤이나 등교 전에 준비물 목록을 알림으로 알려준다.
3. 예) 해결하려는 문제: 방 청소를 자꾸 미루는 문제
   프로그램 이름: 청소 도우미 앱
   누구에게 도움이 되나요?: 청소를 미루거나 잊어버리는 어린이
   프로그램이 하는 일: 청소할 시간이 되면 알림을 주고 할 일 목록을 보여주면 스스로 청소할 수 있게 도와준다.

평가 목표

생활 속에서 프로그래밍으로 해결할 수 있는 문제를 구분하고, 실생활 문제를 프로그램으로 해결하는 아이디어를 표현할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	프로그래밍으로 해결되는 사례를 정확히 구분하고, 실생활 문제를 명확히 찾아 해결 대상과 프로그램의 역할을 구체적으로 표현할 수 있다.
보통	프로그래밍으로 해결되는 사례를 대체로 구분하고, 실생활 문제와 프로그램 이름을 쓸 수 있으나 프로그램의 역할 설명이 다소 간단하다.
노력 요함	프로그래밍으로 해결되는 사례 구분에 어려움을 겪으며, 나만의 문제를 찾거나 표현하는 데 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
1번 문제에서 ②, ⑤를 정답으로 고른 경우	“컴퓨터 프로그램이 하는 일인지, 사람이 직접 하는 일인지 다시 한 번 구분해볼까요?”
프로그램으로 해결할 수 있는 문제를 찾지 못한 경우	“오늘 아침에 불편했던 일이나, 자주 잊어버리는 것이 있었나요? 그것부터 떠올려봐요.”

수업 운영 팁

Q1. 학생들이 프로그래밍으로 해결되는 사례와 그렇지 않은 사례를 구분하기 어려워하면 어떻게 하나요?

A1. 컴퓨터가 스스로 하는 일인지, 사람이 직접 하는 일인지를 기준으로 생각하게 해보세요. 예를 들어 "내비게이션은 누가 길을 알려주나요? 사람인가요, 프로그램인가요?"처럼 친근한 질문으로 기준을 자연스럽게 잡아줄 수 있습니다. 사례를 먼저 함께 분류해보는 전체 활동을 도입에서 충분히 진행한 뒤 개별 활동으로 넘어가면 훨씬 수월합니다.

Q2. 학생들이 나만의 문제를 정하지 못하고, 더 나아가지 못할 때 어떻게 지도하나요?

A2. 오늘 아침에 불편했던 일, 자주 잊어버리는 것, 하기 귀찮은 일 등 일상 경험에서 출발하도록 구체적인 질문을 던져주세요. 모둠 내에서 서로 불편한 점을 먼저 이야기하게 한 뒤, 그 중 하나를 고르는 방식으로 진행하면 혼자 고민하는 것보다 훨씬 쉽게 문제를 찾을 수 있습니다.

Q3. 너무 복잡하거나 구현하기 어려운 문제를 정한 학생은 어떻게 안내하나요?

A3. 문제 자체를 부정하기보다 그 중에서 가장 핵심적인 기능 한 가지만 고른다면 무엇인지 범위를 좁혀주세요. 예를 들어, "자율주행 자동차 만들기"를 원하는 학생에게는 "그 자동차가 하는 일 중 딱 하나만 고른다면?"처럼 질문하여 스스로 범위를 조정하게 도와줍니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
검색 도구	네이버, 구글	키워드별 프로그래밍 사례 탐색
문서·기록 도구	구글 문서, 패들렛, 띵커벨	활동지 디지털 작성
공유·발표 도구	패들렛, 띵커벨, 퀴즈앤 보드	모둠별 사례 발표 및 공유

활용 시 유의점

- 검색 활동 전, 교사가 미리 안전한 검색어와 참고 사이트 목록을 안내하고 검색 시간을 명확히 정해둔다.
- 스마트 기기 사용이 어려운 환경이라면 교사가 사전에 출력한 사례 자료 카드를 대신 활용할 수 있다.

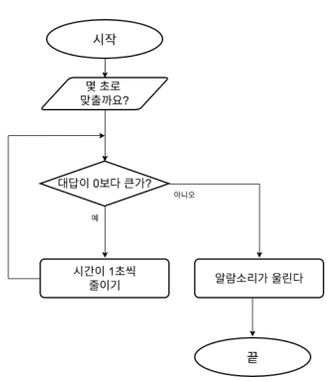
2. 생활 속 문제를 프로그래밍으로 해결하기 위한 알고리즘을 표현해볼까요?

- **관련 성취 기준 :** [06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.
- **학습 목표 :** 프로그래밍으로 해결할 수 있는 생활 속 문제를 알고리즘으로 표현할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 전 차시 학습 내용(알고리즘) 떠올리기 • 학습문제 확인하기	• 문제 해결 과정을 글로 표현하기 • 문제 해결 과정을 그림(순서도)으로 표현하기 • 친구들과 알고리즘 공유하고 피드백 나누기	• 친구의 피드백 정리하기 • 다음 차시 안내하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 알고리즘을 글, 그림, 순서도 등 다양한 방법으로 표현해보게 하여, 같은 문제도 여러 방식으로 표현할 수 있음을 경험하게 한다. • 친구와의 공유 활동을 통해 자신의 알고리즘을 점검하고 보완할 수 있는 기회를 제공한다.	• 학생들에게 가장 친숙한 글로 먼저 알고리즘을 표현하게 한 뒤, 그림이나 순서도로 표현을 확장하도록 지도한다. 필요한 경우 차시를 추가하여 운영할 수 있다. • 교사의 순회 지도와 더불어, 또래 교사를 활용하여 학생 간 상호 도움이 이루어지도록 지도한다. • 친구의 의견을 다음 차시 프로그래밍 과정에서 문제 해결 방법을 개선하는 데 참고 자료로 활용하도록 한다.	• 교사: 알고리즘 표현 예시자료 (글, 그림, 순서도) • 학생: 활동지, 필기도구

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 전 차시 내용 떠올리기 T : 전 차시에서 배운 알고리즘에 대해 떠올려봅시다. S : 알고리즘이란 주어진 문제를 해결하기 위해 필요한 방법을 순서대로 나열한 것입니다. S : 알고리즘은 글, 그림, 순서도 등으로 다양하게 표현할 수 있습니다. T : 지난 시간에 내가 정한 문제와 프로그램 이름을 어떻게 알고리즘으로 표현할 수 있을지 생각해봅시다. ◆ 학습문제 확인하기 프로그래밍으로 해결할 수 있는 생활 속 문제를 알고리즘으로 표현할 수 있다.	
전개	◆ 활동1. 나의 문제 해결과정을 알고리즘으로 표현하기 T : 내가 정한 문제의 해결 과정을 자연어로 표현해봅시다. 예) 타이머 앱: ① 시간을 몇 초로 맞출지 입력한다. ② 대답이 0보다 크면 시간이 줄어든다. ③ 시간이 0이 되면 알람소리가 울린다. 예) 오늘의 명언 출력기 앱: ① 명언 목록 중 하나를 무작위로 고른다. ② 고른 명언을 화면에 글자로 보여준다. ③ 다시 누르면 다른 명언이 랜덤으로 나온다. T : 문제 해결 과정을 의사코드나 순서도로 표현해봅시다. 시작하기 버튼을 클릭했을 때 "몇 초로 맞출까요?"라고 묻기 [대답] 〉 0인 동안 반복하기 만약 [대답] 〉 0 이면 시간을 1초씩 줄이기 아니면 알람 소리가 울린다. 끝  ◆ 활동2. 친구들과 공유하기 T : 짝 또는 모둠 친구에게 내가 표현한 알고리즘을 설명해봅시다. T : 친구의 설명을 듣고 칭찬 또는 제안을 친구의 활동지에 적어 전달합니다. S : "가장 좋았던 점은 ___이야. ___기능이 추가되면 더 좋을 것 같아."	• 준비물: 활동지, 포스트잇 • 학생들에게 가장 친숙한 글로써 알고리즘을 먼저 표현하게 한다. • 교사의 순회 지도와 더불어, 또래 교사를 활용하여 학생 간 상호 도움이 이루어지도록 지도한다. • 활동지의 친구 의견을 추후 문제 개선을 위한 과정에 참고하도록 한다. ※ 교과서 51페이지 정답 ① 알고리즘 ② 순서도
정리	◆ 친구의 피드백 정리 및 다음 차시 안내하기 T : 친구에게 받은 피드백을 읽고, 스스로 알고리즘을 돌아보는 시간을 가져봅시다. T : 다음 시간에는 실생활 속 문제를 엔트리를 통해 직접 프로그래밍 해봅시다.	

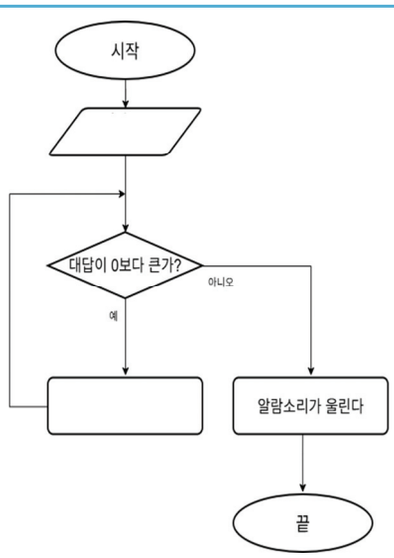
수업 활동지

생활 속 문제를 프로그래밍으로 해결하기 위한 알고리즘을 표현해볼까요?

1. 내가 정한 문제 해결 과정을 자연어로 표현해 봅시다. 선택 1 또는 선택 2를 고른 후, 빈칸에 알맞은 단어를 채우거나 자연어로 완성해 봅시다.

선택 1. 타이머 앱 (자연어)	선택 2. 내가 정한 문제 해결 과정 (자연어)
① (　　)을 누르면 시간을 몇 초로 맞출지 묻고 답한다. ② 대답이 0보다 큰지 확인한다. ③ 대답이 0보다 크면 시간이 (　　)씩 줄어든다. ④ 그렇지 않으면 알람소리가 울린다.	

2. 선택 1, 2 또는 3 중 하나를 선택하여, 의사코드와 순서도 빈칸에 알맞은 단어를 채우거나 의사코드 또는 순서도로 나타내 봅시다.

선택 1. 타이머 앱(의사코드)	선택 2. 타이머 앱(순서도)	선택 3. 내가 정한 문제 해결 과정 (의사코드 또는 순서도)
시작하기 버튼을 클릭했을 때 "몇 초로 맞출까요?"라고 묻고 답하기 [대답] > 0인 동안 반복하기 만약 (　　)이면 시간이 1초씩 줄이기 아니면 (　　) 끝		

3. 친구들에게 나의 설계 과정을 설명해보고, 다양한 의견을 적어봅시다.

이름:	이름:	이름:
의견:	의견:	의견:

수업 평가지

1. 아래 '무작위 메뉴 추천 앱'의 알고리즘 순서가 뒤섞여 있습니다. 올바른 순서가 되도록 번호를 써봅시다.

단계	내용	순서
가	프로그램이 추천한 메뉴가 화면에 나타난다.	( )
나	목록 입력을 마친 후 '추천받기' 버튼을 누른다.	( )
다	먹고 싶은 메뉴 목록을 자유롭게 입력한다.	( )
라	프로그램이 입력된 메뉴 목록 중 하나를 무작위로 고른다.	( )

2. 아래는 '오늘의 점수 칭찬 앱'을 알고리즘으로 표현한 것입니다. 아래 빈칸에 알맞은 단어를 넣어 알고리즘을 완성해봅시다.

[오늘의 점수 칭찬 앱]

[자연어]
① ( )를 묻는다.
② 점수가 80점 이상이면 "목표 달성! 아주 잘했어요!"라고 말한다.
③ 아니면 "다음엔 더 잘할 수 있어요!"라고 말한다.
④ 끝

[의사코드]
시작하기 버튼을 클릭했을 때

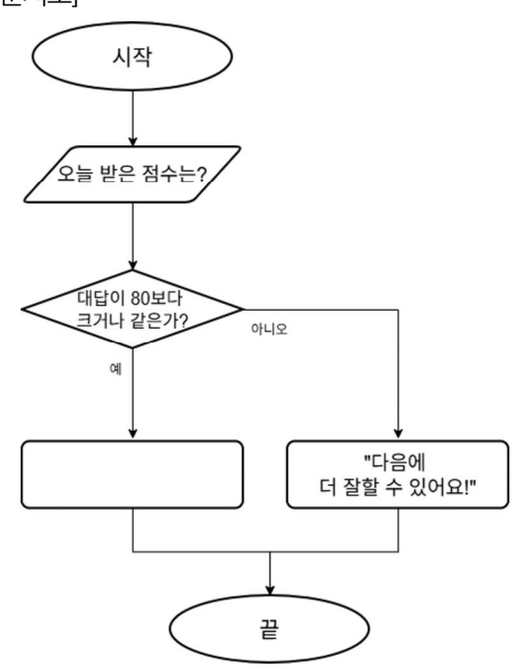
"오늘 받은 점수를 입력해봐요!" 묻기

만약 ( ) 이라면
"목표 달성! 아주 잘했어요!"라고 말하기

아니면
( )라고 말하기

끝

[순서도]



수업 평가지 예시 답안 및 채점 기준

예시 답안

1. 다 → 나 → 라 → 가 (4, 2, 3, 1)
2. 자연어: 오늘 받은 점수
   의사코드: 대답 ≥ 80, “다음엔 더 잘 할 수 있어요!”
   순서도: “목표 달성! 아주 잘했어요!”

평가 목표

생활 속 문제를 해결하기 위해 알고리즘(자연어, 의사코드, 순서도 등)으로 표현할 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	알고리즘의 순서를 논리적으로 정렬할 수 있고, 알고리즘을 자연어, 의사코드 또는 순서도로 정확하게 표현할 수 있다.
보통	알고리즘의 순서를 대체로 정렬할 수 있고, 알고리즘을 자연어, 의사코드 또는 순서도 중 하나로 표현할 수 있으나 조건 흐름 표현이 다소 미흡하다.
노력 요함	문제 해결 과정을 알고리즘으로 표현하는데 어려움을 겪는다.

맞춤형 피드백 예시

학생 반응	피드백 예시
알고리즘 정의는 알지만 직접 적용하지 못하는 경우	내가 만들고 싶은 프로그램을 입력 → 처리 → 출력 순서로 나누어 하나씩 적어볼까요?
자연어는 썼지만 의사코드 또는 순서도로 바꾸지 못한 경우	맨 처음과 마지막에 쓴 내용을 시작/끝으로 표현하고, 조건이 있는 부분에 ‘만약 ~이라면’을 붙여볼까요?

수업 운영 팁

Q1. 자연어와 의사코드의 차이를 학생들이 헷갈려 하면 어떻게 하나요?

A1. 자연어는 친구에게 말하듯 편하게, 의사코드는 컴퓨터에게 설명하듯 조금 더 정확하게 쓰는 것이라고 안내해주세요. 칠판에 같은 내용을 자연어→의사코드 순서로 나란히 써서 비교해주면 차이를 훨씬 빠르게 이해합니다.

Q2. 학생들이 순서도 그리는 것을 어려워하면 어떻게 하나요?

A2. 도형 규칙을 처음부터 엄격하게 요구하지 마세요. 직사각형은 할 일, 마름모는 질문(조건)이라는 두 가지만 먼저 알려주고, 화살표로 흐름을 연결하는 것에 집중하도록 안내하면 부담 없이 시작할 수 있습니다.

Q3. 학생마다 만들고 싶은 프로그램이 너무 달라 교사가 모두 확인하기 어려울 때는 어떻게 하나요?

A3. 모둠별로 한 명씩 발표하게 한 뒤, 나머지 모둠원이 순서가 논리적으로 이어지는지를 체크하는 역할을 맡기세요. 교사는 순회하며 논리적 흐름이 끊긴 부분만 짚어주는 방식으로 효율적으로 운영할 수 있습니다. 또한, 해당 차시는 알고리즘으로 표현하는 것이 학습 목표이기 때문에 학생들이 선택한 문제를 프로그램으로 구현하지 않고 알고리즘으로 표현하는 것만으로도 컴퓨팅 사고력을 기를 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
알고리즘 표현 도구	Draw.io	무료 온라인 순서도 도구로 구글 드라이브를 통한 공동 작업 지원
	보드믹스	드래그 앤 드롭 방식으로 순서도 기호를 배치할 수 있는 온라인 협업 도구

활용 시 유의점

- 디지털로 순서도 제작 도구를 활용할 경우, 교사의 템플릿을 준비하고, 미리 사용 방법을 간단히 시연한 뒤 학생이 활용하도록 안내한다.
- 디지털 작업 중 수시로 저장하는 습관을 갖도록 안내하여, 작업 내용이 사라지는 상황을 예방한다.

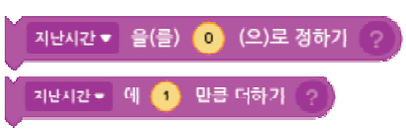
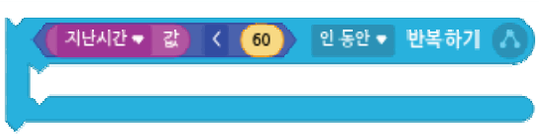
3. 프로그래밍으로 나에게 필요한 프로그램을 만들어 볼까요?

- **관련 성취 기준 :** [06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.
- **학습 목표 :** 나에게 필요한 프로그램을 만들 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 생활 속 시간 관리 경험 나누기 • 학습 문제 확인하기	• 프로그래밍 준비하기 (알고리즘 빈칸 채우기, 블록 살펴보기, 오브젝트·화면 디자인하기) • 1분 스톱워치 프로그래밍하기 (알고리즘대로 코딩하고 실행 확인하기)	• 프로그램 개선 방안 발표하기 • 1분 스톱워치 프로그램 저장하기

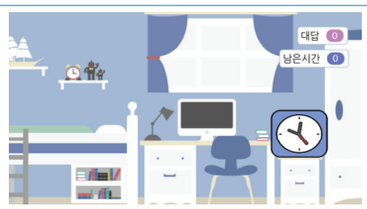
교수·학습 방향	지도상의 유의점	수업 준비물
• 의사코드(알고리즘)의 빈칸을 채우는 과정을 통해 프로그램의 논리적 흐름을 자연스럽게 이해하도록 돕는다. • 변수, 반복, 조건 블록의 역할을 먼저 살펴본 후 직접 연결해보며 블록 코딩의 원리를 체득하도록 한다. • 입력 → 처리 → 출력의 흐름이 프로그램에 올바르게 구현되었는지에 초점을 둔다.	• 알고리즘 및 학습지의 블록을 통해 스스로 프로그래밍할 수 있도록 지도한다. • 오브젝트 및 화면 디자인은 자유롭게 허용하되, 프로그래밍 시간이 부족해지지 않도록 적정 시간을 안내한다. • 학생들의 프로그래밍 수준이 학급마다 다를 수 있으므로, 필요한 경우 짝 또는 모둠별로 프로그래밍 활동을 진행하는 등 학급 상황에 맞게 운영한다.	• 교사: 스마트기기(엔트리) • 학생: 스마트기기(엔트리), 활동지

교수·학습 과정

학습 단계	교수·학습 활동	자료 및 유의점
도입	◆ 생활 속 시간 관리 경험 떠올리기 T : 1분이라는 시간을 정해서 활동해야 했던 경험을 나눠봅시다. S : 1분 동안 손을 씻은 적이 있습니다. S : 1분 스케치를 해본 적이 있습니다. T : 1분이 얼마나 길게 느껴지는지 우리 함께 느껴볼까요? 눈을 감고, 1분이 됐다고 생각되면 조용히 손을 들어보세요. T : 0초부터 시작해서 60초가 되면 알려주는 프로그램을 직접 만들어 봅시다. ◆ 학습문제 확인하기 1분 스톱워치 프로그램을 만들어봅시다.	
전개	◆ 활동1. 프로그래밍 준비하기 T : 완성된 프로그램을 살펴본 뒤 타이머 제작에 필요한 알고리즘을 살펴보고 활동지에 빈칸을 채워봅시다. 시작하기 버튼을 클릭했을 때 "1분 스톱워치입니다."를 말하기 지난시간을 (① 0)으로 정하기 (② 지난시간 < 60) 인 동안 반복하기: 1초 기다리기 (③ 지난시간 1만큼 더하기) 지난시간 + "초 지났어요!" 라고 말하기 반복이 끝나면 "1분 완료!" 라고 4초동안 말하기 T : 타이머 제작에 필요한 변수, 반복 블록들을 살펴봅시다. 변수의 이름을 다양하게 정해보고, 반복 블록을 만들어봅시다. 기능 / 블록 변수  반복 	• 준비물: 활동지, 스마트기기(엔트리) 엔트리 완성 파일 https://naver.me/xdM12MSH • 의사코드로 표현된 알고리즘의 빈칸을 채우면서 프로그램 흐름을 이해하게 한다. • 2가지 이상 블록들을 연결하는 과정을 반복을 통해 익힌다.

T : 타이머 제작에 어울리는 오브젝트를 정하고 실행 화면을 디자인 해봅시다.

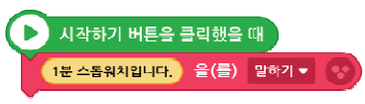
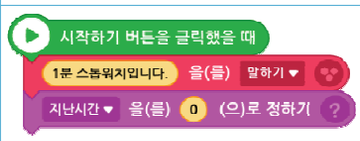
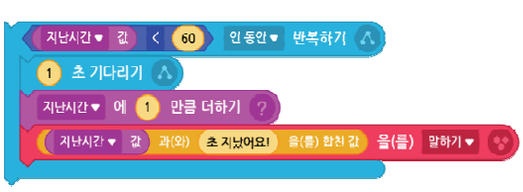
예)



- 자유롭게 오브젝트 및 실행화면을 디자인할 수 있도록 한다.
- 활동지에 알고리즘과 블록을 참고하여 스스로 프로그래밍 할 수 있도록 지도한다.

◆ 활동2. 1분 스톱워치 프로그래밍하기

T : 활동지의 알고리즘, 미리 살펴봤던 블록을 기반으로 1분 스톱워치 프로그램을 만들어봅시다.

알고리즘	블록
"1분 스톱워치입니다." 를 말하기	
지난시간 0으로 정하기	
지난시간 < 60 동안 계속 반복하기	
1초 기다리기 지난시간에 1만큼 더하기 지난시간 + "초 지났어요!" 라고 말하기	
"1분 완료!" 4초 동안 말하기	

T : 프로그램을 실행하여 입력, 처리, 출력 결과를 확인한 뒤 여러 가지 개선 방안을 생각해 활동지에 적어봅시다.

※ 교과서 55페이지 정답
① 프로그래밍

정리

◆ 프로그램 개선방안 발표하기

T : 활동지에 적은 내용을 바탕으로 프로그램을 발전시키기 위한 여러 가지 아이디어를 발표해봅시다.

- 프로그래밍 중간에도 학생들이 스스로 저장할 수 있도록 지도한다.

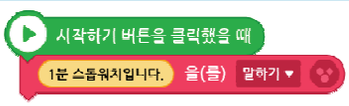
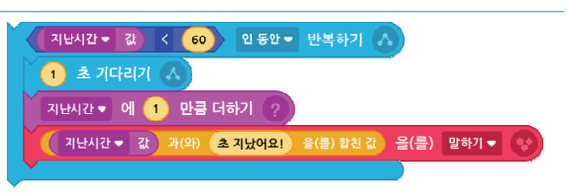
◆ 1분 스톱워치 프로그램 저장하기

T : 제작한 프로그램을 저장해봅시다.

수업 활동지

프로그래밍으로 나에게 필요한 프로그램을 만들어 볼까요?

1. 완성된 프로그램을 보고, 왼쪽 알고리즘의 빈칸을 채워봅시다.

알고리즘	블록
시작하기 버튼을 클릭했을 때 "1분 스톱워치입니다."를 말하기	
지난시간 = (① _____)	
(② _____) 인 동안 반복하기	
1초 기다리기 (③ _____) 지난시간 + "초 지났어요" 라고 말하기	
"1분 완료!" 4초 말하기	

2. 완성한 프로그램을 실행해보고, 더 발전시키고 싶은 점을 적어봅시다.

수업 평가지

1. 다음 자연어 알고리즘을 보고, 의사코드 빈칸에 들어갈 알맞은 답을 적어봅시다.

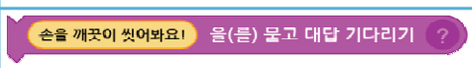
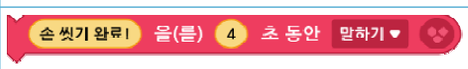
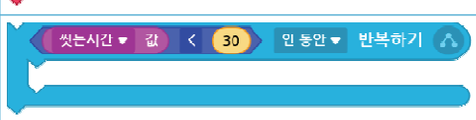
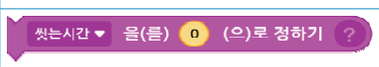
[자연어]

1. 시작 버튼을 누르면 "손을 깨끗이 씻어봐요!"라고 말한다.
2. 씻는시간을 0으로 초기화한다.
3. 씻는시간이 30보다 작은 동안 아래를 반복한다.
   3-1. 1초를 기다린다.
   3-2. 씻는시간에 1을 더한다.
   3-3. 씻는시간과 "초 씻었어요!"를 합쳐서 말한다.
4. 반복이 끝나면 "손 씻기 완료!"를 4초 동안 말한다.

[의사코드]

시작하기 버튼을 클릭했을 때
  "손을 깨끗이 씻어봐요!" 을(를) 말하기
  (① _____) 인 동안 반복하기
    1 초 기다리기
    [씻는시간] 에 1 만큼 더하기
    (② _____) 값 과(와) "초 씻었어요!" 를 합친 값 을(를) 말하기
  "손 씻기 완료!" 을(를) 4 초 동안 말하기

2. 위의 알고리즘 기반 "30초 손 씻기 스톱워치"를 만들었습니다. 다음 프로그래밍에서 필요하지 않은 블록을 골라봅시다. ( )

번호	블록
①	
②	
③	
④	
⑤	

3. 30초 손 씻기 스톱워치에서 더 발전시키거나, 개선 시키고 싶은 부분을 적어봅시다.

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ① 씻는시간 값 < 30 / ② 씻는시간
2. ①
3. 15초가 됐을 때 "절반 지났어요!"라는 중간 알림 추가
   손 씻기가 완료되면 박수 소리나 효과음 재생
   배경색이 시간에 따라 점점 바뀌도록 설정
   시작/정지 버튼을 따로 만들어 중간에 멈출 수 있도록 개선

평가 목표

알고리즘을 이해하고 프로그램 구현에 필요한 블록을 판별하여 프로그래밍할 수 있다.

평가 기준

수준	평가 기준
잘함	프로그램의 입력·처리·출력 흐름을 이해하고, 프로그램에 필요하지 않은 블록을 정확하게 찾아낼 수 있다.
보통	각 블록의 기능을 대체로 이해하고 있으나, 프로그램 흐름과의 연관성을 완전히 파악하지 못해 블록 선택에 다소 어려움을 겪는다.
노력 요함	각 블록의 역할을 이해하지 못해 프로그램에 필요하지 않은 블록을 찾아내지 못한다.

맞춤형 피드백 예시

학생 반응	피드백 예시
자연어는 이해했지만 의사코드 표현을 어려워하는 학생	"자연어에서 맨 처음 일어나는 일과 마지막에 일어나는 일을 먼저 찾아보세요. 그다음 '씻는시간이 30보다 작은 동안'처럼 조건이 있는 부분에 기호(〈)를 붙여서 써볼까요?"
알고리즘 개념은 알지만 빈칸에 적용 못하는 학생	"씻는시간 타이머가 작동하려면 어떤 순서로 이루어져야 할까요? 자연어를 보며 시작 → 처리 → 끝 순서로 하나씩 나누어 빈칸에 적어볼까요?"

수업 운영 팁

Q1. 학생들이 자연어 알고리즘을 의사코드로 바꾸는 것을 어려워하면 어떻게 하나요?

A1. 자연어 알고리즘을 한 문장씩 소리 내어 읽으면서, 각 문장이 의사코드의 어느 부분에 해당하는지 함께 찾아보도록 지도합니다. 예를 들어 "씻는시간이 30보다 작은 동안 반복한다"는 문장을 읽고 "이걸 기호로 쓰면 어떻게 될까요?"라고 질문하며 〈 30 표현을 자연스럽게 유도합니다. 또한 빈칸 바로 앞뒤 문맥을 단서로 활용하도록 안내하고, 전체를 한 번에 채우려 하지 않고 시작 → 반복 조건 → 반복 내용 → 끝 순서로 하나씩 짚어가도록 합니다.

Q2. 학생들이 엔트리 블록 코딩 자체를 어려워하면 어떻게 지도하나요?

A2. 활동지의 의사코드와 블록을 나란히 놓고, 의사코드 한 줄이 어떤 블록에 해당하는지 1:1로 대응시켜 보는 활동을 먼저 진행합니다. 교사가 첫 번째 블록(시작하기 버튼 클릭 → 말하기)을 함께 연결해 시범을 보여준 뒤, 나머지는 학생이 의사코드를 보며 순서대로 따라 연결하도록 지도합니다. 블록을 찾기 어려워하는 학생에게는 변수, 흐름, 생김새 등 카테고리별 색깔을 단서로 활용하도록 안내하고, 먼저 완성한 학생이 옆 친구를 도와주는 협력 학습 방식도 효과적입니다.

Q3. 학생 간 프로그래밍 진행 속도 차이가 크면 어떻게 운영하나요?

A3. 빠르게 완성한 학생에게는 3번 문항(개선 아이디어)과 연계하여 직접 심화 기능을 추가해보도록 안내합니다. 예를 들어 "15초가 됐을 때 중간 알림 말하기 추가하기", "카운트다운(30 → 0)으로 바꿔보기" 등 구체적인 도전 과제를 미리 칠판에 적어두면 별도의 안내 없이도 스스로 도전할 수 있습니다. 속도가 느린 학생에게는 의사코드와 블록 대응표를 다시 확인하도록 안내하고, 교사가 순회하며 막히는 블록 한두 개만 힌트를 주는 방식으로 스스로 완성하는 경험을 갖도록 지원합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치, 메이크 코드	알고리즘을 실제 프로그램으로 구현
공유·발표 도구	엔트리 작품 공유하기, 구글 드라이브, 패들렛	완성한 프로그램을 저장 및 친구들과 공유

활용 시 유의점

- 예기치 않은 오류나 접속 문제로 작업 내용이 사라질 수 있으므로, 프로그래밍 중간중간에 수시로 저장하도록 안내한다.
- 엔트리에 작품을 공유할 경우, 학생의 이름·사진·위치 정보 등 개인정보가 포함되지 않도록 사전에 지도한다.

4. 나에게 필요한 프로그램을 발전시켜 볼까요?

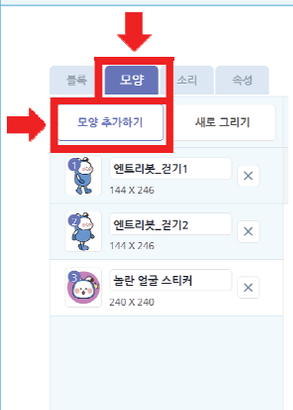
- **관련 성취 기준 :** [06자율-3] 프로그래밍으로 해결된 사례를 찾아보고, 나에게 필요한 프로그램을 만드는 과정을 수행한다.
- **학습 목표 :** 나에게 필요한 프로그램을 다양하게 발전시킬 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 프로그램의 개선점 찾기 • 학습문제 확인하기	• 친구와 함께 개선점 토의하기 • 개선 아이디어를 적용해 프로그램 수정 및 공유하기	• 개선한 프로그램 내용 발표하기 • 프로그램 완성 과정 소감 나누기

교수·학습 방향	지도상의 유의점	수업 준비물
• 다양한 난이도의 개선 아이디어를 구체적인 블록 예시와 함께 학습지로 제공하여, 학생 스스로 자신의 수준에 맞는 도전을 선택하도록 한다. • 친구의 프로그램을 직접 실행해보고 피드백을 나누는 과정을 통해, 같은 문제도 다양한 방식으로 해결할 수 있음을 경험하게 한다.	• 기본 개선사항(예: 소리 효과주기, 스토리 추가하기 등)부터 심화 개선사항(예: 시간 안남았을 때 모양의 변화주기 등)까지 난이도에 따라 안내하여 학생 스스로 도전하도록 한다. • 친구와의 공유 활동 시 활동지와 프로그램을 책상 위에 올려두고 자유롭게 돌아다니도록 하며, 친구 활동지에는 칭찬과 궁금한 점을 중심으로 적도록 안내한다. • 필요한 경우 앞차시와 연차시로 수업을 구성할 수 있다.	• 교사: 개선 아이디어 예시자료 (난이도별), 스마트기기(엔트리) • 학생: 전 차시에 만든 타이머 프로그램, 활동지, 스마트기기(엔트리)

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 프로그램의 개선점 찾기
T : 전 차시에서 만든 프로그램을 여러번 실행해보고, 더 추가하고 싶은 기능들을 발표해봅시다.
예) 시간이 다 되었을 때 소리 등의 효과를 추가하고 싶습니다.
예) 시간이 얼마 남지 않았을 때 화면에 변화를 주고 싶습니다.
◆ 학습문제 확인하기
1분 타이머 프로그램을 다양하게 발전시켜 봅시다.</td><td></td></tr>
<tr><td>전개</td><td>◆ 활동1. 친구와 함께 개선점 토의하기
T : 짝 또는 모둠 친구와 함께 서로의 프로그램을 실행해보고 개선하고 싶은 부분 2-3가지를 활동지에 적어봅니다.
T : 선생님이 제시하는 개선 아이디어 예시들을 비교해보고, 나의 프로그램에 적용할 아이디어를 1개 이상 선택합니다.
<table><tr><th>개선사항</th><th>블록</th></tr><tr><td>소리 효과 추가하기</td><td>
- 소리 탭에서 원하는 소리 선택 및 저장
- [소리] 블록 꾸러미에서 원하는 소리 재생하기 효과 추가하기</td></tr><tr><td>스토리 추가하기, 시간 바꾸기</td><td>
- ‘라면을 끓이기 시작합니다.’ 문구 수정하기
- 120초(2분)로 스톱워치 설정하기
- ‘초 익혔어요!’, ‘라면 완성!’ 문구 수정하기</td></tr></table></td><td>• 준비물: 활동지, 스마트기기(엔트리), 예시자료
• 코드 외에도 실행 화면 디자인, 소리 등 다양한 부분을 수정할 수 있게 지도한다.
• 구현 가능한 아이디어들에 접근할 수 있도록 예시를 제공한다.</td></tr>
</table>

개선사항	블록
시간이 얼마 안 남았을 때 모양 변화 주기	 - [모양] 탭에서 오브젝트의 모양을 다양하게 추가 가능  - [흐름] 블록 꾸러미에서 10초가 남았을 때 ([지난시간] 〉 50) 조건 블록 설정하기 - [생김새] 블록 꾸러미에서 오브젝트의 모양을 다양한 모양으로 바꾸기

◆ **활동2. 개선 아이디어를 적용해 프로그램 수정 및 공유하기**

T : 선택한 개선 아이디어를 엔트리 블록으로 직접 적용해봅니다.

T : 수정된 프로그램을 자리에 두고, 정해진 시간(약 3-5분)동안 친구의 자리에 돌아다니며 직접 프로그램을 실행해봅시다.

T : 인상 깊었던 점이나 궁금한 점을 친구의 활동지에 적어봅시다.

- 스스로 코딩이 어려운 경우 활동지의 예시를 참고하도록 지도한다.
- 충분한 시간을 통해 자유롭게 개선할 시간을 제공한다.
- 활동지와 프로그램을 책상 위에 올려놓고 자유롭게 돌아다니게 한다.

※ 교과서 58페이지 정답
① 친구

정리

◆ **개선한 프로그램 내용 발표하기**

T : 어떤 부분을 개선했는지 발표해봅시다.

◆ **프로그램 완성 과정 소감 나누기**

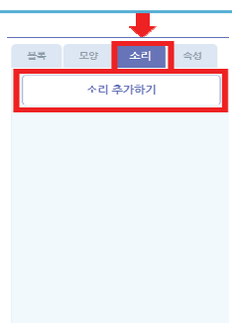
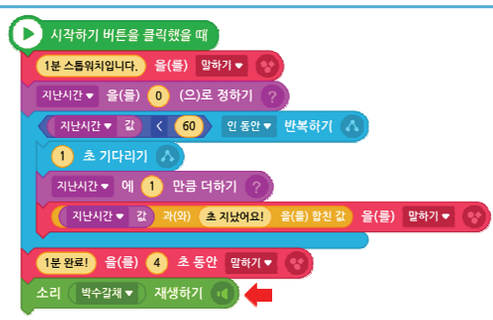
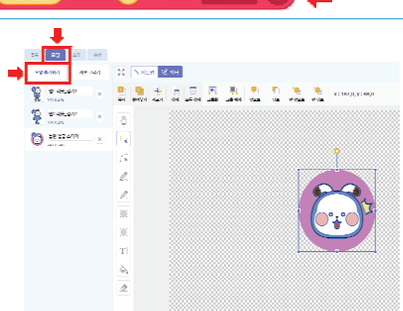
T : 친구들이 작성한 후기 중 가장 기억에 남는 후기를 읽어봅시다.

수업 활동지

나에게 필요한 프로그램을 발전시켜 볼까요?

1. 친구와의 토론을 통해 완성된 프로그램에서 개선하고 싶은 부분 2-3가지를 적어봅시다.

2. 아래의 예시들을 참고하여, 만든 프로그램을 다양하게 수정해봅시다.

개선사항	블록
소리 효과 추가하기	  – [소리] 블록 꾸러미에서 원하는 소리 재생하기 효과 추가하기 – 원하는 시점에 소리가 재생됨
스토리 추가하기, 시간 바꾸기	 – ‘라면을 끓이기 시작합니다.’ 문구 수정하기 – 120초(2분)로 스톱워치 설정하기 – ‘초 익혔어요!’, ‘라면 완성!’ 문구 수정하기
시간이 얼마 안 남았을 때 모양 변화 주기	  – [모양] 탭에서 오브젝트의 모양을 다양하게 추가 가능 – [흐름] 블록 꾸러미에서 10초가 남았을 때 ([지난시간] 〉 50) 조건 블록 설정하기 – [생김새] 블록 꾸러미 에서 오브젝트의 모양을 다양한 모양으로 바꾸기

수업 활동지

나에게 필요한 프로그램을 발전시켜 볼까요?

3. 친구의 프로그램을 직접 실행해보세요! 인상 깊었던 점이나 궁금한 점을 아래 후기에 적어봅시다.

이름			
후기			
이름			
후기			
이름			
후기			
이름			
후기			

수업 평가지

1. 다음은 친구가 만든 “30초 타이머 게임” 프로그램 입니다. 실행해보니 바로 “30초 끝!”이 나타나는 문제가 발생했습니다. 친구의 프로그램에서 잘못된 부분을 찾아 번호를 쓰고, 어떻게 고쳐야 할지 써 봅시다.

[친구의 프로그램]



잘못된 부분 : ( )

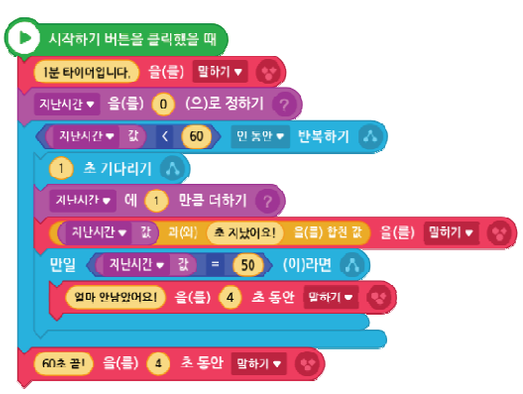
고친 방법 : _____

_____

2. 처음 프로그램과 비교했을 때, 수정 후 프로그램이 어떻게 추가되었는지 설명해보고, 추가로 더 발전시키고 싶은 부분을 작성해봅시다.



처음



수정 후

수정 방향 : _____

추가 수정 방향 : _____

_____

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. 잘못된 부분: ② “지난시간 〉 30”
   고친 방법: “지난시간 〈 30”으로 고쳐야 한다. 지난 시간이 30보다 작은 동안 지난 시간에 1초씩 더해지고, 몇 초가 지났는지 나타날 수 있다.

2. 수정 방향: 지난시간이 50초가 되었을 때, 얼마 안 남았어요! 문구가 나타난다.
   추가 수정 방향: 시간이 다 되었을 때, 소리 효과를 추가할 수 있다.

평가 목표

프로그램의 개선점을 스스로 발견하고, 소리·스토리·모양 변화 등 다양한 기능을 추가하여 나만의 프로그램으로 발전시킬 수 있는지 확인한다.

평가 기준

수준	평가 기준
잘함	프로그램의 개선점을 구체적으로 발견하고, 실현 가능한 내용으로 명확하게 작성할 수 있다.
보통	프로그램의 개선점을 발견하여 작성할 수 있다.
노력 요함	프로그램을 실행해 보았으나 개선점을 스스로 발견하지 못하거나, 프로그램과 직접적인 연관성이 부족하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
개선 아이디어는 찾았지만 프로그래밍 적용에 어려움을 겪는 학생	활동지의 블록 예시를 한 단계씩 따라가 보세요. 어느 블록 꾸러미에서 찾아야 하는지 색깔을 단서로 활용해볼까요?
개선 아이디어를 찾지 못한 학생	프로그램을 친구에게 보여준다고 생각해보세요. 친구가 더 재미있게 쓰려면 어떤 기능이 있으면 좋을까요? 소리, 모양, 문구 중 하나만 골라 생각해볼까요?

수업 운영 팁

Q1. 학생들이 어떤 개선 아이디어를 골라야 할지 몰라 어려움을 겪으면 어떻게 하나요?

A1. 난이도를 기본·심화로 구분하여 칠판에 크게 적어두고, 처음 도전하는 친구는 소리 효과 추가하기부터 시작이라는 출발점을 명확하게 안내합니다. 선택 자체에 시간을 너무 쓰지 않도록 1~2분 안에 고르도록 시간을 제한하고, 고르지 못한 학생에게는 교사가 학생의 프로그램을 함께 보며 "여기에 소리가 나면 어떨까요?"처럼 구체적인 질문으로 방향을 제시해줍니다.

Q2. 공유 활동 시 돌아다니지 않고 자리에만 있는 학생이 있으면 어떻게 하나요?

A2. 공유 활동 전에 최소 2명의 친구 프로그램을 실행해보는 것이 목표라는 명확한 기준을 제시합니다. 활동지에 친구 이름을 적는 칸을 만들어두면 자연스럽게 움직이도록 유도할 수 있습니다. 소극적인 학생에게는 교사가 먼저 "○○이 프로그램 한번 보러 가볼까요?"라고 함께 이동을 제안하며 참여를 이끌어줍니다.

Q3. 학생 간 프로그래밍 진행 속도 차이가 크면 어떻게 운영하나요?

A3. 기본 개선(소리 추가, 문구 바꾸기)만 완성해도 충분히 의미 있는 활동임을 먼저 인정해줍니다. 심화 개선(모양 변화)은 블록 예시를 활동지에 미리 제공하여 따라 하기만 해도 완성할 수 있도록 스캐폴딩을 제공합니다. 빠르게 완성한 학생에게는 "친구의 개선을 도와주는 선생님 역할"을 부여하여 상호 학습이 이루어지도록 운영합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치, 메이크 코드	개선 아이디어를 블록으로 직접 적용하고 실행하여 결과 확인
공유·발표 도구	엔트리 작품 공유하기, 패들렛	완성된 프로그램을 친구들과 공유하고 피드백 남기기

활용 시 유의점

- 개선 작업 중간중간에 수시로 저장하도록 안내하여, 오류나 접속 문제로 작업 내용이 사라지는 상황을 예방한다.
- 엔트리 작품 공유하기나 패들렛에 작품을 공유할 경우, 학생의 이름·사진·위치 정보 등 개인정보가 포함되지 않도록 사전에 지도한다.
- 친구의 프로그램에 피드백을 남길 때 칭찬과 궁금한 점 위주로 작성하도록 사전에 안내하여, 부정적인 표현이 나오지 않도록 지도한다.

5. 생활 속 문제를 해결하는 방법을 알고리즘으로 표현해 볼까요?

- **관련 성취 기준 :** [6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.
- **학습 목표 :** 생활 속 문제를 해결하기 위한 절차를 알고리즘으로 표현할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 생활 속에서 일정한 기준에 따라 판단이 필요한 상황 이야기 나누기 • 조건에 따라 안내해 주는 프로그램이 필요한 상황 확인하기 • 학습문제 확인하기	• 현재 상태와 목표 상태 정하기 • 프로그램이 해야 할 일을 입력, 처리, 출력으로 나누기 • 문제 해결 과정을 알고리즘으로 표현하기	• 알고리즘 표현 방법 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 생활 속 문제 상황의 현재 상태와 목표 상태를 구분하며, 프로그램이 해결해야 할 문제를 명확히 이해하도록 한다. • 입력 → 처리 → 출력의 흐름에 따라 프로그램이 해야 할 일을 나누어 보고, 문제 해결 절차를 알고리즘으로 표현하도록 한다. • 조건에 따라 서로 다른 결과가 나타나는 과정을 자연어, 의사코드, 블록과 연결하여 이해하도록 한다.	• 현재 상태는 문제가 아직 해결되지 않은 지금의 상태, 목표 상태는 프로그램을 통해 도달하고 싶은 결과임을 안내한다. • 입력, 처리, 출력은 각각 사용자가 값을 넣는 과정, 프로그램이 판단하는 과정, 결과를 안내하는 과정으로 구분하여 지도한다. • 알고리즘은 엔트리 블록을 그대로 옮겨 적는 것이 아니라, 프로그램이 해야 할 일을 사람이 이해할 수 있도록 순서대로 나타내는 것임을 안내한다.	• 교사: 스마트기기(엔트리) • 학생: 스마트기기(엔트리), 활동지

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 생활 속에서 일정한 기준에 따라 판단이 필요한 상황 이야기 나누기
T : 생활 속에서 어떤 기준에 따라 다르게 행동해야 하는 경우가 있나요?
S : 비가 많이 오면 우산을 챙겨야 합니다.
S : 시간이 많이 지나면 교실을 환기해야 합니다.
◆ 조건에 따라 안내해 주는 프로그램이 필요한 상황 확인하기
T : 교실에서 창문을 닫고 수업하다 보면 어떤 불편함이 생길 수 있을까요?
S : 환기한 지 얼마나 지났는지 기억하기 어렵습니다.
S : 언제 창문을 열어야 할지 놓칠 수 있습니다.
T : 환기한 지 몇 분이 지났는지 입력하면, 창문을 열어야 할지 알려 주는 프로그램이 있다면 어떤 점이 좋을까요?
S : 환기할 시간을 놓치지 않을 수 있습니다.
S : 창문을 열어야 할 때를 쉽게 알 수 있습니다.
◆ 학습문제 확인하기
생활 속 문제를 해결하기 위한 절차를 알고리즘으로 표현해봅시다.</td><td></td></tr>
<tr><td>전개</td><td>◆ 활동1. 현재 상태와 목표 상태 정하기
T : 교실 환기 상황에서 현재 불편한 점은 무엇인가요?
S : 환기한 지 몇 분이 지났는지 기억하기 어렵습니다.
S : 언제 창문을 열어야 할지 놓칠 수 있습니다.
S : 창문을 닫고 오래 지나도 바로 알기 어렵습니다.
T : 프로그램이 생긴 뒤 바라는 모습은 무엇인가요?
S : 환기한 지 몇 분이 지났는지 입력하면 안내해 줍니다.
S : 120분 이상 지났으면 창문을 열라고 알려 줍니다.
S : 120분이 지나지 않았으면 아직 열지 않아도 된다고 알려 줍니다.
<table><tr><th>현재 상태</th><th>목표 상태</th></tr><tr><td>창문을 닫고 시간이 지났는데도 언제 환기해야 할지 놓칠 수 있다.</td><td>환기한 지 몇 분이 지났는지 입력하면, 창문을 열어야 할지 알려 준다.</td></tr></table>◆ 활동2. 프로그램이 해야 할 일을 입력, 처리, 출력으로 나누기
T : 이 프로그램은 사용자에게 무엇을 물어보아야 할까요?
S : 환기한 지 몇 분이 지났는지 물어보아야 합니다.
S : 창문을 닫고 몇 분이 지났는지 물어보아야 합니다.
T : 입력한 값을 가지고 프로그램은 무엇을 판단해야 할까요?</td><td>• 준비물: 활동지, 스마트기기(엔트리)
• 현재 상태와 목표 상태를 한 문장으로 간단히 쓰게 한다.
• 자연어 알고리즘을 먼저 읽고, 그 내용을 바탕으로 의사코드의 빈칸을 채우게 한다.
• 빈칸은 기준값, 조건을</td></tr>
</table>

S : 입력한 시간이 120분 이상인지 판단해야 합니다.
S : 2시간이 지났는지 확인해야 합니다.
T : 판단한 뒤에는 어떤 문장을 말하면 좋을까요?
S : 120분 이상이면 "창문을 열어주세요."라고 말하면 좋겠습니다.
S : 120분이 지나지 않았으면 "아직은 창문을 열지 마세요."라고 말하면 좋겠습니다.

입력	환기한 지 몇 분이 지났는지 묻고 대답을 입력받는다.
처리	입력한 시간이 120분 이상인지 판단한다.
출력	120분 이상이면 "창문을 열어주세요."라고 말하고, 그렇지 않으면 "아직은 창문을 열지 마세요."라고 말한다.

만족할 때의 출력, 조건을 만족하지 않을 때의 출력으로 제한하여 학생 부담을 줄인다.

• 이 차시에서는 엔트리 블록과 직접 연결하기보다, 문제 해결 절차가 순서대로 표현되었는지에 초점을 둔다.

◆ 활동3. 의사코드의 빈칸을 채워 알고리즘 완성하기

T : 교실 환기 안내 프로그램의 자연어 알고리즘을 살펴봅시다. 이 알고리즘은 어떤 순서로 이루어져 있나요?
S : 먼저 환기한 지 몇 분이 지났는지 묻습니다.
S : 입력한 시간이 120분 이상인지 확인합니다.
S : 120분 이상이면 창문을 열라고 말합니다.
S : 120분 미만이면 아직 창문을 열지 말라고 말합니다.

• 조건을 만족할 때와 만족하지 않을 때 실행되는 결과가 다르다는 점을 확인하게 한다.

예) 교실 환기 안내 프로그램 자연어 알고리즘
① 시작하기 버튼을 클릭한다.
② "환기한 지 몇 분이 지났나요?"라고 묻고 대답을 기다린다.
③ 대답한 값이 120 이상인지 확인한다.
④ 120 이상이면 "창문을 열어주세요."라고 말한다.
⑤ 그렇지 않으면 "아직은 창문을 열지 마세요."라고 말한다.

T : 자연어 알고리즘을 참고하여 의사코드 활동지의 빈칸을 채워 봅시다.

• 자연어 알고리즘을 먼저 읽고, 그 내용을 바탕으로 의사코드의 빈칸을 채우게 한다.

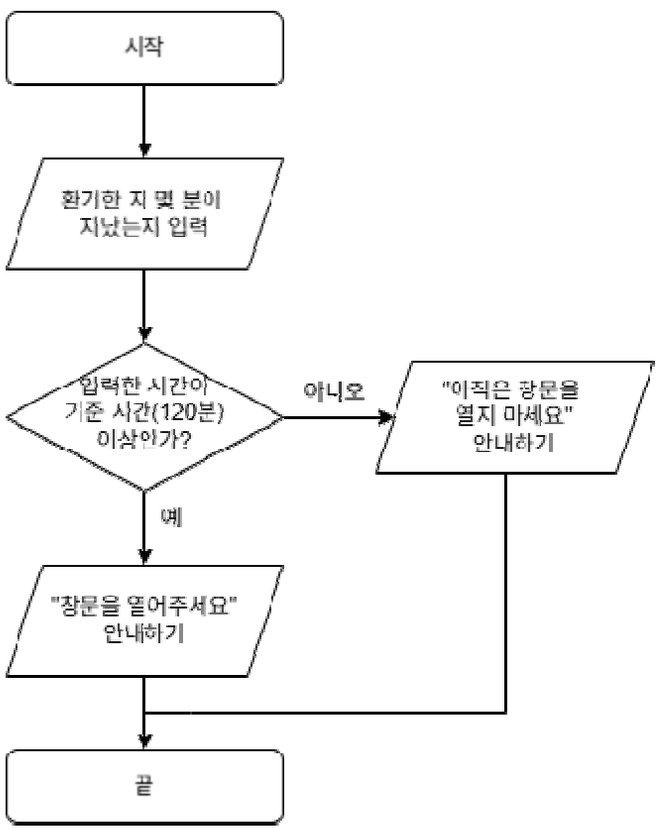
시작
사용자에게 _____을 묻는다.
입력한 시간이 기준 시간 이상인지 확인한다.
기준 시간 이상이면 _____을 안내한다.
기준 시간보다 적으면 _____을 안내한다.
끝

• 빈칸은 기준값, 조건을 만족할 때의 출력, 조건을 만족하지 않을 때의 출력으로 제한하여 학생 부담을 줄인다.

T : 빈칸에 어떤 내용이 들어가야 할까요?
S : 첫 번째 빈칸에는 120이 들어갑니다.
S : 120 이상이면 "창문을 열어주세요."가 들어갑니다.
S : 아니면 "아직은 창문을 열지 마세요."가 들어갑니다.
T : 자연어 알고리즘을 참고하여 순서도 활동지의 빈칸을 채워 봅시다.

교실 환기 안내 프로그램 자연어 알고리즘

① 시작하기 버튼을 클릭한다.
② “환기한 지 몇 분이 지났나요?”라고 묻고 대답을 기다린다.
③ 대답한 값이 120 이상인지 확인한다.
④ 120 이상이면 “창문을 열어주세요.”라고 말한다.
⑤ 그렇지 않으면 “아직은 창문을 열지 마세요.”라고 말한다.



- 이 차시에서는 엔트리 블록과 직접 연결하기보다, 문제 해결 절차가 순서대로 표현되었는지에 초점을 둔다.

※ 교과서 63페이지 정답
① 입력
② 알고리즘

정리

◆ **알고리즘 표현 방법 정리하기**

T : 오늘 생활 속 문제를 해결하기 위한 절차를 어떻게 표현해 보았는지 다시 한번 생각해봅시다.

◆ **교실 환기 안내 프로그램 저장하기**

T : 제작한 프로그램을 저장해봅시다.

- 프로그래밍 중간에도 학생들이 스스로 저장할 수 있도록 지도한다.

수업 활동지

생활 속 문제를 해결하는 방법을 알고리즘으로 표현해볼까요?

1. 교실 환기 상황에서 현재 상태와 목표 상태를 적어봅시다.

문제상황

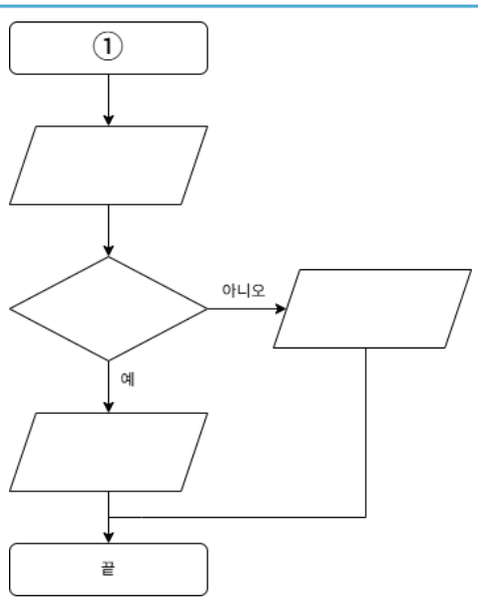
학교에서는 2시간 마다 한 번씩 환기를 하면 좋다고 한다. 교실에서 창문을 닫고 있으면 시간이 지났는데도 깜빡 잊어버리고 환기하지 않는 경우가 있다. 그래서 환기한 지 몇 분이 지났는지 시간을 입력했을 때 환기를 해야 하는지, 아닌지 알려주는 프로그램을 만들고 싶다.

현재 상태	목표 상태

2. 완성된 프로그램을 보고, 순서도의 빈칸에 번호를 채워봅시다.

교실 환기 안내 프로그램 자연어 알고리즘

① 시작하기
② "창문을 열어주세요" 출력하기
③ 환기한 지 몇 분이 지났는지 입력하기
④ 입력한 시간이 120분 이상인지 확인하기
⑤ “아직은 창문을 열지 마세요.” 출력하기



수업 평가지

1. 다음 중 교실 환기 안내 프로그램을 만드는 과정으로 적절하지 않은 것은 무엇입니까?

① 환기한 지 몇 분이 지났는지 묻는다.
② 입력한 시간이 120분 이상인지 판단한다.
③ 120분을 2시간으로 바꿔서 다시 안내한다.
④ 120분 이상이면 창문을 열어 달라고 안내한다.
⑤ 120분이 지나지 않았으면 아직 창문을 열지 말라고 안내한다.

정답 : _____

2. 교실 환기 안내 프로그램의 현재 상태와 목표 상태를 적어봅시다.

현재 상태	목표 상태

3. 교실 환기 안내 프로그램의 입력, 처리, 출력을 바르게 연결해봅시다.

〈보기〉

㉠	입력한 시간이 120분 이상인지 판단한다.
㉡	“환기한 지 몇 분이 지났나요?”라고 묻고 대답을 입력받는다.
㉢	조건에 따라 “창문을 열어주세요.” 또는 “아직은 창문을 열지 마세요.”라고 말한다.

입력 · · ㉠
처리 · · ㉡
출력 · · ㉢

4. 다음 교실 환기 안내 알고리즘의 빈칸을 채워봅시다.

시작
“환기한 지 몇 분이 지났나요?”라고 묻는다.
대답한 값이 _____ 분 이상이면
“ _____ ”라고 말한다.
아니면
“ _____ ”라고 말한다.
끝

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ③
2. 현재 상태: 창문을 닫고 시간이 지나도 환기한 지 몇 분이 지났는지 기억하기 어렵다.
   목표 상태: 환기한 지 몇 분이 지났는지 입력하면, 120분 이상인지 판단하여 창문을 열어야 하는지 알려 준다.
3. 입력: ㉡, 처리: ㉠ 출력: ㉢
4. 시작
   "환기한 지 몇 분이 지났나요?"라고 묻는다.
   대답한 값이 120분 이상이면
   "창문을 열어주세요."라고 말한다.
   아니면
   "아직은 창문을 열지 마세요."라고 말한다.
   끝

평가 목표

생활 속 문제 상황의 현재 상태와 목표 상태를 구분하고, 입력·처리·출력의 흐름에 따라 조건 판단을 포함한 알고리즘을 표현할 수 있다.

평가 기준

수준	평가 기준
잘함	교실 환기 안내 프로그램의 현재 상태와 목표 상태를 명확히 구분하고, 입력·처리·출력과 조건 판단을 정확하게 연결하여 알고리즘을 완성할 수 있다.
보통	교실 환기 안내 프로그램의 현재 상태와 목표 상태를 대체로 구분하고, 입력·처리·출력 또는 조건 판단을 간단히 표현할 수 있다.
노력 요함	현재 상태와 목표 상태, 입력·처리·출력, 조건 판단을 이해하는 데 어려움을 겪으며 교사의 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
현재 상태와 목표 상태를 구분하지 못하는 경우	현재 상태는 지금 불편한 점이고, 목표 상태는 프로그램이 도와준 뒤의 모습이에요, 지금은 무엇이 불편한가요?, 프로그램이 있으면 어떻게 좋아질까요?로 나누어 생각해봅시다."

수업 운영 팁

Q1. 학생들이 현재 상태와 목표 상태를 구분하기 어려워하면 어떻게 하나요?

A1. 현재 상태는 지금 겪는 불편함이고, 목표 상태는 프로그램을 사용한 뒤 바라는 모습이라고 안내합니다. 예를 들어 "지금은 환기 시간을 놓칠 수 있다"는 현재 상태이고, "프로그램이 창문을 열 때를 알려 준다"는 목표 상태입니다. 학생이 문장을 길게 쓰기 어려워하면 '지금 불편한 점'과 '프로그램이 도와줄 점'으로 나누어 짧게 적게 합니다.

Q2. 학생들이 안내 프로그램의 입력, 처리, 출력 과정을 왜 써야 하는지 물어보면 어떻게 설명하나요?

A2. 입력, 처리, 출력은 프로그램이 어떤 순서로 작동하는지 쉽게 나누어 보는 방법이라고 안내합니다. 안내 프로그램은 보통 사용자가 어떤 값을 넣고, 프로그램이 그 값을 기준과 비교한 뒤, 결과를 알려 주는 흐름으로 작동합니다. 예를 들어 교실 환기 안내 프로그램에서는 '환기한 지 몇 분이 지났는지 대답하는 것'이 입력, '120분 이상인지 판단하는 것'이 처리, '창문을 열어주세요' 또는 '아직은 창문을 열지 마세요'라고 말하는 것이 출력입니다. 이렇게 나누어 생각하면 프로그램을 만들 때 무엇을 먼저 해야 하고, 어떤 기준으로 판단해야 하며, 어떤 결과를 보여 주어야 하는지 더 분명하게 알 수 있습니다.

Q3. 학생들이 알고리즘을 엔트리 블록처럼 그대로 쓰려고 하면 어떻게 하나요?

A3. 알고리즘은 블록 이름을 외워 쓰는 것이 아니라, 프로그램이 해야 할 일을 사람이 이해할 수 있도록 순서대로 나타낸 것이라고 안내합니다. 먼저 "묻는다 → 판단한다 → 말한다"처럼 짧은 문장으로 쓰게 하고, 이후 자연어 알고리즘과 의사코드를 비교하게 하면 부담을 줄일 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
문서·기록 도구	그림판, Drawio	순서도 그리기
문서·기록 도구	한글, 구글 Docs, 패들렛	현재 상태, 목표 상태를 입력

활용 시 유의점

- 엔트리 프로그램을 직접 작성하기 전에, 먼저 활동지에 현재 상태와 목표 상태를 적고 알고리즘을 표현하게 한다.
- 예시 프로그램은 정답을 바로 보여 주기보다, 입력값에 따라 결과가 달라지는 것을 확인하는 자료로 활용한다.

6. 알고리즘에 맞게 프로그램을 완성해 볼까요?

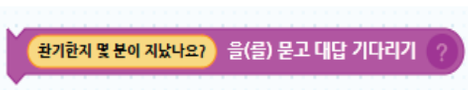
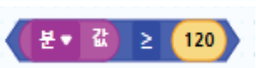
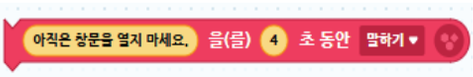
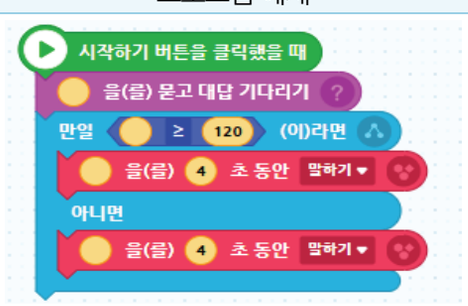
- **관련 성취 기준 :** [6실05-03] 실생활의 문제를 해결하는 프로그램을 협력하여 작성하고, 산출물을 타인과 공유한다.
- **학습 목표 :** 알고리즘에 맞게 프로그램을 완성하고, 실행 결과를 확인할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 지난 차시 알고리즘 떠올리기 • 학습문제 확인하기	• 알고리즘과 블록 연결하기 • 프로그램의 빈칸 채우기 • 실행 결과 확인하고 오류 수정하기	• 완성한 프로그램의 흐름 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 의사코드의 빈칸을 블록으로 연결하는 과정을 통해 프로그램의 논리적 흐름을 자연스럽게 이해하도록 돕는다. • 묻고 기다리기, 대답, 조건, 말하기 블록의 역할을 먼저 살펴본 후 직접 연결해보며 블록 코딩의 원리를 체득하도록 한다. • 입력 → 처리 → 출력의 흐름이 프로그램에 올바르게 구현되었는지에 초점을 둔다.	• 프로그램을 처음부터 모두 작성하기보다, 알고리즘과 연결되는 핵심 블록의 빈칸을 채우는 활동으로 운영한다. • 대답은 사용자가 입력한 값이며, 조건 판단에 사용되는 값임을 안내한다. • 대답 ≥ 120은 입력한 시간이 120분 이상인지 판단하는 조건임을 확인하게 한다.	• 교사: 스마트기기(엔트리) • 학생: 스마트기기(엔트리), 활동지

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 지난 차시 알고리즘 떠올리기
T : 지난 시간에 교실 환기 안내 프로그램의 알고리즘을 어떻게 표현했나요?
S : 환기한 지 몇 분이 지났는지 묻도록 했습니다.
T : 이 알고리즘에서 입력, 처리, 출력은 각각 무엇이었나요?
S : 입력은 환기한 지 몇 분이 지났는지 대답하는 것입니다.
S : 처리는 대답한 값이 120 이상인지 판단하는 것입니다.
S : 출력은 조건에 따라 안내 문장을 말하는 것입니다.

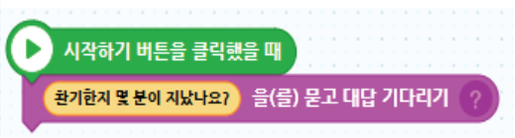
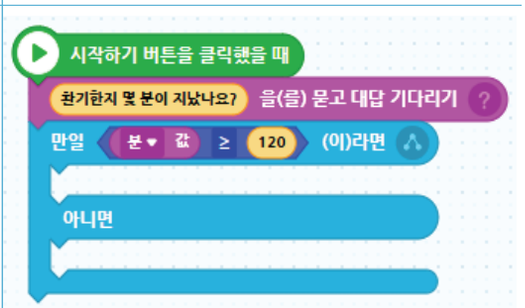
◆ 학습문제 확인하기
알고리즘에 맞게 프로그램의 빈칸을 채우고 실행 결과를 확인해 봅시다.</td><td></td></tr>
<tr><td>전개</td><td>◆ 활동1. 알고리즘과 블록 연결하기
T : 교실 환기 안내 알고리즘이 엔트리 프로그램에서 어떤 블록과 연결되는지 살펴봅시다.
<table><tr><th>알고리즘</th><th>블록</th></tr><tr><td>“환기한 지 몇 분이 지났나요?” 라고 묻는다.</td><td></td></tr><tr><td>대답한 값이 120 이상인지 판단한다.</td><td></td></tr><tr><td>“창문을 열어주세요.” 라고 말한다.</td><td></td></tr><tr><td>“아직은 창문을 열지 마세요.” 라고 말한다.</td><td></td></tr></table>◆ 활동2. 알고리즘에 맞게 프로그램 완성하기
T : 교실 환기 안내 프로그램을 살펴봅시다. 비어 있는 곳에는 어떤 내용이 들어가야 할까요?
<table><tr><th>프로그램 예제</th></tr><tr><td></td></tr></table></td><td>• 준비물: 활동지, 스마트기기(엔트리)
엔트리 완성 파일
https://naver.me/5B0MZem3

• 블록의 이름을 외우기보다 각 블록이 질문, 입력값, 조건 판단, 출력 중 어떤 역할을 하는지 이해하게 한다.

• 빈칸은 질문 문장, 조건식, 출력 문장으로 제한하여 학생 부담을 줄인다.
• 조건을 만족하는 경우와 만족하지 않는 경우의 말하기 문장을 바꾸어 넣지 않도록 확인한다.</td></tr>
</table>

◆ **활동3. 알고리즘에 맞게 코드 완성 과정 살펴보기**

T : 교실 환기 안내 프로그램이 알고리즘에 따라 어떻게 코드로 완성되는지 살펴봅시다.

알고리즘	블록
시작하기 버튼을 클릭한다. "환기한 지 몇 분이 지났나요?"라고 묻는다.	
대답한 값이 120 이상인지 판단한다.	
"창문을 열어주세요." 라고 말한다. 아니면 "아직은 창문을 열지 마세요."라고 말한다.	
전체 완성	

T : 완성된 코드가 입력, 처리, 출력의 흐름을 잘 나타내는지 확인해봅시다.

- 코드 전체를 한 번에 제시하기보다, 알고리즘의 각 단계가 블록으로 어떻게 추가되는지 순서대로 보여 준다.
- 학생들이 블록 모양을 외우기보다, 각 블록이 입력, 처리, 출력 중 어떤 역할을 하는지 이해하도록 지도한다.

※ 교과서 67페이지 정답
① 대답
② 만일

정리

◆ **완성한 프로그램의 흐름 정리하기**

T : 오늘 완성한 프로그램은 어떤 순서로 동작하나요?

T : 프로그램의 빈칸을 채울 때 중요한 점은 무엇인가요?

◆ **교실 환기 안내 프로그램 저장하기**

T : 제작한 프로그램을 저장해봅시다.

- 프로그래밍 중간에도 학생들이 스스로 저장할 수 있도록 지도한다.

수업 활동지

알고리즘에 맞게 프로그램의 빈칸을 채워볼까요?

1. 교실 환기 안내 프로그램의 알고리즘을 다시 살펴봅시다.

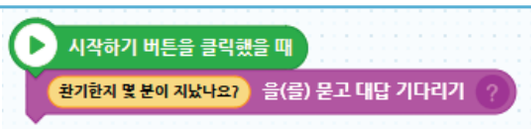
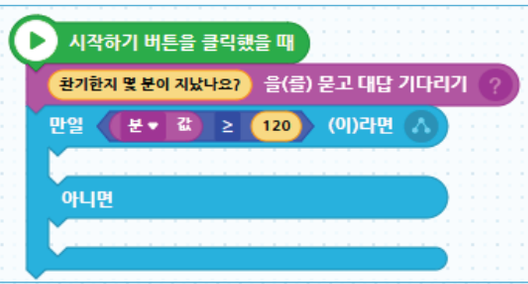
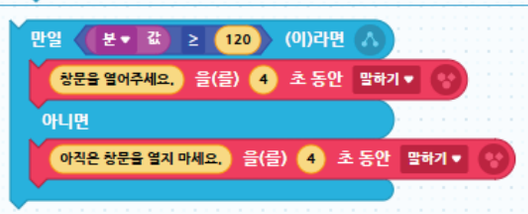
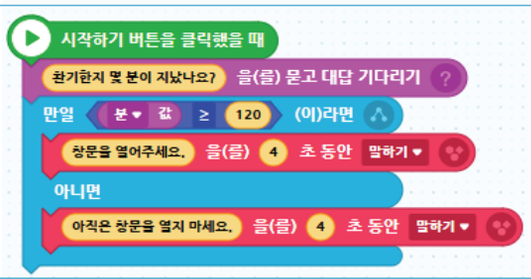
시작
"환기한 지 몇 분이 지났나요?"라고 묻는다.

대답한 값이 120분 이상이면
" 창문을 열어주세요."라고 말한다.

아니면
" 아직은 창문을 열지 마세요. "라고 말한다.

끝

2. 완성된 교실 환기 안내 프로그램을 보고 빈칸을 채워봅시다.

알고리즘	블록
시작하기 버튼을 클릭했을 때 "환기한 지 몇 분이 지났나요?"라고 묻는다.	
[대답] = (①)	
[대답] ≥ (②) 이면 (③) 말하기 [대답] ≥ (②) 이 아니면 (④) 말하기	
전체 코드 완성하기	

수업 평가지

1. 다음 중 '대답' 블록이 나타내는 것으로 알맞은 것은 무엇입니까?

① 프로그램의 제목
② 창문을 여는 동작
③ 사용자가 질문에 입력한 값
④ 처음부터 정해져 있는 숫자 120
⑤ 프로그램이 화면에 말하는 문장

정답 : _____

2. 다음 중 교실 환기 안내 프로그램의 실행 결과로 알맞은 것을 고르시오.

입력값이 119일 때 출력되어야 하는 문장은 무엇입니까?

① 창문을 열어주세요.
② 창문을 닫아주세요.
③ 환기가 끝났습니다.
④ 120분이 지났습니다.
⑤ 아직은 창문을 열지 마세요.

정답 : _____

3. 다음 교실 환기 안내 프로그램의 빈칸을 채워봅시다.

프로그램



수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ③

2. ⑤

3. 만일 대답 ≥ 120 이라면
   “창문을 열어주세요.” 4초 동안 말하기
   아니면
   “아직은 창문을 열지 마세요.” 4초 동안 말하기

평가 목표

알고리즘에 맞게 프로그램의 빈칸을 채우고, 입력값에 따라 조건 판단과 출력 결과가 올바르게 나타나는지 확인할 수 있다.

평가 기준

수준	평가 기준
잘함	‘대답’ 블록의 의미를 이해하고, 조건식과 출력 문장을 정확하게 채우며 실행 결과를 바르게 확인할 수 있다.
보통	‘대답’ 블록과 조건 판단의 의미를 대체로 이해하고, 프로그램의 주요 빈칸을 채울 수 있다.
노력 요함	‘대답’ 블록, 조건식, 출력 결과를 이해하는 데 어려움을 겪으며 교사의 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
‘대답’ 블록의 의미를 이해하지 못하는 경우	‘대답’은 사용자가 질문에 입력한 값이에요. 질문에 119를 입력하면 ‘대답’은 119가 됩니다.
조건을 만족할 때와 만족하지 않을 때의 문장을 바꾸어 쓴 경우	조건이 참일 때는 120분 이상 지난 경우입니다. 이때 나와야 할 문장을 먼저 생각해봅시다.

수업 운영 팁

Q1. 학생들이 '대답' 블록의 의미를 이해하지 못하면 어떻게 하나요?

A1. '대답'은 사용자가 질문에 입력한 값을 프로그램이 기억하고 있는 것이라고 안내합니다. 예를 들어 "환기한 지 몇 분이 지났나요?"라고 물었을 때 학생이 120이라고 입력하면, 대답 블록에는 120이라는 값이 들어갑니다. 이후 프로그램은 이 값을 조건식에서 사용하여 120분 이상인지 판단합니다.

Q2. 학생들이 조건을 만족할 때와 만족하지 않을 때의 출력 문장을 바꾸어 넣으면 어떻게 하나요?

A2. 먼저 조건이 참인 상황을 말로 풀어 보게 합니다. "대답 ≥ 120"은 환기한 지 120분 이상 지났다는 뜻이므로, 이때는 "창문을 열어주세요."가 나와야 합니다. 반대로 조건을 만족하지 않는 경우는 120분이 지나지 않은 상황이므로 "아직은 창문을 열지 마세요."가 나와야 합니다.

Q3. 학생들이 실행 결과만 보고 오류 원인을 찾지 못하면 어떻게 하나요?

A3. 오류를 한 번에 찾게 하기보다 질문, 조건식, 출력 문장의 순서로 나누어 확인하게 합니다. 먼저 질문 문장이 알맞은지, 다음으로 조건식이 대답 ≥ 120인지, 마지막으로 조건을 만족할 때와 만족하지 않을 때의 말하기 문장이 맞게 들어갔는지 확인하게 하면 오류 원인을 찾기 쉽습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치, MakeCode	알고리즘을 실제 프로그램으로 구현

활용 시 유의점

- 교사는 수업 전에 빈칸이 표시된 엔트리 예시 프로그램을 준비한다.
- 학생들이 처음부터 전체 프로그램을 만들기보다, 알고리즘과 연결되는 핵심 빈칸만 채우도록 한다.
- 스마트기기 사용이 어려운 환경에서는 엔트리 블록 화면을 출력하여 빈칸 채우기 활동으로 대체할 수 있다.

7. 친구의 프로그램을 탐색하고 의견을 나눠볼까요?

- **관련 성취 기준 :** [06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.
- **학습 목표 :** 친구의 프로그램을 살펴보고, 좋은 점과 그 이유를 말할 수 있다.
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 기본 프로그램의 아쉬운 점 생각하기 • 학습문제 확인하기	• 추가할 기능 생각하기 • 선택한 기능을 프로그램에 추가하기 • 친구의 프로그램을 탐색하고 의견 나누기	• 기능 추가와 프로그램 탐색 방법 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 기본 프로그램을 사용자의 관점에서 살펴보고, 더 필요한 기능이 무엇인지 생각하도록 한다. • 교사가 제시한 기능 예시와 학생의 아이디어 중에서 한 가지를 선택하여 프로그램에 추가하도록 한다. • 친구의 프로그램을 탐색하며 추가된 기능의 좋은 점과 그 이유를 구체적으로 말하도록 한다.	• 기본 프로그램이 틀렸다는 관점이 아니라, 사용자에게 더 도움이 되도록 기능을 추가하는 관점으로 접근한다. • 모든 학생이 어려운 기능을 추가하도록 요구하지 않고, 간단한 문장 수정이나 말하기 블록 추가도 기능 개선으로 인정한다. • 친구의 프로그램을 탐색할 때에는 추가된 기능이 사용자에게 어떤 도움을 주는지 중심으로 살펴보게 한다.	• 교사: 스마트기기(엔트리) • 학생: 스마트기기(엔트리), 활동지

교수·학습 과정

<table>
<tr><th>학습 단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td>도입</td><td>◆ 기본 프로그램의 아쉬운 점 생각하기
T : 지난 시간에 만든 교실 환기 안내 프로그램은 어떤 일을 했나요?
S : 환기한 지 몇 분이 지났는지 물었습니다.
T : 이 프로그램을 사용하는 사람이 더 알고 싶어 할 수 있는 정보는 무엇일까요?
S : 몇 분 후에 창문을 열면 되는지 알고 싶을 것 같습니다.
S : 소리로 알려 주면 더 잘 알 수 있을 것 같습니다.
S : 남은 시간을 타이머로 알려 주면 좋겠습니다.

◆ 학습문제 확인하기
프로그램에 필요한 기능을 추가하고 친구의 프로그램을 탐색해봅시다.</td><td>• 기본 프로그램은 이미 작동하는 프로그램이며, 이번 차시는 더 유용하게 만드는 활동임을 안내한다.

• 기능 추가는 어려운 새 프로그램 만들기가 아니라, 사용자가 더 편리하게 쓸 수 있도록 보완하는 활동임을 안내한다.</td></tr>
<tr><td>전개</td><td>◆ 활동1. 추가할 기능 생각하기
T : 교실 환기 안내 프로그램에 어떤 기능을 추가하면 더 유용할까요?
S : 아직 120분이 되지 않았을 때 몇 분 후에 창문을 열면 되는지 알려 주면 좋겠습니다.
S : 120분 이상일 때 소리로 알려 주면 좋겠습니다.
T : 다음 기능 예시를 살펴보고, 내가 추가하고 싶은 기능을 골라 봅시다.
<table><tr><td>예시1</td><td>남은 시간 안내 기능
120분이 지나지 않았을 때, 몇 분 후에 창문을 열면 되는지 알려 준다.
예) 100을 입력하면 “20분 후에 열어주세요.”라고 말한다.</td></tr><tr><td>예시2</td><td>환기 후 행동 안내 기능
120분 이상이 되었을 때, 창문을 열라는 안내와 함께 환기할 때 필요한 행동을 알려 준다.
예) “창문을 열어주세요.”라고 말한 뒤 “5분 동안 환기해 주세요.”라고 말한다.</td></tr></table>T : 두 가지 예시 외에 내가 생각한 기능이 있나요?
S : 앞선 시간에 배운 타이머 기능을 활용하여 남은 시간을 알려 주면 좋겠습니다.</td><td>• 준비물: 활동지, 스마트기기(엔트리)
엔트리 완성 파일
https://naver.me/5B0MZem3

• 학생 아이디어는 문장 수정, 소리 추가, 기준 시간 변경처럼 간단한 수준도 허용한다.

• 기능 예시는 2가지로 제한하여 학생들이 선택하기 쉽게 한다.</td></tr>
</table>

◆ 활동2. 선택한 기능을 프로그램에 추가하기

T : 내가 선택한 기능을 프로그램에 추가해봅시다.

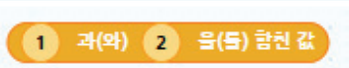
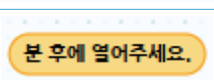
T : 남은 시간 안내 기능을 선택한 경우, 120분이 되기까지 몇 분이 남았는지 어떻게 구할 수 있을까요?

S : 120에서 입력한 값을 빼면 됩니다.

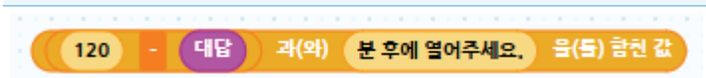
S : 120 – 대답으로 계산할 수 있습니다.

예) 남은 시간 안내 기능: 기존 문장: “아직은 창문을 열지 마세요.”
수정 문장: 120 – 대답 과(와) “분 후에 열어주세요.”를 합친 값 말하기

필요한 블록

①+②	
①	
②	

완성 블록



T : 환기 후 행동 안내 기능을 선택한 경우, 어떤 말하기 블록을 추가하면 좋을까요?

S : “10분 동안 환기해 주세요.”를 추가하면 좋겠습니다.

완성 블록



◆ 활동3. 친구의 프로그램을 탐색하고 의견 나누기

T : 친구가 추가한 기능을 실행해 보고, 어떤 점이 더 좋아졌는지 살펴보고 활동지에 적어봅시다.

T : 활동지에 적은 내용을 보고 내 프로그램에도 추가하고 싶은 것을 써 봅시다.

- 준비물: 활동지, 스마트기기(엔트리)
- 학생이 기능을 선택하여 추가하도록 하되, 구현이 어려운 경우 교사가 준비한 예시 블록을 활용하게 한다.
- 남은 시간 안내 기능은 계산식의 의미를 이해하는 데 초점을 두고, 블록 조립이 지나치게 어렵지 않도록 지원한다.
- 다른 기능을 추가하고자 하는 학생이 있다면 스스로 도전하도록 독려한다.
- 좋은 점과 개선할 점을 각각 1가지 이상 찾도록 하되, 개선 의견은 구체적으로 쓰게 한다.

※ 교과서 71페이지 정답
① 유용하고 편리
② 이유

정리

◆ 기능 추가와 프로그램 탐색 의견 정리하기

T : 오늘 기본 프로그램에 어떤 기능을 추가해 보았는지 발표해봅시다.

T : 친구의 프로그램의 좋은 점과 그 이유를 발표해봅시다.

◆ 교실 환기 안내 프로그램 저장하기

T : 제작한 프로그램을 저장해봅시다.

- 친구의 프로그램 탐색과 의견 나누기는 프로그램을 함께 발전시키기 위한 활동임을 강조한다.

수업 활동지

친구의 프로그램을 탐색하고 의견을 나눠볼까요?

1. 기본 교실 환기 안내 프로그램에 추가하면 좋을 기능을 생각하여 표 안의 빈칸에 적어봅시다.

추가 기능	기능 설명
남은 시간 안내 기능	120분이 되기까지 몇 분이 남았는지 알려 준다.
환기 후 행동 안내 기능	창문을 연 뒤 무엇을 하면 좋은지 알려 준다.
나의 아이디어	

2. 내가 추가할 기능을 선택하고 이 기능을 추가하면 좋은 점을 써 봅시다.

내가 추가할 기능	
이 기능을 추가하면 좋은 점	

3. 친구가 추가한 기능이 무엇인지 확인해봅시다.

친구가 추가한 기능		
확인할 내용	추가한 기능이 실행되나요?	예 □ | 아니오 □
	사용자에게 더 도움이 되나요?	예 □ | 아니오 □
	기능을 이해하고 사용하기 쉬운가요?	예 □ | 아니오 □

4. 친구 프로그램의 좋은 점과 그 이유를 적어봅시다.

좋은 점	
이유	

수업 평가지

1. 다음 중 교실 환기 안내 프로그램에 추가하면 좋은 기능으로 가장 알맞은 것은 무엇입니까?

① 프로그램 제목만 더 길게 바꾼다.
② 사용자가 입력한 값을 사용하지 않는다.
③ 환기와 관련 없는 게임 점수를 보여 준다.
④ 몇 분 후에 창문을 열면 되는지 알려 준다.
⑤ 입력값과 관계없이 항상 같은 말을 하게 한다.

정답 : _____

2. 남은 시간 안내 기능을 추가하였습니다. 입력값이 100일 때 알맞은 안내 문장은 무엇입니까?

① 창문을 닫아주세요.
② 20분 후에 열어주세요.
③ 100분 후에 열어주세요.
④ 120분 후에 열어주세요.
⑤ 아직은 입력하지 마세요.

정답 : _____

3. 친구의 프로그램을 탐색하고 의견을 나눌 때 주의할 점으로 바르지 않은 것은 무엇입니까?

① 친구의 프로그램을 존중하며 살펴본다.
② 추가된 기능이 사용자에게 도움이 되는지 확인한다.
③ 좋은 점을 먼저 찾고, 개선할 점은 구체적으로 말한다.
④ 마음에 들지 않는 부분이 있으면 "별로야"라고 말한다.
⑤ 프로그램을 직접 고치기보다 좋은 점과 개선할 점을 기록한다.

정답 : _____

4. 친구가 만든 프로그램의 좋은 점과 그 이유를 한 가지 이상 써봅시다.

좋은 점	
이유	

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. ④
2. ②
3. ④
4. 남은 시간을 알려 주어서 더 편리한 점이 좋았습니다.
   또는
   환기 후 행동을 안내해 주어서 사용자가 이해하기 쉬운 점이 좋았습니다.
   또는
   안내 문장을 조금 더 짧고 분명하게 바꾸면 좋을 것 같습니다.

평가 목표

기본 프로그램에 필요한 기능을 추가하고, 친구의 프로그램을 탐색하여 좋은 점과 개선할 점을 구체적으로 말할 수 있다.

평가 기준

수준	평가 기준
잘함	기본 프로그램에 필요한 기능을 적절히 추가하고, 친구 프로그램의 좋은 점과 그 이유를 구체적으로 말할 수 있다.
보통	기본 프로그램에 기능을 추가할 수 있고, 친구 프로그램의 좋은 점과 그 이유를 간단히 말할 수 있다.
노력 요함	기능 추가나 친구 프로그램 탐색, 의견 표현에 교사의 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
추가할 기능을 고르지 못하는 경우	사용자가 프로그램을 쓸 때 더 알고 싶어 할 정보를 생각해 봅시다. 몇 분 후에 창문을 열면 되는지, 창문을 연 뒤 무엇을 하면 좋은지 생각해 볼 수 있습니다.
기능 추가를 꾸미기와만 연결하는 경우	배경이나 글자 모양도 바꿀 수 있지만, 이번 시간에는 사용자가 더 편리하게 쓸 수 있는 기능을 생각해 봅시다.

수업 운영 팁

Q1. 학생들이 어떤 기능을 추가해야 할지 어려워하면 어떻게 하나요?

A1. 사용자의 입장에서 "이 프로그램을 사용할 때 더 알고 싶은 정보가 무엇일까?"라고 질문합니다. 교실 환기 안내 프로그램에서는 몇 분 후에 창문을 열면 되는지, 창문을 연 뒤 무엇을 하면 좋은지, 안내 문장이 충분히 친절한지 등을 생각하게 할 수 있습니다. 학생이 어려워하면 교사가 제시한 두 가지 기능 중 하나를 선택하게 합니다.

Q2. 학생들이 너무 어려운 기능을 추가하려고 하면 어떻게 하나요?

A2. 학생의 아이디어를 부정하지 말고, 한 차시 안에 만들 수 있는 작은 기능으로 범위를 줄이도록 안내합니다. 예를 들어 "자동으로 시간을 재는 타이머"는 구현이 복잡하므로, 이번 시간에는 "몇 분 후에 열면 되는지 안내하기"나 "환기 후 행동을 말하기"처럼 간단한 기능으로 바꾸어 구현하게 합니다.

Q3. 학생들이 친구 프로그램을 탐색할 때 코드만 보려고 하면 어떻게 하나요?

A3. 먼저 사용자의 입장에서 실행해 보고, 추가된 기능이 실제로 도움이 되는지 살펴보게 합니다. 코드가 복잡한지보다, 입력값에 따라 안내가 잘 나오는지, 안내 문장이 이해하기 쉬운지, 사용자가 더 편리하게 쓸 수 있는지를 기준으로 의견을 남기게 합니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치, MakeCode	알고리즘을 실제 프로그램으로 구현
공유 전시 도구	엔트리, 구글 드라이브, 패들렛 등	완성한 프로그램을 저장 및 친구들과 공유 및 재창작

활용 시 유의점

- 교사는 수업 전에 기본 프로그램과 기능 추가 예시를 준비한다.
- 학생들이 모든 기능을 새로 만들기보다, 교사가 제시한 기능 예시 중 하나를 선택하여 추가할 수 있게 한다.
- 남은 시간 안내 기능은 블록 조립이 어려울 수 있으므로 예시 블록을 제공할 수 있다.
- 친구 프로그램을 탐색할 때는 직접 수정하지 않고, 실행 결과와 의견을 활동지에 기록하게 한다.
- 디지털 공유가 어려운 경우에는 모둠별로 한 기기에서 프로그램을 실행하고, 다른 학생들이 결과를 관찰하며 의견을 쓰도록 운영할 수 있다.

8. 공유된 프로그램을 더 유용하게 수정해 볼까요?

- **관련 성취 기준 :** [06자율-4] 공유된 타인의 프로그램을 탐색해 보고 수정하는 활동을 수행한다.
- **학습 목표 :** 공유된 프로그램을 탐색하고, 친구의 의견을 바탕으로 프로그램을 수정할 수 있다
- **수업 로드맵**

도입 (5분)	전개 (30분)	정리 (5분)
• 친구에게 받은 의견 떠올리기 • 학습문제 확인하기	• 수정할 부분 정하기 • 프로그램을 수정하고 실행해 보기 • 수정 전후를 비교하여 공유하기	• 프로그램 수정과 공유 과정 정리하기

교수·학습 방향	지도상의 유의점	수업 준비물
• 친구에게 받은 의견을 바탕으로 프로그램을 더 유용하게 수정하는 경험을 하도록 한다. • 프로그램을 무작정 바꾸기보다, 사용자에게 어떤 도움이 되는지 생각하며 수정 계획을 세우도록 한다. • 수정 전후를 비교하며 프로그램이 어떻게 개선되었는지 말로 설명하도록 한다.	• 수정 활동은 완전히 새로운 프로그램을 만드는 것이 아니라, 기존 프로그램을 더 유용하게 발전시키는 활동임을 안내한다. • 학생이 모든 의견을 반영해야 한다고 느끼지 않도록 한다. • 타이머 기능처럼 반복, 기다리기, 시간 감소가 복합적으로 필요한 기능은 심화 활동으로만 안내한다. • 친구의 프로그램을 참고하거나 수정할 때에는 원래 만든 사람의 아이디어를 존중하도록 지도한다.	• 교사: 스마트기기(엔트리) • 학생: 스마트기기(엔트리), 활동지

교수·학습 과정

<table>
<tr><th>학습
단계</th><th>교수·학습 활동</th><th>자료 및 유의점</th></tr>
<tr><td rowspan="2">도입</td><td>◆ 친구의 의견 떠올리기
T : 지난 시간에 친구의 프로그램을 탐색하며 어떤 의견을 주고받았나요?
S : 남은 시간을 알려 주면 더 편리하다는 의견을 들었습니다.
S : 안내 문장을 더 친절하게 바꾸면 좋겠다는 의견을 들었습니다.
S : 환기 후 행동을 알려 주면 좋겠다는 의견을 들었습니다.
S : 소리가 나면 더 쉽게 알 수 있을 것 같다는 의견을 들었습니다.
T : 친구의 의견을 모두 반영해야 할까요?
S : 모두 반영하지 않아도 됩니다.
S : 필요한 의견을 골라서 반영하면 됩니다.
S : 내가 만들고 싶은 방향에 맞는 의견을 선택하면 됩니다.</td><td rowspan="2">• 친구의 의견은 프로그램을 더 좋게 만들기 위한 참고 자료임을 안내한다.

• 의견을 반영할 때에는 사용자에게 어떤 도움이 되는지 생각하게 한다.</td></tr>
<tr><td>◆ 학습문제 확인하기
공유된 프로그램을 탐색하고 친구의 의견을 바탕으로 프로그램을 수정해 봅시다.</td></tr>
<tr><td>전개</td><td>◆ 활동1. 수정할 부분 정하기
T : 내 프로그램에서 더 좋게 바꾸고 싶은 부분을 친구들의 의견을 반영하여 한 가지 정해봅시다.
<table><tr><th>수정할 부분</th><th>수정 방향</th></tr><tr><td>안내 문장</td><td>사용자가 이해하기 쉽게 더 친절하게 바꾼다.</td></tr><tr><td>남은 시간 안내</td><td>120분이 되기까지 몇 분 남았는지 알려 준다.</td></tr><tr><td>환기 후 행동 안내</td><td>창문을 연 뒤 무엇을 하면 좋은지 알려 준다.</td></tr><tr><td>소리 알림</td><td>환기 시간이 되었을 때 소리로 알려 준다.</td></tr><tr><td>기준 시간</td><td>교실 상황에 맞게 환기 기준 시간을 조정한다.</td></tr></table>T : 이 수정이 사용자에게 어떤 도움을 줄 수 있을까요?
S : 몇 분 뒤에 창문을 열면 되는지 알 수 있습니다.
S : 환기할 때 무엇을 해야 하는지 알 수 있습니다.
S : 안내 문장을 더 쉽게 이해할 수 있습니다.
S : 소리가 나서 환기 시간을 더 잘 알 수 있습니다.
◆ 활동2. 프로그램을 수정하고 실행해 보기
T : 내가 정한 수정 방향에 맞게 프로그램을 고쳐봅시다.</td><td>• 준비물: 활동지, 스마트기기(엔트리)

• 수정할 부분은 한 차시 안에 완성할 수 있도록 한다.

• 기능의 복잡성보다 사용자에게 주는 도움이 무엇인지 설명하게 한다.

• 친구의 의견과 자신의 생각을 연결하여 수정 방향을 정하게 한다.

• 준비물: 활동지, 스마트기기(엔트리)</td></tr>
</table>

	T : 수정한 프로그램을 실행해 보고, 의도한 대로 작동하는지 확인해 봅시다. 확인할 내용 입력값에 따라 결과가 알맞게 나오나요? 추가하거나 수정한 기능이 잘 실행되나요? 수정한 부분이 원래 프로그램의 목적과 어울리나요? ◆ 활동3. 수정 전후를 비교하여 공유하기 T : 수정하기 전과 수정한 후의 프로그램을 비교해봅시다. 무엇을 바꾸고 어떻게 더 좋아졌나요? S : 안내 문장을 더 친절하게 바꿔 사용자가 더 쉽게 이해할 수 있습니다. S : 120분이 되지 않았을 때 남은 시간을 알려 주도록 바꿔 몇 분 후에 창문을 열면 되는지 알 수 있습니다. S : 창문을 연 뒤 5분 동안 환기하라는 문장을 추가했습니다. S : 환기 시간이 되었을 때 소리가 나도록 하여 환기 시간이 되었음을 더 잘 알 수 있습니다. T : 수정한 후의 프로그램을 공유하고 친구들과 서로 의견을 나눠봅시다.	• 발표는 짧게 하되, 수정한 부분과 수정 이유를 반드시 말하게 한다. • 단순히 "좋아졌습니다"라고 말하기보다, 사용자에게 어떤 도움이 되는지 설명하게 한다. • 친구의 수정 결과를 들을 때에는 좋은 점을 중심으로 반응하게 한다. ※ 교과서 75페이지 정답 ① 공유 ② 실행
정리	◆ 프로그램 수정과 공유 과정 정리하기 T : 프로그램을 수정할 때 중요한 점은 무엇인가요? S : 사용자의 입장에서 필요한 기능을 생각해야 합니다. S : 수정한 뒤 실행해 보며 잘 작동하는지 확인해야 합니다. T : 친구와 프로그램을 공유하면 어떤 점이 좋나요? S : 프로그램을 더 좋게 만들 수 있습니다. S : 서로 도와가며 프로그램을 발전시킬 수 있습니다. ◆ 교실 환기 안내 프로그램 저장하기 T : 제작한 프로그램을 저장해봅시다.	• 프로그램 수정은 오류를 고치는 활동뿐만 아니라, 더 유용하게 발전시키는 활동임을 정리한다. • 공유와 피드백을 통해 프로그램을 개선할 수 있음을 강조한다.

수업 활동지

공유된 프로그램을 더 유용하게 수정해 볼까요?

1. 친구의 의견을 바탕으로 내 프로그램에 반영할 부분을 선택해 봅시다.

친구의 의견	*예) 안내문장을 더 친절하게 바꿔줘.*
반영할 부분	*예) "10분 뒤에 환기해주세요"로 문장을 바꾼다.*

2. 프로그램을 수정한 뒤 실행 결과를 확인해 봅시다.

입력값에 따라 결과가 알맞게 나오나요?	예 □ | 아니오 □
추가하거나 수정한 기능이 잘 실행되나요?	예 □ | 아니오 □
수정한 부분이 원래 프로그램의 목적과 어울리나요?	예 □ | 아니오 □

3. 수정 전과 수정 후를 비교해 봅시다.

수정 전에는		했습니다.
수정 후에는		합니다.
그래서 사용자는		할 수 있습니다.

수업 평가지

1. 프로그램을 수정할 때 친구들의 의견을 반영하면 좋은 점을 써 봅시다.

2. 친구의 의견 중 어떤 의견을 반영했는지 써 봅시다.

3. 프로그램을 실행한 후 결과를 확인해 봅시다.

입력값에 따라 결과가 알맞게 나오나요?	예 □ | 아니오 □
추가하거나 수정한 기능이 잘 실행되나요?	예 □ | 아니오 □
수정한 부분이 원래 프로그램의 목적과 어울리나요?	예 □ | 아니오 □

4. 친구가 공유한 프로그램에서 어떤 기능을 추가하면 좋을지 써 봅시다.

수업 평가지 예시 답안 및 채점 기준

예시 답안

1. 내가 생각하지 못한 부분을 알 수 있어 사용자에게 더 도움이 되는 프로그램으로 수정할 수 있었다 등
2. 안내 문장을 더 친절하게 바꿔 달라는 의견을 반영했다, 안내 문장을 소리로 구현하면 좋겠다는 의견을 반영했다 등
3. 실행 후 세 가지 확인 항목의 점검 방법 안내
4. 환기 시간이 되었을 때 소리로 알려 주는 기능, 화면에 크게 환기하라는 멘트가 나오는 기능, 환기하라는 멘트를 확인하지 않으면 다시 한번 알려주는 기능 등

평가 목표

친구의 의견을 바탕으로 프로그램을 더 유용하게 수정하고, 수정한 부분과 이유를 설명할 수 있다.

평가 기준

수준	평가 기준
잘함	친구의 의견이나 사용자 관점을 반영하여 프로그램을 적절히 수정하고, 수정한 부분과 이유를 구체적으로 설명할 수 있다.
보통	프로그램의 일부를 수정하고, 수정한 부분이나 이유를 간단히 설명할 수 있다.
노력 요함	프로그램 수정 방향을 정하거나 수정한 이유를 설명하는 데 교사의 도움이 필요하다.

맞춤형 피드백 예시

학생 반응	피드백 예시
수정할 부분을 정하지 못하는 경우	친구가 말해 준 개선할 점을 다시 살펴봅시다. 사용자가 더 편리하게 쓸 수 있는 부분을 한 가지 골라봅시다.
꾸미기만 수정하려는 경우	꾸미기도 좋지만, 이번에는 사용자가 더 쉽게 이해하거나 편리하게 사용할 수 있는 기능을 생각해봅시다.

수업 운영 팁

Q1. 학생들이 프로그램을 어떻게 수정해야 할지 어려워하면 어떻게 하나요?

A1. 친구에게 받은 개선 의견을 먼저 다시 보게 합니다. 의견이 많다면 한 가지만 선택하게 하고, 안내 문장 수정, 남은 시간 안내, 환기 후 행동 안내, 소리 알림처럼 간단한 예시 중에서 고르게 합니다. 수정할 부분을 작게 정하면 한 차시 안에 완성하기 쉽습니다.

Q2. 학생들이 친구의 의견을 모두 반영하려고 하면 어떻게 하나요?

A2. 모든 의견을 반영할 필요는 없다고 안내합니다. 프로그램의 목적과 맞고, 한 차시 안에 수정할 수 있으며, 사용자에게 도움이 되는 의견을 한 가지 선택하게 합니다. 선택한 의견을 반영한 이유를 설명할 수 있으면 충분합니다.

Q3. 학생들이 타이머 기능처럼 어려운 기능을 만들고 싶어 하면 어떻게 하나요?

A3. 타이머 기능은 반복, 기다리기, 시간 감소, 화면 표시 등이 함께 필요하므로 난이도가 높다고 안내합니다. 아이디어 자체는 좋지만 이번 시간에는 "남은 시간 안내하기"처럼 더 간단한 기능으로 줄여서 구현하게 합니다. 빠르게 완성한 학생에게는 심화 아이디어로 안내할 수 있습니다.

디지털 도구 활용 가이드

활용 가능한 도구

도구 유형	도구 예시	활용 방법
블록 기반 프로그래밍 언어	엔트리, 스크래치, MakeCode	아이디어를 수정 및 반영
공유 전시 도구	엔트리 작품 공유하기, 구글 드라이브, 패들렛 등	완성한 프로그램을 저장 및 친구들과 공유, 리메이크

활용 시 유의점

- 수업 전에 학생들이 수정할 수 있는 기본 프로그램 또는 복사본을 준비한다.
- 수정 전 프로그램을 보존할 수 있도록 필요한 경우 복사본을 만들어 수정하게 한다.
- 수정 활동은 한 가지 기능 또는 한 가지 문장 개선으로 제한하여 수업 시간 안에 완성할 수 있게 한다.
- 수정 후에는 반드시 프로그램을 실행하여 기존 기능과 새 기능이 모두 잘 작동하는지 확인하게 한다.
- 공유할 때에는 수정한 부분, 수정한 이유, 사용자에게 도움이 되는 점을 함께 말하게 한다.

III

부록



1. 정보 교과 교수·학습 모형 ····· 144

2. 수업 배경지식 ····· 154

3. 정보 용어 사전 ····· 165

4. 저작권 및 AI 윤리 자료 ····· 167

5. 참고 문헌 및 사이트 ····· 170

6. 생성형 AI 활용 내역 ····· 172

1 정보 교과 교수·학습 모형

1-1 PRIMM 모형(Sentance & Waite, 2017)

① 특징

PRIMM 모형은 프로그래밍 교육에서 학생들이 프로그램을 단순히 실행하거나 코드를 암기하는 데 그치지 않고 코드의 의미와 구조를 단계적으로 이해하도록 돕기 위해 제안된 교수·학습 모형이다. 학생들은 먼저 주어진 코드의 실행 결과를 예측하고 실제 실행 결과와 비교하며 코드의 구성 요소를 조사하고 일부를 수정한 뒤 최종적으로 자신만의 프로그램을 만든다.

이 모형은 처음부터 빈 화면에서 코드를 작성하게 하기보다, 완성되었거나 부분적으로 완성된 코드를 바탕으로 학습을 시작한다는 점에서 초등학생의 프로그래밍 학습 부담을 줄이는 데 효과적이다. PRIMM 모형은 학생들이 프로그램의 동작을 추측하고 실행 결과를 확인하며 코드의 구조를 분석하는 과정을 통해 컴퓨팅사고력의 주요 요소인 알고리즘적 사고, 추상화, 디버깅, 일반화 능력을 기를 수 있도록 한다. 또한 기존 코드를 수정하고 새롭게 만드는 과정에서 창의적 문제 해결 경험을 제공할 수 있다.

② 학습 과정

단계	설명
Predict (예측하기)	• 교사는 학생들에게 완성된 코드나 일부 코드가 포함된 프로그램을 제시한다. • 학생들은 코드를 실행하기 전에 프로그램이 어떤 결과를 나타낼지 예상한다. • 이때 교사는 “이 코드를 실행하면 어떤 일이 일어날까요?”, “반복 블록은 몇 번 실행될까요?”, “조건이 참일 때와 거짓일 때 결과가 어떻게 달라질까요?”와 같은 발문을 통해 학생들이 코드의 흐름을 생각하도록 돕는다.
Run (실행하기)	• 학생들은 주어진 코드를 실제로 실행해 보고 자신이 예측한 결과와 비교한다. • 교사는 학생들이 단순히 결과만 확인하지 않도록 하고, 예측과 실제 결과가 같거나 다른 이유를 말해 보게 한다. • 이 과정에서 학생들은 코드의 실행 순서, 입력과 출력, 조건과 반복의 작동 방식을 경험적으로 확인한다.
Investigate (조사하기)	• 학생들은 코드의 각 부분이 어떤 역할을 하는지 자세히 살펴본다. • 교사는 변수, 조건, 반복, 이벤트, 함수, 입력값, 출력 결과 등 주요 요소를 중심으로 코드를 나누어 분석하도록 안내한다. • 학생들은 “이 블록은 어떤 역할을 하나요?”, “이 값을 바꾸면 결과가 어떻게 달라질까요?”, “이 명령이 없으면 프로그램은 어떻게 실행될까요?”와 같은 질문을 통해 코드의 의미를 탐구한다.
Modify (수정하기)	• 학생들은 기존 코드를 일부 수정하여 새로운 결과를 만들어 본다. • 교사는 수정 범위를 적절히 제시하여 학생들이 지나치게 막막함을 느끼지 않도록 한다. • 예를 들어 캐릭터의 움직임 바꾸기, 반복 횟수 변경하기, 조건 추가하기, 출력 메시지 바꾸기, 새로운 기능 한 가지 추가하기 등의 활동을 할 수 있다. • 학생들은 수정한 코드가 예상대로 작동하는지 확인하고 오류가 발생하면 원인을 찾아 고쳐 본다.
Make (만들기)	• 학생들은 앞선 단계에서 이해하고 수정한 내용을 바탕으로 자신의 아이디어를 반영한 프로그램을 만든다. • 교사는 학생들이 완전히 새로운 프로그램을 만들기 전에 해결할 문제, 사용할 명령, 필요한 조건과 반복, 예상되는 실행 결과를 간단히 설계하도록 지도한다. • 학생들은 자신만의 프로그램을 작성하고 실행해 보며, 친구들과 결과를 공유하고 개선점을 찾아 수정한다.

③ 모형 적용상의 유의점

- PRIMM 모형을 적용할 때에는 학생들이 처음부터 코드를 작성하는 데 부담을 느끼지 않도록 학습 목표에 맞는 적절한 예제 코드를 제공해야 한다. 예제 코드는 너무 복잡하지 않아야 하며, 학생들이 눈으로 읽고 실행 결과를 예측할 수 있을 정도의 수준이어야 한다.
- 예측하기 단계에서는 정답을 맞히는 것보다 코드의 흐름을 생각해 보는 과정이 중요하다. 따라서 교사는 학생들의 다양한 예측을 허용하고, 왜 그렇게 생각했는지 설명하도록 유도해야 한다. 예측이 틀린 경우에도 오류로 지적하기보다 실제 실행 결과와 비교하면서 코드 이해를 심화할 수 있도록 지도한다.
- 실행하기 단계에서는 단순히 "실행해 보기"에 그치지 않도록 유의한다. 학생들이 자신의 예측과 실제 결과를 비교하고, 차이가 발생한 원인을 코드에서 찾아볼 수 있도록 해야 한다. 이를 위해 실행 전 예상 결과를 간단히 기록하게 하거나, 실행 후 달라진 점을 말로 설명하게 할 수 있다.
- 조사하기 단계에서는 코드의 모든 요소를 한꺼번에 분석하기보다, 수업 목표와 관련된 핵심 개념을 중심으로 살펴보도록 한다. 예를 들어 반복을 배우는 수업에서는 반복 횟수와 실행 순서에 초점을 두고, 조건을 배우는 수업에서는 조건식과 분기 결과에 초점을 두는 것이 적절하다.
- 수정하기 단계에서는 학생들의 수준에 따라 수정 범위를 조절해야 한다. 초등학생에게는 "한 가지 값 바꾸기", "블록 하나 추가하기", "조건 하나 바꾸기"와 같이 구체적인 수정 과제를 제시하는 것이 효과적이다. 이후 점차 학생 스스로 수정할 부분을 선택하도록 확장할 수 있다.
- 만들기 단계에서는 완성도 높은 산출물보다 학생이 코드의 원리를 이해하고 자신의 아이디어를 프로그램으로 표현하는 과정에 초점을 두어야 한다. 교사는 학생들이 기존 코드를 참고하되 단순 복사에 머무르지 않도록 하고, 자신이 수정하거나 새롭게 만든 부분을 설명하게 함으로써 코드 이해와 표현을 함께 평가할 수 있다.

1-2 UMC모형(Lee 등, 2011)

① 특징

UMC 모형은 컴퓨터 교육에서 학생들이 컴퓨팅 사고력과 프로그래밍 역량을 점진적으로 기를 수 있도록 설계된 교수·학습 모형이다. UMC는 Use, Modify, Create의 세 단계로 구성되며, 학생들이 기존 프로그램을 사용해 보고, 이를 수정해 보며, 최종적으로 자신만의 프로그램이나 산출물을 창작하도록 안내한다.

이 모형은 처음부터 학생들에게 새로운 프로그램을 만들도록 요구하기보다, 이미 완성된 프로그램이나 예제 코드를 먼저 경험하게 한다는 점에서 초등학생에게 적합하다. 학생들은 프로그램을 실행하면서 동작 원리를 관찰하고, 일부 요소를 바꾸어 보면서 코드의 의미를 이해한 뒤, 자신만의 아이디어를 반영한 프로젝트를 설계하고 구현하게 된다.

UMC 모형은 프로그래밍 학습에서 학생들이 느끼는 인지적 부담을 줄이고 단계적으로 자신감을 형성할 수 있도록 돕는다. 또한 학생들이 기존 코드를 단순히 따라 하는 데 그치지 않고 코드를 변형하고 확장하며 자신의 문제 해결 아이디어를 프로그램으로 표현하도록 한다. 특히 초등학교 정보 수업에서는 학생들이 프로그램의 구조를 직접 설계하기 전에 먼저 프로그램의 기능을 체험하고, 작은 수정을 통해 코드와 결과 사이의 관계를 이해한 뒤, 창의적 산출물을 만드는 과정으로 수업을 전개할 수 있다는 장점이 있다.

② 학습 과정

단계	설명
Use (사용)	• 교사는 학습 목표와 관련된 예제 프로그램, 완성된 코드, 블록 조합, 피지컬 컴퓨팅 장치의 작동 예시 등을 제시한다. • 학생들은 제시된 프로그램을 실행해 보고, 프로그램이 어떤 기능을 수행하는지 관찰한다. 이 단계에서는 코드를 자세히 수정하기보다 프로그램의 목적, 입력과 출력, 실행 결과, 주요 기능을 파악하는 데 초점을 둔다. • 교사는 “이 프로그램은 어떤 일을 하나요?”, “입력하면 어떤 결과가 나타나나요?”, “어떤 순서로 동작하는 것 같나요?”와 같은 발문을 통해 학생들이 프로그램의 전체적인 흐름을 이해하도록 돕는다.
Modify (수정)	• 학생들은 기존 프로그램의 일부를 수정하여 실행 결과가 어떻게 달라지는지 확인한다. • 교사는 학생들의 수준에 따라 수정할 부분을 제한하거나 선택지를 제공할 수 있다. 예를 들어 캐릭터의 움직임 바꾸기, 반복 횟수 변경하기, 조건 추가하기, 변수값 조정하기, 출력 메시지 바꾸기, 센서 입력값에 따른 동작 바꾸기 등의 활동을 할 수 있다. • 학생들은 수정 전과 수정 후의 결과를 비교하면서 코드와 실행 결과 사이의 관계를 이해한다. 이 과정에서 오류가 발생하면 원인을 찾고 다시 수정하면서 디버깅 경험을 하게 된다.
Create (창작)	• 학생들은 앞선 단계에서 사용하고 수정한 경험을 바탕으로 자신만의 프로그램이나 프로젝트를 설계하고 구현한다. • 교사는 학생들이 해결하고 싶은 문제, 만들고 싶은 기능, 사용할 명령이나 블록, 필요한 입력과 출력 등을 간단히 계획하도록 지도한다. • 학생들은 자신의 아이디어를 반영하여 프로그램을 만들고, 실행 결과를 확인하며, 필요에 따라 개선한다. 완성 후에는 친구들과 결과물을 공유하고, 자신이 만든 프로그램의 기능과 수정·창작한 부분을 설명한다.

③ 모형 적용상의 유의점

- UMC 모형을 적용할 때에는 학생들이 처음 접하는 프로그램이 지나치게 복잡하지 않도록 해야 한다. Use 단계에서 제시하는 예제는 학생들이 실행 결과를 쉽게 관찰할 수 있고, 주요 기능을 직관적으로 이해할 수 있는 수준이어야 한다. 예제 프로그램이 너무 어렵거나 코드의 양이 많으면 학생들이 프로그램의 전체 구조를 파악하기보다 실행 결과만 보는 데 그칠 수 있다.
- Use 단계에서는 학생들이 단순히 프로그램을 실행해 보는 활동에 머무르지 않도록 해야 한다. 교사는 프로그램의 목적, 입력과 출력, 실행 순서, 주요 기능을 말이나 그림으로 설명하게 하여 학생들이 프로그램의 동작 방식을 이해하도록 지도한다. 필요한 경우 실행 결과를 관찰한 뒤 "무엇이 반복되었는가?", "어떤 조건에서 결과가 달라졌는가?", "사용자가 조작한 부분은 무엇인가?"와 같은 질문을 제시할 수 있다.
- Modify 단계에서는 수정 범위를 학생 수준에 맞게 조절하는 것이 중요하다. 초등학생에게는 처음부터 자유롭게 수정하도록 하기보다, "숫자 하나 바꾸기", "명령 블록 하나 추가하기", "조건 하나 바꾸기"와 같이 구체적인 수정 과제를 제시하는 것이 효과적이다. 이후 학생들이 익숙해지면 스스로 수정할 부분을 선택하거나 새로운 기능을 추가하도록 확장할 수 있다.
- 수정 활동에서는 학생들이 수정한 내용과 실행 결과의 변화를 연결하여 설명하도록 해야 한다. 단순히 결과가 바뀌었다는 사실보다, 어떤 코드를 어떻게 바꾸었기 때문에 결과가 달라졌는지를 말할 수 있도록 지도해야 한다. 이를 통해 학생들은 코드의 의미를 이해하고, 프로그램의 동작 원리를 점진적으로 파악할 수 있다.
- Create 단계에서는 산출물의 완성도보다 학생이 자신의 아이디어를 프로그램으로 표현하는 과정에 초점을 두어야 한다. 교사는 학생들이 기존 예제를 그대로 복사하지 않고, 자신이 새롭게 구성하거나 변형한 부분을 포함하도록 안내해야 한다. 또한 학생들이 계획, 구현, 실행, 수정, 공유의 과정을 경험할 수 있도록 충분한 시간을 제공하는 것이 필요하다.
- UMC 모형은 초등학생의 프로그래밍 학습에서 자신감을 형성하는 데 효과적이지만, Create 단계로 자연스럽게 확장되기 위해서는 Use와 Modify 단계에서 충분한 이해와 탐색이 이루어져야 한다. 따라서 교사는 각 단계가 단절되지 않도록, 사용한 프로그램의 기능이 수정 활동으로 이어지고, 수정한 경험이 창작 활동의 아이디어로 연결되도록 수업을 설계해야 한다.

1-3 짝 프로그래밍 모형(Kent Beck, 1999)

① 특징

짝 프로그래밍 모형은 두 명의 학생이 하나의 컴퓨터를 함께 사용하여 프로그래밍 과제를 해결하는 협력적 교수·학습 모형이다. 이 모형은 켄트 벡(Kent Beck)이 제안한 익스트림 프로그래밍(Extreme Programming, XP)의 핵심 실천 방법 중 하나로 제시되었으며, 이후 Laurie Williams와 Robert Kessler 등이 짝 프로그래밍을 컴퓨터 교육 맥락에 적용하면서 교육 현장에서도 널리 활용되기 시작하였다.

짝 프로그래밍에서는 한 학생이 드라이버(driver) 역할을 맡아 키보드와 마우스를 조작하며 직접 코드를 작성하고, 다른 학생은 내비게이터(navigator) 역할을 맡아 코드의 흐름, 오류 가능성, 문제 해결 전략을 점검한다. 두 학생은 일정 시간마다 역할을 교대하면서 코드 작성자와 검토자의 관점을 모두 경험한다.

이 모형은 프로그래밍을 개인의 독립적 활동으로만 다루지 않고, 문제를 함께 분석하고 해결 전략을 논의하며 코드를 검토하는 협력적 사고 과정으로 확장한다는 특징이 있다. 학생들은 짝과 대화하면서 자신의 생각을 설명하고, 친구의 설명을 듣고, 오류를 함께 찾으며 프로그래밍 개념을 보다 깊이 이해할 수 있다.

초등학교 정보 수업에서는 짝 프로그래밍을 통해 학생들의 코딩 부담을 줄이고, 프로그래밍 과정에서 발생하는 오류를 함께 해결하도록 지도할 수 있다. 또한 의사소통 능력, 협력적 문제 해결력, 알고리즘적 사고, 디버깅 능력, 코드 검토 능력을 함께 기르는 데 효과적이다.

② 학습 과정

단계	설명
준비 단계	• 교사는 짝 프로그래밍의 목적과 활동 방법을 안내한다. 학생들에게 두 명이 하나의 컴퓨터를 함께 사용하되, 한 명만 키보드와 마우스를 조작하고 다른 한 명은 코드의 흐름과 오류를 점검하는 방식임을 설명한다. • 교사는 수업 목표에 맞는 프로그래밍 과제를 제시하고, 학생들이 해결해야 할 문제 상황, 사용할 명령이나 블록, 완성해야 할 기능을 안내한다. • 짝을 구성할 때에는 학생들의 프로그래밍 수준, 협력 태도, 성별, 수업 맥락 등을 고려하여 구성하거나, 활동 목적에 따라 무작위로 구성할 수 있다.
역할 설정	• 교사는 드라이버와 내비게이터의 역할을 구체적으로 안내한다. • 드라이버는 키보드와 마우스를 직접 조작하여 코드를 작성하고, 내비게이터의 의견을 들으며 코드를 수정한다. • 내비게이터는 드라이버가 작성하는 코드의 논리적 흐름을 살펴보고, 오류 가능성을 점검하며, 더 나은 해결 방법이나 아이디어를 제안한다. • 교사는 학생들이 역할을 단순히 나누는 데 그치지 않고, 서로 질문하고 설명하며 협력해야 함을 강조한다.
문제해결 및 프로그래밍	• 학생들은 주어진 문제를 함께 분석하고 해결 방법을 논의한다. • 드라이버는 논의한 내용을 바탕으로 코드를 작성하고, 내비게이터는 코드가 문제 해결 목표에 맞게 작성되고 있는지 점검한다. • 학생들은 "이 명령이 먼저 실행되어야 할까?", "반복을 사용하면 더 간단해질까?", "조건이 참일 때와 거짓일 때 결과가 어떻게 달라질까?"와 같은 질문을 주고받으며 알고리즘과 프로그램 구조를 함께 구성한다. • 교사는 짝별 활동을 관찰하며 역할이 한 학생에게 치우치지 않도록 지도한다.
검토 및 토의	• 코드 작성 후 두 학생은 프로그램을 실행해 보고, 의도한 대로 작동하는지 함께 확인한다. 오류가 발생하면 어느 부분에서 문제가 생겼는지 함께 찾고, 코드의 순서, 조건, 반복, 변수, 입력과 출력 등을 점검한다. • 학생들은 오류를 수정하면서 디버깅 과정을 경험하고 더 효율적인 방법이나 다른 해결 전략이 있는지 토의한다. • 교사는 학생들이 정답 코드만 찾는 데 그치지 않고, 왜 그렇게 수정했는지 설명하도록 유도한다.
역할교대 및 반복	• 일정 시간 또는 과제의 일정 부분이 끝난 뒤 드라이버와 내비게이터의 역할을 교대한다. 일반적으로 10~20분 간격으로 역할을 바꿀 수 있으며, 초등학생의 경우 과제 난이도와 집중 시간을 고려하여 더 짧게 운영할 수도 있다. • 새로운 드라이버는 이전 드라이버와 내비게이터가 논의한 내용을 이어받아 코드를 작성하고, 새로운 내비게이터는 코드의 흐름을 점검하며 피드백을 제공한다. 이 과정을 반복하면서 두 학생 모두 코드 작성과 코드 검토를 균형 있게 경험한다.
발표 및 성찰	• 짝 프로그래밍 활동이 끝나면 학생들은 완성한 프로그램의 기능과 해결 과정을 공유한다. • 교사는 학생들이 결과물뿐만 아니라 협력 과정, 역할 수행, 오류 해결 과정, 새롭게 알게 된 점을 발표하도록 지도한다. • 학생들은 "드라이버 역할에서 어려웠던 점", "내비게이터 역할에서 도움이 되었던 점", "짝과 함께 해결한 오류", "다음 활동에서 개선하고 싶은 점" 등을 성찰한다. • 교사는 프로그래밍 결과와 함께 협력 태도, 의사소통 과정, 문제 해결 전략을 종합적으로 피드백한다.

③ 모형 적용상의 유의점

- 짝 프로그래밍 모형을 적용할 때에는 먼저 드라이버와 내비게이터의 역할을 학생들이 명확히 이해하도록 해야 한다. 드라이버는 단순히 혼자 코드를 작성하는 사람이 아니며, 내비게이터는 옆에서 구경하는 사람이 아니다. 두 역할 모두 문제 해결 과정에 적극적으로 참여해야 하며, 서로의 생각을 말로 표현하고 조율하는 과정이 중요하다.
- 짝을 구성할 때에는 학생 간 수준 차이를 신중하게 고려해야 한다. 수준 차이가 너무 클 경우 한 학생이 활동을 주도하고 다른 학생은 소극적으로 참여할 수 있다. 반대로 두 학생 모두 과제 해결에 어려움을 느끼는 경우 활동이 정체될 수 있다. 따라서 교사는 과제의 난이도, 학생의 프로그래밍 경험, 협력 태도 등을 고려하여 짝을 구성해야 한다.
- 역할 교대는 반드시 이루어져야 한다. 역할을 교대하지 않으면 한 학생은 계속 코드를 입력하고, 다른 학생은 계속 관찰만 하게 되어 학습 경험이 불균형해질 수 있다. 교사는 일정 시간이나 과제의 단계에 따라 역할을 바꾸도록 안내하고, 역할 교대 시 현재까지 작성한 코드와 해결 중인 문제를 서로 설명하게 해야 한다.
- 내비게이터의 역할이 지나치게 소극적으로 변하지 않도록 유의해야 한다. 내비게이터는 드라이버의 실수를 지적하는 역할만 하는 것이 아니라 코드의 방향을 함께 생각하고, 오류를 미리 예상하며 더 나은 해결 전략을 제안하는 역할을 수행해야 한다. 이를 위해 교사는 내비게이터가 사용할 수 있는 질문 예시나 점검 기준을 제공할 수 있다.
- 드라이버가 내비게이터의 의견을 무시하거나 혼자서만 코드를 작성하지 않도록 지도해야 한다. 드라이버는 코드를 작성하기 전에 자신의 생각을 말로 설명하고, 내비게이터의 제안을 들으며 코드를 수정해야 한다. 교사는 학생들이 “내 생각에는 이렇게 하면 될 것 같아”, “왜 이 블록을 사용했어?”, “다른 방법도 있을까?”와 같은 상호작용을 하도록 격려한다.
- 짝 프로그래밍에서는 결과물 평가와 함께 협력 과정에 대한 평가가 필요하다. 프로그램이 제대로 작동했는지만 평가하면 한 학생이 대부분의 작업을 수행해도 높은 평가를 받을 수 있다. 따라서 교사는 역할 수행, 의사소통, 공동 문제 해결, 오류 수정 과정, 성찰 내용을 함께 평가해야 한다.
- 초등학생을 대상으로 할 때에는 활동 규칙을 구체적으로 제시하는 것이 효과적이다. 예를 들어 “드라이버만 키보드를 사용한다”, “내비게이터는 반드시 이유를 말하며 제안한다”, “역할 교대 시간이 되면 하던 작업을 설명하고 바꾼다”, “친구의 의견을 듣고 결정한다”와 같은 규칙을 사전에 안내할 수 있다.
- 짝 프로그래밍 모형은 협력적 학습 효과가 크지만 학생 간 갈등이나 참여 불균형이 발생할 수 있다. 따라서 교사는 활동 중 순회 지도를 통해 짝별 상호작용을 관찰하고, 필요한 경우 역할을 재조정하거나 발문을 통해 대화를 촉진해야 한다. 또한 활동 후에는 결과물 발표뿐만 아니라 협력 과정에 대한 성찰 시간을 마련하여 학생들이 다음 활동에서 더 나은 협력 방식을 익힐 수 있도록 해야 한다.

1-4 청사진 프로그래밍 모형(Papert, 1993)

① 특징

청사진 프로그래밍 모형은 프로그램을 작성하기 전에 문제 해결 절차와 전체 구조를 미리 계획하고, 그 계획에 따라 순차적으로 프로그램을 구현하는 교수·학습 모형이다. 건축에서 건물을 짓기 전에 청사진을 설계하듯이, 프로그래밍에서도 전체 구조, 필요한 기능, 명령의 순서, 입력과 출력, 조건과 반복의 사용 방식을 먼저 구상한 후 코드를 작성한다.

이 모형은 문제를 즉흥적으로 해결하기보다 해결해야 할 문제를 분석하고 알고리즘을 설계한 뒤 이를 코드로 구현하는 계획 중심의 접근을 강조한다. 따라서 학생들이 프로그램을 만들기 전에 "무엇을 만들 것인가", "어떤 순서로 동작해야 하는가", "어떤 명령이나 블록이 필요한가", "어떤 결과가 나와야 하는가"를 먼저 생각하도록 지도하는 데 적합하다.

청사진 프로그래밍은 큰 문제를 작은 절차로 나누고 전체 흐름을 먼저 정한 뒤 세부 코드를 작성하는 탑다운 방식과 관련이 깊다. 또한 소프트웨어 개발에서 요구사항 분석, 설계, 구현, 테스트의 순서로 진행되는 워터폴 모델과 유사한 특성을 지닌다.

이 모형은 학생들이 코드를 작성하기 전에 충분히 사고하도록 돕는다는 장점이 있다. 특히 순차, 선택, 반복 구조를 체계적으로 학습하거나, 정해진 문제를 정확하게 해결해야 하는 프로그래밍 활동에서 효과적이다. 다만 학생의 창의적 탐색과 시행착오를 제한하지 않도록, 계획과 구현 사이에서 적절한 수정과 피드백이 이루어지도록 지도할 필요가 있다.

② 학습 과정

단계	설명
문제 이해하기	• 교사는 학생들에게 해결해야 할 문제 상황이나 만들 프로그램의 목표를 제시한다. • 학생들은 문제에서 요구하는 기능, 입력해야 할 값, 출력되어야 할 결과, 해결 조건을 확인한다. • 교사는 "이 프로그램은 어떤 문제를 해결해야 할까요?", "사용자가 입력하는 것은 무엇인가요?", "프로그램이 보여주어야 할 결과는 무엇인가요?"와 같은 발문을 통해 학생들이 문제의 핵심을 파악하도록 돕는다.
전체 구조 설계하기	• 학생들은 프로그램의 전체 흐름을 먼저 구상한다. • 교사는 학생들이 프로그램을 시작, 입력, 처리, 조건, 반복, 출력, 종료 등의 구조로 나누어 생각하도록 안내한다. • 학생들은 자연어, 순서도, 의사코드 등을 활용하여 프로그램이 어떤 순서로 실행될지 계획한다. • 이 단계에서는 바로 코드를 작성하기보다 전체적인 실행 흐름을 명확히 하는 데 초점을 둔다.
세부 절차 계획하기	• 학생들은 전체 구조를 바탕으로 각 단계에서 필요한 명령이나 블록을 구체적으로 정한다. 예를 들어 반복이 필요한 부분, 조건에 따라 달라지는 부분, 변수를 사용해야 하는 부분, 계산이 필요한 부분을 표시한다. • 교사는 학생들이 "어떤 명령을 먼저 사용해야 할까?", "조건이 참일 때와 거짓일 때 각각 무엇을 해야 할까?", "반복은 몇 번 또는 언제까지 해야 할까?"를 생각하도록 지도한다.
코드 구현하기	• 학생들은 앞서 작성한 계획에 따라 프로그램을 구현한다. 블록 기반 프로그래밍에서는 설계한 순서에 맞게 블록을 조립하고, 텍스트 기반 프로그래밍에서는 의사코드나 순서도를 참고하여 코드를 작성한다. • 교사는 학생들이 계획과 코드가 일치하는지 확인하도록 돕고, 필요한 경우 설계한 절차를 다시 살펴보게 한다. • 이 단계에서는 계획한 내용을 코드로 정확하게 표현하는 데 중점을 둔다.
실행 및 검토하기	• 학생들은 완성한 프로그램을 실행하여 계획한 대로 작동하는지 확인한다. • 실행 결과가 예상과 다를 경우, 학생들은 오류가 발생한 부분을 찾아 계획과 코드를 비교한다. • 교사는 학생들이 단순히 코드를 고치는 데 그치지 않고 처음 세운 계획이 적절했는지 함께 검토하도록 지도한다. • 필요한 경우 알고리즘, 조건, 반복 구조, 입력과 출력 과정을 수정한다.
정리 및 공유하기	• 학생들은 자신이 작성한 프로그램의 설계 과정과 실행 결과를 정리한다. • 교사는 학생들이 프로그램의 전체 흐름, 사용한 알고리즘, 계획과 실제 구현의 차이, 수정한 부분을 설명하도록 한다. • 학생들은 친구들과 결과물을 공유하고, 다른 학생의 설계 방법과 비교하면서 더 효율적이거나 이해하기 쉬운 방법을 찾아본다.

③ 모형 적용상의 유의점

- 청사진 프로그래밍 모형을 적용할 때에는 학생들이 코드를 작성하기 전에 문제를 충분히 이해하도록 해야 한다. 초등학생은 문제 상황을 정확하게 파악하지 못한 상태에서 바로 블록을 조립하거나 코드를 입력하려는 경우가 많다. 따라서 교사는 문제의 조건, 필요한 입력, 기대되는 출력, 해결 순서를 먼저 말이나 그림으로 표현하게 해야 한다.
- 전체 구조를 설계하는 단계에서는 학생 수준에 맞는 표현 방식을 활용해야 한다. 초등학생에게 처음부터 복잡한 순서도나 의사코드를 요구하기보다는, "먼저-다음-마지막"과 같은 자연어 절차, 간단한 그림, 표, 카드 배열 등을 활용할 수 있다. 이후 학년이나 수준에 따라 순서도, 알고리즘 표현, 의사코드로 점차 확장하는 것이 적절하다.
- 이 모형은 계획을 강조하지만 처음 세운 계획을 반드시 그대로 따라야 한다는 의미는 아니다. 학생들이 구현 과정에서 오류를 발견하거나 더 나은 방법을 찾을 경우 계획을 수정하고 다시 구현할 수 있도록 해야 한다. 계획은 고정된 답이 아니라 프로그래밍을 돕는 사고 도구로 활용되어야 한다.
- 청사진 프로그래밍은 정해진 조건에 따라 정확한 결과를 산출해야 하는 활동에 적합하다. 예를 들어 수학 공식을 프로그램으로 표현하기, 로봇을 정해진 경로로 이동시키기, 조건에 따라 다른 메시지를 출력하기, 반복 구조를 활용하여 규칙적인 도형을 그리기 등의 활동에서 효과적으로 활용할 수 있다.
- 반면, 창의적 탐색이나 자유로운 프로젝트 활동에서는 계획 중심 접근이 학생들의 실험과 상상력을 제한할 수 있다. 따라서 창작 중심 수업에서는 전체 계획을 지나치게 엄격하게 요구하기보다 핵심 기능만 먼저 설계하고 구현 과정에서 아이디어를 수정하거나 확장할 수 있도록 허용해야 한다.
- 교사는 학생들이 계획과 코드 사이의 관계를 이해하도록 지도해야 한다. 학생들이 작성한 순서도나 절차가 실제 코드에서 어떤 명령이나 블록으로 구현되었는지 연결해 보게 하면 알고리즘적 사고와 프로그래밍 이해를 함께 기를 수 있다. 이를 위해 "이 부분은 코드의 어느 블록과 연결되나요?", "계획에는 있었지만 코드에는 빠진 부분이 있나요?", "코드에는 있지만 계획에는 없었던 부분은 무엇인가요?"와 같은 질문을 활용할 수 있다.
- 평가에서는 완성된 프로그램의 작동 여부뿐만 아니라 문제 분석, 알고리즘 설계, 절차 표현, 계획과 구현의 일치성, 오류 수정 과정을 함께 살펴보아야 한다. 특히 초등학교 정보 수업에서는 결과물의 완성도보다 학생이 문제를 어떻게 나누어 생각하고, 어떤 순서로 해결하려 했는지를 평가하는 것이 중요하다.

1-5 창발적 프로그래밍 모형(Papert, 1996)

① 특징

창발적 프로그래밍 모형은 처음부터 완성된 계획을 세우고 그 계획에 따라 프로그램을 작성하기보다 작은 아이디어나 간단한 기능에서 출발하여 실행, 관찰, 수정, 확장을 반복하면서 프로그램을 점진적으로 발전시켜 가는 교수·학습 모형이다. 이 모형은 프로그래밍을 정해진 절차를 그대로 구현하는 활동으로만 보지 않고 학습자가 직접 시도하고 실험하면서 새로운 가능성을 발견하는 탐색적 문제 해결 과정으로 본다.

청사진 프로그래밍이 전체 구조를 먼저 설계한 후 순차적으로 구현하는 탑다운 방식에 가깝다면 창발적 프로그래밍은 작은 기능이나 부분적인 아이디어에서 시작하여 점차 전체 프로그램으로 확장해 가는 바텀업 방식에 가깝다. 학생들은 간단한 명령이나 블록을 실행해 보고, 예상한 결과와 실제 결과를 비교하며 오류나 예상치 못한 결과를 바탕으로 프로그램을 수정하고 개선한다.

이 모형은 계획보다 실험과 수정의 과정을 중시한다. 학생들은 처음부터 완성도 높은 프로그램을 만들기보다 작게 만들어 보고 실행해 본 뒤 필요한 기능을 하나씩 추가한다. 이 과정에서 시행착오, 디버깅, 아이디어 확장, 기능 개선이 자연스럽게 이루어진다. 또한 창발적 프로그래밍은 현대 소프트웨어 개발에서 작은 단위로 기능을 만들고 실행과 피드백을 통해 점진적으로 개선하는 애자일 또는 스크럼 방식과 유사한 특성을 지닌다.

② 학습 과정

단계	설명
아이디어 떠올리기	• 교사는 학생들이 만들고 싶은 기능이나 해결하고 싶은 간단한 문제를 떠올리도록 안내한다. 이 단계에서는 완성된 프로그램 전체를 설계하기보다, "캐릭터가 움직이게 하기", "버튼을 누르면 소리가 나게 하기", "센서 값에 따라 불이 켜지게 하기", "간단한 게임 규칙 만들기"와 같이 작은 출발점을 정하는 데 초점을 둔다. • 교사는 학생들이 부담 없이 시도할 수 있도록 문제를 작게 나누어 제시하거나 여러 가지 선택 가능한 아이디어를 제공할 수 있다.
작게 만들어 보기	• 학생들은 떠올린 아이디어를 바탕으로 가장 기본적인 기능을 먼저 만들어 본다. 블록 기반 프로그래밍에서는 필요한 블록 몇 개를 조합해 보고, 피지컬 컴퓨팅에서는 센서나 출력 장치 하나를 작동시켜 보는 것부터 시작할 수 있다. • 교사는 학생들이 처음부터 완성된 프로그램을 만들려고 하기보다 실행 가능한 작은 단위를 먼저 구성하도록 지도한다.
실행하고 관찰하기	• 학생들은 자신이 만든 프로그램을 실행하고 결과를 관찰한다. 예상한 대로 작동하는지, 예상과 다른 결과가 나타나는지, 오류가 발생하는지 확인한다. • 교사는 "무엇이 예상대로 되었나요?", "생각과 다른 점은 무엇인가요?", "왜 이런 결과가 나왔을까요?"와 같은 발문을 통해 학생들이 실행 결과를 학습의 단서로 활용하도록 돕는다.
수정하고 개선하기	• 학생들은 실행 결과를 바탕으로 프로그램을 수정한다. • 오류가 발생한 경우에는 원인을 찾아 고치고 예상과 다른 결과가 나타난 경우에는 코드를 바꾸어 다시 실행해 본다. 또한 더 흥미로운 결과를 만들기 위해 기능을 일부 변경하거나 추가할 수 있다. • 교사는 학생들이 시행착오를 실패로 받아들이지 않고 프로그램을 발전시키는 과정으로 이해하도록 지도한다.
확장하기	• 학생들은 기본 기능이 작동하면 새로운 기능을 하나씩 추가하며 프로그램을 확장한다. 예를 들어 움직임에 점수 기능을 추가하거나, 조건에 따라 다른 동작을 하게 하거나, 반복 구조를 넣어 더 복잡한 동작을 만들 수 있다. • 교사는 학생들이 한꺼번에 많은 기능을 넣기보다 하나의 기능을 추가한 뒤 실행하고 확인하는 방식으로 확장하도록 안내한다.
공유하고 성찰하기	• 학생들은 자신이 만든 프로그램을 친구들과 공유하고, 어떤 아이디어에서 시작했는지, 어떤 부분을 수정하고 확장했는지 설명한다. • 교사는 학생들이 완성 결과뿐만 아니라 시도한 과정, 발생한 오류, 해결 방법, 새롭게 발견한 점을 발표하도록 지도한다. • 학생들은 친구들의 프로그램을 보며 새로운 아이디어를 얻고 자신의 프로그램을 더 개선할 방향을 생각한다.

③ 모형 적용상의 유의점

- 창발적 프로그래밍 모형을 적용할 때에는 학생들이 처음부터 완벽한 계획이나 완성된 프로그램을 만들도록 요구하지 않아야 한다. 이 모형의 핵심은 작은 시도에서 출발하여 실행과 수정을 반복하면서 점차 프로그램을 발전시키는 데 있다. 따라서 교사는 학생들이 부담 없이 시작할 수 있는 작은 과제나 기능을 제시하고, 이를 확장할 수 있도록 안내해야 한다.
- 탐색과 시행착오를 허용하는 수업 분위기를 조성하는 것이 중요하다. 학생들은 프로그램이 예상대로 작동하지 않으면 실패했다고 느낄 수 있다. 교사는 오류나 예상치 못한 결과가 프로그래밍 학습에서 자연스러운 과정이며, 새로운 발견과 개선의 기회가 될 수 있음을 강조해야 한다.
- 실행과 수정의 주기를 짧게 운영하는 것이 효과적이다. 초등학생은 오랫동안 코드를 작성한 뒤 한 번에 실행하는 방식보다, 작은 기능을 만들고 바로 실행해 보며 결과를 확인하는 방식에서 더 잘 학습할 수 있다. 교사는 학생들이 "만들기-실행하기-관찰하기-수정하기"의 과정을 반복하도록 지도해야 한다.
- 확장 단계에서는 학생들이 기능을 지나치게 많이 추가하여 프로그램 구조가 복잡해지지 않도록 주의해야 한다. 학생들이 새로운 기능을 추가할 때에는 한 번에 하나의 기능을 추가하고, 실행 결과를 확인한 뒤 다음 기능으로 넘어가게 하는 것이 좋다. 이를 통해 오류의 원인을 쉽게 찾고 프로그램의 구조를 이해할 수 있다.
- 교사는 학생의 자유로운 탐색을 존중하되 학습 목표와 연결될 수 있도록 적절한 발문과 비계를 제공해야 한다. 예를 들어 반복 구조를 학습하는 수업에서는 학생들이 자유롭게 캐릭터 움직임을 만들더라도 반복 블록을 활용해 동작을 단순화할 수 있도록 안내할 수 있다. 조건 구조를 학습하는 수업에서는 특정 상황에서 다른 동작이 일어나도록 기능을 확장하게 할 수 있다.
- 창발적 프로그래밍은 창의적 산출물을 만드는 데 효과적이지만 수업이 지나치게 자유 활동으로 흐르지 않도록 해야 한다. 교사는 활동의 최소 조건, 사용할 핵심 개념, 완성해야 할 기본 기능, 공유할 내용 등을 명확히 제시해야 한다. 예를 들어 "반복 블록을 한 번 이상 사용하기", "조건에 따라 결과가 달라지게 하기", "자신이 수정한 부분 설명하기"와 같은 기준을 제공할 수 있다.
- 평가에서는 최종 산출물의 완성도뿐만 아니라 학생이 어떤 아이디어에서 출발했는지, 어떤 시행착오를 경험했는지, 오류를 어떻게 해결했는지, 프로그램을 어떻게 확장했는지를 함께 살펴보아야 한다. 특히 초등학교 정보 수업에서는 학생의 탐색 과정, 문제 발견 과정, 수정과 개선 과정, 창의적 표현 과정을 평가하는 것이 중요하다.

2 수업 배경지식

2-1 초등학교 정보교육의 흐름(전 단원)

제5차 교육과정 (1987년 발표 / 1989년 적용)

경제 발전에 따른 소프트산업 대두와 정보화 사회 도래를 배경으로 개정되었으며, 기본방향으로 ① 기초교육의 강화, ② 정보화 사회에 대응하는 교육 강화, ③ 교육과정의 효율성 제고를 제시했다. 초등학교에서는 실과교과를 통해 컴퓨터 기초 내용이 처음 도입되었다. 실과는 4~6학년에 편성되었고, 정보교육이 처음 도입됨에 따라 각 학년·영역별로 관련 내용이 반영되었다.

학년	영역	지도요소
4학년	일과 직업의 이해	컴퓨터와 일의 세계
5학년	소비와 절약	컴퓨터의 종류와 쓰임새
6학년	생활기능	컴퓨터의 기초, 컴퓨터 다루기

제6차 교육과정 (1992년 발표 / 1995년 적용)

학교 수준 교육과정 편성·운영이 강조된 시기로, 특별활동이 학교재량시간으로 변경되었다. 초등학교 3학년 이상 주당 1시간으로 편성된 학교재량시간에 많은 학교가 컴퓨터를 선택하면서, 실과교과의 정보교육영역과 함께 이원화된 형태로 정보교육이 실시되었다. 초등 3학년부터 주당 1시간씩 총 136시간 운영, 중학교 컴퓨터교과와 연계되는 체계를 갖췄다. 또한5차의 '이해' 중심에서 컴퓨터를 실제로 활용·적용하는 활동(컴퓨터 다루기, 컴퓨터 관리하기, 컴퓨터로 글쓰기 등) 중심으로 변화하였다.

제7차 교육과정 (1997년 발표 / 2000년 적용)

'학교 재량 시간'을 '재량 활동'으로 명칭 변경, 초등 1학년부터 주당 2시간 편성. 추구하는 인간상으로 "21세기 세계화·정보화 시대를 주도할 자율적이고 창의적인 한국인 육성"을 제시했다. 2000년 정부의 '초·중등학교 정보통신기술교육 운영지침' 발표로 "세계에서 컴퓨터를 가장 잘 쓰는 국민" 육성을 위해 재량활동에서 주당 1시간을 편성, 6년간 200시간의 정보교육이 운영되었다. 이 운영지침을 통해 학년군에 따른 단계별 내용체계가 완성되었고, 2005년 개정으로 컴퓨터과학 내용을 강화한 역량 중심 소양교육으로 보완되었다. 실과교과에서는 '생활 기술' 영역을 통해 5~6학년에서 컴퓨터 다루기(구성·자판·글쓰기), 컴퓨터 활용하기(그림·통신)가 편성되었다.

시수 확보 지침: 1~4학년은 재량활동, 5·6학년은 재량활동·특별활동·실과의 컴퓨터 관련 영역을 활용해 연간 34시간 이상(1학년 30시간) 확보. 교과 활용교육 10% 이상 반영으로 200+α 운영.

2007 개정 교육과정 (2007년 발표 / 2009년 적용)

수시 개정 체제로 전환된 첫 교육과정으로 7차의 기본방향을 견지했다. 재량활동이 '교과 재량 활동'과 '창의적 재량 활동'으로 구성. 주5일 수업제 정착으로 초등 3~6학년 연간 34시간이 감축되었다. 2008년 학교규제지침 일괄정비계획 발표로 정보통신기술교육의 편성·운영·시수 권한이 시·도교육청으로 이관되었고, 그 결과 시수가 축소되었다.

실과에서는 '기술의 세계' 영역에서 5학년 "정보 기기와 사이버 공간", 6학년 "인터넷과 정보"로 편성. '컴퓨터'에서 '정보'라는 용어로 전환되며 정보 탐색·활용과 함께 정보윤리 내용이 제시되었다.

2009 개정 교육과정 (2009년 발표 / 2011년 적용)

'학년군'·'교과군' 개념 도입, 특별활동과 창의적 재량활동을 통합한 '창의적체험활동'으로 개편하였다. 글로벌 창의 인재"를 인간상으로 정의했으나, 이전까지 인간상에 제시되던 '정보화' 내용이 빠지게 된 결정적 분기점이다. 정보통신활용교육은 39개 범교과학습 영역의 하나로 설정되어 학교 재량에 맡겨졌고, 7차의 200시간에 비해 절대적으로 부족한 여건이 되었다. 실과에서는 5~6학년군으로 편성되며 사실상 6학년 한 개 단원("생활과 정보": 정보 기기와 사이버 공간, 멀티미디어 자료 만들기)으로 위축되었다.

2015 개정 교육과정 (2015년 발표 / 2017년 적용)

창의융합형 인재 양성을 목표로 6개 핵심역량(자기관리, 지식정보처리, 창의적 사고, 심미적 감성, 의사소통, 공동체 역량)을 제시했다. 정보교육 측면의 핵심 변화는 소프트웨어(SW)교육 강화로, 기존 실과의 정보교육 내용을 별도 단원으로 편성해 5~6학년에서 17시간 이상 학습하도록 기준을 제시한 것이다. 이로써 정보교육이 초등학교 필수교육 내용으로 반영되었다. 한편 정보교육은 범교과학습 주제에서는 제외되었고, 실제 편성은 6학년 17시간으로 운영되었다.

내용체계상 SW교육은 실과의 '기술 시스템' 영역, '소통' 핵심개념 아래 편성되었으며, 일반화된 지식은 "통신 기술은 정보를 생산·가공하여 다양한 수단과 장치를 통해 송수신하고 공유한다"로 제시되었다. 내용요소는 소프트웨어의 이해, 절차적 문제해결, 프로그래밍 요소와 구조의 세 가지로 구성되었다. 성취기준은 소프트웨어 사례 이해([6실04-07]), 절차적 문제해결([6실04-08]), 프로그래밍 체험·설계([6실04-09~11]) 등 5개로, 블록 기반 교육용 프로그래밍 언어(스크래치·엔트리)를 활용해 입력-처리-출력 과정과 순차·선택·반복 구조를 다루도록 했다. 교수학습 측면에서는 구체적 조작기라는 인지 발달 단계를 고려해 언플러그드 활동을 포함하고, 컴퓨팅 사고력 향상을 정보교육의 목표로 설정했다.

2022 개정 교육과정 (2022년 발표 / 2024~2027년 연차 적용)

'기초소양' 개념을 처음 도입하여 언어·수리·디지털 소양을 모든 교과 학습의 기초로 설정했다. 디지털 소양은 "디지털 도구로 정보를 수집·분석·관리하고 문제를 해결하는 능력"으로 정의되었다. 초등 정보교육 시수는 17시간에서 34시간으로 확대되었다(중학교 34 → 68시간). 추가된 17시간은 별도 시수로 배당되지 않고 단위 학교가 학교교육과정 편성을 통해 확보하도록 제시되었으며, 확보 방법으로 ①교과(군) 시수 20%

조정(과학/실과 340 → 374시간), ②학교자율시간을 활용한 별도 과목 개설이 안내되었다.

내용 면에서는 2015의 프로그래밍 중심 구성에 데이터·인공지능 영역이 추가되어 성취기준이 5개에서 12개로 늘었다. 12개 성취기준은 실과에 5개, 부록에 7개로 분리 편성되었으며, 실과의 정보교육은 5번째 영역 디지털 사회와 인공지능에 해당한다. 핵심 아이디어는 프로그래밍·데이터·인공지능으로 구성되고, 내용요소는 컴퓨터의 개념, 문제 찾기와 문제 해결 절차, 컴퓨터에게 명령하는 방법, 데이터의 종류와 표현, 생활 속 인공지능으로 편성되었다. 2015 대비 중학교에 있던 변수 개념이 초등으로 조정되었고, 프로그래밍 구조 순서가 "순차·선택·반복"에서 "순차·반복·선택"으로 변경되었다.

구분	2015 개정	2022 개정
초등 시수	17시간	34시간
편성 형태	실과 내 단원	실과 17차시 + 부록 17차시
내용 영역	프로그래밍 중심	프로그래밍, 데이터, 인공지능
성취 기준 수	5개	12개

출처 : 교육부(1987, 1992, 1997, 2007, 2009, 2015, 2022), 신승기(2021, 2025)

2-2 컴퓨터의 종류 (1단원)

구분	의미
개인용 컴퓨터	• 개인이 문서 작성, 정보 검색, 학습, 자료 저장 등에 활용하는 컴퓨터이다. 데스크톱 컴퓨터와 노트북 컴퓨터가 대표적이다.
휴대용 컴퓨터	• 스마트폰, 태블릿 PC처럼 이동하면서 사용할 수 있는 컴퓨터이다. 통신, 검색, 사진 촬영, 학습 등 다양한 기능을 수행한다.
임베디드 컴퓨터	• 자동차, 로봇 청소기, 세탁기, 냉장고, 엘리베이터 등 특정 기능을 수행하는 기기 안에 들어 있는 컴퓨터이다.
메인 프레임 컴퓨터	• 은행, 기업, 공공기관 등에서 많은 데이터를 안정적으로 처리하기 위해 사용하는 대형 컴퓨터이다.
슈퍼 컴퓨터	• 과학 연구, 기상 예측, 우주·공학 계산 등 매우 복잡하고 빠른 계산이 필요한 분야에 활용되는 고성능 컴퓨터이다.

2-3 컴퓨팅 사고력(Computational Thinking) (전 단원)

'컴퓨팅 사고력'이라는 표현은 Seymour Papert가 1980년 『Mindstorms』에서 처음 사용한 것으로 알려져 있다. 다만 Papert는 컴퓨팅 사고력을 하나의 문장으로 정의하지는 않았다. 그는 컴퓨터가 형식적인 것을 구체화하고 개인화할 수 있으며, 학습자가 컴퓨터로 만든 대상을 '생각하기 위한 대상'으로 활용할 수 있다고 보았다.

Jeannette Wing(2011)은 컴퓨팅 사고를 문제와 그 해결 방법을 구성하여, 해결 방법이 정보 처리 에이전트에 의해 효과적으로 수행될 수 있는 형태로 표현되도록 하는 사고 과정으로 설명하였다. 이 해결 방법은 사람이나 기계 또는 사람과 기계가 함께 수행할 수 있다.

Lodi와 Martini는 Papert와 Wing의 관점이 모두 초·중등 컴퓨터과학 교육에 중요하다고 보았다. Papert의 관점은 컴퓨터를 활용하여 형식적이고 추상적인 지식을 구체적으로 다루는 학습을 강조하며, Wing의 관점은 문제와 해결 방법을 수행 가능한 형태로 표현하는 사고 과정을 강조한다.

출처 : Papert(1980), Wing(2011), Lodi·Martini(2021)

2-4 컴퓨팅 사고력의 구성 요소 (전 단원)

컴퓨팅 사고력은 논리적 추론을 통해 문제를 해결하고, 대상·절차·시스템을 더 잘 이해하는 인지적 사고 과정이다. 컴퓨팅 사고에는 다음과 같은 능력이 포함된다.

- 알고리즘적으로 생각하는 능력
- 문제를 부분으로 나누어 생각하는 능력
- 패턴을 찾아 활용하고 일반화하는 능력
- 적절한 표현을 선택하여 추상적으로 생각하는 능력
- 해결 방법을 평가하는 능력

출처 : Csizmadia et al.(2015)

2-5 컴퓨팅 사고력의 구성 요소 - 추상화 (전 단원)

추상화는 Wing(2008)이 컴퓨팅 사고에서 가장 중요하고 고차원적인 사고 과정으로 제시한 개념이다. 추상화란 여러 대상이나 상황에서 본질적이고 공통적인 속성만을 추려내고, 그 외의 세부 사항은 의도적으로 덜어 내는 사고 과정을 말한다. Wing은 추상화가 패턴을 정의하고, 개별 사례로부터 일반적인 원리를 끌어내며, 구체적인 값 대신 변할 수 있는 요소를 매개변수로 다루는 활동을 통해 이루어진다고 보았다. 추상화의 핵심은 '무엇을 남기고 무엇을 덜어 낼 것인가'를 판단하는 데 있다. 특정 사례의 세부 사항에 얽매이지 않고 하나의 추상화로 여러 경우를 포괄할 수 있도록 하는 것이 추상화의 힘이다.

출처 : Wing(2006), Wing(2008)

2-6 컴퓨팅 사고력의 구성 요소 - 자동화 (전 단원)

자동화는 추상화와 짝을 이루는 컴퓨팅 사고력의 또 다른 핵심 요소이다. Wing(2008)은 "컴퓨팅이란 우리가 만든 추상화를 자동화하는 것"이라고 설명하며, 추상화와 자동화를 컴퓨팅 사고의 두 축으로 제시하였다. 자동화란 사람이 설계한 추상화, 즉 문제 해결의 절차나 방법을 정보 처리 에이전트가 대신 수행하도록 맡기는 것을 의미한다.

자동화가 이루어지기 위해서는 추상화를 해석하고 실행할 수 있는 컴퓨터가 필요하다. 이때 컴퓨터는 반드시 기계만을 가리키는 것이 아니라, 사람이나 기계, 또는 사람과 기계가 함께 수행하는 경우를 모두 포함한다. 자동화가 가능하려면 추상화가 기계나 사람이 그대로 따라 실행할 수 있을 만큼 명확하고 정밀하게 표현되어 있어야 한다. 자동화는 사람이 직접 수행하기 어려운 작업을 빠르고 정확하게, 그리고 대규모로 처리할 수 있게 해 준다는 점에서 추상화에 힘을 더해 준다.

출처 : Wing(2008), Wing(2011)

2-7 컴퓨팅 사고력과 코딩의 관계 (1단원)

컴퓨팅 사고는 단순히 프로그램 코드를 작성하는 기능만을 의미하지 않는다. 컴퓨팅 사고는 문제의 핵심을 파악하고, 문제를 작은 부분으로 나누며, 해결 과정을 컴퓨터나 사람이 실행할 수 있는 명확한 단계로 표현하는 사고 과정이다.

K-12 Computer Science Framework는 컴퓨팅 사고를 “컴퓨터가 실행할 수 있는 계산 단계나 알고리즘으로 해결 방법을 표현하는 데 관련된 사고 과정”으로 제시한다.

컴퓨팅 사고는 프로그램 코드를 작성하기 전부터 시작된다. 해결하려는 문제가 무엇인지 정하고, 필요한 정보와 불필요한 정보를 구분하며, 해결 절차를 설계하는 활동도 컴퓨팅 사고에 포함된다. 따라서 컴퓨터를 사용하지 않는 언플러그드 활동에서도 컴퓨팅 사고를 기를 수 있다. 반대로 컴퓨터를 사용했다는 사실만으로 컴퓨팅 사고가 이루어졌다고 볼 수는 없다.

출처: K-12 Computer Science Framework Steering Committee(2016)

2-8 초등학교 수준의 알고리즘 학습 (전 단원)

2022 개정 실과(기술·가정)/정보과 교육과정에서는 문제 해결을 위한 명령에는 명확한 절차가 필요하다고 보고, 일상생활의 문제를 해결하기 위한 알고리즘을 구상하고 다양한 방법으로 표현하도록 한다. 특히 언플러그드 활동 등을 통한 문제 해결 과정을 알고리즘으로 작성하기 위해 자연어, 의사코드(pseudocode), 순서도 등으로 표현하도록 제시하고 있다.

CSTA(Computer Science Teachers Association)에 따르면 초등학교 3~5학년 수준에서는 같은 과제를 해결하는 여러 알고리즘을 비교하고 개선하여 가장 적절한 알고리즘을 정하도록 한다. 또한 문제를 작고 다루기 쉬운 하위 문제로 나누고, 프로그램이나 알고리즘이 의도한 대로 작동하는지 테스트하며 오류를 찾아 고치도록 한다.

출처: 교육부 실과(기술·가정)/정보과 교육과정, CSTA(2017)

2-9 알고리즘 - 문제 해결의 절차 (2단원)

알고리즘은 어떤 문제를 해결하기 위해 거쳐야 할 단계들을 순서대로 정리한 명확한 절차다. 2022 개정 교육과정 실과(정보영역)는 성취기준 [6실05-01]의 해설에서, 언플러그드 활동 등을 통한 문제 해결 과정을 알고리즘으로 작성하기 위해 자연어, 의사코드(pseudocode), 순서도 등으로 문제 해결 과정을 표현하도록 안내하고 있다. 즉 알고리즘은 컴퓨터 사용 여부와 무관하게 '문제를 해결하는 절차'를 가리키며, 컴퓨터를 쓰지 않는 활동에서도 충분히 다룰 수 있는 사고의 산물이다.

이러한 관점은 국제적으로 통용되는 컴퓨터과학 교육 기준과도 일치한다. 미국의 K-12 Computer Science Framework는 컴퓨팅 사고를 컴퓨터가 수행할 수 있는 계산 단계나 알고리즘으로 해결 방법을 표현하는 데 관련된 사고 과정으로 정의하면서, 컴퓨팅 사고는 컴퓨터의 능력을 이해하고, 컴퓨터로 다룰 문제를 정의하며, 컴퓨터가 실행할 수 있는 알고리즘을 설계하는 것을 요구한다고 설명한다.

출처: 교육부 실과(기술·가정)/정보과 교육과정; K-12 CSTA(2016)

2-10 프로그램의 세 가지 구조 - 순차 구조 (2단원)

순차는 가장 기본이 되는 구조로, 명령이 작성된 순서대로 위에서 아래로 하나씩 실행되는 방식이다. 순차에서는 명령이 선형적이고 예측 가능하게 하나씩 차례로 실행되어 명확한 흐름을 보장한다. 라면을 끓일 때 물을 끓이고, 면을 넣고, 스프를 넣는 순서를 지켜야 하듯, 순서가 바뀌면 결과가 달라지는 일상의 절차가 순차 구조에 해당한다. 학생이 블록을 위에서 아래로 쌓아 실행하는 가장 처음의 경험이 바로 이 구조를 익히는 과정이다.

출처: Bohm & Jacopini(1966)

2-11 프로그램의 세 가지 구조 - 선택 구조 (2단원)

선택은 주어진 조건이 참인지 거짓인지에 따라 실행할 명령을 달리하는 구조다. 선택, 즉 조건문은 프로그램이 참 또는 거짓이라는 두 갈래의 논리적 경로 중 하나를 고르도록 한다. "비가 오면 우산을 챙기고, 그렇지 않으면 그냥 나간다"처럼 상황을 판단해 다르게 행동하는 일상의 의사결정이 선택 구조에 대응한다. 선택 구조는 프로그램이 단순히 정해진 길만 가는 것이 아니라, 상황에 반응하며 다르게 동작할 수 있게 해 주는 핵심 장치다. 선택 구조는 매우 근본적이어서 사실상 거의 모든 다른 프로그래밍 패러다임에서도 유지되고 있다.

출처: Bohm & Jacopini(1966)

2-12 프로그램의 세 가지 구조 - 반복 구조 (2단원)

반복은 같은 명령을 조건이 만족되는 동안, 혹은 정해진 횟수만큼 되풀이하는 구조다. 반복, 즉 루프는 정해진 형식의 순환을 통해 코드 블록을 제어된 방식으로 되풀이하게 한다. 같은 동작을 여러 번 적어야 할 때, 반복 구조를 쓰면 명령을 단 한 번만 작성하고도 여러 번 실행할 수 있어 프로그램이 간결해진다. 반복 구조는 되풀이를 멈추는 기준을 무엇으로 두느냐에 따라 크게 횟수 반복과 조건 반복으로 나눌 수 있다.

① 횟수 반복(count-controlled loop)

횟수 반복은 정해진 횟수만큼 명령을 되풀이하는 방식이다. 반복을 시작하기 전에 몇 번 되풀이할지가 이미 정해져 있으며, 반복한 횟수를 세는 변수를 두어 그 값이 정해진 횟수에 이르면 반복이 끝난다. "구구단 2단을 1부터 9까지 외운다"처럼 되풀이할 횟수가 분명하게 정해진 상황이 횟수 반복의 좋은 예다. 같은 작업을 수를 바꿔 가며 정해진 만큼 되풀이할 때 적합하다.

② 조건 반복(condition-controlled loop)

조건 반복은 주어진 조건이 참인 동안 명령을 계속 되풀이하는 방식이다. 횟수 반복과 달리 몇 번 되풀이할지가 미리 정해져 있지 않으며, 매 반복마다 조건을 확인하여 조건이 더 이상 만족되지 않으면 반복을 멈춘다. 따라서 상황에 따라 되풀이되는 횟수가 달라진다. "정답을 맞힐 때까지 문제를 다시 푼다"처럼 끝나는 시점이 조건에 따라 달라지는 상황이 조건 반복에 해당한다. 되풀이할 횟수를 미리 알 수 없고, 어떤 조건이 충족될 때까지 작업을 이어 가야 할 때 적합하다.

이처럼 횟수 반복과 조건 반복은 '되풀이를 멈추는 기준'에서 차이가 있다. 횟수 반복은 정해진 횟수를 기준으로, 조건 반복은 조건의 만족 여부를 기준으로 반복을 제어한다. 학생이 블록을 위에서 아래로 쌓아 실행하는 경험에서 출발하여, 두 가지 반복 방식을 모두 다루어 보면 상황에 맞는 반복 구조를 선택하는 사고를 기를 수 있다.

출처: Bohm & Jacopini(1966), CSTA(2017)

2-13 의사코드 (전 단원)

의사코드(pseudocode)는 특정 프로그래밍 언어의 문법에 얽매이지 않고, 알고리즘의 논리적 흐름을 자연어와 프로그래밍 언어의 중간 형태로 표현하는 방식이다. '의사(疑似, pseudo)'는 '가짜' 또는 '유사한'이라는 뜻으로, 실제로 컴퓨터에서 실행되지는 않지만 코드처럼 보이는 표현임을 나타낸다. 의사코드라는 용어는 1960~70년대 알고리즘 설계와 구조적 프로그래밍 연구가 활발해지면서 학술 문헌에 등장하였다. 특히, 도널드 크누스(Donald Knuth) 교수의 저서 'The Art of Computer Programming (1968)'에서 알고리즘을 특정 언어에 종속되지 않게 설명하는 방법으로 사용되면서 널리 알려졌다.

의사코드는 프로그래밍 언어를 처음 배우는 학습자가 문법 오류에 신경 쓰지 않고 문제 해결의 절차와 논리에 집중할 수 있도록 돕는다는 점에서 교육적 가치가 크다. 2022 개정 교육과정에서도 알고리즘 표현 방법의 하나로 의사코드를 제시하고 있으며, 자연어와 순서도와 함께 초등학교 수준의 알고리즘 학습에서 중요한 표현 도구로 활용된다.

출처: Knuth(1968), 교육부(2022)

2-14 프로그램 공유 문화 (3단원)

프로그램은 개인이 혼자 완성하고 끝내는 결과물만이 아니라, 다른 사람과 공유하고 서로의 아이디어를 참고하면서 계속 발전할 수 있는 산출물이다. 특히 Scratch와 같은 교육용 프로그래밍 환경에서는 자신이 만든 프로젝트를 공유하고, 다른 사람이 만든 프로젝트를 살펴보며, 그 아이디어를 바탕으로 새로운 프로그램을 만들어 가는 문화를 중요하게 다룬다.

이러한 프로그램 공유 문화는 Mitchel Resnick이 강조한 창의적 학습의 관점과 연결된다. 레즈닉은 학습자가 아이디어를 상상하고, 직접 만들고, 실행하며 실험하고, 다른 사람과 공유한 뒤, 그 경험을 성찰하면서 다시 새로운 아이디어로 나아가는 과정을 중요하게 보았다. 이때 공유는 완성된 결과물을 보여 주는 마지막 단계가 아니라, 다른 사람의 반응과 아이디어를 통해 다시 생각하고 새롭게 만들어 가는 과정의 일부이다.

Scratch에서 말하는 Remix는 이러한 공유 문화를 잘 보여 주는 사례이다. Remix는 다른 사람이 공유한 프로젝트의 복사본을 만들고, 여기에 자신의 아이디어를 더해 새로운 프로젝트로 발전시키는 활동을 뜻한다. 이는 단순히 다른 사람의 작품을 베끼는 것이 아니라, 기존 프로그램의 구조와 표현을 살펴보고, 그 위에 새로운 기능이나 표현을 덧붙이는 창작 활동으로 볼 수 있다.

프로그램 공유 문화는 학생들이 다른 사람의 프로그램을 통해 새로운 아이디어를 얻고, 프로그램이 어떻게 구성되어 있는지 탐색하며, 자신의 산출물을 더 나은 방향으로 개선하는 데 도움을 준다. 또한 프로그램을 혼자만의 결과물로 이해하는 데서 나아가, 서로의 아이디어와 피드백을 바탕으로 함께 발전시킬 수 있는 창작물로 바라보게 한다.

출처 : Resnick(2017), Scratch Help Center(2024), Dasgupta, Hale, Monroy-Hernández, & Hill(2016).

2-15 Seymour Papert의 구성주의(Constructionism) (전 단원)

페퍼트의 구성주의(constructionism)는 학습자가 지식을 수동적으로 전달받는 것이 아니라, 자신에게 의미 있는 산출물을 직접 만들고 발전시키는 과정에서 지식을 구성해 간다는 관점이다. 페퍼트는 이를 간단히 "만들면서 배우기"로 설명할 수 있다고 보았지만, 동시에 그런 짧은 표현만으로는 구성주의의 의미를 충분히 담기 어렵다고 보았다. 여기서 중요한 것은 단순히 무엇인가를 손으로 만드는 활동 자체가 아니다. 학습자가 자신의 아이디어를 바깥으로 드러나는 산출물로 만들고, 그 산출물을 다시 살펴보고, 고치고, 다른 사람과 나누는 과정 속에서 생각이 더 깊어지는 것이 핵심이다.

페퍼트는 학습을 지식이 한 사람에게서 다른 사람에게 그대로 전달되는 과정으로 보지 않았다. 교사가 설명한 내용이 학생의 머릿속에 그대로 복사되는 것이 아니라, 학생은 자신의 경험과 관심, 사고방식에 따라 지식을 스스로 구성한다. 이 점에서 페퍼트의 구성주의는 학습자가 능동적으로 지식을 구성한다는 일반적인 구성주의 관점과 연결된다. 그러나 페퍼트는 여기에 한 가지를 더 강조하였다. 학습은 특히 학습자가 공적으로 드러나는 산출물을 만들 때 더욱 풍부하게 일어난다는 점이다. 그 산출물은 모래성, 이야기, 그림, 로봇, 프로그램, 또는 하나의 이론이 될 수도 있다.

이때 산출물은 학습이 끝난 뒤 남는 결과물에 그치지 않는다. 산출물은 학습자의 생각이 바깥으로 표현된 대상이며, 학습자가 자신의 생각을 다시 들여다보게 하는 매개이다. 머릿속에 있을 때는 막연했던 아이디어도 산출물로 표현되면 눈으로 보고, 실행해 보고, 설명하고, 수정할 수 있는 대상이 된다. 학생은 자신이 만든 것을 보면서 "내가 생각한 대로 되었는가?", "어떤 점이 다르게 작동하는가?", "무엇을 바꾸면 더 나아질 수 있는가?"를 생각하게 된다. 따라서 산출물을 만드는 과정은 단순한 활동 수행이 아니라, 자신의 사고를 외부화하고 다시 재구성하는 과정이다.

페퍼트는 이러한 학습을 설명하기 위해 비누 조각 수업의 사례를 제시하였다. 일반적인 수학 수업에서는 학생들이 짧은 문제를 풀고, 맞거나 틀린 결과를 확인한 뒤 다음 문제로 넘어가는 경우가 많다. 반면 비누 조각 수업에서는 학생들이 여러 주에 걸쳐 자신만의 작품을 만들었다. 그 과정에는 생각할 시간, 바라볼 시간, 새로운 아이디어를 떠올릴 시간, 시도하고 포기하거나 지속할 시간, 다른 사람의 작품을 보고 자신의 작품에 대한 반응을 들을 시간이 있었다. 페퍼트는 수학 수업도 바로 이런 방식에 가까워지기를 바랐다. 즉, 수학도 학생에게 외부에서 주어진 문제풀이가 아니라, 자신의 아이디어를 표현하고 다듬어 가는 창작 과정이 될 수 있어야 한다고 본 것이다.

프로그래밍은 페퍼트의 구성주의를 잘 보여 주는 대표적인 학습 활동이다. 학생이 프로그램을 만든다는 것은 단순히 명령어를 순서대로 배열하는 일이 아니다. 그것은 자신의 생각과 문제 해결 절차를 컴퓨터가 실행할 수 있는 형태로 표현하는 일이다. 학생이 캐릭터를 움직이게 하거나, 조건에 따라 다른 반응이 나타나게 하거나, 로봇이 특정한 동작을 하도록 만들 때, 학생의 사고는 코드, 블록, 화면 움직임, 소리, 조건, 반복, 변수와 같은 구체적인 형태로 드러난다. 이때 프로그램은 학생의 생각이 담긴 산출물이자, 학생이

자신의 생각을 다시 살펴볼 수 있게 하는 대상이 된다.

프로그램은 실행된다는 점에서 더욱 중요한 학습 산출물이 된다. 학생이 머릿속으로 생각한 절차가 실제 프로그램으로 만들어지면, 그 생각은 곧바로 실행 결과로 나타난다. 예상대로 작동할 수도 있지만, 그렇지 않을 수도 있다. 캐릭터가 원하는 방향으로 움직이지 않거나, 반복이 멈추지 않거나, 조건이 제대로 작동하지 않거나, 로봇이 의도와 다른 방향으로 움직일 수 있다. 이때 오류는 단순한 실패가 아니다. 오류는 학생이 자신의 생각과 실행 결과를 비교하게 만드는 중요한 단서가 된다. 학생은 프로그램을 다시 살펴보며 자신의 절차적 사고를 점검하고, 필요한 부분을 수정하면서 사고를 더 정교하게 만들어 간다.

따라서 디버깅은 페퍼트의 구성주의 관점에서 중요한 학습 과정이다. 디버깅은 단순히 틀린 부분을 찾아 고치는 기술적 절차가 아니다. 학생이 자신의 생각이 어떻게 표현되었는지 확인하고, 예상과 실제 결과 사이의 차이를 발견하며, 그 차이를 줄이기 위해 사고를 조정하는 과정이다. 학생은 오류를 고치면서 절차, 조건, 반복, 변수, 이벤트와 같은 컴퓨팅 개념을 자신의 경험 속에서 이해하게 된다. 이 과정에서 학생은 정답 코드를 외우는 것이 아니라, 자신이 만든 산출물과 상호작용하면서 개념을 구성한다.

페퍼트는 LEGO/Logo와 같은 환경을 통해 학생들이 수학, 과학, 상상력을 함께 사용하여 산출물을 만들 수 있다고 보았다. 학생들이 LEGO/Logo로 뱀을 만들 때, 그들은 단순히 장난감을 조립한 것이 아니라, 자신이 상상한 생명체의 움직임을 표현하기 위해 수학과 과학을 사용하였다. 비누 조각에서 칼이 비누의 형태를 만들어 내는 도구였다면, LEGO/Logo에서는 수학이 뱀의 움직임을 만들고, 물리 개념이 그 구조를 이해하게 하는 도구가 된다. 이 사례는 프로그래밍과 로봇 활동이 단순한 기능 학습이 아니라, 상상력과 수학적·과학적 사고가 만나는 표현 활동이 될 수 있음을 보여 준다.

또한 페퍼트는 모든 학생이 같은 방식으로 배우거나 같은 방식으로 문제를 해결한다고 보지 않았다. 어떤 학생은 미리 계획을 세우고 위계적으로 절차를 구성하는 방식으로 프로그램을 만들 수 있다. 반면 어떤 학생은 마치 화가가 그림을 그리듯이, 먼저 화면에 무언가를 만들고, 그것을 보면서 조금씩 수정하고 확장해 갈 수 있다. 페퍼트는 이러한 다양한 작업 방식이 모두 학습에서 의미를 가질 수 있다고 보았다. 중요한 것은 학생이 하나의 정해진 방식으로만 사고하도록 강요받는 것이 아니라, 자신에게 맞는 방식으로 산출물을 만들고 발전시킬 수 있는 환경을 제공받는 것이다.

이 점에서 페퍼트의 구성주의는 교수주의(instructionism)와 대비된다. 교수주의는 지식을 잘게 나누어 교사가 설명하고, 학생이 그것을 받아들이도록 하는 방식에 가깝다. 페퍼트가 비판한 것은 설명이나 지도가 모두 나쁘다는 뜻이 아니다. 핵심은 교육의 혁신이 단지 더 효과적인 설명 방법을 찾는 데만 머물러서는 안 된다는 것이다. 학생이 배우고 싶어지는 맥락, 즉 자신이 만들고 싶은 것을 만들기 위해 수학, 과학, 프로그래밍 개념을 필요로 하는 상황을 제공할 때, 학습은 더 깊어질 수 있다. 학생은 외부에서 주어진 지식을 억지로 익히는 것이 아니라, 자신의 산출물을 만들기 위해 필요한 지식을 능동적으로 사용하게 된다.

따라서 초등 프로그래밍 교육에서 페퍼트의 구성주의는 중요한 의미를 갖는다. 초등학생에게 프로그래밍은 전문 개발 기술을 조기에 익히는 일이 아니라, 자신의 생각을 절차적으로 표현하고, 실행

결과를 보며 사고를 조정하고, 산출물을 발전시키는 경험이어야 한다. 학생이 만든 애니메이션, 게임, 로봇 동작, 퀴즈 프로그램, 발표 타이머 등은 단순한 과제 결과물이 아니라 학생의 사고가 드러난 학습 산출물이다. 학생은 그 산출물을 통해 자신의 생각을 확인하고, 오류를 수정하고, 더 나은 방식으로 개선하면서 컴퓨팅 개념을 경험 속에서 이해하게 된다.

결국 페퍼트의 구성주의에서 프로그램은 완성해야 할 과제가 아니라, 학생의 생각이 드러나는 산출물이며, 학생이 자신의 생각을 다시 살펴보게 하는 매개이다. 프로그래밍 수업은 정답 코드를 따라 쓰고 결과를 확인하는 활동에 머물러서는 안 된다. 학생이 자신의 아이디어를 프로그램으로 표현하고, 실행하고, 오류를 발견하고, 수정하고, 다시 발전시키는 과정이 중심이 되어야 한다. 이러한 과정 속에서 학생은 컴퓨팅 개념을 단순히 아는 것을 넘어, 자신의 생각을 표현하고 발전시키는 도구로 활용하게 된다.

출처: Harel & Papert(1991)

3 정보 용어 사전

용어	의미
컴퓨터	① 1개 이상의 처리 장치와 주변 기기로 구성되며, 내부에 저장된 프로그램에 의해 제어받고, 산술 연산이나 논리 연산과 같은 여러 종류의 실질적인 계산 작업을 수행할 수 있는 장치이다. ② 정보를 입력하여 그 정보를 정해진 과정대로 처리하고 그 결과를 제공할 수 있는 기기. 구성 요소는 입출력 장치, 기억 장치, 연산 장치, 제어 장치이다. 출처: 한국정보통신기술협회, TTA 정보통신용어사전
알고리즘	알고리즘은 원하는 결과를 얻기 위해 수행해야 할 작업의 절차를 말하며, 입력이 주어진 초기 상태로부터 출력이 산출된 최종 상태에 이를 때까지 수행해야 할 연산들의 순차이다. 출처: 한국교육학술정보원(KERIS), 소프트웨어 교육을 위한 용어 해설 자료집(2015)
자연어	컴퓨터에서 사용하는 프로그램 작성 언어 또는 기계어와 구분하기 위해 인간이 일상생활에서 의사 소통을 위해 사용하는 언어를 가리키는 말이다. 출처: 한국정보통신기술협회, TTA 정보통신용어사전
의사 코드	프로그래밍 언어와 비슷한 형식으로, 사람이 읽기 쉽게 명령의 순서를 적은 것이다. 출처: 한국정보통신기술협회, TTA 정보통신용어사전
순서도	연산, 자료, 흐름, 도구를 표현하는 기호를 이용한 정의, 분석, 문제의 해결책에 대한 도식이다. 출처: 한국정보통신기술협회, TTA 정보통신용어사전
디버깅	컴퓨터 프로그램의 오류를 수정하는 일련의 작업이다. 하나의 작은 에러가 많은 문제를 불러 올 수 있다. 프로그램 에러를 버그(bug)라 부르고 이런 에러를 찾아 수정·제거하는 과정을 일컫는다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)
프로그램	컴퓨터 시스템이 특정한 작업을 수행하거나 원하는 결과를 얻을 수 있도록 일련의 명령어들을 논리적으로 구성한 것이다. 출처: 한국정보통신기술협회, TTA 정보통신용어사전
프로그래밍	특정 목적을 달성하기 위해 컴퓨터가 실행할 수 있는 프로그램을 설계하는 과정이다. 문제 상황에서 해결 방법을 생각하고, 알고리즘을 설계한 뒤, 이를 프로그래밍 언어로 표현하는 일련의 과정을 말한다. 프로그래밍 언어로 코드를 작성한다는 의미에서 코딩(coding)이라고도 부른다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)
프로그래밍 언어	사람이 컴퓨터에게 명령을 전달하기 위해 사용하는 언어로, 컴퓨터가 수행해야 할 명령문을 작성할 수 있도록 도와주는 기호와 문법으로 구성된다. 컴퓨터는 사람의 말을 바로 이해하지 못하므로, 컴퓨터에게 내릴 명령어를 사람이 이해할 수 있는 형태로 만들어 놓은 것이 프로그래밍 언어다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)

용어	의미
순차 구조	알고리즘이 작성된 순서대로 명령이 수행되는 구조. 대부분의 프로그래밍 언어에서는 특별한 표기 방법이 존재하지 않으며 작성된 순서대로, 즉 위에서 아래로 혹은 왼쪽에서 오른쪽으로 수행되도록 한다. 출처: 교육부, KERIS, 소프트웨어와 함께하는 창의력 여행(2016)
반복 구조	특정 조건이 만족되는 동안, 또는 정해진 횟수만큼 같은 명령을 되풀이하는 프로그램 구조이다. 같은 명령을 여러 번 작성하지 않아도 되므로 프로그램을 간결하게 만들 수 있다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)
선택 구조	주어진 조건의 참·거짓 여부에 따라 서로 다른 명령을 실행하는 프로그램 구조이다. 조건이 참이면 A를 실행하고, 거짓이면 B를 실행하는 방식으로, 상황에 따라 프로그램이 다르게 동작할 수 있도록 한다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)
변수	프로그램 실행 중에 변할 수 있는 값을 저장하는 공간이다. 이름을 붙여 값을 기억하고, 필요할 때 꺼내 쓰거나 새로운 값으로 바꿀 수 있다. 출처: 한국교육학술정보원·한국정보과학교육연합회, 소프트웨어 교육을 위한 용어 해설 자료집(2015)

4 저작권 및 AI 윤리 자료

관련 단원	내용
	문화체육관광부 및 한국저작권위원회에서 출판한 자료로, 청소년저작권 자료를 개발 및 배포하였습니다. 하단 링크를 통해 접속 가능하며, 배움터, 놀이터, 나눔터에서 다양한 방식으로 저작권 교육을 실시할 수 있습니다. 특히, 나눔터에서는 학교급별, 학년군별 교안을 제공하여 학교 실정에 맞는 저작권 교육이 가능합니다. 출처: 한국저작권위원회 https://edu-copyright.or.kr/user/schl/selectYothInfo.do?p_menuId=242&menuId=394
	2020년 '윤리적 AI을 위한 국가정책 수립' 정책연구를 시작으로 과학기술정보통신부는 현장 연구에 부합하는 AI 윤리기준을 제시하고, 실천을 위한 구체적 지침을 제공하고 있습니다. 이에 발맞춰 학교급에 알맞은 AI 윤리교재를 개발·보급하였습니다. 출처: 인공지능 윤리 소통채널 https://ai.kisdi.re.kr/aieth/bbs/B0000082/list.do?menuNo=400011
	학년 초 학생들의 균형 잡힌 디지털 활용 습관과 책임 있는 시민의식 함양을 위한 디지털 활용 능력 및 디지털 윤리 역량과 관련한 도움 자료집입니다. 학생들과 수업에 바로 활용할 수 있는 교수학습자료(PPT)와 수업지도안이 탑재되어 있어 현장에 바로 활용할 수 있습니다. 출처: 서울시교육청 창의미래교육과 https://drive.google.com/drive/folders/1HDiMhzeCN_fQ6rC4k3T1KJppAfqk4k86?usp=drive_link

관련 단원	내용
	서울형 인공지능 윤리 교육 내용체계에 따라 초등학생에게 적합한 성취기준을 선정하여 학년군별 계열성을 고려해 개발한 자료입니다. 인공지능 윤리의 3대 주제 10개 하위 영역에 대해 3-4학년 2차시, 5-6학년 2차시의 자료로 개발되었습니다. 수업지도안과 교수학습자료(PPT)를 제공합니다. 출처: 서울시교육청 교육혁신과 https://mail.sen.go.kr/webhard/popup/webLinkDownload.do?lang=ko&usedate=0&a=NjhiM2EzMjIzZmRjNmVlNDY1YjljYzM2
	디지털시민 One-Team에서 개발한 디지털 시민 프레임워크의 3대 핵심가치 및 9개 하위주제를 바탕으로 개발된 자료입니다. 디지털 안전, 디지털 공존, 디지털 책임 3대 핵심가치를 중심으로 초등학교 3학년부터 6학년까지 학년별 위계성과 연계성을 고려하여 학교 현장에서 바로 적용 가능한 디지털 시민성 교육 내용을 체계적으로 제시하고 있습니다. 출처: 서울시교육청 디지털혁신미래교육과 https://mail.sen.go.kr/webhard/popup/webLinkDownload.do?lang=ko&usedate=0&a=NjhhZmQ2NzYzZmNkNDdmZmI3ZjU3ZjI4
	Common Sense Media(CSM) 이하의 디지털 시민성 교육과정을 바탕으로 개발된 한국형 디지털 시민성 교육과정의 목적과 목표, 내용체계, 교육과정 구성을 따라 개발된 교육자료입니다. 수업지도안, 강의안, 활동자료(학생용), 퀴즈, 활동지를 필수적으로 포함하고 있어 창의적 체험활 혹은 학교 자율시간 등 다양한 교육활동 시간을 활용하여 디지털 시민성 교육을 운영할 수 있습니다. 출처: 서울시교육청 창의미래교육과 https://mail.sen.go.kr/webhard/popup/webLinkDownload.do?lang=ko&usedate=0&a=NjhiM2E1MTczZmQ4ODg3OWEyYWE0OTE2

관련 단원	내용
	경기도교육청이 학생들의 주도성과 책임감 함양을 위해 개발한 인공지능 윤리 교육 장학자료입니다. 과학기술정보통신부의 AI 윤리 기준을 바탕으로 초·중등 교원들이 직접 연구·개발에 참여하였으며, 인공지능의 원리 이해를 넘어 올바른 활용과 오남용 예방, 사회적 문제 해결을 위한 윤리적 접근을 체험할 수 있도록 구성되었습니다. 초등용은 생활 속 AI 체험과 토론 활동 위주로, 중등용은 알고리즘 편향성·개인정보 보호·딥페이크 등 윤리적 쟁점을 다루며, 초·중등 수업 지도안 및 학습지 60편이 포함되어 있어 교사가 현장에서 즉시 활용할 수 있습니다. 출처: 경기도교육청 https://www.goe.go.kr/resource/old/BBSMSTR_000000030127/BBS_20220823114424839 2.pdf
	충남교육청이 2020년부터 매년 발간하고 있는 교과융합형 인공지능 교육 도움자료입니다. 학생이 미래핵심역량을 바탕으로 삶의 문제를 스스로 해결하는 참학력 인재로 성장하도록 지원하는 것을 목적으로 하며, 인공지능 이해·활용·융합·윤리의 4가지 영역을 모든 교과 교육과정에 실제적으로 반영할 수 있도록 재구성되었습니다. 본 자료는 초·중등 30주제 57차시로 구성되며, XR 콘텐츠·홀로그램 등 실감형 콘텐츠 교육 도구를 활용한 수업 사례를 초등 1학년부터 고등 2학년까지 학년별로 제공하고 있어 교사가 교과교육과정을 재구성하는 데 바로 활용할 수 있습니다. 출처: 충청남도교육청 https://www.3dbank.xyz/_docs/3DBANK-2025-CNE-AI-Help-XRCLASS.pdf

5 참고 문헌 및 사이트

- 교육부. (1987). 제5차 교육과정.
- 교육부. (1992). 제6차 교육과정.
- 교육부. (1997). 제7차 교육과정.
- 교육인적자원부. (2007). 2007 개정 교육과정.
- 교육과학기술부. (2009). 2009 개정 교육과정.
- 교육부. (2015). 실과(기술·가정)/정보과 교육과정 (교육부 고시 제2015-74호 [별책 10]).
- 교육부. (2016). 소프트웨어와 함께하는 창의력 여행: 지도서.
- 교육부. (2022). 실과(기술·가정)/정보과 교육과정 (교육부 고시 제2022-33호 [별책 10]).
- 김진숙, 한선관, 김수환, 정순원, 양재명, 장의덕, 김정남 (2015). SW교육 교수학습모델 개발 연구 (연구보고 CR 2015-35). 한국교육개발원·한국교육학술정보원.
- 신승기. (2021). 초등학교 정보교육의 국가수준 교육과정 편성에 대한 내용체계 구성 연구. 정보교육학회논문지, 25(1), 13-31.
- 신승기. (2025). 2022 개정교육과정에 따른 초등학교 정보교육의 학습공백 분석 연구. 창의정보문화연구, 11(4), 469-483.
- 한국교육학술정보원·한국정보과학교육연합회. (2015). 소프트웨어 교육을 위한 용어 해설 자료집 (교육자료 GM 2015-9).
- 한국저작권위원회. 청소년 저작권 교육 자료. 저작권 e-배움터. https://edu-copyright.or.kr/user/schl/selectYothInfo.do?p_menuId=242&menuId=394
- 한국정보통신기술협회. TTA 정보통신용어사전. https://terms.tta.or.kr
- Beck, K. (1999). Extreme Programming Explained: Embrace Change
- Bohm, C., & Jacopini, G. (1966). Flow diagrams, Turing machines and languages with only two formation rules. Communications of the ACM, 9(5), 366-371.
- Computer Science Teachers Association. (2017). CSTA K-12 computer science standards: Revised 2017. https://csteachers.org/k12standards/
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking: A guide for teachers. Computing At School.
- CSTA. (2017). CSTA K-12 Computer Science Standards.
- Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). Remixing as a pathway to computational thinking. Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, 1438-1449.

- Encyclopaedia Britannica. Encyclopaedia Britannica. https://www.britannica.com/
- Harel, I., & Papert, S. (Eds.). (1991). Constructionism: Research reports and essays, 1985-1990. Ablex Publishing Corporation.
- Irene Lee, Fred Martin, and Katie Apone. 2014. Integrating computational thinking across the K-8 curriculum. Acm Inroads 5, 4 (2014), 64-71.
- K-12 Computer Science Framework Steering Committee. (2016). K-12 computer science framework. https://k12cs.org/
- Laurie Williams & Robert Kessler (2002). Pair Programming Illuminated
- Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. ACM Inroads, 5(4), 64-71.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32-37.
- Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. Science & Education, 30, 883-908.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books.
- Papert, S. (1993). The children's machine: Rethinking school in the age of the computer. Basic Books.
- Resnick, M. (2017). Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play. MIT Press.
- Scratch Help Center. (2024). Remixing: What is it and why can every project on Scratch be remixed?
- Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. Proceedings of the 12th Workshop on Primary and Secondary Computing Education. ACM.
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. Computer Science Education, 29(2-3), 136-176.
- Williams, L., & Kessler, R. (2002). Pair programming illuminated. Addison-Wesley.
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A, 366(1881), 3717-3725.
- Wing, J. M. (2011, March 6). Research notebook: Computational thinking—What and why?

6 생성형 AI 활용 내역

번호	그림 위치	생성형 AI 모델	프롬프트
1	1단원 삽화	ChatGPT, GPT 5.5	초등학교 실과 정보교육 단원「생활 속 컴퓨터와 문제 해결」에 들어갈 간결한 교과서 삽화. 흰색 또는 연한 크림색 배경의 따뜻한 파스텔톤 벡터 일러스트. 초등학교 여학생이 책상 앞에서 태블릿 PC를 보며 생활 속 문제를 컴퓨터로 해결하는 모습을 표현한다. 여학생은 태블릿 PC를 들고 있고 학생 위쪽에는 생활 속 컴퓨터 활용을 상징하는 작은 아이콘 두 개를 배치한다: 지도 앱 위치 표시, 버스 도착 정보 표시. 전체적으로 복잡하지 않고 여백이 충분하며, 교과서 단원 표지에 넣기 좋은 부드럽고 친근한 분위기. 글자나 말풍선은 넣지 않는다. 장식 요소는 최소화하고, 초등학교 5~6학년 교재에 어울리는 깔끔한 가로형 삽화.
2	1단원 2차시 교수·학습 과정 내 삽화	ChatGPT, GPT 5.5	초등학교 실과 정보교육 단원「생활 속 컴퓨터와 문제 해결」활동지에 들어갈 가로형 교과서 삽화. 한 장의 그림을 좌우 두 칸으로 나누어, 왼쪽은 '현재 상태', 오른쪽은 '목표 상태'를 나란히 보여 준다. 가운데에 옅은 회색 세로 구분선을 둔다. 두 칸 모두 같은 여자아이(밤색 단발머리, 노란 티셔츠)가 등장해 동일 인물·동일 화풍으로 그린다. 흰색 또는 연한 크림색 배경의 따뜻한 파스텔톤 벡터 일러스트. 왼쪽 칸(현재 상태): 여자아이가 갈림길 앞에서 도서관을 찾지 못해 두리번거리며 난처해하는 모습. 머리 위에 물음표(?) 한두 개. 손에 든 스마트폰 화면은 비어 있다. 길을 몰라 막막한 분위기. 오른쪽 칸(목표 상태): 같은 여자아이가 스마트폰 지도 앱 화면을 보며 환하게 웃는 모습. 화면에는 출발지에서 도서관(책 모양 아이콘이 있는 건물)까지 이어지는 파란색 경로 선과, 걸리는 시간을 나타내는 작은 시계 아이콘이 그려져 있다. 길을 찾아 안심한 분위기. 글자·문자·숫자·말풍선은 넣지 않는다(걸리는 시간은 시계 아이콘으로만 표현). 장식은 최소화하고 여백을 충분히 둔다. 초등학교 5~6학년 교재에 어울리는 깔끔하고 친근한 가로형 삽화.
3	1단원 4차시 교수·학습 과정 내 삽화 (집)	ChatGPT, GPT 5.5	교실 보드게임에 활용하기 좋은 깔끔하고 간결한 어린이 친화적 집 삽화. 흰색 배경의 밝고 부드러운 파스텔톤 벡터 일러스트. 정면을 바라보는 귀엽고 일반적인 집 아이콘을 중심에 배치한다. 삼각형 지붕, 가운데 문, 창문을 포함하고, 단순하면서도 굵은 외곽선과 둥글고 친근한 형태로 표현한다. 전체적으로 복잡하지 않고 초등학생이 쉽게 알아볼 수 있도록 구성한다. 배경 풍경은 넣지 않고, 장식 요소는 최소화한다. 글자, 로고, 브랜드 요소는 넣지 않는다. 초등학생용 교실 보드게임 자료에 어울리는 깔끔하고 친근한 분위기의 삽화.

번호	그림 위치	생성형 AI 모델	프롬프트
4	1단원 4차시 교수·학습 과정 내 삽화 (고양이)	ChatGPT, GPT 5.5	교실 보드게임에 활용하기 좋은 깔끔하고 간결한 어린이 친화적 마스코트 스타일의 고양이 삽화. 흰색 배경의 밝고 부드러운 파스텔톤 벡터 일러스트. 정면을 바라보는 일반적인 주황색 만화풍 고양이 캐릭터를 중심에 배치하고, 두 발로 서 있는 모습으로 표현한다. 고양이는 웃는 표정을 짓고 있으며, 크고 생동감 있는 눈, 둥글고 친근한 형태, 단순하면서도 굵은 외곽선을 지니도록 한다. 전체적으로 초등학생이 쉽게 알아볼 수 있을 만큼 친숙하고 이해하기 쉬운 모습으로 구성한다. 단순한 색상과 평면적인 벡터 스타일로 표현하고, 배경 풍경은 넣지 않는다. 글자, 로고, 브랜드 요소는 넣지 않는다. 전신이 모두 보이도록 하며, 화면 중앙에 배치한다.
5	1단원 4차시 교수·학습 과정 내 삽화 (장애물)	ChatGPT, GPT 5.5	교실 보드게임에서 장애물로 활용하기 좋은 깔끔하고 간결한 어린이 친화적 바리케이드 삽화. 흰색 배경의 밝고 부드러운 벡터 스타일 일러스트로 표현한다. 정면을 바라보는 일반적인 도로 바리케이드를 중심에 배치하고, 가로로 긴 직사각형 판을 두 개의 튼튼한 다리가 받치고 있는 형태로 그린다. 바리케이드에는 주황색과 흰색이 번갈아 나타나는 사선 줄무늬를 적용하고, 모서리는 둥글게 처리하며, 진하고 또렷한 외곽선과 단순한 색상을 사용한 평면적인 벡터 스타일로 표현한다. 전체적으로 초등학생이 쉽게 알아볼 수 있도록 명확하고 친숙하게 구성하되, 위험하거나 지나치게 사실적으로 보이지 않도록 한다. 배경 풍경은 넣지 않고, 흰색 배경으로 유지하며, 글자, 경고 표지, 로고, 브랜드 요소는 넣지 않는다. 바리케이드 전체가 잘 보이도록 화면 중앙에 단독으로 배치한다.
6	2단원 삽화	ChatGPT, GPT 5.5	컴퓨터 화면에서 다채로운 블록형 코딩을 활용하여 프로그램을 만드는 밝고 활발한 한국 초등학생 남학생 삽화. 흰색 배경의 밝고 친근한 평면 일러스트 스타일로, 한국 초등학교 교과서에 어울리는 분위기로 표현한다. 학생은 컴퓨터 모니터를 집중해서 바라보며 마우스를 사용하고 있다. 화면 왼쪽에는 블록 코딩 작업 공간이 보이고, 오른쪽에는 로봇 또는 거북이 캐릭터가 움직이는 모습이 나타나도록 한다. 전체적으로 친근하고 이해하기 쉬운 분위기로 구성하며, 글자는 넣지 않는다.
7	3단원 삽화	ChatGPT, GPT 5.5	노트북 앞에서 엔트리 소프트웨어로 협력하여 코딩하는 2명의 밝은 한국 남자 초등학생 1명과 여자 초등학생 1명(6학년) 삽화. 흰색 배경의 따뜻한 파스텔톤 평면 벡터 일러스트로, 교육적인 느낌이 잘 드러나도록 표현한다. 학생들은 함께 노트북 화면을 보며 협력하고 있으며, 한 학생은 간단한 블록 코딩 프로그램이 보이는 화면을 바라보고 있다. 학생들 주변에는 알람시계, 우산, 학교 가방과 같은 생활 속 물건 아이콘이 작은 말풍선처럼 떠 있도록 구성한다. 전체적으로 친근하고 부드러우며, 초등학교 교재에 어울리는 학습 분위기로 표현한다. 글자는 넣지 않는다.

기 획 김경숙 교육연구관(교육부 인공지능인재지원국 인공지능교육지능과)
서지연 본부장(한국과학창의재단 AI인재양성본부)
김형진 실장(한국과학창의재단 AI·융합교육실)
김혜미 연구원(한국과학창의재단 AI·융합교육실)

집 필 신승기(서울교육대학교)
최현재(서울마천초등학교)
황지영(서울해누리초등학교)
최문석(대광초등학교)
성윤호(내리숲초등학교)
박다빈(서울증산초등학교)

초등학교 정보교육

학습 보완 자료

발 간 2026년 8월
지은이 서울교육대학교산학협력단
펴낸곳 교육부·한국과학창의재단
디자인·인쇄 세일포커스㈜ 02-2275-6894

내용의 일부와 전부를 무단 전재하거나 복제를 금지합니다.