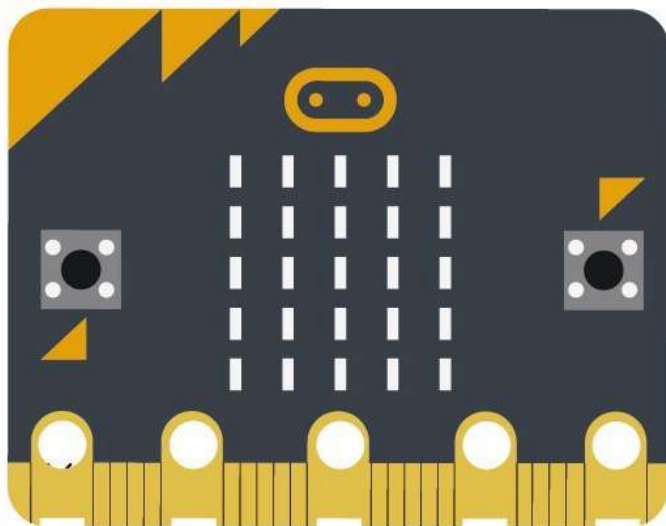


micro:bit

망치는 마이크로비트 배우기

디바이스마트 지음

기초편



DEVICEMART



변화를 꿈꾸는 사람들의 혁신을 위한 움직임
엔티렉스 각 사업부는 새로운 변화를 꿈꾸며,
더 나은 내일을 위해 혁신적으로 움직이고자 합니다.

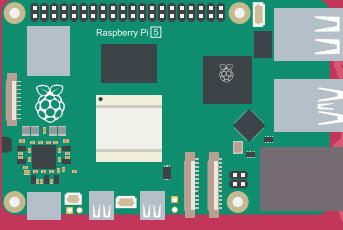
디바이스마트

NO.1!! 국내 최대·최다 전자, 로봇, 기계부품 전문 쇼핑몰 디바이스마트
최대 500만개 제품 데이터와 콘텐츠 제공을 통해 최고의 서비스와 고객만족을 제공합니다.

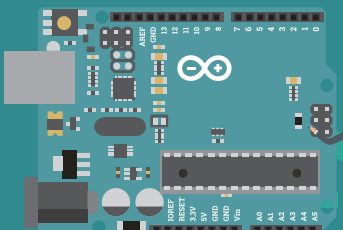
국내·외 유명 브랜드 공식 대리점



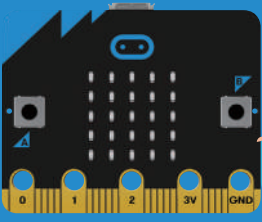
4차 산업의 선두주자 Raspberry Pi
라즈베리 파이



전세계 메이커에게 가장 사랑받는
오픈소스 플랫폼 아두이노



쉽고 재미있는 프로그래밍의 시작!
작은 컴퓨터 마이크로 비트



디바이스마트는 가장 유명한 프로그래밍 하드웨어





국내 유일 공식 리셀러는
오직 디바이스마트입니다.

TEL : 070-7019-8887 FAX : 02-6008-4953
E-MAIL : ebiz@ntrex.co.kr
www.devicemart.co.kr

디바이스마트가 사랑받는 이유!



후불 결제 서비스 운영

학교, 공공기관인 경우
물품 선수령 후 결제하는 서비스!



공모전 개최

이공계 분야 활성화를 위한 캡스톤 공모전,
ICT 융합 프로젝트 공모전 등을 주체!



메이커 서비스 운영

국내 최저가 샘플 PCB 제작 및 DIY 메이크
활동에 대한 모든것을 지원



중국 심천 지사 설립

중국 현지 원스탑 상품 개발, 소싱, 포장
국내 최저 가격 경쟁력 확보



물류 시스템 업그레이드

연면적 4500㎡, 지상 4층
첨단 통합 관리 시스템(입고, 보관, 피킹, 검수, 출고)
다양한 포장 시스템 개발



디바이스마트 블로그&유튜브 운영

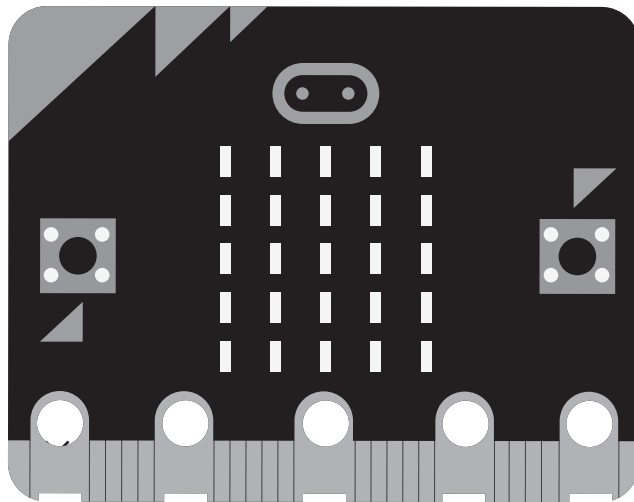
교육 자료 및 예제, 제품 활용 방법 및 예시,
동영상 자료, 메뉴얼 등 무료 배포!

micro:bit

맛있는 마이크로비트 배우기

디바이스마트 지음

기초편



DEVICEMART

목차

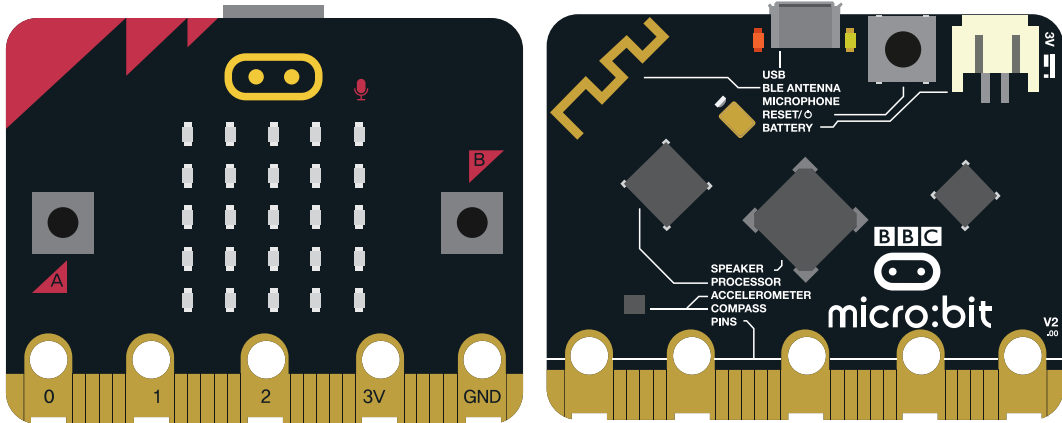
Chapter 1	마이크로비트 소개	5
Flavor 1	마이크로비트?	6
Chapter 2	기본 기능 익히기	14
Flavor 2	두근두근 하트 만들기	15
Flavor 3	가위바위보 게임 만들기	20
Flavor 4	주사위 만들기	25
Chapter 3	V2 기능 익히기	32
Flavor 5	미러볼 만들기	33
Flavor 6	메트로놈 만들기	38
Flavor 7	스톱워치 만들기	42
Chapter 4	마이크로비트 활용하기	49
Flavor 8	전자 촛불 만들기	50
Flavor 9	나침반 만들기	58
Flavor 10	마이크로비트 키우기	65
Chapter 5	응용하기 정답편	73

#01

마이크로비트 소개

01 FLAVOR

마이크로비트란? 마이크로비트에 대해 알아봅시다.



마이크로비트는 영국의 BBC에서 출시한 초등 교육용 마이크로 컨트롤러 보드입니다.

마이크로비트는 손에 들고 프로그래밍할 수 있는 작은 컴퓨터로서 로봇, 전자 악기 등 다양한 종류의 창작물을 만들어낼 수 있습니다.

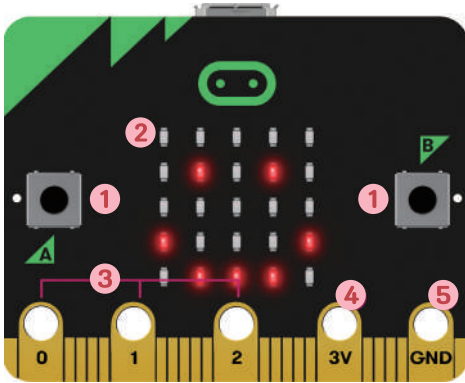
마이크로비트는 별도의 소프트웨어를 설치할 필요 없이, 웹브라우저에서 블록, 자바스크립트, 파이썬, 스크래치 등을 이용해 프로그래밍할 수 있습니다.

마이크로비트는 아래와 같은 하드웨어적 특징을 가지고 있습니다.

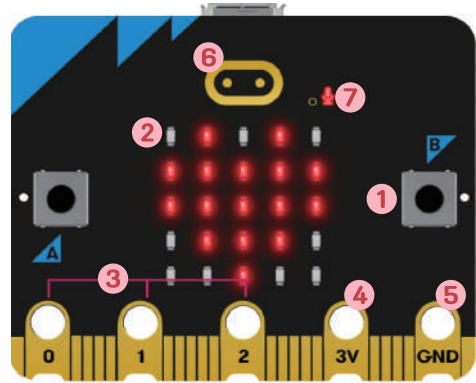
- 25개의 5×5 LED 매트릭스
- 2개의 버튼
- 입출력 핀과 전원 핀
- 터치센서
- 마이크와 스피커
- 조도 센서와 온도 센서
- 가속도 센서와 나침반 센서
- 라디오와 블루투스 통신
- USB 인터페이스



마이크로비트 V1과 V2 어떻게 다를까요?



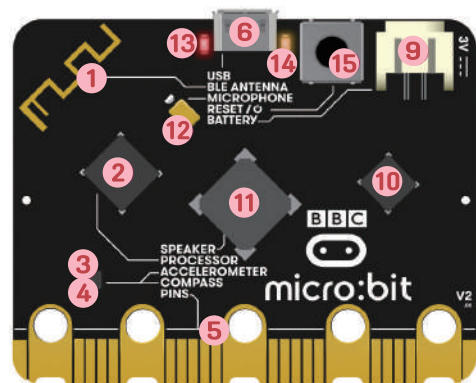
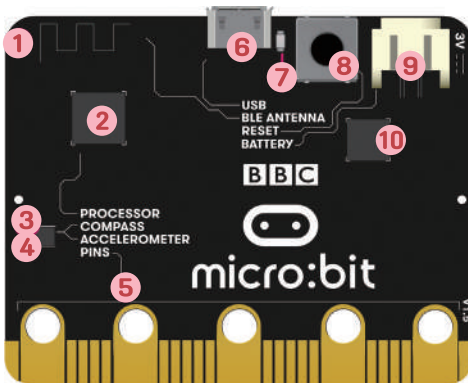
오리지널 마이크로비트 (V1)



새로운 마이크로비트 (V2)

전면

1	A, B 버튼	4	3V VCC핀	7	마이크 LED NEW
2	5X5 LED	5	GND 핀		
3	GPIO 핀	6	터치 센서 로고 NEW		

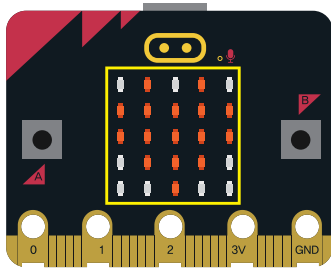


후면

1	라디오 및 블루투스 안테나	6	micro 5핀 USB 커넥터	11	스피커 NEW
2	CPU	7	다운로드 표시 LED	12	마이크 NEW
3	나침반	8	리셋 버튼	13	빨간색 전원 LED NEW
4	가속도계	9	배터리 소켓	14	노란색 다운로드 LED NEW
5	GPIO 핀	10	USB 인터페이스 칩	15	리셋 및 전원 LED NEW

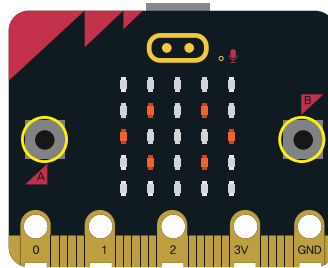


마이크로비트 기능 살펴보기



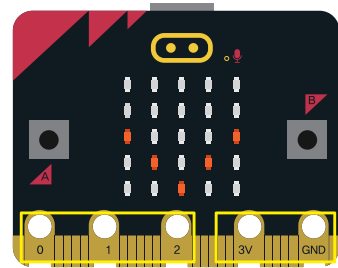
5 x 5 LED 매트릭스

마이크로비트 앞면에는 5x5 LED 매트릭스가 있습니다. 25개의 LED에 숫자나 텍스트, 간단한 이미지 등을 표현할 수 있습니다.



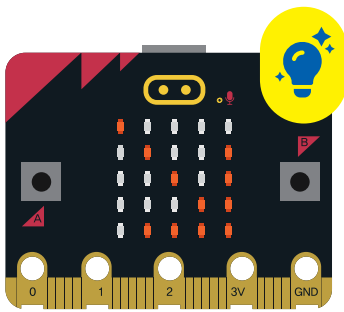
A, B 버튼

마이크로비트 앞면에는 A, B 두 개의 버튼이 있습니다. A, B 버튼으로 외부 입력을 할 수 있습니다.



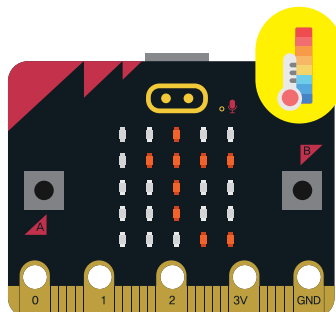
3개의 입출력 핀과 2개의 전원 핀

마이크로비트 앞면에는 3개의 입출력 핀과 2개의 전원 핀이 있습니다.



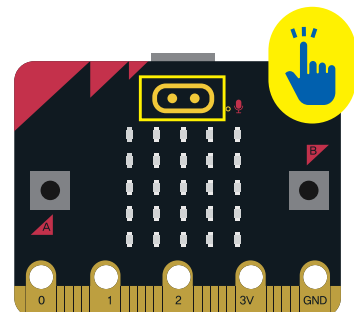
CDS 센서

마이크로비트에는 CDS (조도) 센서가 내장되어 있습니다. 조도 센서로 주변 빛의 양을 감지할 수 있습니다.



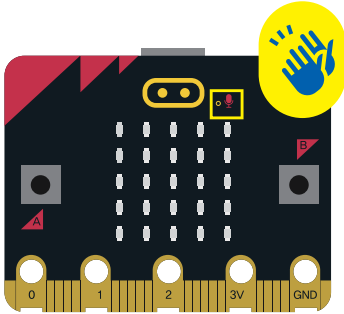
온도 센서

마이크로비트에는 디지털 온도 센서가 내장되어 있습니다. 온도 센서로 주변의 섭씨온도를 감지할 수 있습니다.



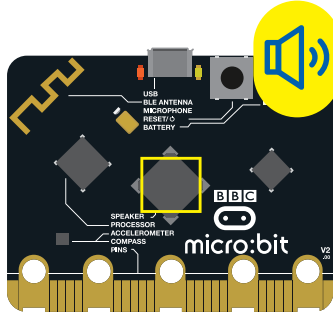
터치 센서

마이크로비트 V2에는 정전식 터치 센서가 내장되어 있습니다. 스마트폰, 태블릿PC처럼 손가락으로 터치해 버튼처럼 제어 할 수 있습니다.



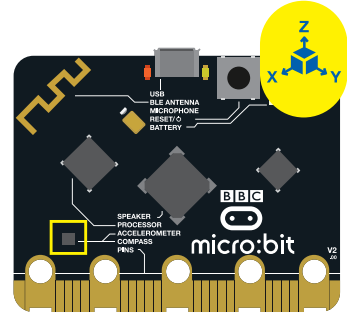
내장 마이크

마이크로비트 V2에는 마이크가 내장되어 있습니다. 박수를 치는 것과 같은 간단한 동작만으로 작동시킬 수 있습니다. 보드 앞면에는 마이크 소리 구멍과 LED가 있으며 마이크 본체는 뒷면에 부착되어 있습니다.



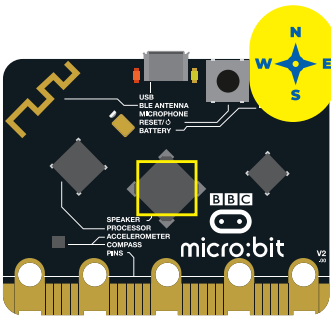
내장 스피커

마이크로비트 V2에는 스피커 모듈이 내장되어 있습니다. 원하는 소리를 출력할 수 있고 헤드셋 연결 및 음소거도 가능합니다.



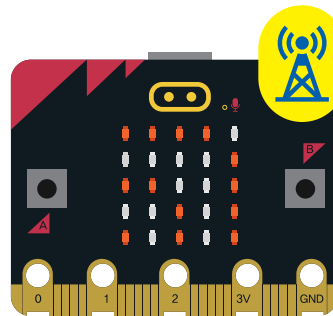
가속도계

마이크로비트에는 가속도계가 내장되어 있습니다. 마이크로비트의 흔들림이나 자유낙하, 기울기 정도 등의 동작을 감지할 수 있게 됩니다.



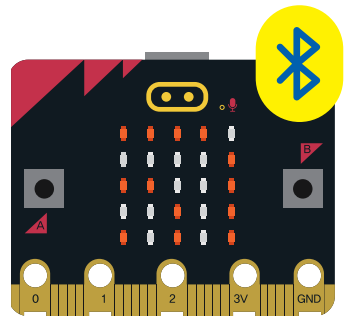
나침반

마이크로비트에 내장된 나침반은 지구 자기장을 감지해서 마이크로비트가 어느 방향을 향하고 있는지 알 수 있습니다. 나침반을 사용하기 전에는 보정작업이 필요합니다.



라디오 모듈

라디오 통신 기능을 이용하면 마이크로비트 간에 무선으로 통신을 할 수 있습니다. 마이크로비트끼리 통신을 하면 멀티 플레이어게임 등 다양한 프로젝트를 할 수 있습니다.



블루투스

블루투스 기능을 이용하면 마이크로비트가 PC나 스마트폰, 태블릿 등과 무선으로 통신이 가능해집니다. 블루투스를 이용해서 소스코드를 무선으로 보내거나 마이크로비트를 제어 할 수 있습니다.

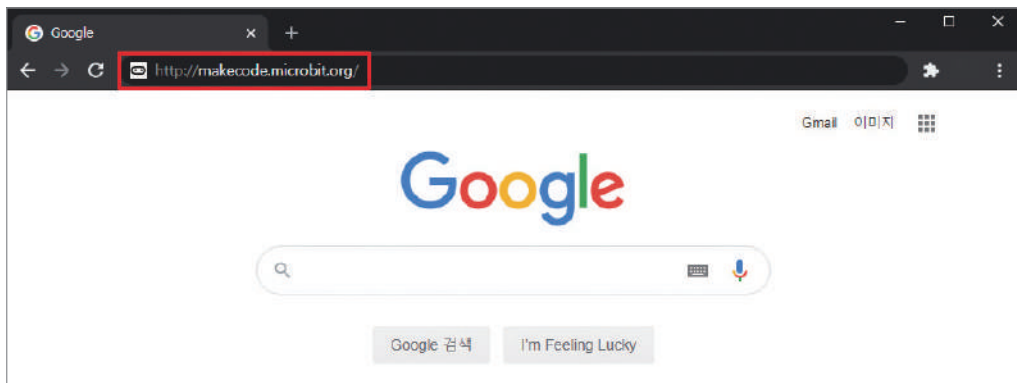


메이크 코드 (Make Code)

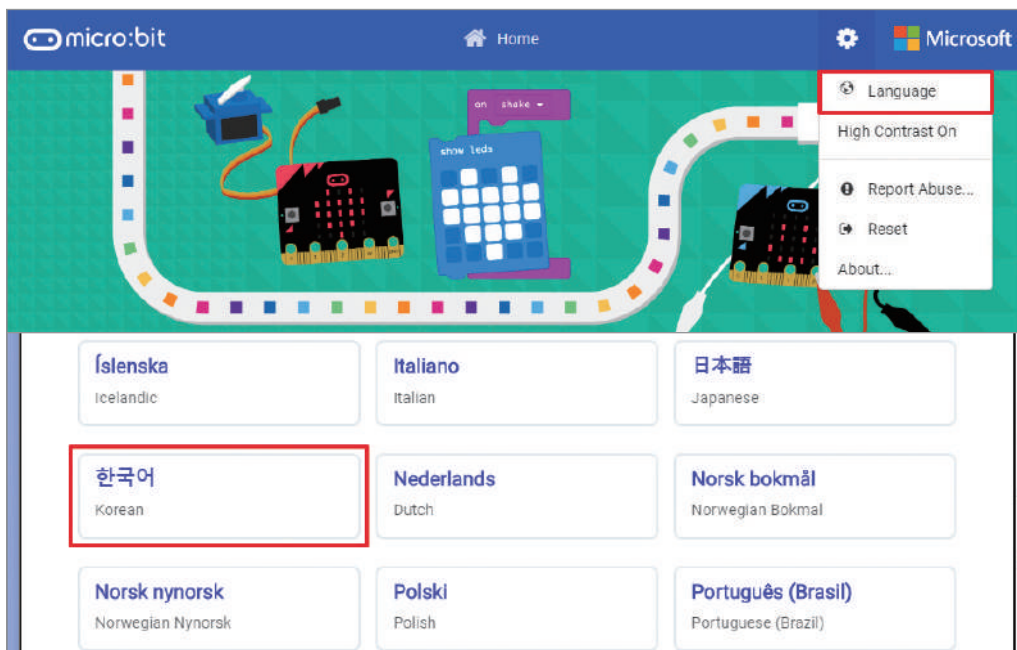
메이크 코드란?

마이크로비트는 메이크 코드를 이용해 쉽게 프로그래밍을 할 수 있습니다. 메이크 코드는 이미 만들어진 블록 언어를 모아놓은 블록 코딩 에디터로 비전공자, 초보자도 쉽고 간단하게 프로그래밍 할 수 있으며, 자바스크립트 언어를 지원하기 때문에 심화 학습도 가능합니다.

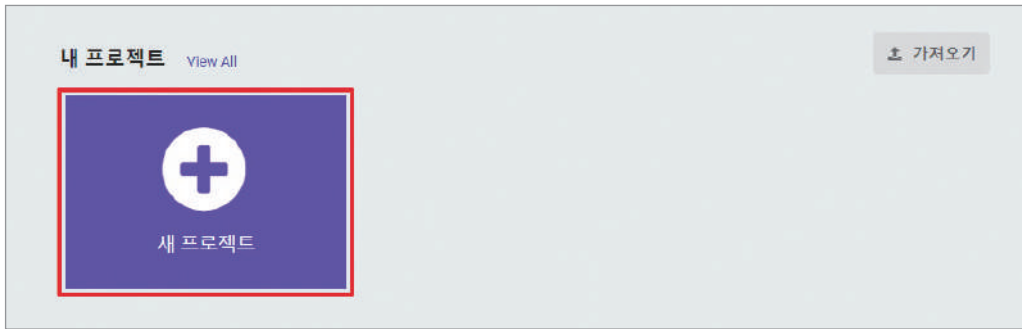
메이크 코드 사용하기



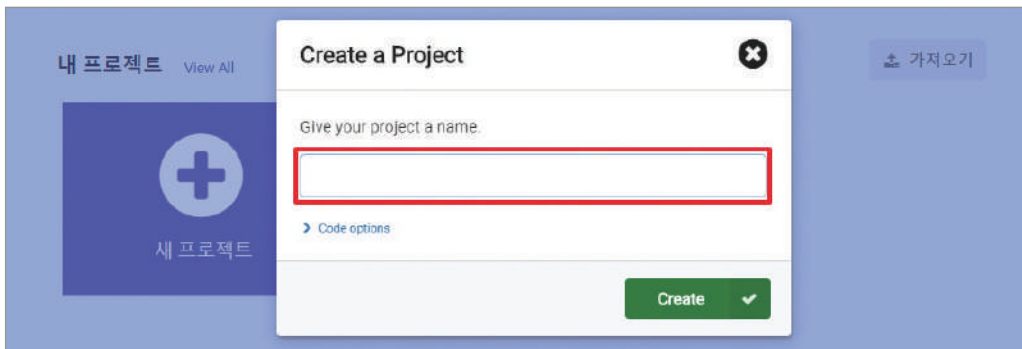
1. 마이크로비트 메이크 코드 사이트(<https://makecode.microbit.org/>)에 접속합니다.



2. 오른쪽 상단 ⚙ 을 누른 후 [Language] 설정에서 [한국어]를 선택해 사용 언어를 변경합니다.



3. [내 프로젝트] – [새 프로젝트]를 누릅니다.



4. 새롭게 생성할 프로젝트의 파일명을 정한 뒤, [Create]를 누릅니다.

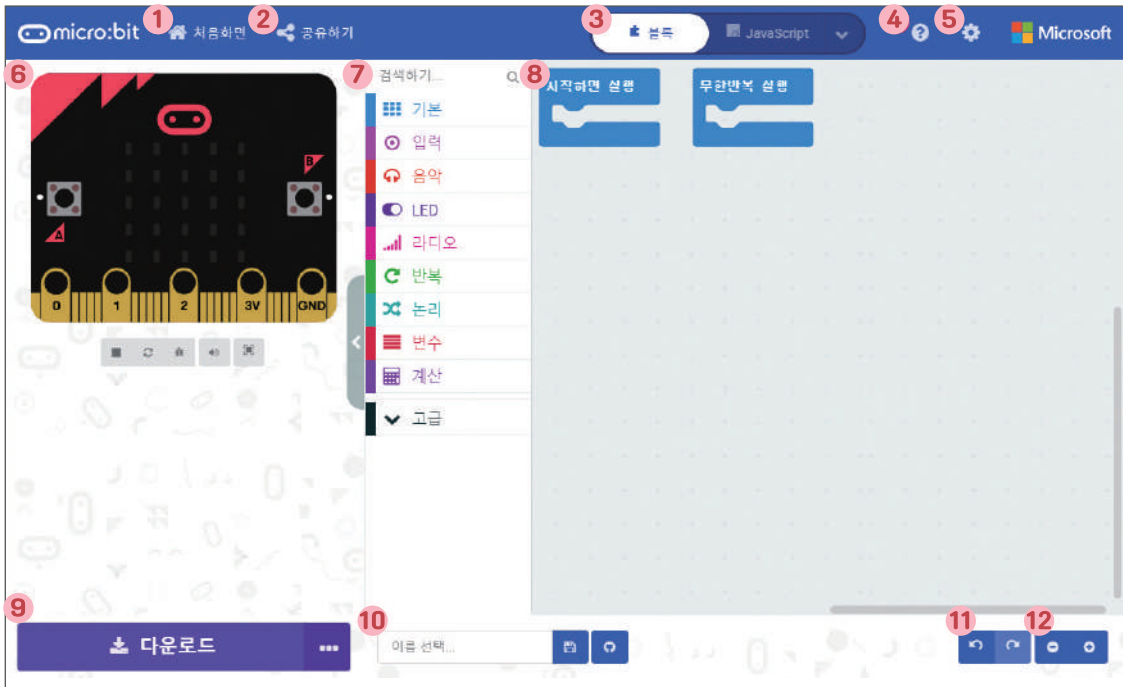


5. 새 프로젝트 창이 활성화되면 메이크 코드를 이용해 코딩을 할 수 있습니다.



메이크 코드 (Make Code)

메이크코드 자세히 보기



① 처음 화면

프로젝트 선택 화면으로 돌아갑니다.

② 공유하기

내가 만든 프로젝트의 URL과 QR 코드를 생성해 공유할 수 있습니다.

③ 코딩 언어 변환

사용 중인 코딩 언어 (블록→자바스크립트)를 변환합니다.

④ 도움말

메이크 코드를 이용하는데 필요한 도움말을 볼 수 있습니다.

⑤ 더 보기

프로젝트 설정, 언어 변경 등 사용 환경 설정을 할 수 있습니다.

⑥ 시뮬레이터

작성한 프로젝트를 시뮬레이션할 수 있습니다.

7 코딩 블록

기본

기본적인 기능의 블록들을 제공합니다

입력

버튼이나 센서 등의 정보를 가져올 수 있습니다.

음악

음악과 관련된 기능을 제어할 수 있습니다.

LED

내장된 LED 매트릭스를 제어할 수 있습니다.

라디오

라디오 무선 통신을 설정할 수 있습니다.

반복

반복 실행 코드를 활용할 수 있습니다.

논리

조건문, 비교, 참/거짓 연산을 할 수 있습니다.

변수

변수를 만들고 활용할 수 있습니다.

계산

사칙연산, 절댓값 등의 계산 코드를 활용할 수 있습니다.

고급

함수, 배열, 시리얼 통신 등의 고급 제어 기능이 있습니다.

8 코딩 영역

코딩 블록을 이용해 프로젝트를 작성할 수 있습니다.

9 다운로드

작성한 프로젝트를 다운로드합니다.

10 프로젝트 이름

프로젝트 이름을 설정하고 저장합니다.

11 이전 상태 / 복구

작성한 코드를 되돌리거나 다시 실행할 수 있습니다.

12 블록 크기 / 블록 작게

블록 크기를 조절합니다.

The background is a light yellow color with a repeating pattern of stylized yellow lemons with green leaves and orange slices. There are also small, faint icons scattered throughout, including a compass, a stopwatch, a candle, a globe, a hand, and a die.

#02
**기본 기능
익히기**

02 FLAVOR

두근두근 하트 만들기

기본 코딩 블록을 활용해 두근거리는 하트를 만들어봅시다.

학습 목표

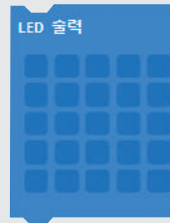
1. 메이크코드를 이용해 프로그램을 작성하고 마이크로비트에 업로드할 수 있다.
2. 기본적인 블록을 익혀 블록 코딩을 할 수 있다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.



작성한 프로그램을 무한 반복 실행합니다.



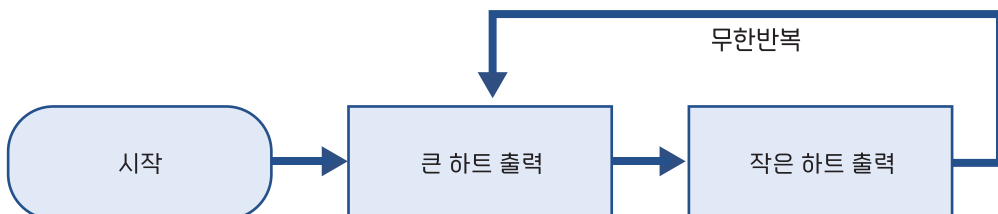
LED를 원하는 대로 출력할 수 있습니다.

동작 원리

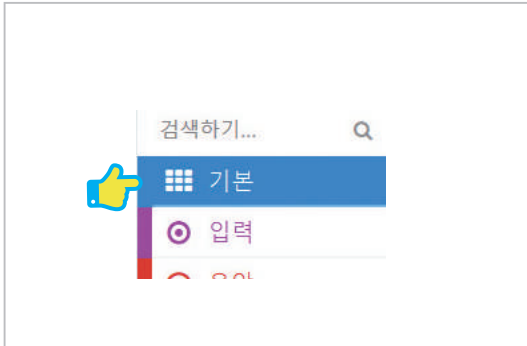
- 마이크로비트의 전원이 OFF 되기 전까지 계속 반복하기 위해 **무한 반복 실행 블록**을 사용합니다.
- 두근거리는 하트 모양을 나타내기 위해 **LED출력 블록**에 에 큰 하트와 작은 하트 그림을 그려줍니다.

생각해보기

- 프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.

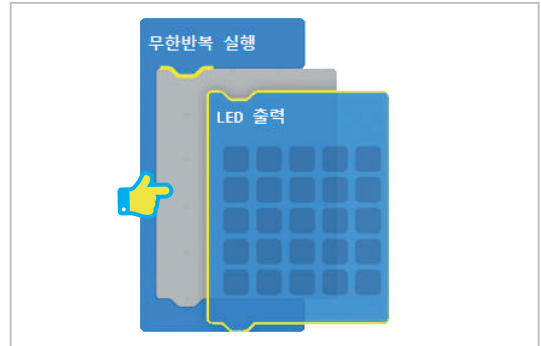


두근두근 하트 만들기



01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



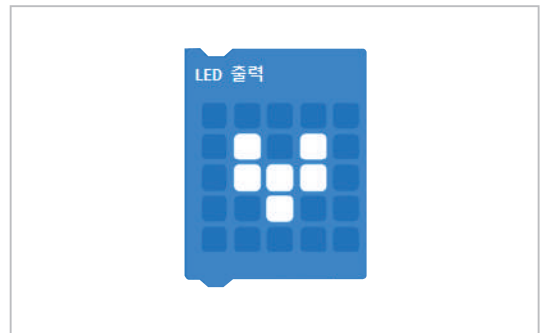
02_

LED 출력 블록을 드래그 앤 드롭하여 코딩 영역에 있는 **무한반복 실행** 블록 아래 놓습니다.



03_

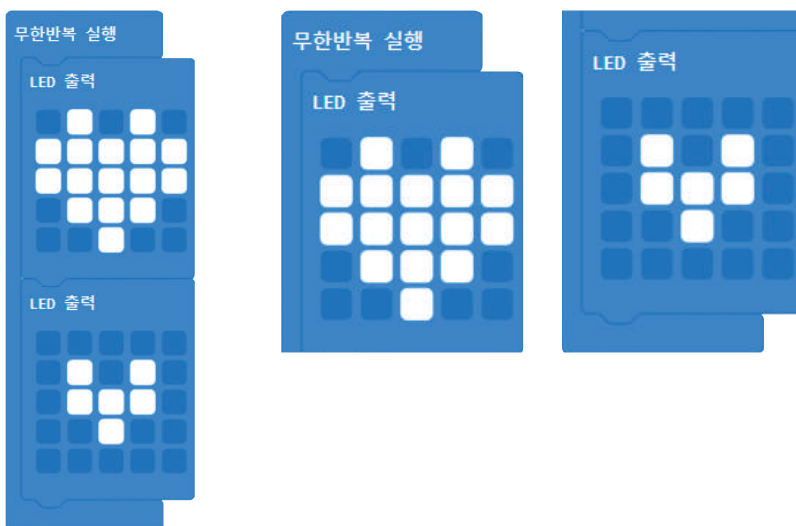
LED 출력 블록에 큰 하트 모양을 그려줍니다.



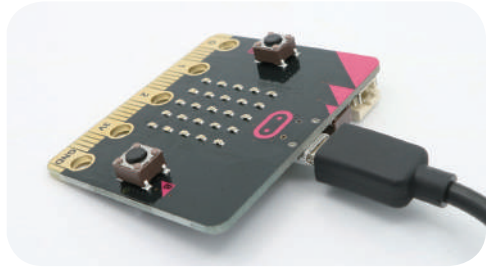
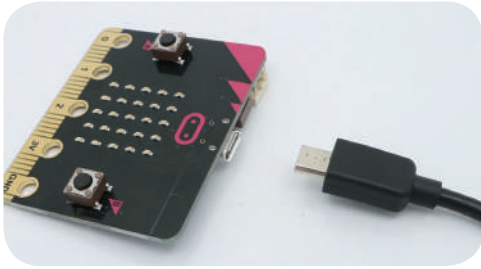
04_

큰 하트를 그려 넣은 **LED 출력** 블록 아래에 **LED 출력** 블록을 놓고 작은 하트를 그려줍니다.

완성 예제



마이크로비트에 프로젝트 다운로드하기



1. 마이크로비트에 마이크로 5핀 USB 케이블을 꽂아주고 반대쪽은 PC와 연결해줍니다.



2. 프로젝트 이름을 정하고 ①, 다운로드 버튼을 ② 눌러 마이크로비트에 프로젝트를 다운로드해줍니다.



3. 다운로드 중 화면으로 전환되고 **Download completed!** 문구와 함께 프로젝트 다운로드가 완료됩니다.



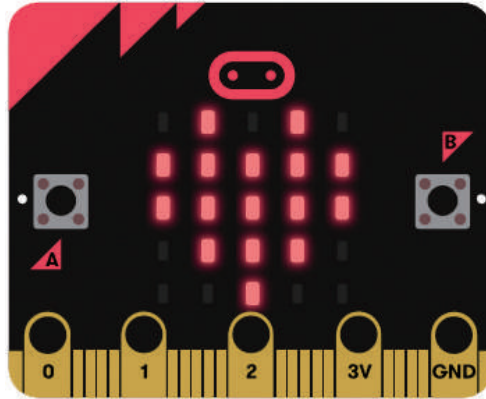
TIP

다운로드 중에는 마이크로비트 뒷면 노란 LED도 반짝거려요!



마무리하기

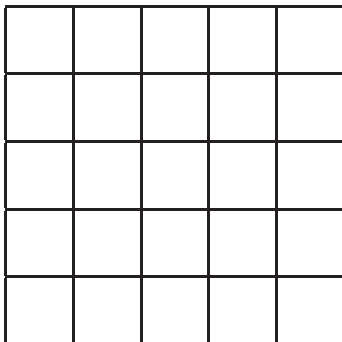
마이크로비트가 잘 동작하는지 확인해봅시다.



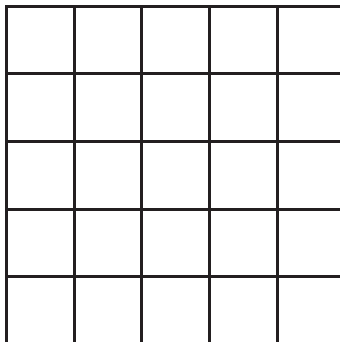
제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

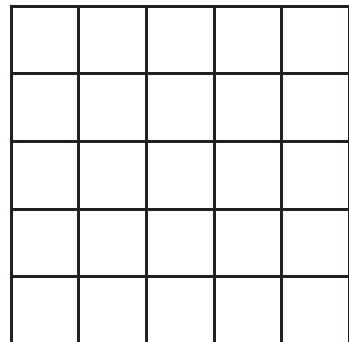
마이크로비트에 내장된 5×5 LED로 가위바위보 모양을 그려봅시다.



[가위]



[바위]



[보]

응용하기 정답은 72쪽 응용하기 해답편에서 확인해주세요!

마이크로비트에 hex 파일 다운로드하기

작성한 프로젝트를 hex 파일로 저장하고 저장한 파일을 이용해 마이크로비트에 프로젝트를 다운로드할 수 있어요! 저장한 파일은 친구들과 공유도 가능하고 보관도 할 수 있어 언제든지 프로젝트를 다운로드할 수 있답니다!



프로젝트 이름 옆  버튼을 눌러 프로젝트를 hex 파일로 저장해줍니다.



다운로드받은 프로젝트 파일(.hex)을 [내 컴퓨터] - [MICROBIT] 폴더에 넣어줍니다.

03 FLAVOR

가위바위보 게임 만들기

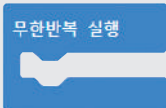
마이크로비트의 버튼과 LED를 이용해 가위바위보 게임을 만들어봅시다.

학습 목표

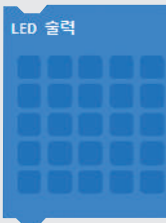
1. 상황에 맞는 블록을 활용할 수 있다.
2. 버튼을 누르면 실행 블록을 이용한 코딩을 할 수 있다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.



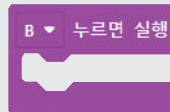
작성한 프로그램을 무한 반복 실행합니다.



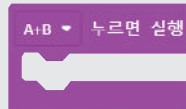
LED를 원하는 대로 출력할 수 있습니다.



A 버튼이 눌리면 작성한 프로그램을 실행합니다.



B 버튼이 눌리면 작성한 프로그램을 실행합니다.



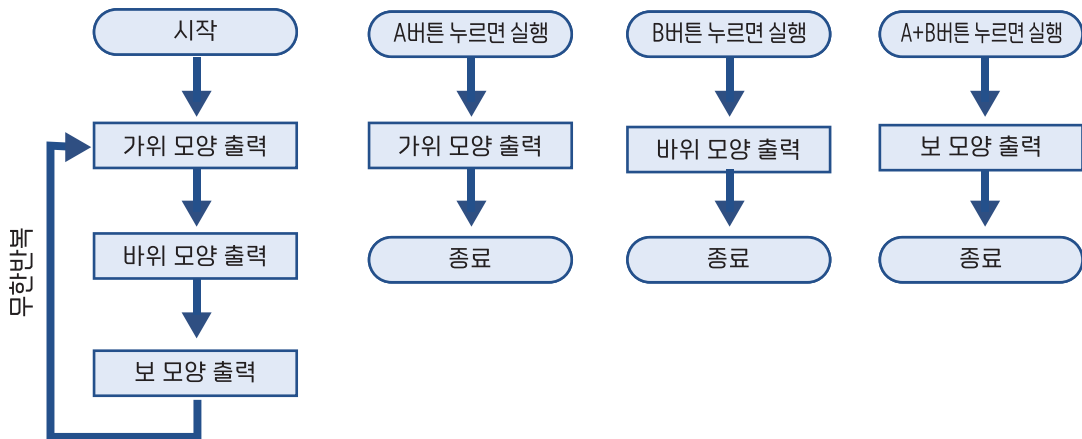
A 버튼과 B 버튼이 함께 눌리면 작성한 프로그램을 실행합니다.

동작 원리

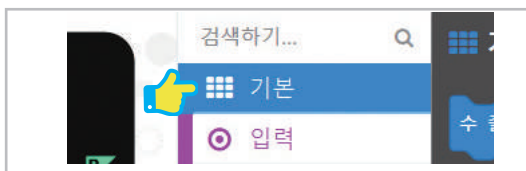
- 가위, 바위, 보 모양을 나타내기 위해 LED 출력 블록에 가위, 바위, 보 모양의 그림을 그려줍니다.
- 마이크로비트의 전원이 켜지면 가위, 바위, 보 모양의 LED가 순차적으로 무한 반복 점등합니다.
- A 버튼을 누르면 가위, B 버튼을 누르면 바위, A+B 버튼을 누르면 보를 출력합니다.

생각해보기

프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.

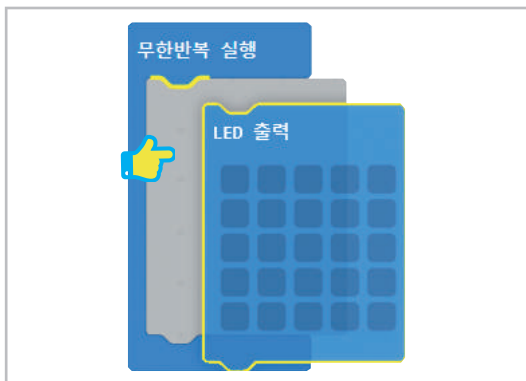


가위바위보 게임 만들기



01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



02_

LED 출력 블록을 드래그 앤 드롭하여 코딩 영역에 있는 무한반복 실행 블록 아래 놓습니다.



03_

LED 출력 블록에 가위 모양을 그려줍니다.



04_

같은 방법으로 바위와 보를 그린
LED 출력 블록을 넣어줍니다.

05_

코딩 블록 메뉴에서 **입력** 탭을 선택합니다.

06_

A 누르면 실행 블록을 드래그 앤 드롭하여
코딩 영역에 놓아줍니다.

07_

A 누르면 실행 블록 아래에 LED 출력 블록을
놓고 가위 모양을 그려줍니다.



08_

A 누르면 실행 블록을 하나 더 놓아준 뒤,
A 옆의 화살표를 눌러 B 버튼으로 변경해줍니다.

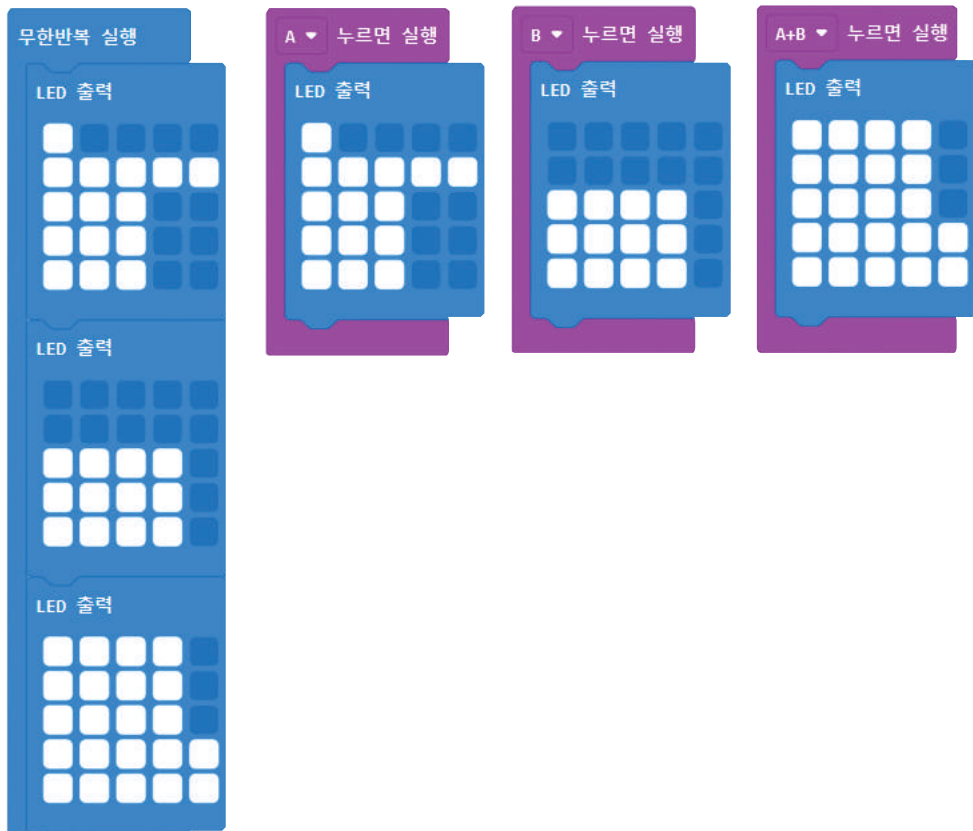
09_

같은 방법으로 A+B 누르면 실행 블록도 추가해줍니다.

10_

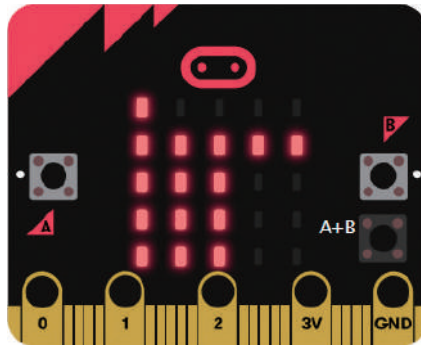
B 누르면 실행 블록에 LED 출력 블록을 놓고
바위 모양을 그려주고, A+B 누르면 실행 블록에
LED 출력 블록을 놓고 보 모양을 그려줍니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

버튼에 따라 출력이 바뀌는 나만의 명함을 만들어봅시다.

A 버튼을 누르면 영어 이름, B 버튼을 누르면 나이, A+B 버튼을 누르면 휴대폰 번호가 나오게 하려면 어떻게 해야 할까요?

응용하기 정답은 72쪽 응용하기 해답편에서 확인해주세요!



TIP!

문자열 출력 블록을 이용해보세요!

04 FLAVOR

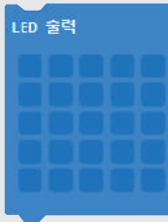
주사위 만들기

마이크로비트의 가속도 센서와 LED를 이용해 주사위를 만들어봅시다.

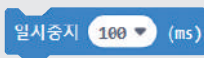
학습 목표

1. 상황에 맞는 다양한 블록을 사용할 수 있다.
2. 조건문을 이해하고 프로그램 작성에 활용할 수 있다.

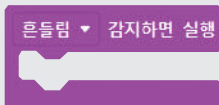
살펴보기



LED를 원하는 대로 출력할 수 있습니다.



원하는 시간만큼 프로그램 실행을 일시 중지 시킵니다.



흔들림이 감지되면 작성한 프로그램을 실행합니다.



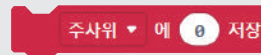
지정한 범위 내의 정수를 랜덤으로 출력합니다.



else if 조건문으로 원하는 조건을 설정하고 설정한 조건을 만족했을 때 / 만족하지 않았을 때 작성한 프로그램을 실행합니다.



'주사위' 이름을 가진 변수로 원하는 값을 담을 수 있습니다



변수에 원하는 값을 저장합니다.

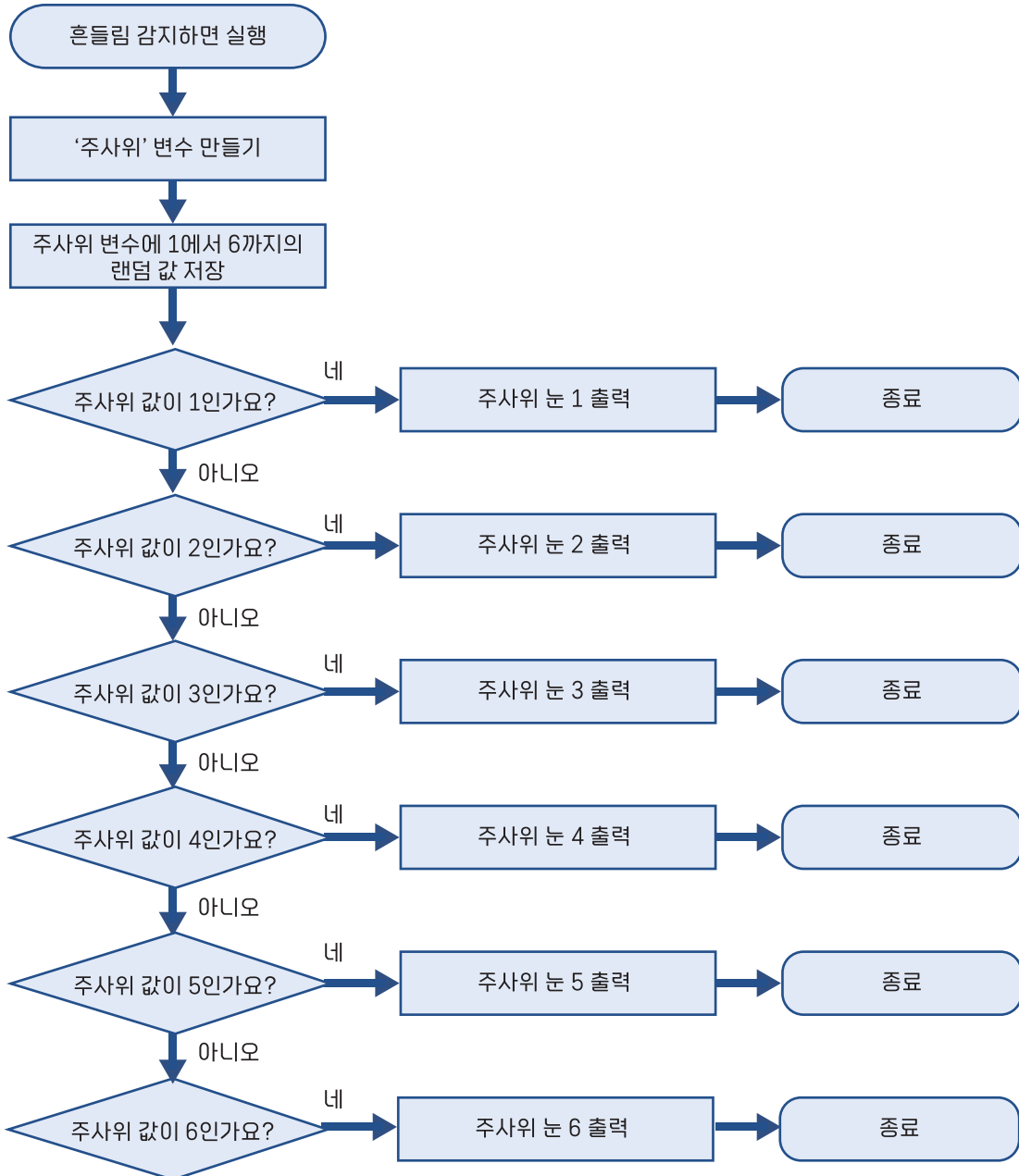
동작 원리

- 주사위 값을 담을 수 있는 '주사위' 변수를 생성합니다.
- 1부터 6까지의 랜덤 값을 주사위 변수에 담아줍니다.
- LED 출력 블록에 1부터 6까지의 주사위 눈을 그려줍니다.
- 마이크로비트가 흔들림을 감지했을 때, 주사위 눈 LED를 랜덤으로 출력합니다.

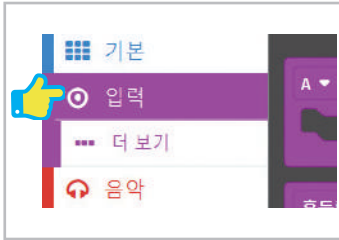
이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

생각해보기

프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.



주사위 만들기



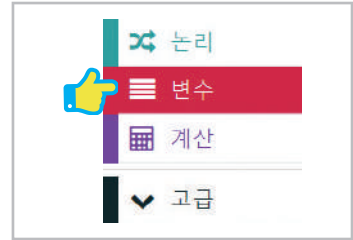
01_

코딩 블록 메뉴에서 **입력** 탭을 선택합니다.



02_

흔들림 감지하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



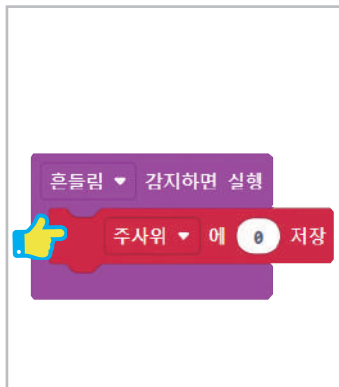
03_

코딩 블록 메뉴에서 **변수** 탭을 선택합니다.



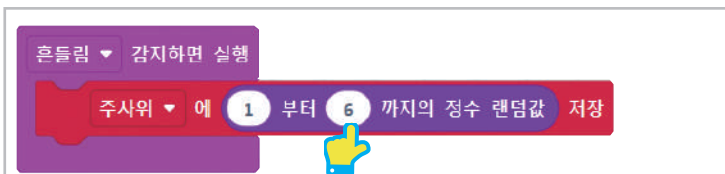
04_

변수 만들기 버튼을 누르고 '주사위' 변수를 만들어줍니다.



05_

흔들림 감지하면 실행 블록 아래에 **변수 탭 > 주사위에 값 저장** 블록을 놓아줍니다.



06_

주사위에 값 저장 블록에 **계산** 탭의 정수의 랜덤 값 블록을 놓고 범위를 1부터 6까지로 지정해줍니다.



07_

논리 탭의 **else if** 조건문을 놓아줍니다.



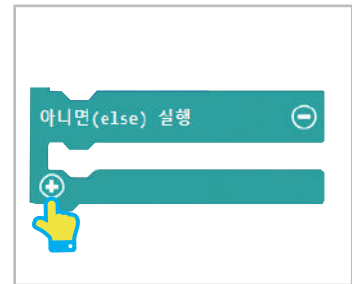
08_

else if 조건문의 조건에 **논리**
탭 > **비교 연산** 블록을 놓아줍니다.



09_

비교연산 블록에 **변수** 탭에서
생성한 '주사위' 변수를 넣고 '주사
위 = 1' 로 조건을 설정해줍니다.



10_

'주사위 = 1 이면 실행' 조건문 아래에
코딩 블록 메뉴 > **기본** > LED
출력 블록을 놓고 주사위 1을 그려
줍니다.

11_

코딩블록메뉴 > **기본** > 일시
중지 블록을 놓고 LED 출력 상태
를 유지할 시간을 설정해줍니다.

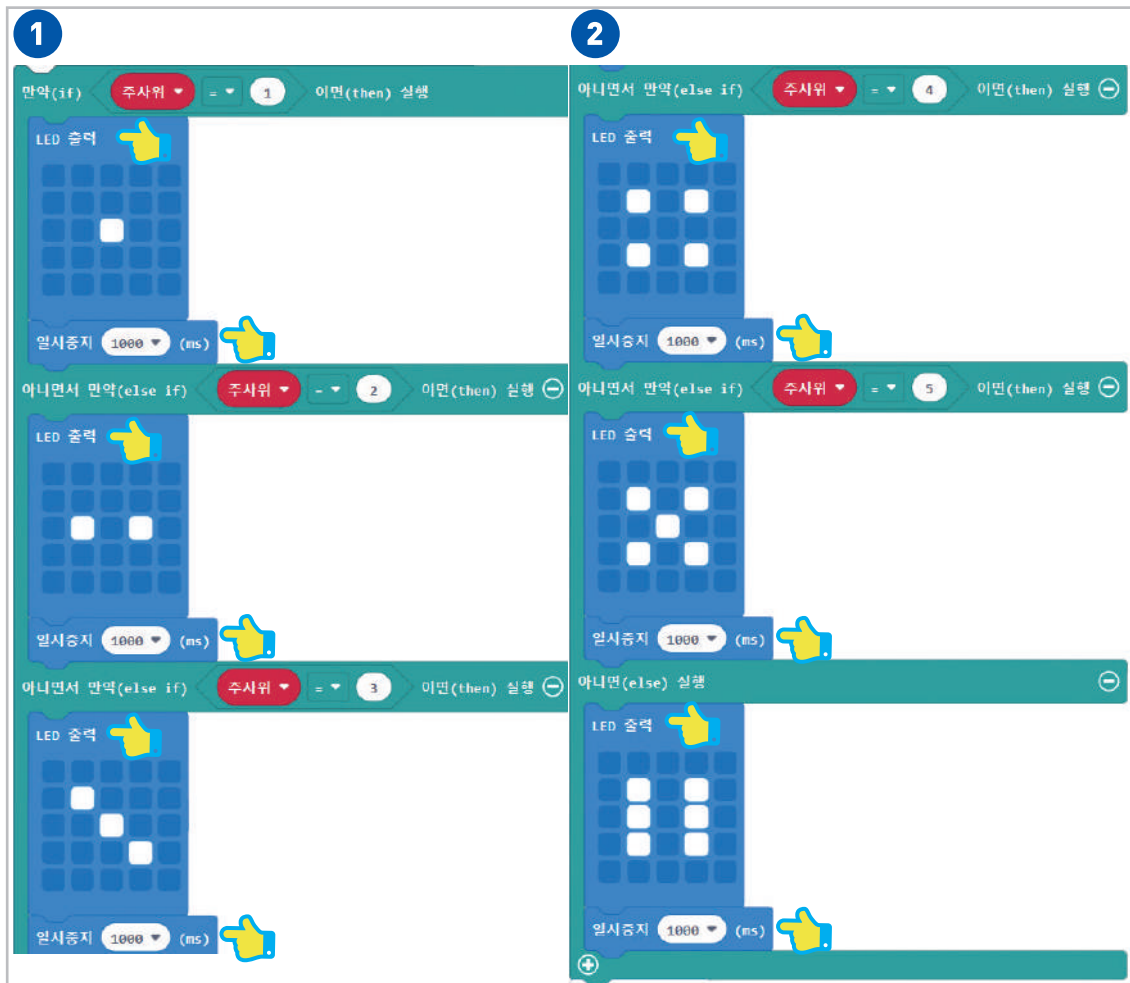
12_

if 조건문 아래에 **+** 버튼을 눌러
else if 조건을 주사위 눈의 수 -1개
만큼 추가해줍니다.



13_

추가한 else if 조건문에 주사위
변수 값이 2일 때부터 5일 때까지의
조건을 설정해줍니다.



14_

위와 마찬가지로 조건마다 LED출력 블록과 일시정지 블록을 놓고 LED를 그려줍니다.

완성 예제

1

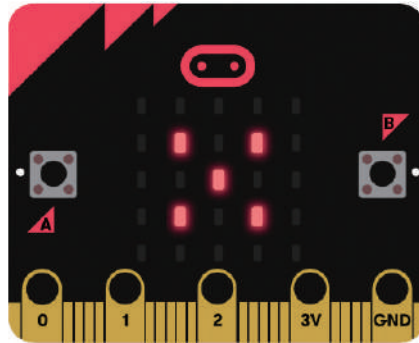


2



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

마이크로비트 **흔들림 감지 블록**을 활용해 만보기를 만들어봅시다.

마이크로비트가 **흔들림을 감지할 때마다 값이 1씩 증가**하는 만보기를 만든 후,

A 버튼을 눌렀을 때 만보기 값을 나타내고 **B 버튼을 눌렀을 때** 만보기 값을 초기화하려면 어떻게 해야 할까요?

응용하기 정답은 72쪽 응용하기 해답편에서 확인해주세요!

The background is a solid teal color. It is decorated with several stylized peach illustrations. Some peaches are whole, with a pink outer skin and an orange inner section. Others are sliced open, revealing a dark brown pit. Green leaves are attached to the stems of the peaches. Scattered throughout the background are faint, light-colored icons, including a clock, a hand holding a heart, a globe, a fork and knife, and a hand making a peace sign.

#03

V2 기능익히기

05 FLAVOR

미러볼 만들기

마이크로비트 V2에 추가된 기능을 활용해 소리에 따라 밝기가 변하는 미러볼을 만들어봅시다.

학습 목표

1. 마이크로비트 V2의 마이크 기능을 이해한다.
2. 여러 개의 블록을 상황에 맞게 연계해서 사용할 수 있다.

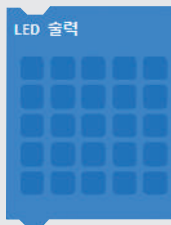
살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행



마이크로비트를 시작하면 작성한 프로그램을 실행합니다.



LED를 원하는 대로 출력할 수 있습니다.

무한반복 실행



작성한 프로그램을 무한 반복 실행합니다.

소리 크기

소리 크기 값을 변수로 사용할 수 있습니다.

LED 스크린 밝기를 0으로 설정

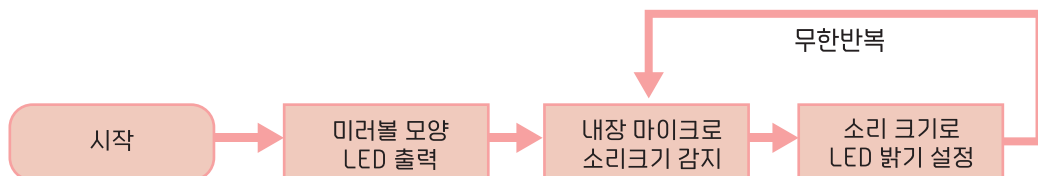
LED 스크린의 밝기를 조절할 수 있습니다.

동작 원리

1. 마이크로비트가 시작되면 **미러볼 모양의 LED**를 출력합니다.
2. 마이크로비트의 마이크 기능을 활용해 **소리 크기를 감지**하고 소리 크기로 **LED 스크린 밝기를 조절**합니다.

생각해보기

1. 프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.

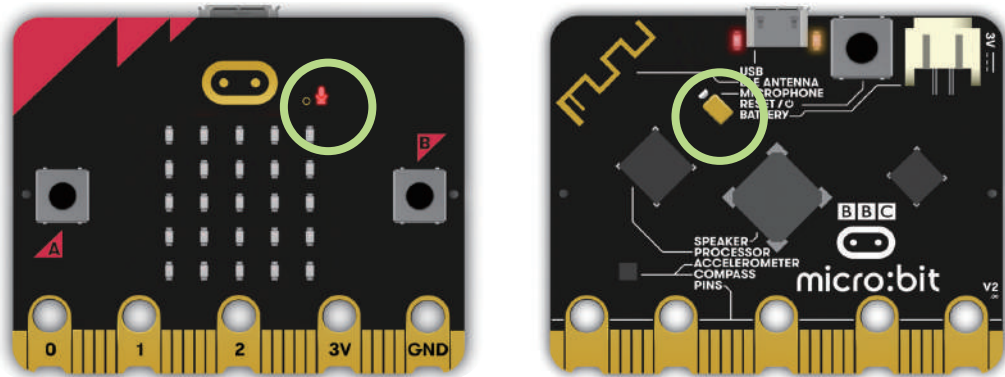


자세히 보기

마이크로비트 V2에 추가된 마이크 기능에 대해 알아보시다.

하드웨어

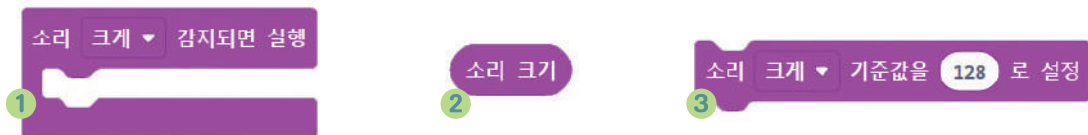
마이크로비트 V2 보드에는 보드 앞면에 마이크 상태 표시 LED가 부착되어 있고 보드 뒷면에 마이크 모듈이 부착되어 있습니다.



마이크 모듈을 사용하는 프로그램을 작성하면 마이크 상태 표시 LED가 켜지게 됩니다.

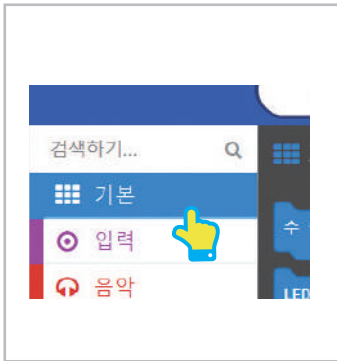
소프트웨어

마이크 모듈 관련 코딩 블록 3가지가 추가되었습니다.



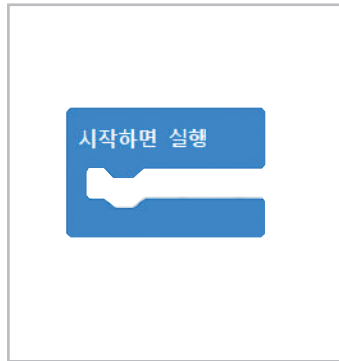
- 1 마이크 모듈이 소리를 감지하면 작성한 프로그램을 실행합니다.
- 2 마이크 모듈이 감지하는 소리 크기 값을 담는 변수입니다.
- 3 마이크 모듈이 감지하는 소리 크기의 기준값을 설정할 수 있습니다.

미러볼 만들기



01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



02_

시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



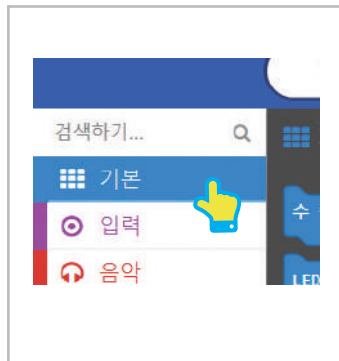
03_

시작하면 실행 블록 아래에 코딩 블록 메뉴 > **기본** 탭에서 **LED 출력** 블록을 놓습니다.



04_

LED 출력 블록에 미러볼 모양의 LED를 그려줍니다.



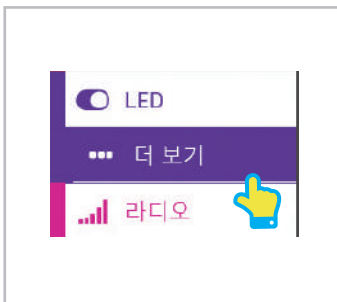
05_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



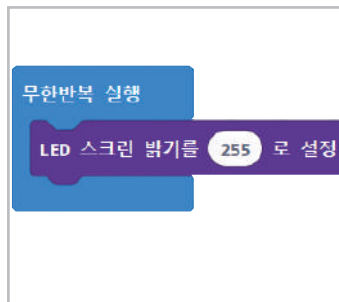
06_

무한반복 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



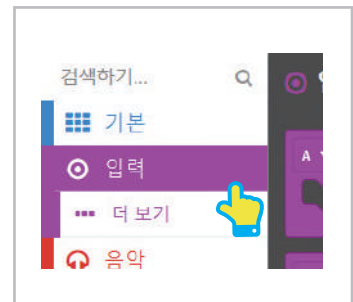
07_

코딩 블록 메뉴에서 **LED** 탭을 선택한 뒤 바로 아래 **더 보기** 탭을 선택합니다.



08_

무한반복 실행 블록 아래에 **스크린 밝기 설정** 블록을 놓습니다.



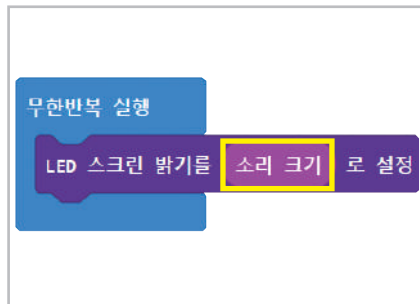
09_

코딩 블록 메뉴에서 **입력** 탭을 선택합니다.



10_

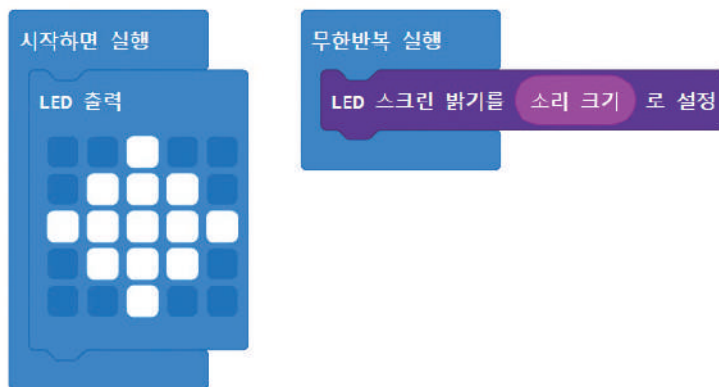
입력 탭 블록 메뉴 하단
micro:bit (V2) 전용 블록에서
소리 크기 블록을 선택합니다.



11_

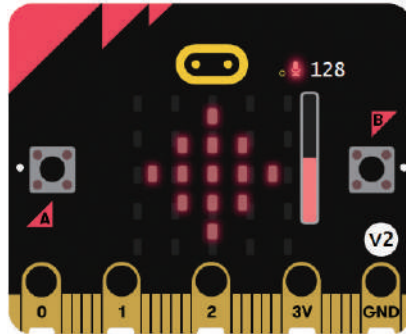
소리 크기 블록을 드래그 앤 드롭하여
LED 스크린 밝기 값으로 지정해줍니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

소리 감지 블록을 활용해 **소음 감지기**를 만들어봅시다.

마이크로비트가 시작되면 **소리 조용하게 기준값을 30**으로 **소리 크게 기준값을 80**으로 설정한 뒤,
소리 조용하게 감지되면 **행복한 표정**을, 소리 크게 감지되면 **슬픈 표정**을 그리려면 어떻게 해야 할까요?

소리 크기 감지되면 실행 블록, 소리 기준값 설정 블록을 활용해보세요!

응용하기 정답은 73쪽 응용하기 해답편에서 확인해주세요!

06 FLAVOR

메트로놈 만들기

마이크로비트 V2에 추가된 기능을 활용해 일정한 박자를 세어주는 메트로놈을 만들어봅시다.

학습 목표

1. 마이크로비트 V2의 스피커 기능을 이해한다.
2. 메이크 코드의 음악 블록을 활용하여 원하는 소리를 출력할 수 있다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행

마이크로비트를 시작하면 작성한 프로그램을 실행합니다.

무한반복 실행

작성한 프로그램을 무한 반복 실행합니다.

아이콘 출력

원하는 LED 모양을 출력할 수 있습니다.

음량을 127로 설정

스피커의 음량을 설정할 수 있습니다.

빠르기(분당 박자 개수)를 120으로 설정

빠르기(bpm)를 원하는 속도로 설정할 수 있습니다.

도 1 박자 출력

원하는 음을 원하는 시간만큼 출력할 수 있습니다.

1 박자 (ms) 유지

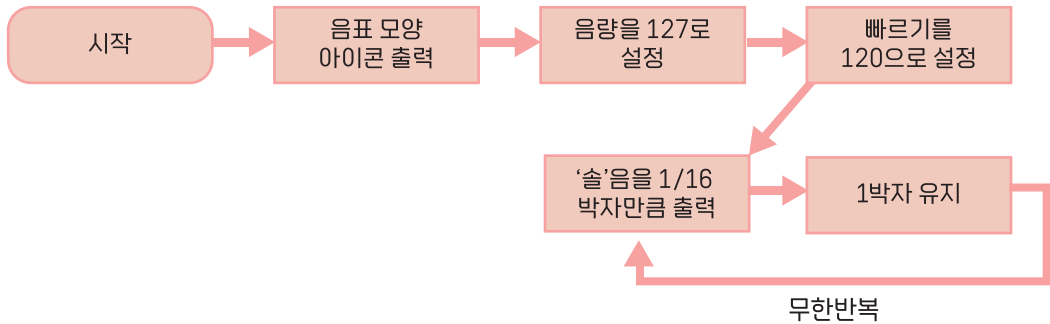
소리를 원하는 시간만큼 유지할 수 있습니다.

동작 원리

- 마이크로비트가 시작되면 **음표 모양 아이콘을 출력**합니다.
- **스피커 음량은 127, 빠르기는 120**으로 설정합니다.
- '솔' 음을 **1/16 박자만큼 출력**하고 **1박자 유지**하는 프로그램을 **무한 반복 실행**합니다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.



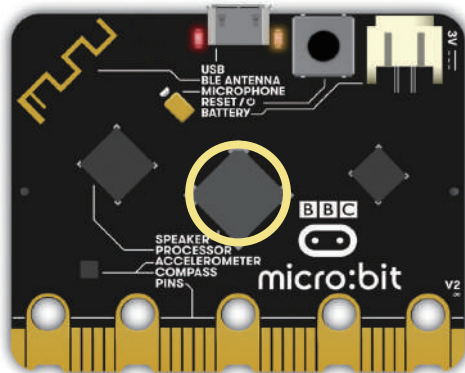
자세히 보기

마이크로비트 V2에 추가된 스피커 기능에 대해 알아봅시다.

하드웨어

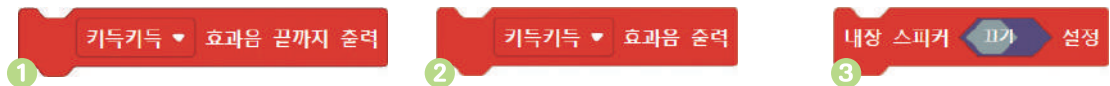
마이크로비트 V2 보드 뒷면 중앙에는 스피커 모듈이 장착되어 있습니다.

이 스피커 모듈은 마이크로비트의 내장 스피커로 추가 모듈 설치 없이 다양한 소리를 출력할 수 있습니다.



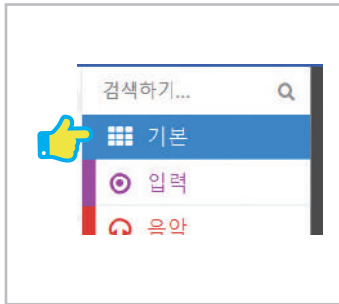
소프트웨어

내장 스피커 관련 코딩 블록 3가지가 추가되었습니다.

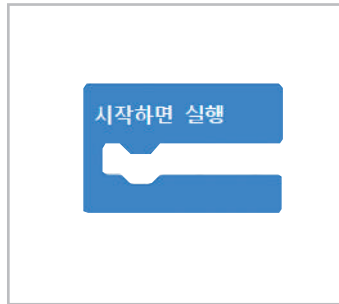


- 1 내장 스피커로 효과음을 끝까지 출력합니다. 효과음 출력 중 다른 코딩 블록이 개입할 수 없습니다.
- 2 내장 스피커로 효과음을 출력합니다. 다른 코딩 블록과 동시에 출력이 가능합니다.
- 3 내장 스피커를 ON/OFF 설정할 수 있습니다.

메트로놈 만들기



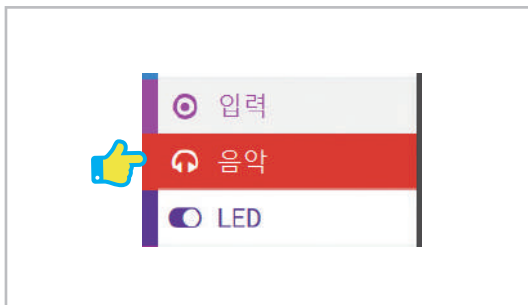
01_
코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



02_
시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



03_
시작하면 실행 블록 아래에 코딩 블록 메뉴 > **기본** 탭에서 **아이콘 출력** 블록을 놓고 음표 아이콘을 선택해줍니다.



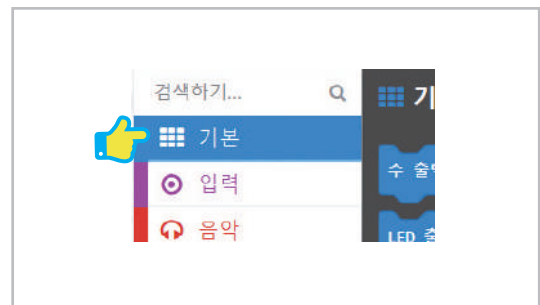
04_
코딩 블록 메뉴에서 **음악** 탭을 선택합니다.



05_
음량 설정 블록을 드래그 앤 드롭해 **아이콘 출력** 블록 아래에 놓고 원하는 소리 크기로 설정해줍니다.



06_
코딩 블록 메뉴 > **음악** 탭에서 **빠르기 설정** 블록을 드래그 앤 드롭해 **음량 설정** 블록 아래에 놓고 원하는 **빠르기(기본값)**로 설정해줍니다.



07_
코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



08_

무한반복 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



09_

무한반복 실행 블록 아래에 코딩 블록 메뉴 > | 음악 탭에서 소리 출력 블록을 놓고 원하는 음과 시간을 지정해줍니다.



10_

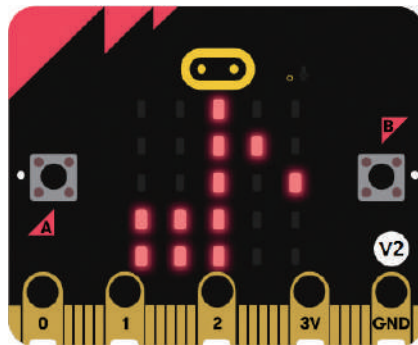
무한반복 실행 블록 아래에 코딩 블록 메뉴 > | 음악 탭에서 지속 시간 블록을 놓고 원하는 지속 시간을 설정해줍니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

속도 조절이 가능한 메트로놈을 만들어봅시다.

A 버튼을 누르면 느리게, B 버튼을 누르면 빠르게 조절하고 A+B 버튼을 누르면 현재 빠르기를 볼 수 있는 메트로놈을 만들려면 어떻게 해야할까요?

응용하기 정답은 73쪽 응용하기 해답편에서 확인해주세요!

07 FLAVOR

스톱워치 만들기

마이크로비트 V2에 추가된 터치 센서 기능을 활용해 스톱워치를 만들어봅시다.

학습 목표

1. 마이크로비트 V2의 터치 기능을 이해한다.
2. While 반복문을 활용해 변수 값을 제어할 수 있다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행



마이크로비트를 시작하면 작성한 프로그램을 실행합니다.

수 출력

입력한 수를 출력합니다.

LED 출력



LED를 원하는 대로 출력할 수 있습니다.

일시 중지 1000 (ms)

원하는 시간만큼 프로그램 실행을 일시 중지 시킵니다.

A 누르면 실행



A / B / A+B 버튼을 누르면 작성한 프로그램을 실행합니다.

흔들림 감지하면 실행



흔들림이 감지되면 작성한 프로그램을 실행합니다.

로고 누름 시 실행



로고 버튼을 누르면 작성한 프로그램을 실행합니다.

반복(while): 참(true) 인 동안



설정된 조건만큼 작성한 프로그램을 반복 실행합니다.



비교 연산 조건을 설정합니다.

타이머 로고상태

원하는 값을 담을 수 있는 변수입니다.

타이머 에 저장

변수에 원하는 값을 저장합니다.

타이머 값 1 증가

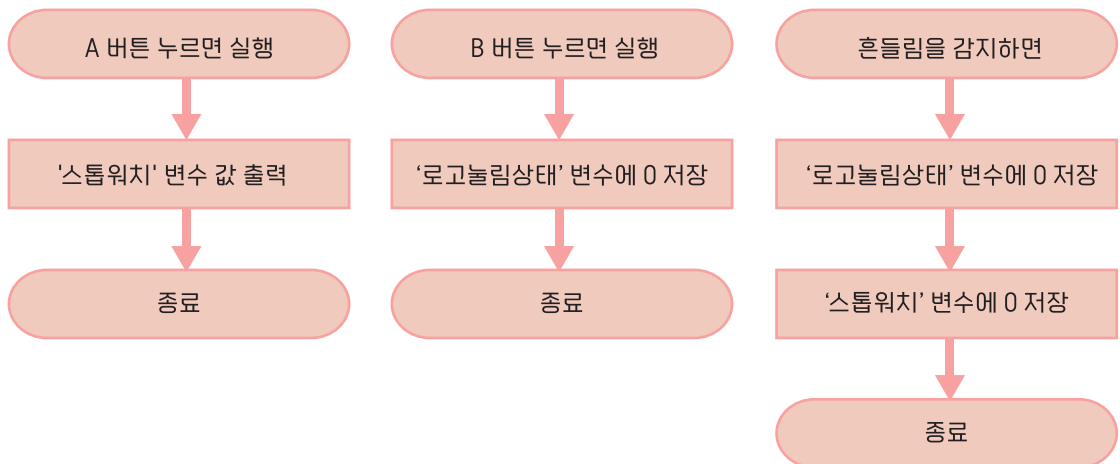
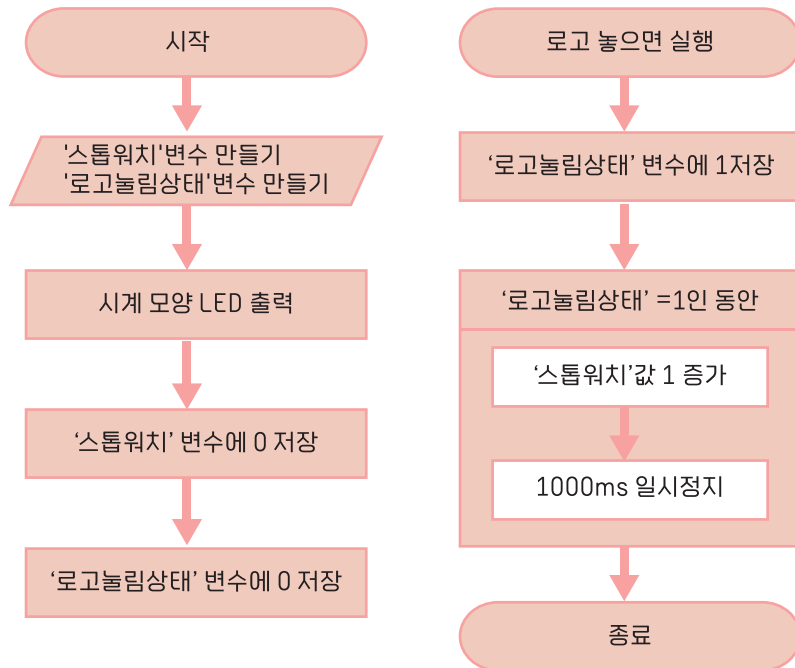
변수 값을 증가시킵니다.

동작 원리

- 마이크로비트가 시작되면 시계 모양 LED가 출력됩니다.
- 마이크로비트의 로고를 눌렀다가 놓았을 때 '스톱워치' 가 1초마다 1씩 증가합니다.
- A 버튼을 누르면 스톱워치값을 출력하고 B 버튼을 누르면 일시 정지합니다.
- 흔들림을 감지하면 스톱워치를 초기화합니다.

살펴보기

프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.

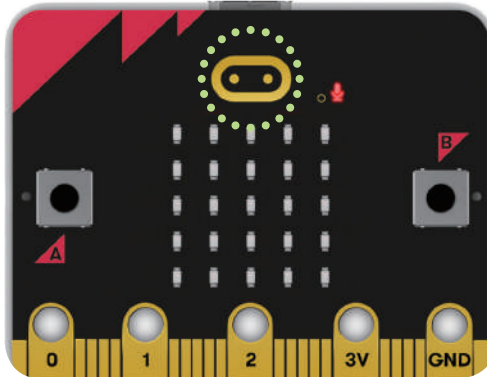


자세히 보기

마이크로비트 V2에 추가된 터치 센서에 대해 알아보시다.

하드웨어

마이크로비트 V2 보드 앞면 로고에는 터치 기능이 내장되어 있습니다.
이 터치 센서로 다양한 이벤트를 발생시키고 프로그램을 제어할 수 있습니다.



소프트웨어

터치 센서 관련 코딩 블록 3가지가 추가되었습니다.



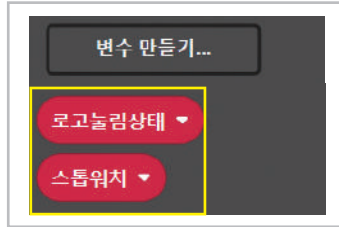
- 1 로고를 누르거나 터치하거나 놓았을 때 프로그램을 실행합니다.
- 2 로고 눌림 상태를 확인합니다.
- 3 터치 모드를 정전식/감압식으로 설정합니다.

스톱워치 만들기



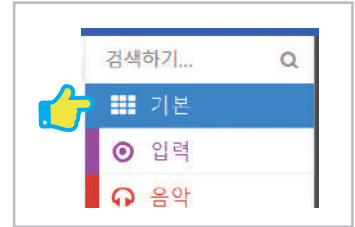
01_

코딩 블록 메뉴 > ≡ 변수 탭에서 변수 만들기를 선택합니다.



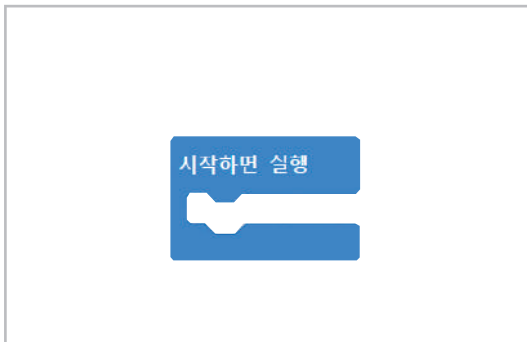
02_

'스톱워치' 변수와 '로고 놀림 상태' 변수를 생성합니다.



03_

코딩 블록 메뉴에서 ≡ 기본 탭을 선택합니다.



04_

시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



05_

시작하면 실행 블록 아래에 코딩 블록 메뉴 > ≡ 기본 탭에서 LED 출력 블록을 놓고 시계 모양을 그려줍니다.



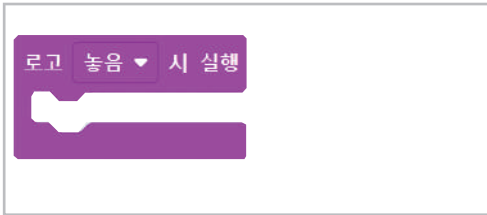
06_

코딩 블록 메뉴 > ≡ 변수 탭에서 생성한 타이머 변수와 로고상태 변수를 놓고 변수에 '0'을 저장합니다.
(변수 값 초기화 역할)



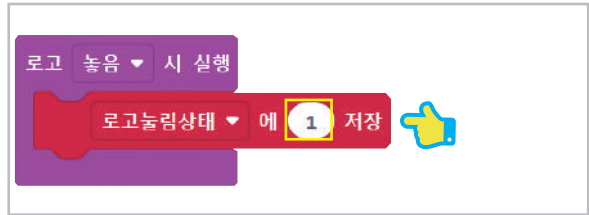
07_

코딩 블록 메뉴에서 ≡ 입력 탭을 선택합니다.



08_

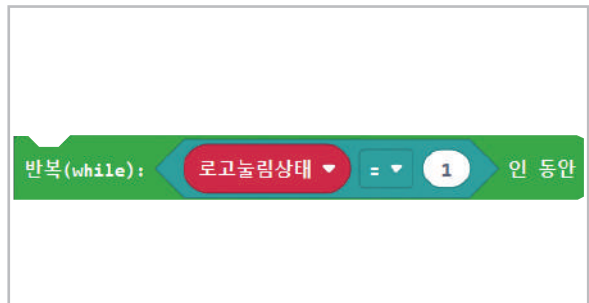
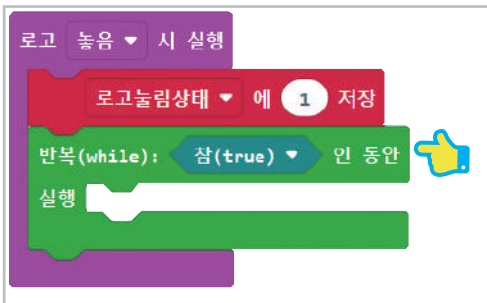
로그 누름 시 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓고 '로그 놓음 시 실행'으로 변경해줍니다.



09_

코딩 블록 메뉴 > 변수 탭에서 변수에 값 저장 블록을 놓고 '로그 누림 상태' 변수에 '1'을 저장해줍니다.

(로그 누림 = 1, 눌리지 않음 = 0)



10_

코딩 블록 메뉴 > 반복 탭에서 조건 반복문을 가져옵니다.

11_

반복문 조건을 '로그 누림 상태가 1일 때'로 설정합니다.



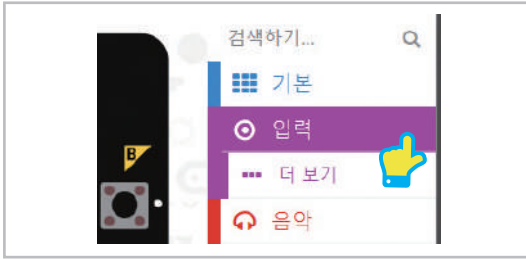
12_

코딩 블록 메뉴 > 기본 탭에서 일시중지 블록을 드래그 앤 드롭하여 반복문 안에 놓고 1000ms (1초)로 설정합니다.



13_

코딩 블록 메뉴 > 변수 탭에서 '변수 값 증가' 블록을 가져온 뒤, '스톱워치' 변수 값을 1 증가 시켜줍니다.



14_

코딩 블록 메뉴에서 **입력** 탭을 선택합니다.



16_

코딩 블록 메뉴 > **기본** 탭에서 수 출력 블록을 드래그 앤 드롭하여 A 누르면 실행 블록 아래에 놓고 출력 값을 '스톱워치' 변수 값으로 설정해줍니다.



18_

코딩 블록 메뉴 > **변수** 탭에서 '변수 값 저장' 블록을 가져온 뒤, 로그인 상태 변수에 0을 저장합니다.



20_

코딩 블록 메뉴 > **변수** 탭에서 '변수 값 저장' 블록을 가져온 뒤, '로그인 상태' 변수에 0을 저장하고 '스톱워치' 변수에도 0을 저장합니다.



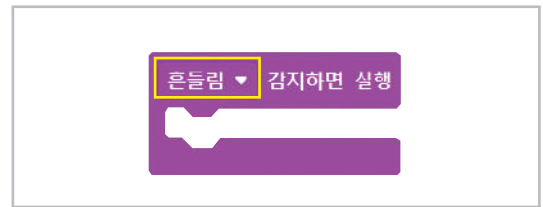
15_

A 누르면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



17_

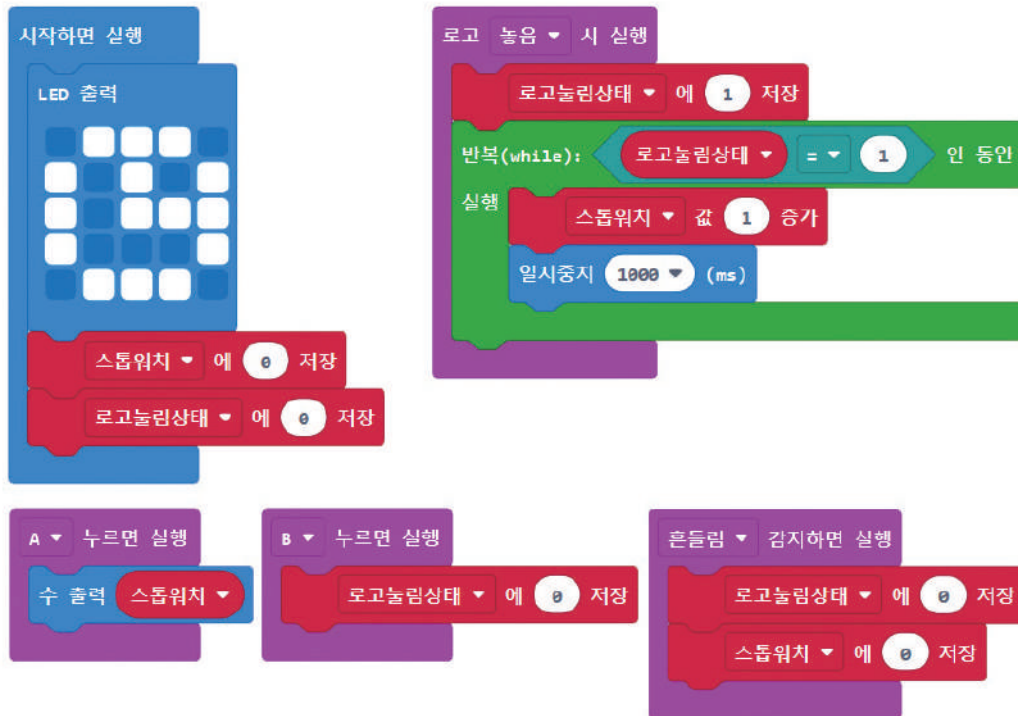
코딩 블록 메뉴 > **입력** 탭에서 A 누르면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓고 'B 누르면 실행'으로 설정해줍니다.



19_

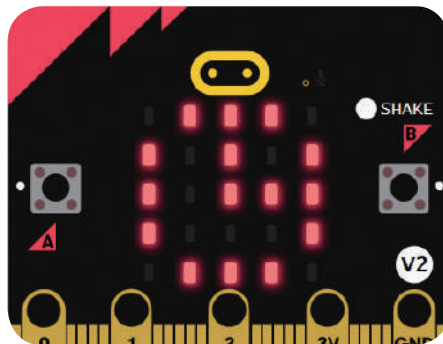
코딩 블록 메뉴 > **입력** 탭에서 흔들림 감지하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

스톱워치와 반대로 작동하는 타이머를 만들어봅시다.

A 버튼과 B 버튼으로 원하는 시간을 설정하고 A+B 버튼을 누르면 설정한 시간만큼 동작하는 타이머를 만들고 시간이 다 되었을 때 알람을 울릴 수 있도록 하려면 어떻게 해야 할까요?

응용하기 정답은 74쪽 응용하기 해답편에서 확인해주세요!

#04

마이크로비트 활용하기

08 FLAVOR

전자 촛불 만들기

마이크로비트로 소리 크기에 따라 불꽃이 변하는 전자 촛불을 만들어봅시다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행

마이크로비트를 시작하면 작성한 프로그램을 실행합니다.

무한반복 실행

작성한 프로그램을 무한 반복 실행합니다.

LED 출력

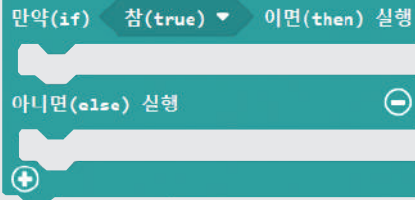
LED를 원하는 대로 출력할 수 있습니다.

소리 크기

소리 크기 값을 변수로 사용할 수 있습니다.

LED 스크린 밝기를 0로 설정

LED 스크린의 밝기를 조절할 수 있습니다.



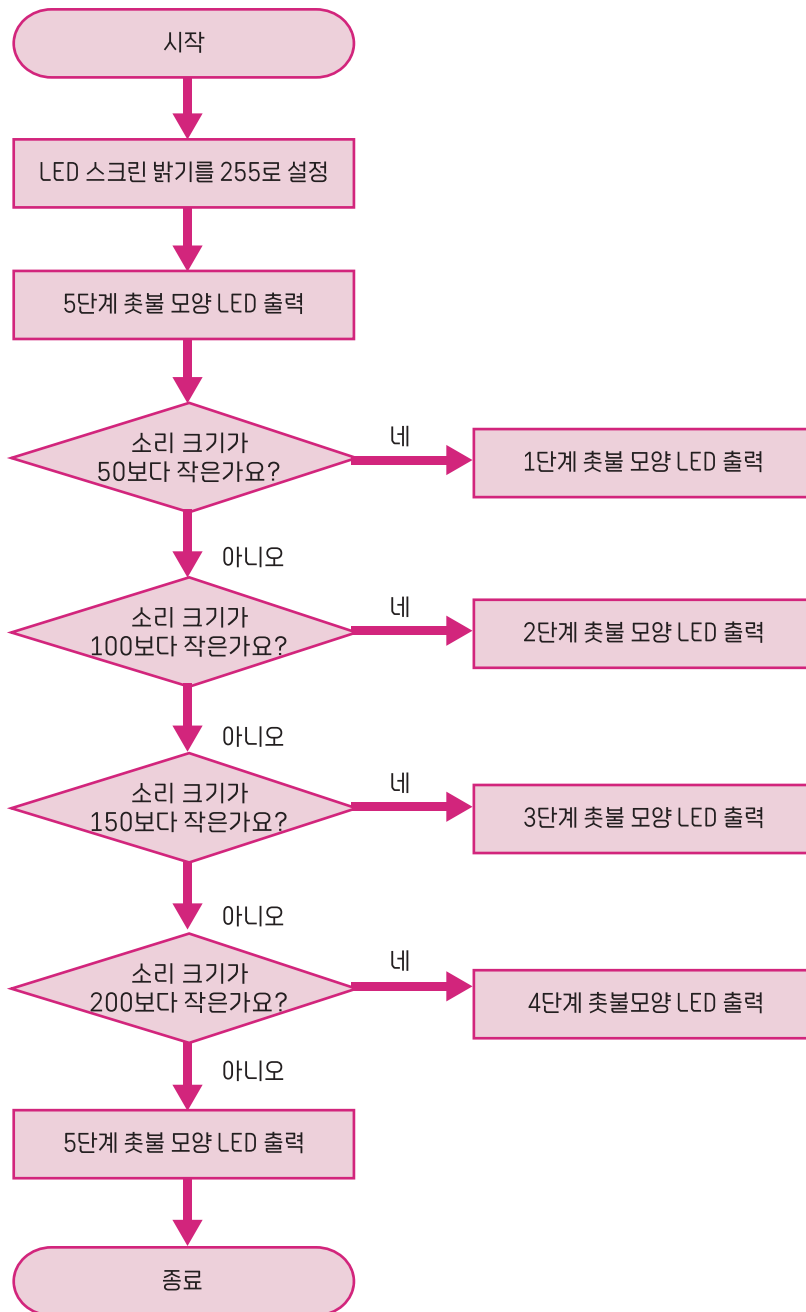
else if 조건문으로 원하는 조건을 설정하고, 설정한 조건을 만족했을 때 / 만족하지 않았을 때 작성한 프로그램을 실행합니다.

동작 원리

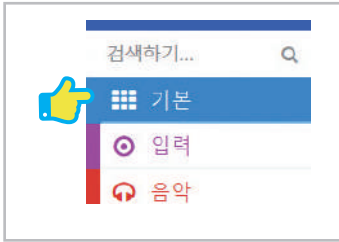
- 마이크로비트가 시작되면 **LED 스크린 밝기를 최대**로 설정합니다.
- 마이크로비트에 소리 크기가 감지되면 **소리 크기에 따라 촛불의 크기를 5단계로 출력**합니다.
(감지된 소리 크기가 작을수록 불꽃이 약하고 클수록 불꽃이 강하게 설정합니다.)

생각해보기

프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.

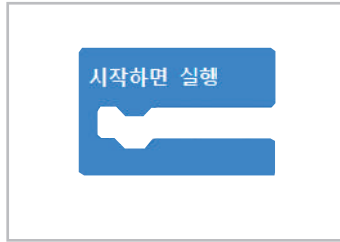


전자 촛불 만들기



01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



02_

시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



03_

코딩 블록 메뉴에서 **LED** 탭을 선택한 뒤 바로 아래 **더 보기** 탭을 선택합니다.



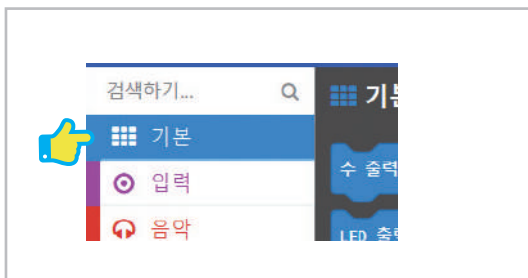
04_

스크린 밝기 설정 블록을 시작하면 실행 블록 아래에 놓고 촛불의 밝기를 설정해줍니다.
(0 : 꺼짐, 255 : 가장 밝음)



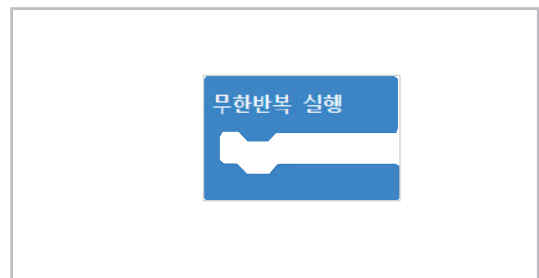
05_

시작하면 실행 > 스크린 밝기 설정 블록 아래에 코딩 블록 메뉴 > **기본** 탭에서 **LED 출력** 블록을 놓고 촛불 모양을 그려줍니다.



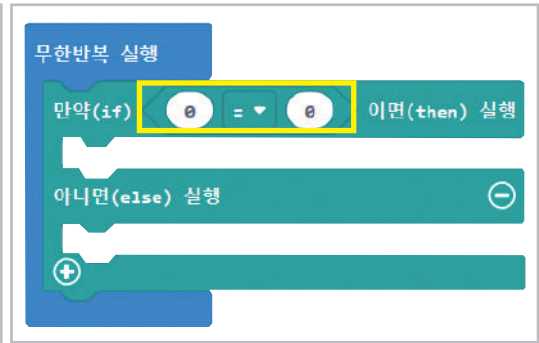
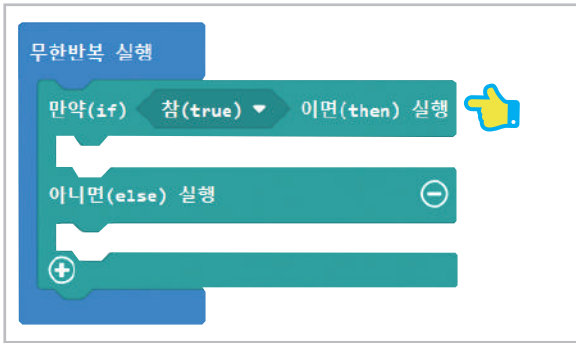
06_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



07_

무한반복 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.

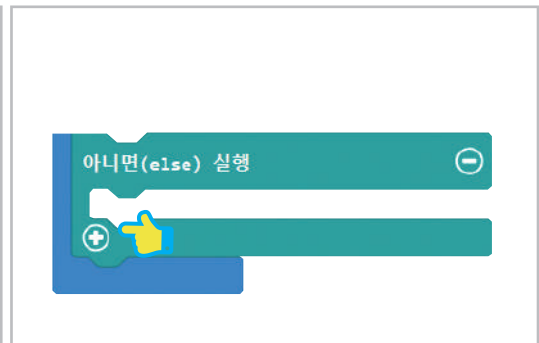
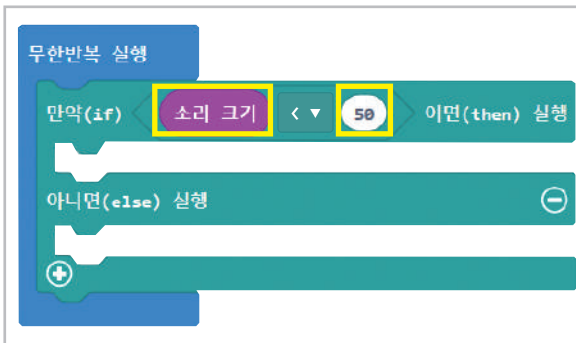


08_

무한반복 실행 블록 아래에 코딩 블록 메뉴 > 논리 탭의 else if 조건문을 놓아줍니다.

09_

else if 조건문 블록의 if문 조건에 코딩 블록 메뉴 > 입력 탭 > 비교 연산 블록을 놓아줍니다.



10_

비교 연산 블록에 코딩 블록 메뉴 > 입력 탭의 소리 크기 블록을 넣고 "소리 크기가 50보다 작을 때" 로 조건을 설정해줍니다.

11_

else if 조건문 아래에 + 버튼을 눌러 else if 조건문을 추가해줍니다.



12_

추가한 else if 조건문에 원하는 촛불의 세기 단계만큼 조건을 추가해줍니다.

※ 예제에서는 총 5단계로 설정했습니다. (1단계 불꽃 약함, 5단계 불꽃 강함)



13.

'소리 크기 < 50' 조건문 아래에 코딩 블록 메뉴 > 기본 > LED 출력 블록을 놓고 1단계 촛불 세기를 그려줍니다.

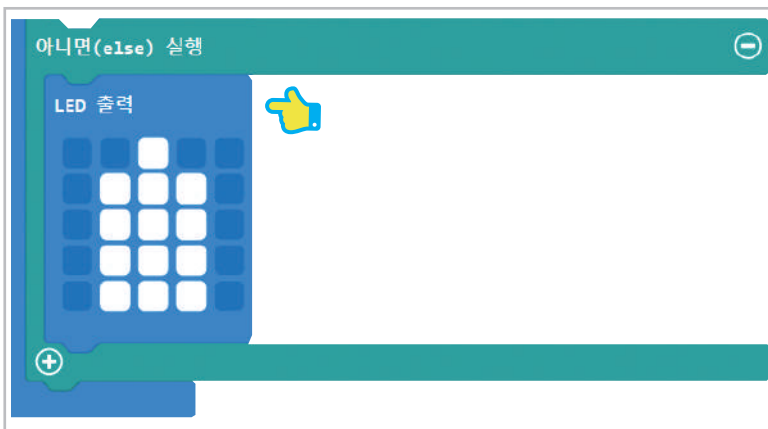
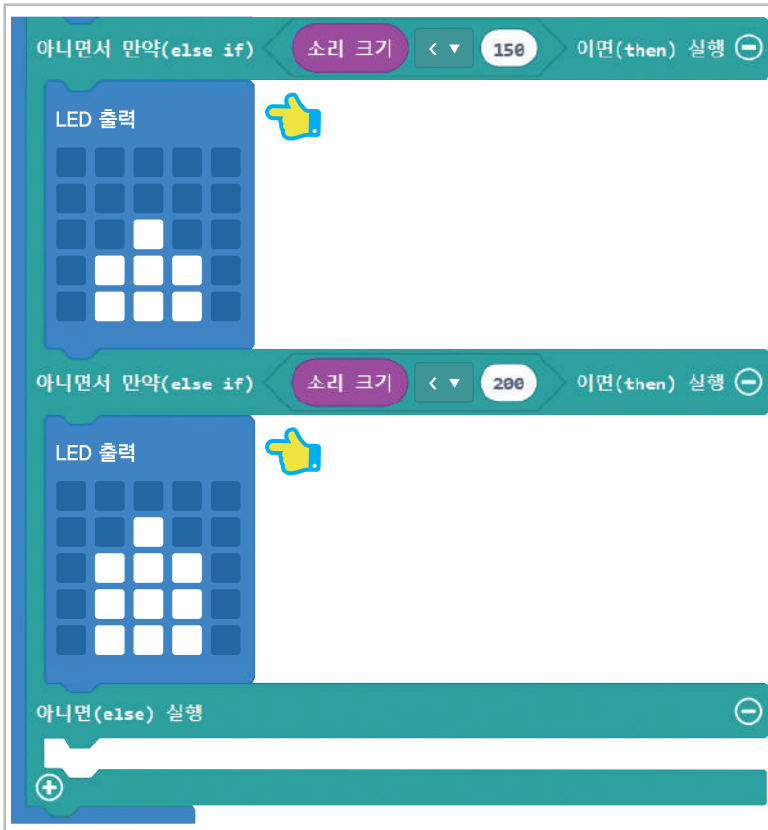


14.

'소리 크기 < 50' 조건문과 마찬가지로 각 조건에 맞게 촛불 세기를 LED로 그려줍니다.

예제에서 지정한 조건은 아래와 같습니다.

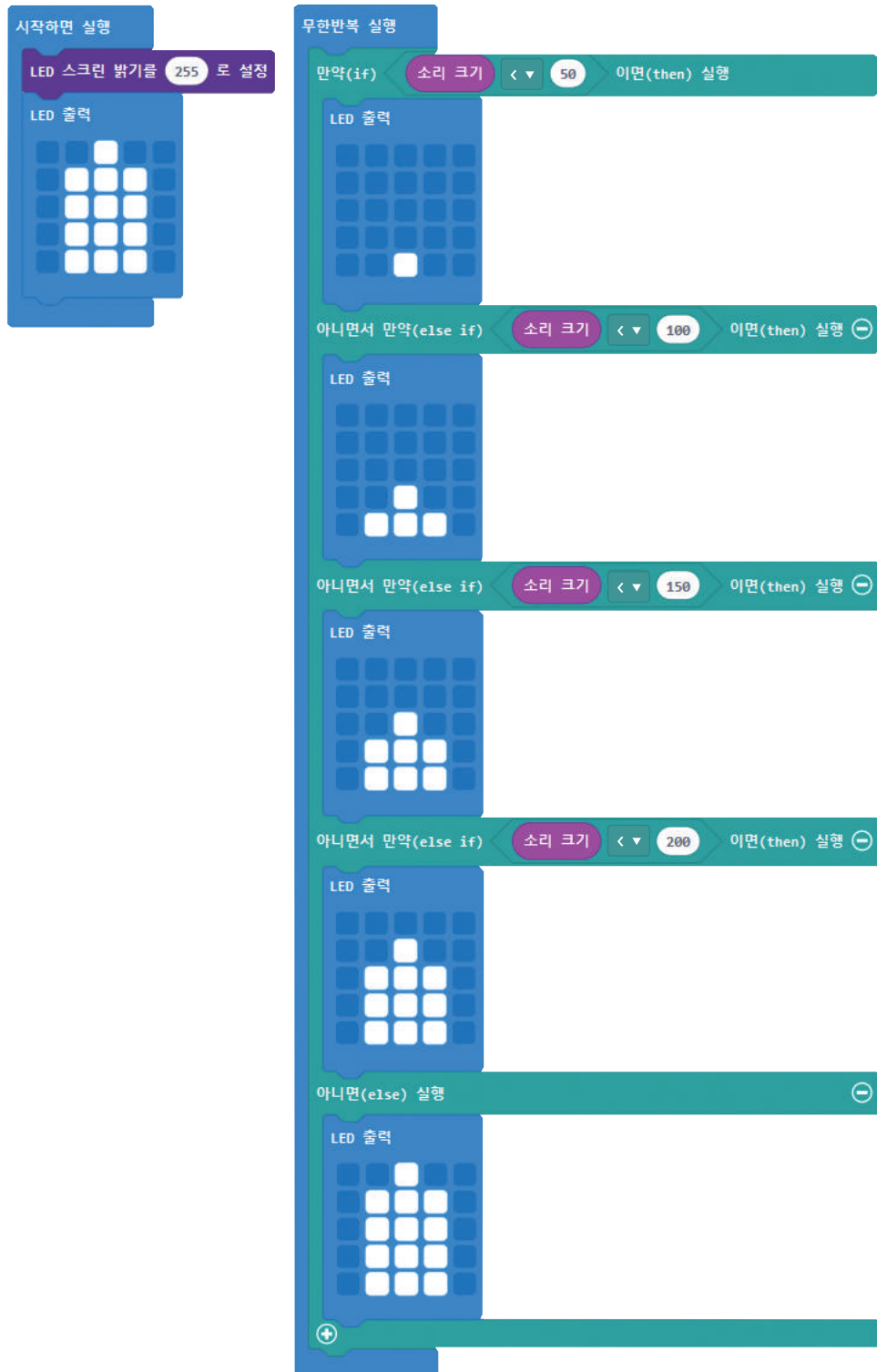
- 소리 크기 < 50 : 불꽃 1단계
- 소리 크기 < 100 : 불꽃 2단계
- 소리 크기 < 150 : 불꽃 3단계
- 소리 크기 < 200 : 불꽃 4단계
- 소리 크기 > 200 : 불꽃 5단계



15_

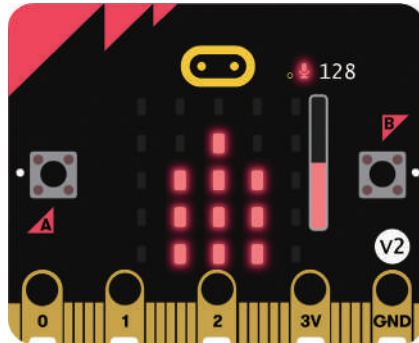
else 문은 위에서 작성한 조건 (**if**, **else if**)에 맞지 않을 때 실행되므로 5단계 촛불 세기를 그려줍니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

조도 센서를 활용해 **빛의 세기에 따라 불꽃 세기가 변하는 스마트 촛불**을 만들어봅시다.

빛의 세기가 100 이상일 때 약한 촛불이 켜지고, 빛의 세기가 50 이하일 때 강한 촛불이 켜지게 하려면 어떻게 해야 할까요?

응용하기 정답은 75쪽 응용하기 해답편에서 확인해주세요!

나침반 만들기

마이크로비트로 나침반을 만들어봅시다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행

마이크로비트를 시작하면 작성한 프로그램을 실행합니다.

무한반복 실행

작성한 프로그램을 무한 반복 실행합니다.

수 출력

입력한 수를 출력합니다.

문자열 출력

입력한 문자열을 출력합니다.

A 누르면 실행

A 버튼을 누르면 작성한 프로그램을 실행합니다

자기센서 값(°)

자기 센서 값을 사용할 수 있습니다.

음량을 127로 설정

스피커의 음량을 설정할 수 있습니다.

안녕 효과음 출력

정해진 효과음을 출력할 수 있습니다.

반복(repeat): 4회

실행

설정된 횟수만큼 작성한 프로그램을 반복 실행합니다.

만약(if) 참(true)이면(then) 실행

if 조건문으로 원하는 조건을 설정하고 설정한 조건을 만족했을 때 작성한 프로그램을 실행합니다.

또는(or)

OR 연산자로 두 조건 중 하나만 만족해도 '참'입니다.

방향

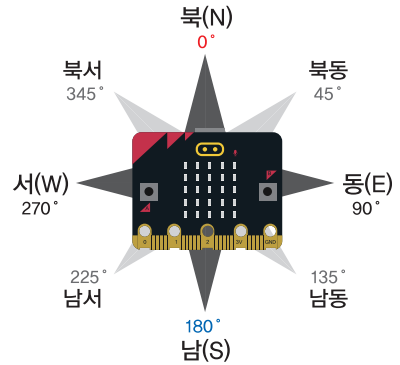
원하는 값을 담을 수 있는 변수입니다.

방향에 0 저장

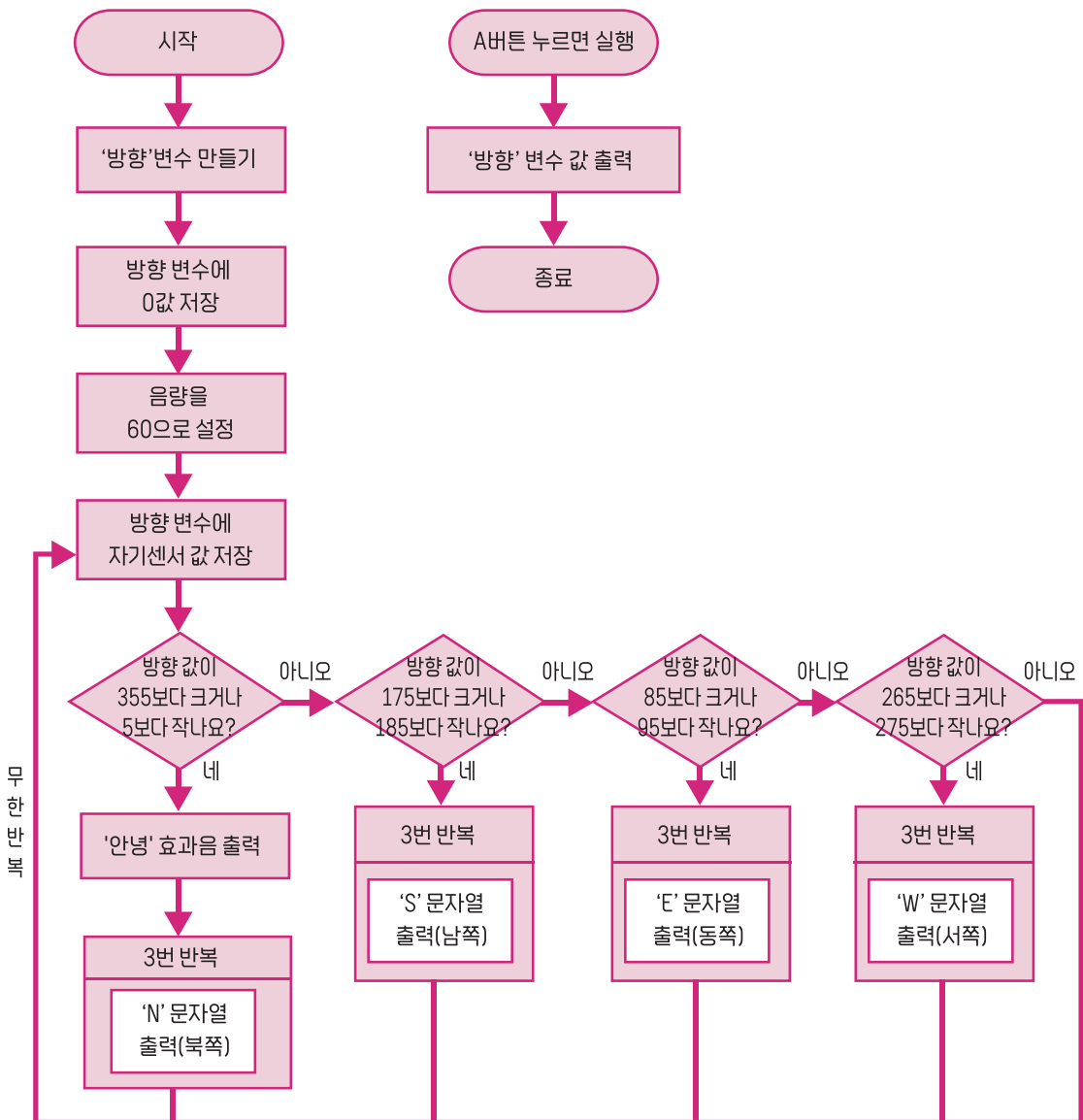
변수에 원하는 값을 저장합니다.

동작 원리

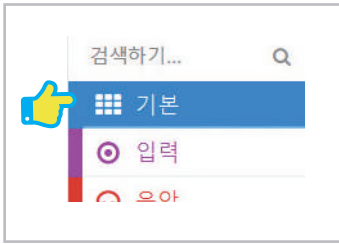
- 자기 센서 값을 담을 수 있는 '방향' 변수를 생성합니다.
- 마이크로비트가 시작되면 음량을 설정하고 '방향' 변수를 초기화합니다.
- 자기 센서 값을 '방향' 변수에 저장하고 동, 서, 남, 북 방향에 해당하는 값이 감지되면 감지된 방향을 문자열로 3번 출력, '북' 쪽을 감지했을 때는 '안녕' 효과음을 함께 출력합니다.
- A 버튼을 누르면 현재 방향 값이 나타납니다.



생각해보기

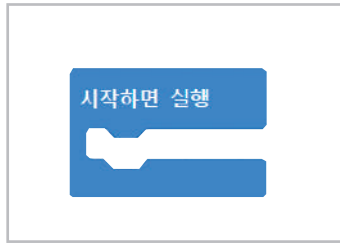


나침반 만들기



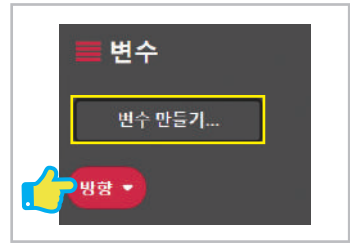
01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



02_

시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



03_

코딩 블록 메뉴 > **변수** 탭에서 변수 만들기를 하고 '방향' 변수를 생성합니다.



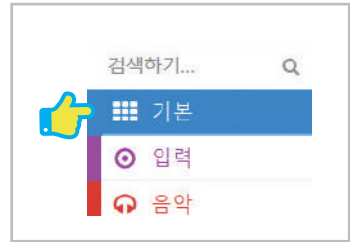
04_

시작하면 실행 블록 아래에 코딩 블록 메뉴 > **변수** > 변수에 값 저장 블록을 놓고 방향 변수에 '0' 을 저장합니다. (변수 값 초기화 역할)



05_

코딩 블록 메뉴 > **음악** > 음량 설정 블록을 놓고 음량을 60으로 설정합니다.



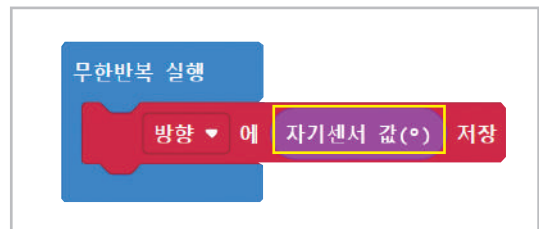
06_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



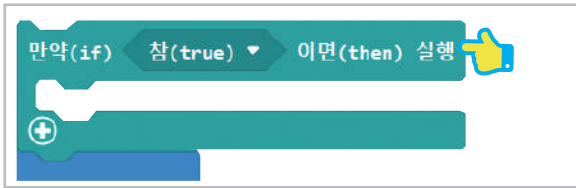
07_

무한반복 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



08_

무한반복 실행 블록 아래에 코딩 블록 메뉴 > **변수** > 변수에 값 저장 블록을 놓고 코딩 블록 메뉴 > **입력** 탭의 자기 센서 값을 저장합니다.



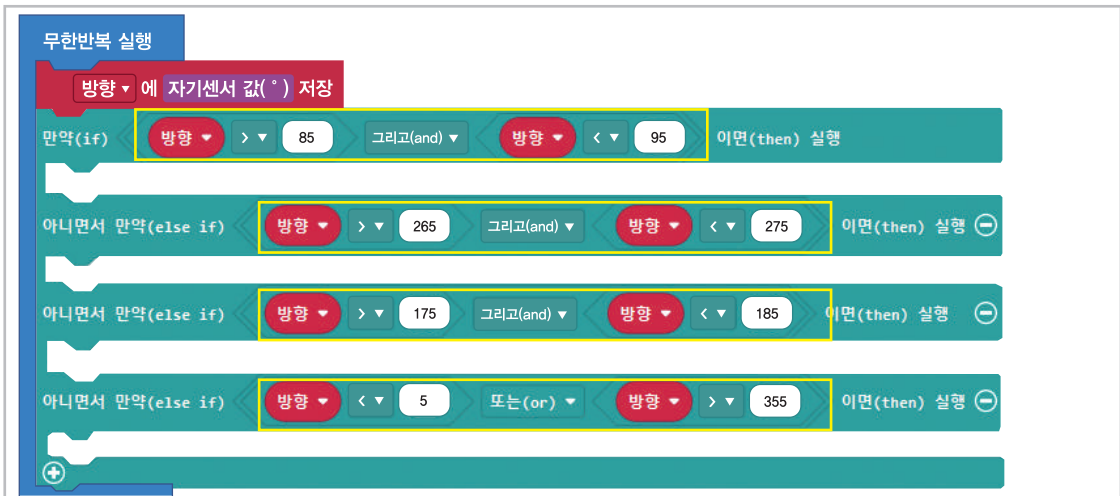
09_

코딩 블록 메뉴 > 논리 > if 조건문을
놓아줍니다.



10_

프로젝트에서 설정한 조건의 개수만큼 조건문을
추가해줍니다. (동, 서, 남, 북 4개)



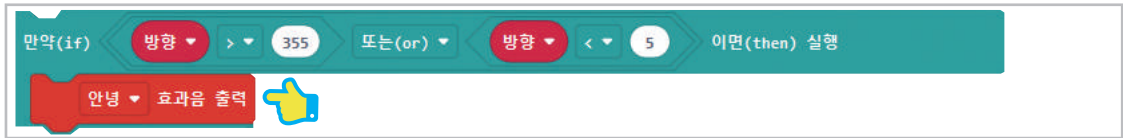
11_

각 방향에 맞는 방향 값의 범위를 설정해줍니다.

동 : 85 ~ 95° / 서 : 275 ~ 285° / 남 : 175 ~ 185° / 북 : 355 ~ 5°

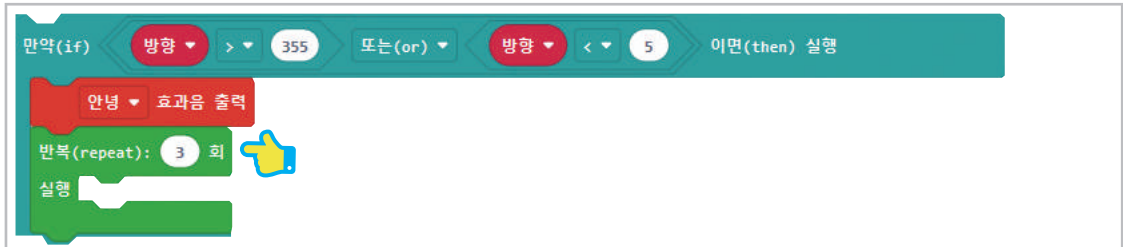
(사이 값의 경우 AND 연산문 사용)

(‘북’의 경우 사이값이 아니므로 OR 연산문 사용)



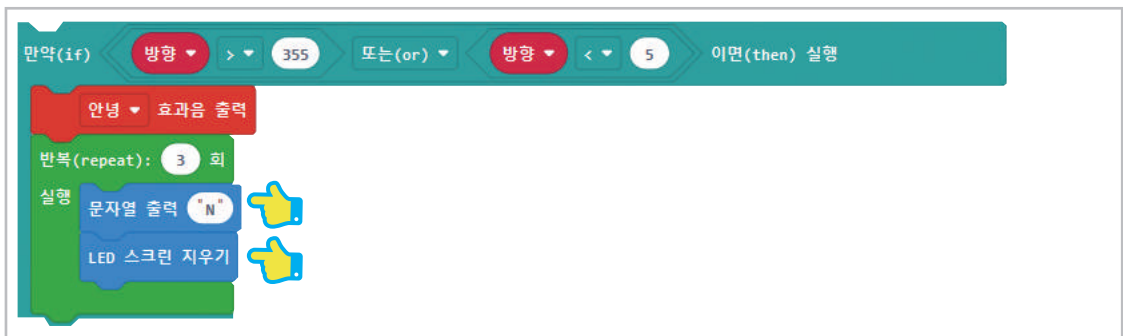
12.

북쪽에 해당하는 경우, **음악 > 효과음 출력** 블록을 놓고 안녕 효과음을 출력합니다.



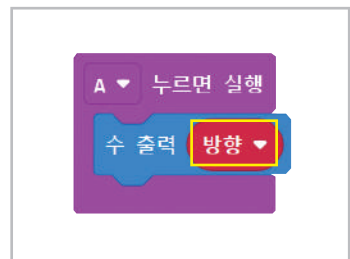
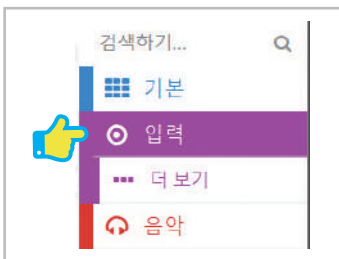
13.

코딩 블록 메뉴 > **반복 > 반복문**을 놓아주고 횟수를 3회로 설정합니다.



14.

코딩 블록 메뉴 > **기본 > 문자열 출력** 블록과 **LED 스크린 지우기** 블록을 놓아주고 방향에 맞는 문자열을 깜빡이며 출력합니다. (동 : E, 서 : W, 남 : S, 북 : N)



15.

코딩 블록 메뉴 > **입력** 탭을 선택합니다.

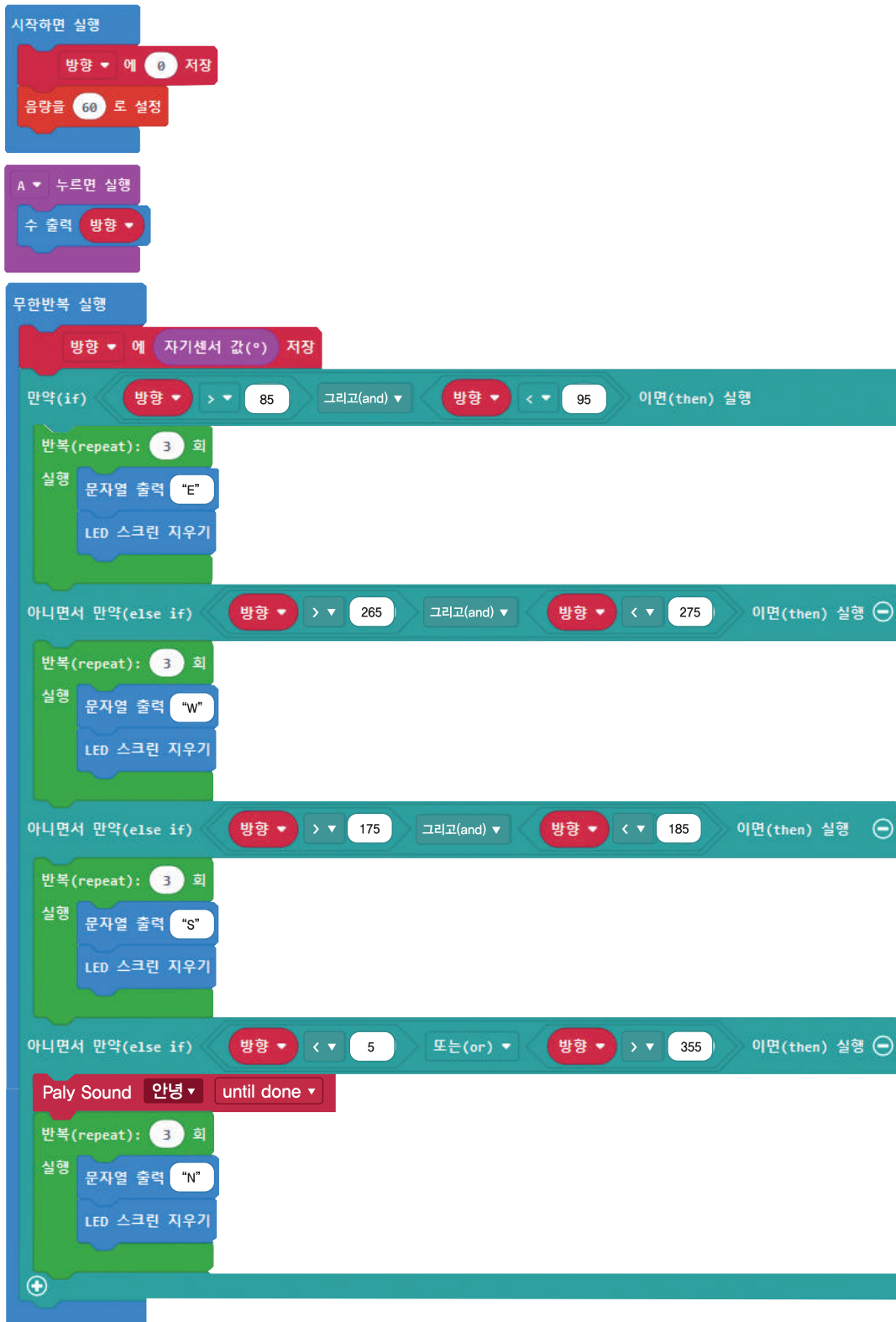
16.

버튼 누르면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.

17.

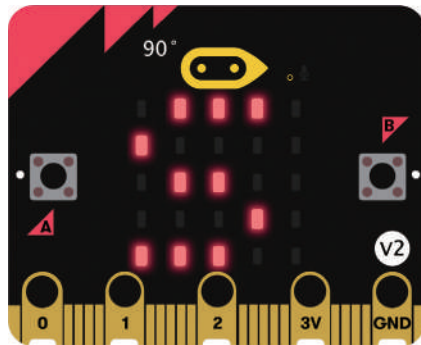
코딩 블록 메뉴 > **기본 > 수 출력** 블록을 놓고 방향 값을 저장합니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

좌회전, 우회전을 자동으로 감지하는 방향 지시등을 만들어봅시다.

마이크로비트의 가속도 센서 x축 값을 활용해 좌회전일 때 왼쪽 화살표, 우회전일 때 오른쪽 화살표를 나타내려면 어떻게 해야 할까요?

(가속도 센서 x축 값이 1024에 가까울수록 오른쪽 방향이며 -1024에 가까울수록 왼쪽 방향입니다.)

응용하기 정답은 76쪽 응용하기 해답편에서 확인해주세요!

10 FLAVOR

마이크로비트 키우기

다양한 입력 블록을 이용해 마이크로비트를 키우는 프로젝트를 작성해봅시다.

살펴보기

이번 프로젝트에서 사용하게 될 블록을 살펴봅시다.

시작하면 실행



마이크로비트를 시작하면 작성한 프로그램을 실행합니다.

무한반복 실행



작성한 프로그램을 무한 반복 실행합니다.

아이콘 출력



선택한 아이콘을 출력합니다.

수 출력



입력한 수를 출력합니다.

일시중지



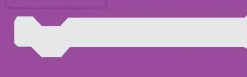
원하는 시간만큼 프로그램 실행을 일시 중지시킵니다.

A+B 누르면 실행



A+B 버튼을 누르면 작성한 프로그램을 실행합니다

흔들림 감지하면 실행



흔들림을 감지하면 작성한 프로그램을 실행합니다

로고 놓음 시 실행



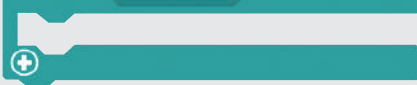
로고를 놓으면 작성한 프로그램을 실행합니다.

다다듬 멜로디 한 번 출력



정해진 멜로디를 출력할 수 있습니다.

만약(if) 참(true) 이면(then) 실행



if 조건문으로 원하는 조건을 설정하고 설정한 조건을 만족했을 때 작성한 프로그램을 실행합니다.



비교 연산자로 두 값을 비교할 수 있습니다.

애정도



원하는 값을 담을 수 있는 변수입니다.

애정도 에 저장



변수에 원하는 값을 저장합니다.

애정도 값 1 증가



변수 값을 증가(감소) 시킵니다.

재시작



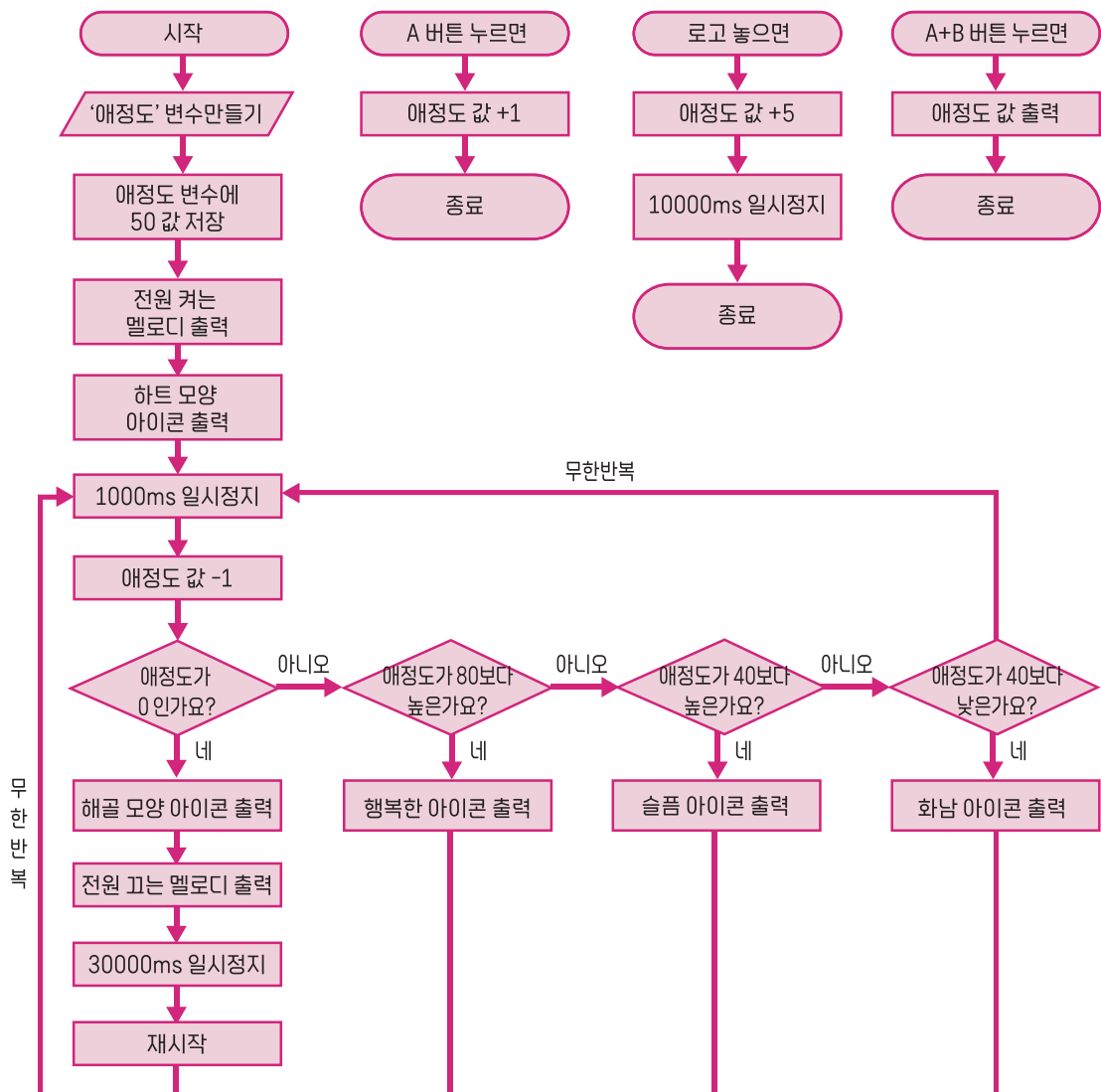
고급 제어 블록으로 마이크로비트를 재시작합니다.

동작 원리

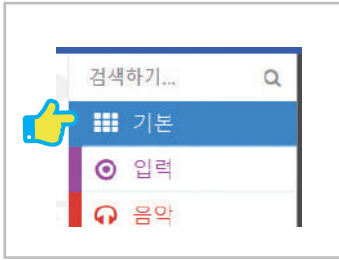
- 마이크로비트와의 애정도를 나타내는 '애정도' 변수를 생성하고 초기값으로 50을 저장합니다.
- 마이크로비트가 시작되면 전원을 켜는 멜로디와 하트 모양 아이콘을 출력합니다.
- 1초마다 애정도가 1씩 감소하며 애정도가 80보다 높을 때 행복함, 애정도가 40보다 높을 때 슬픔, 애정도가 40보다 낮을 때 화남을 표시합니다.
- 애정도가 0이 되면 해골 모양 아이콘과 함께 전원 끄는 멜로디를 출력하며 30초 후 재시작합니다.
- 마이크로비트와 놀아주면 (흔들림 감지) 애정도가 1 증가하고 간식을 주면 (로고 놓음) 애정도가 5 증가합니다. (간식은 10초에 한 번 줄 수 있음)
- A+B 버튼을 누르면 현재 애정도 값을 출력합니다.

생각해보기

프로젝트 알고리즘을 생각해보고 순서도를 작성해봅시다.



마이크로비트 키우기



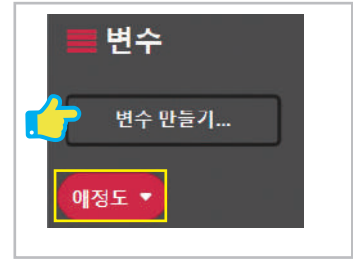
01_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



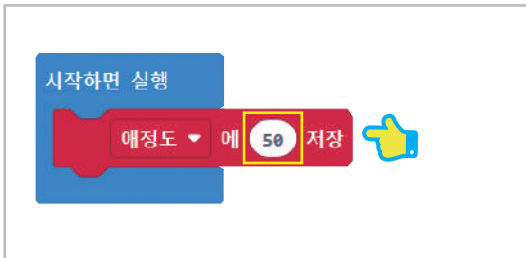
02_

시작하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



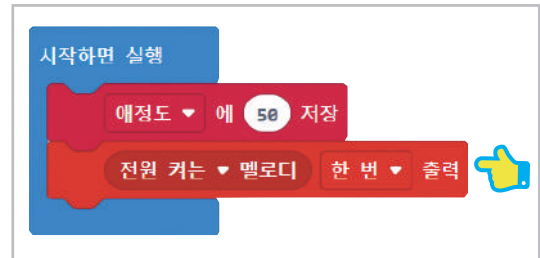
03_

코딩 블록 메뉴 > **변수** 탭에서 변수 만들기를 선택하고 '애정도' 변수를 생성합니다.



04_

코딩 블록 메뉴 > **변수** > 변수에 값 저장 블록을 놓고 '애정도'에 '50'을 저장합니다.



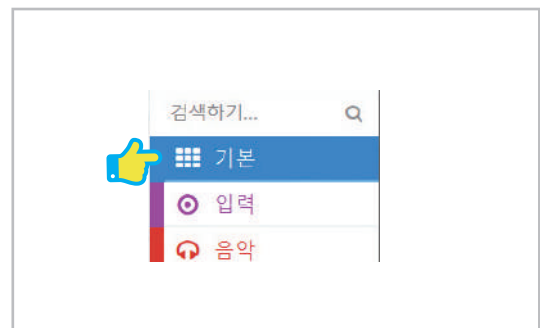
05_

코딩 블록 메뉴 > **음악** > 멜로디 출력 블록을 놓고 '전원 켜는 멜로디'를 '한 번' 출력합니다.



06_

코딩 블록 메뉴 > **기본** > 아이콘 출력 블록을 놓고 하트 모양을 출력합니다.



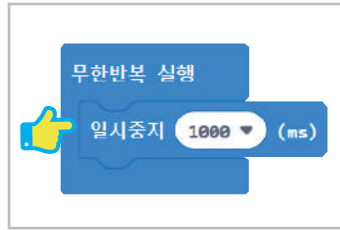
07_

코딩 블록 메뉴에서 **기본** 탭을 선택합니다.



08_

무한반복 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



09_

코딩 블록 메뉴 > 기본 > 일시중지 블록을 놓고 1초 일시중지로 설정합니다. (1초마다 애정도를 1씩 감소시키기 위함)



10_

코딩 블록 메뉴 > 변수 > 변수 값 증가 블록을 놓고 '애정도 값을 -1 증가' 시켜줍니다.



11_

무한반복 실행 블록 아래에 코딩 블록 메뉴 > 논리 탭의 if 조건문을 놓고 아래 버튼을 눌러 if 조건문을 3개 추가해줍니다.

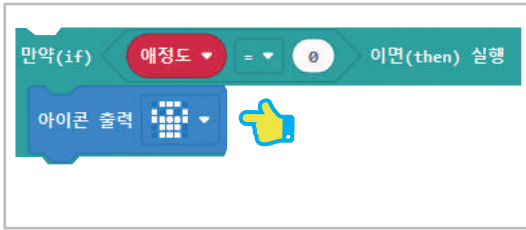


12_

추가한 if 조건문에 애정도에 따른 조건을 설정합니다.

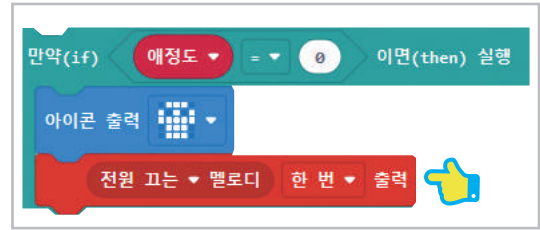
예제에서 지정한 조건은 총 4개 입니다.

- 애정도 = 0,
- 애정도가 80보다 클 때,
- 애정도가 40보다 클 때,
- 애정도가 40보다 작을 때



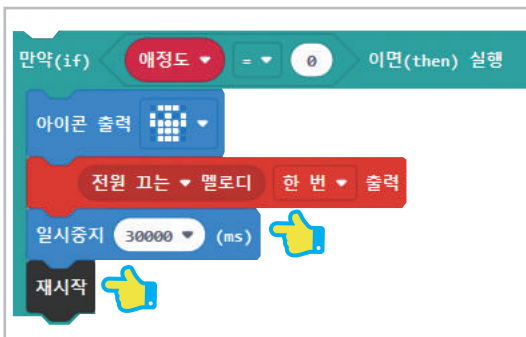
13_

'애정도 = 0' 조건문 아래에 코딩 블록 메뉴 > **기본** > 아이콘 출력 블록을 놓고 해골 모양을 선택합니다.



14_

코딩 블록 메뉴 > **음악** > 멜로디 출력 블록을 놓고 '전원 끄는 멜로디'를 '한 번' 출력합니다.



15_

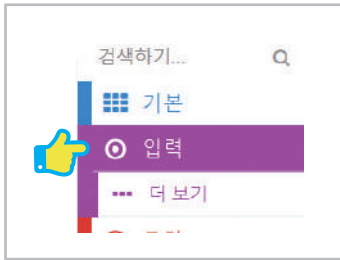
'애정도 = 0' 조건문 아래에 코딩 블록 메뉴 > **기본** > 일시 중지 블록과 코딩 블록 메뉴 > **고급** > **고급제어** > 재시작 블록을 놓고 '30초 일시중지 후 재시작'으로 설정합니다.



16_

코딩 블록 메뉴 > **기본** > 아이콘 출력 블록을 활용 해 각 조건에 맞는 아이콘을 출력해줍니다.

- 애정도 = 0일 때 해골 아이콘,
- 전원 끄는 멜로디, 30초 후 재시작
- 애정도 > 80 (행복)
- 40 < 애정도 < 80 (슬픔)
- 애정도 < 40 (화남)



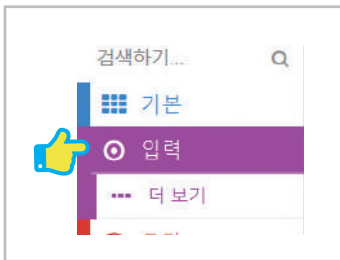
17_
코딩 블록 메뉴 > **입력** 탭을 선택합니다.



18_
흔들림 감지하면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓습니다.



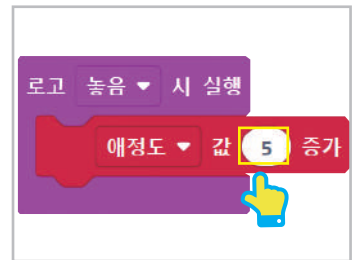
19_
코딩 블록 메뉴 > **변수** > '변수 값 증가' 블록을 놓고 '애정도 값을 +1 증가' 시켜줍니다. (흔들림 감지 = 돌아주기 = 애정도 1 증가)



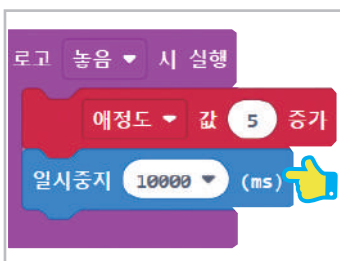
20_
코딩 블록 메뉴 > **입력** 탭을 선택합니다.



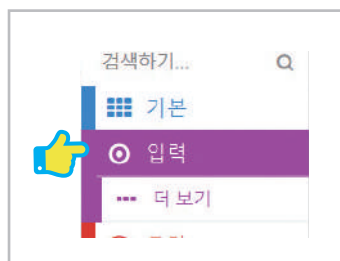
21_
로고 누르면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓고 로고 누름 시 실행으로 설정합니다.



22_
코딩 블록 메뉴 > **변수** > '변수 값 증가' 블록을 놓고 '애정도 값을 +5 증가' 시켜줍니다. (로고 누름 = 간식주기 = 애정도 5 증가)



23_
코딩 블록 메뉴 > **기본** > 일시 중지 블록을 놓고 10초 일시 중지 합니다. (간식은 10초에 한 번만 줄 수 있도록 설정)



24_
코딩 블록 메뉴 > **입력** 탭을 선택합니다.



25_
A를 누르면 실행 블록을 드래그 앤 드롭하여 코딩 영역에 놓고 A+B 누르면 실행으로 설정합니다.



26_

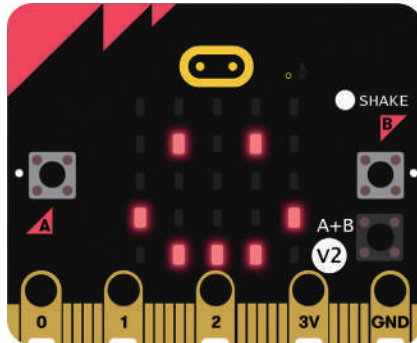
코딩 블록 메뉴 > 기본 > 수 출력 블록을 놓고 애정도 값을 출력합니다.

완성 예제



마무리하기

마이크로비트가 잘 동작하는지 확인해봅시다.



제대로 동작하지 않는다면 무엇이 잘못되었는지 생각해보고 잘 동작할 수 있도록 수정해보세요!

응용해보기

마이크로비트 키우기에 **낮잠 자기 모드**를 추가해봅시다.

마이크로비트 **스크린이 바닥을 향하면 낮잠 자기 모드**가 되며 **5초마다 애정도가 1 증가**하도록 하고, 마이크로비트에 **'일어나!'** 하고 **속삭이면 낮잠 자기 모드가 해제**되도록 하려면 어떻게 해야 할까요? 그 동안 배웠던 **if 문**, **While 문**, **변수** 등을 활용해보세요!

#05

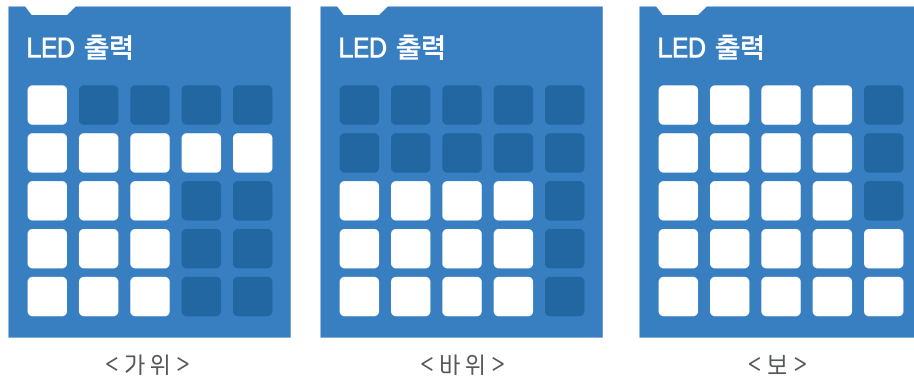
응용하기 해답편

11 FLAVOR

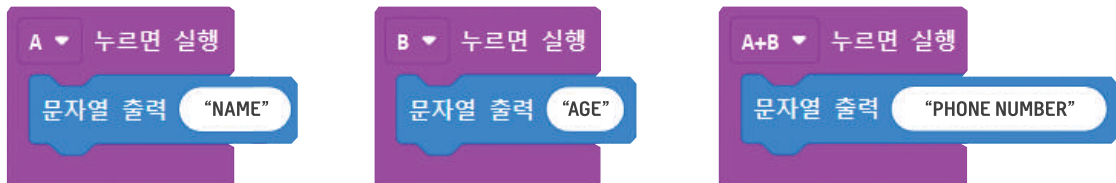
응용하기 해답

앞에서 배운 마이크로비트 응용해보기 해답들을 알아봅시다.

Flavor 2. 가위바위보 그리기



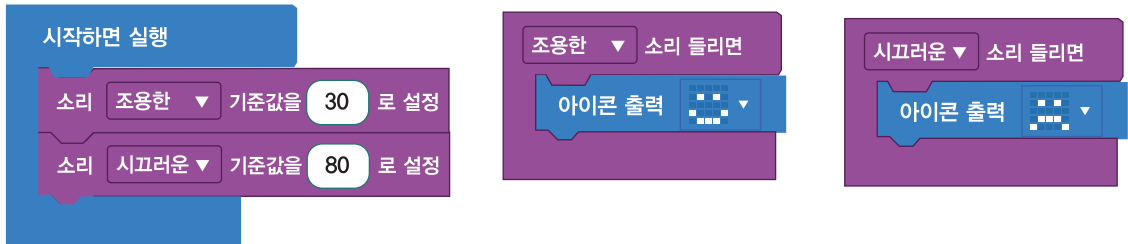
Flavor 3. 나만의 명함만들기



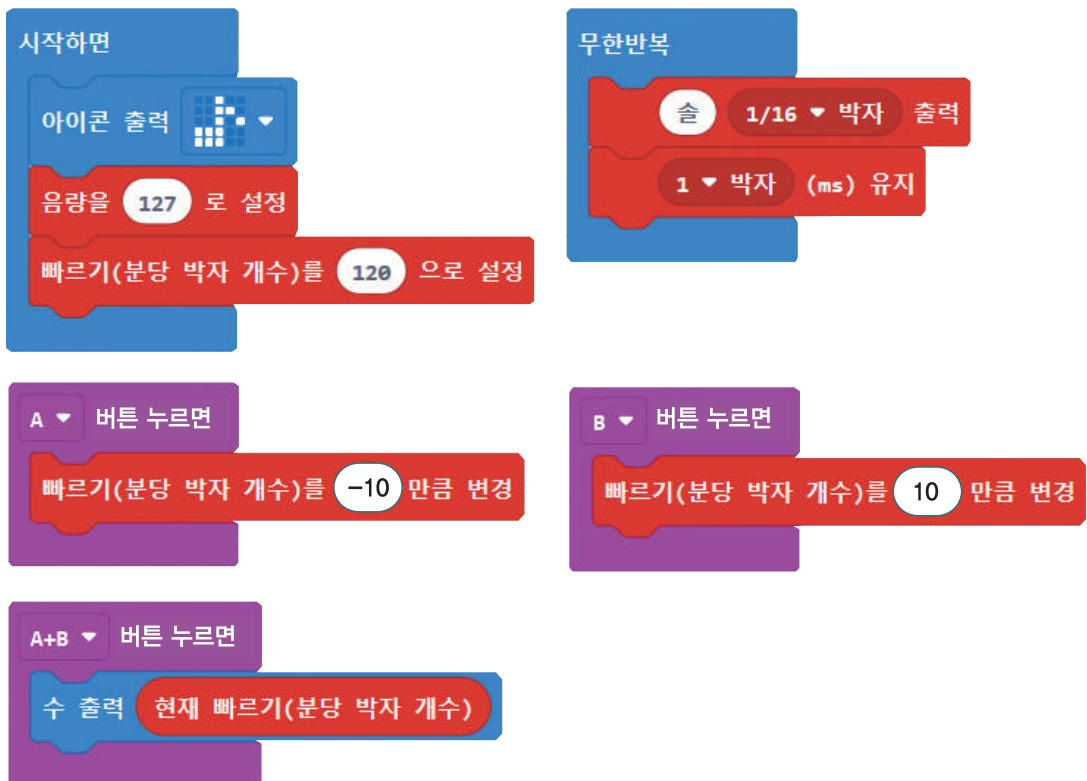
Flavor 4. 만보기 만들기



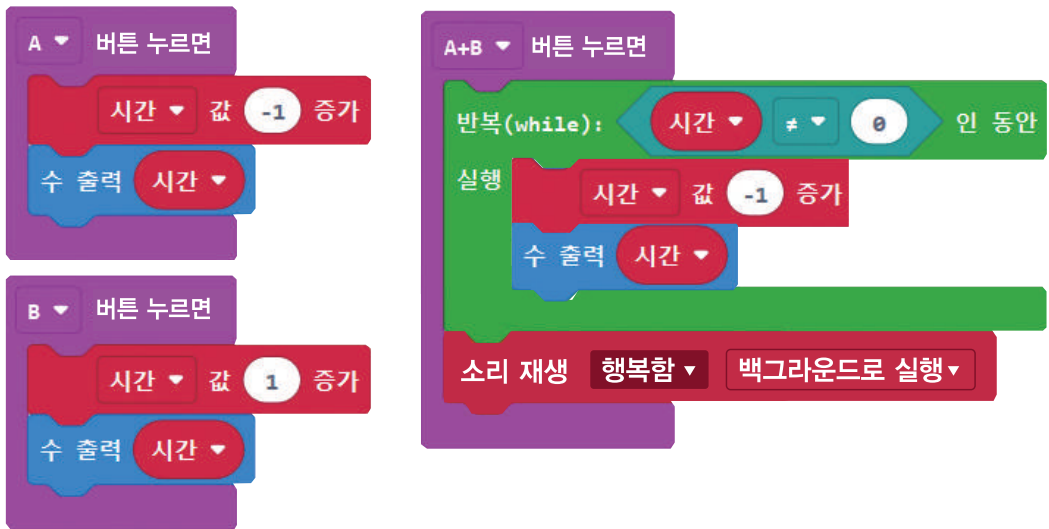
Flavor 5. 소음 감지기 만들기



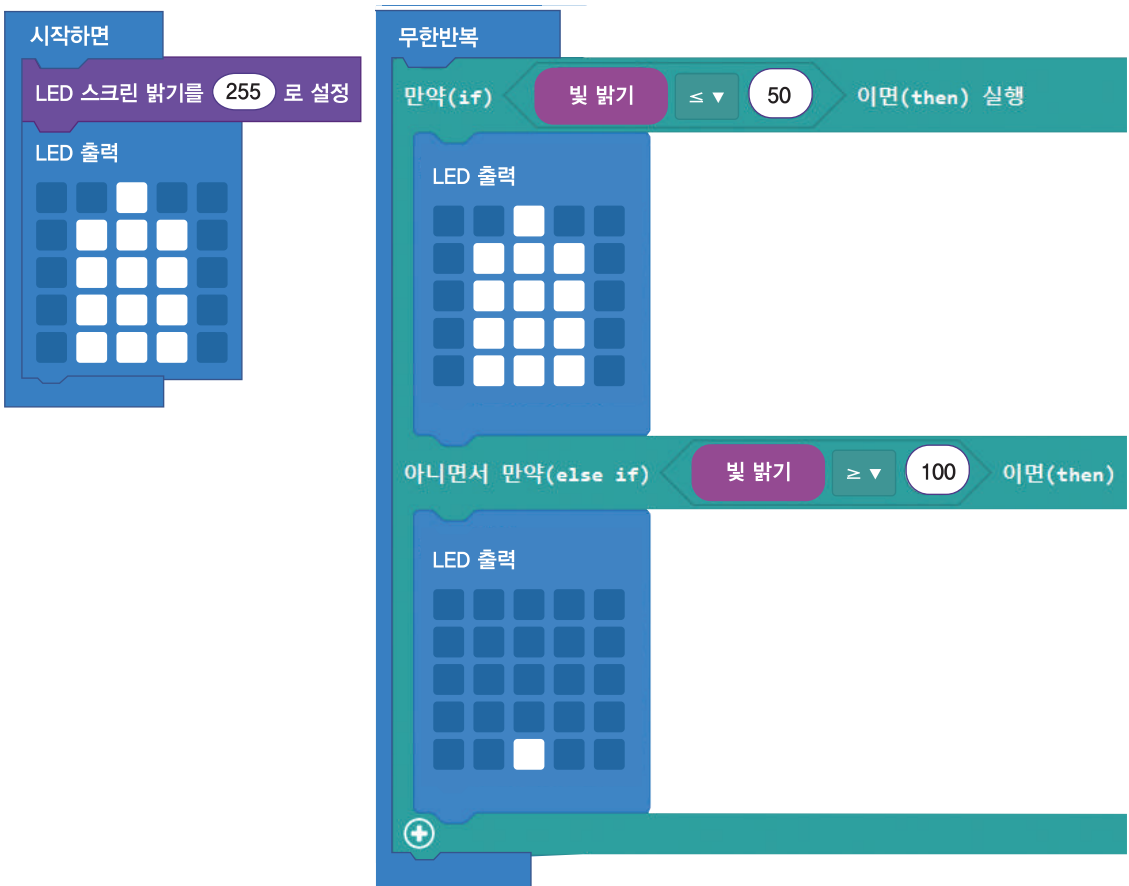
Flavor 6. 속도 조절 메트로놈 만들기



Flavor 7. 타이머 만들기



Flavor 8. 조도센서에 반응하는 스마트 촛불 만들기

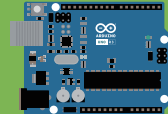
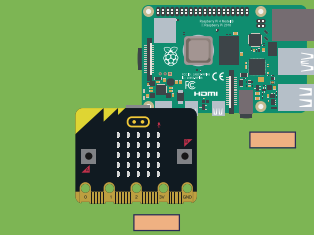


Flavor 9. 방향지시등 만들기



Flavor 10. 낮잠자기 모드





Only 디바이스마트에서만 만나볼 수 있는

디바이스마트

자체 제작 키트 DIY

개념과 실습 모두 한방에 잡는 방법!

아두이노(Arduino)



WEKIT AI 코딩 친구 KIT

상품번호: 14587099



IoT 스마트 무드등 키트

상품번호: 13519826



음성인식 키트

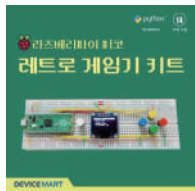
상품번호: 12526506



구독자 카운터 키트

상품번호: 14486353

라즈베리파이(Raspberry Pi)



피코 레트로 게임기 키트

상품번호: 14594158



피코 GPS 속도측정 키트

상품번호: 14622392



피코 입문 키트

상품번호: 15162348



프리미엄 키트

상품번호: 1279309

마이크로비트(Micro:bit)



미세먼지 측정 키트

상품번호: 12391833



심박수 측정 키트

상품번호: 12737672



나만의 파워 서플라이 만들기

상품번호: 14483410



레트로 아케이드 DIY 키트

상품번호: 13904242



DIY 메이커(Maker)

디바이스마트 키트 시리즈

MY:BIT or ARDUBOX



WK 마이크로비트 화분 키우기 키트 [WK-MB-K01]

· 상품번호: 14918370 · 가격: 35,000원 (VAT 미포함)

마이크로비트 코딩 교육용 화분 키트입니다.
자동 물주기, 토양 습도 측정 기능으로 손쉽게 화분을 키워보아요.



MY:Bit ~기상 관측편~ [MYBit-WT]

· 상품번호: 15023591 · 가격: 40,000원 (VAT 미포함)

5종의 센서를 이용해 7가지 기상을 관측할 수 있습니다.
온도, 습도, 기압, 풍향, 풍속, 미세먼지, 지진을 모두 감지합니다.



아두이노 음악분수 키트 [KIT-HWD001]

· 상품번호: 15207607 · 가격: 32,000원 (VAT 미포함)

나만의 화려한 음악 분수대를 만들어보아요!
버튼을 이용하여 LED, 워터펌프, 부저를 제어 합니다.



MY:Bit ~친환경 에너지편~ [MYBit-GE]

· 상품번호: 15266235 · 가격: 33,000원 (VAT 미포함)

풍력발전과 태양광에 대해 이해하고 실습해봅시다.
모터와 태양광 패널을 통해 LED를 점등하고 발전된 전기량을 계산합니다.



