



13주차 2차시

컴퓨팅사고력과 코딩

틴커카드 서킷 활용하기



동양미래대학교



동의과학대학교



영진전문대학교
YEUNGJIN UNIVERSITY



전주비전대학교



충북보건과학대학교
CHUNGBUK HEALTH & SCIENCE UNIVERSITY

학습목표

- 틴커캐드 서킷을 이용하여 신호등과 같이 3개의 LED (Red, Green, Blue)를 교대로 깜박이도록 코딩할 수 있다.

학습목차

[응용예제] LED 제어 시뮬레이터

생각하기

[응용예제] LED 제어 시뮬레이터

[문제상황]

- 3개의 LED(RED, GREEN, BLUE)를 이용하여 신호등과 같이 LED가 교대로 On, Off 하려면 어떻게 해야 할까?
 - 아두이노에 LED 광원 3개를 연결하여 하드웨어를 구성한 후 1초마다 각 LED 광원이 차례대로 깜박이도록 코딩하자!



문제정의 및 분해

1. 문제정의

- 아두이노 보드에 LED 광원 3개를 연결한 후 매초마다 교대로 깜박이도록 코딩하기

2. 문제분해

- 아두이노 보드에 LED 3개를 연결하여 하드웨어 구성하기
- 틴커캐드를 이용하여 LED 제어 시뮬레이터 구현하기
 - ▶ 각 LED On, 대기 시간, Off 설정
 - ▶ 빨간색 LED 깜박이고, 녹색 LED 깜박인 다음 파란색 LED가 깜박이도록 제어하기



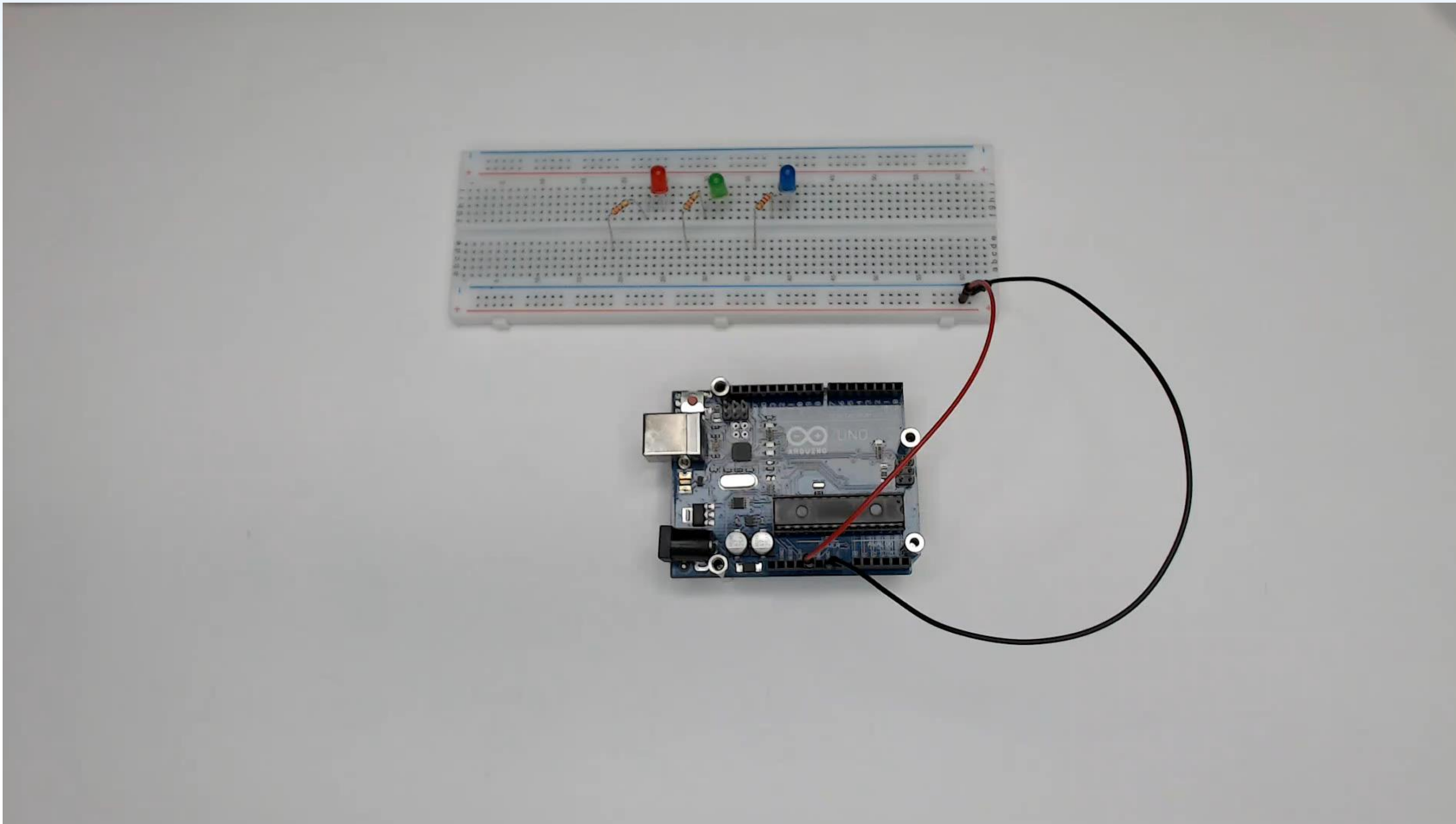
알고리즘

- 빨간색(RED), 초록색(GREEN), 파란색(BLUE) LED 마다 아래 명령어 반복하기
 - ▶ LED를 On(HIGH)으로 설정
 - ▶ 1초 대기
 - ▶ LED를 Off(LOW)로 설정
 - ▶ 1초 대기

코딩하기

[응용예제] LED 제어 시뮬레이터

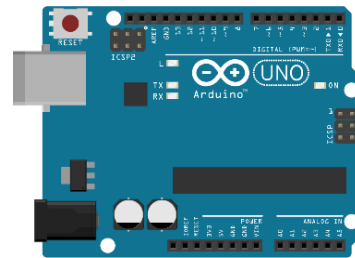
▶ 결과영상 (아두이노 활용) 미리보기



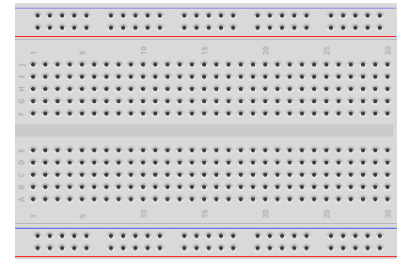
[사전준비]

- 아두이노 보드
- 브레드 보드
- 저항(220옴) X 3개
- LED 광원 X 3개
(RED, GREEN, BLUE)

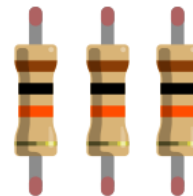
실습에 필요한 부품



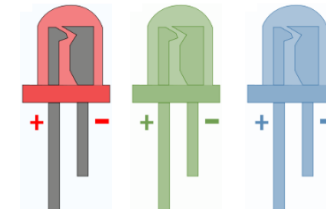
아두이노(UNO) 보드



브레드 보드



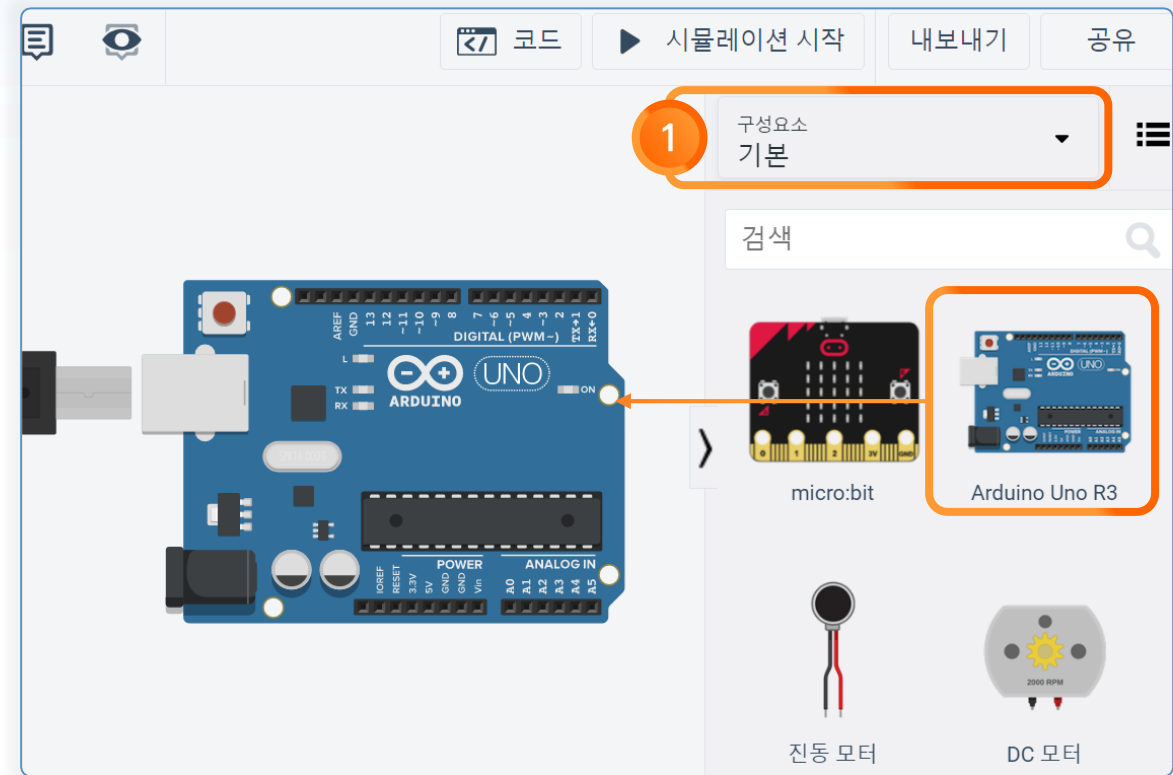
저항(220Ω) X 3



LED X 3
(RED, GREEN, BLUE)

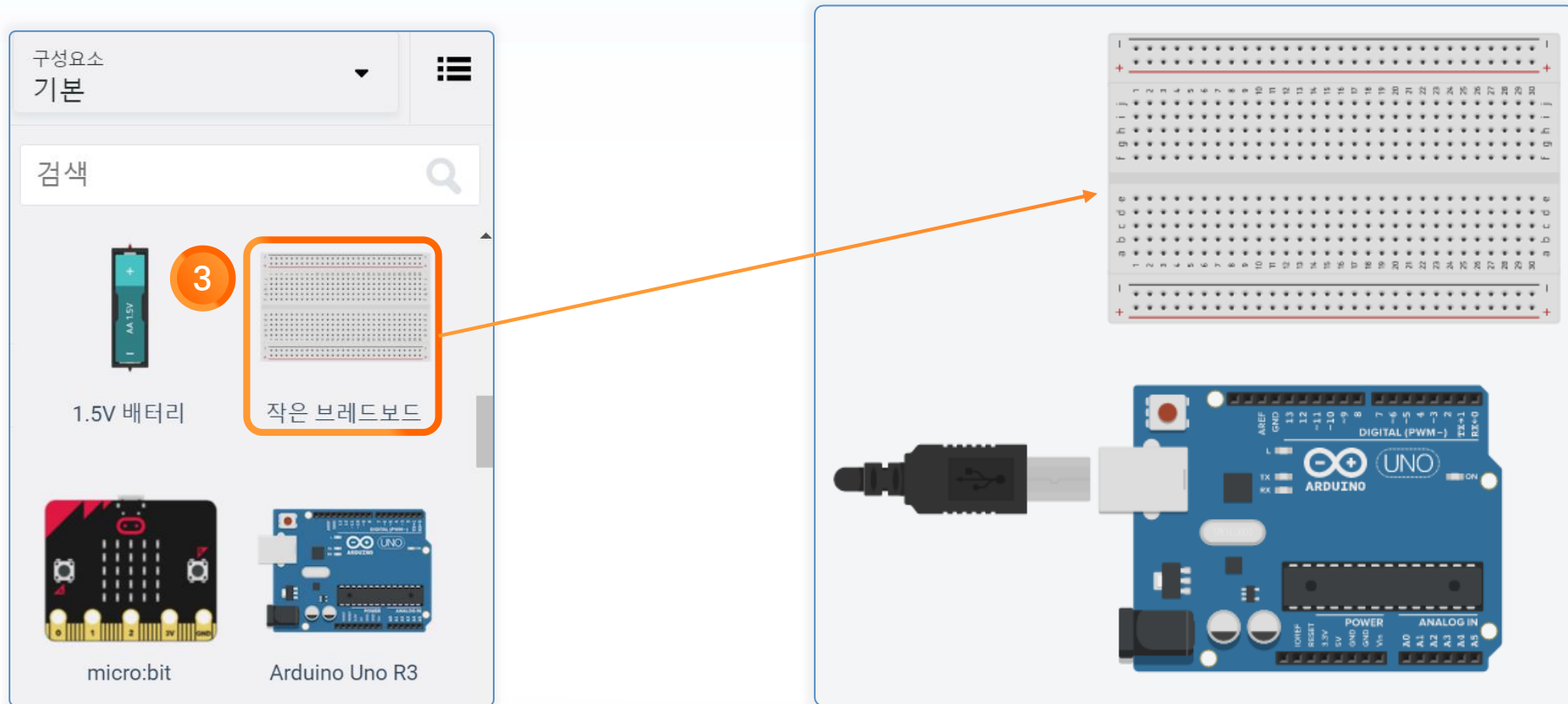
[사전준비 - 하드웨어 연결]

- 1 화면 상단 우측에 있는 구성요소 - [기본] 클릭
- 2 [Arduino Uno R3] 선택 후 작업 영역으로 드래그



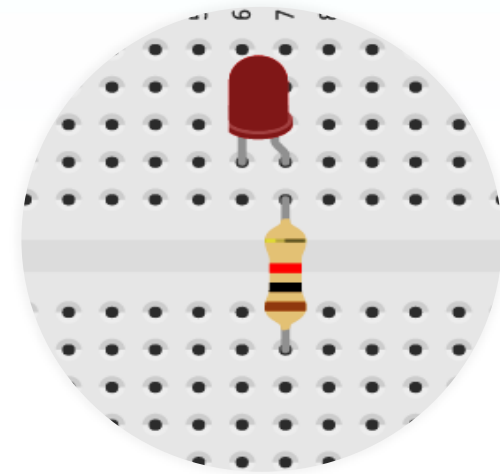
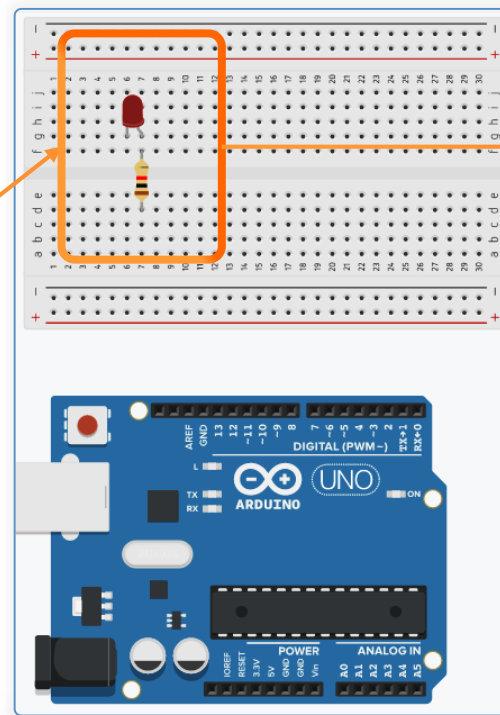
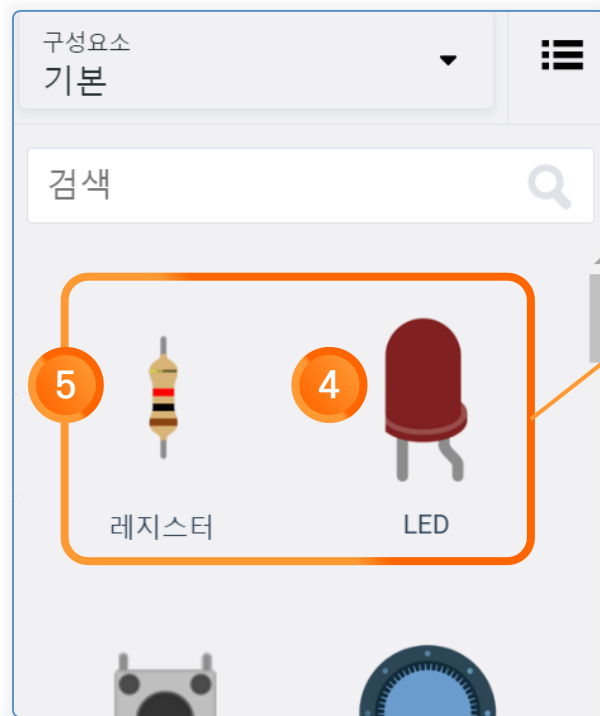
[사전준비 - 하드웨어 연결]

3 [작은 브레드보드] 선택 후 작업 영역으로 드래그



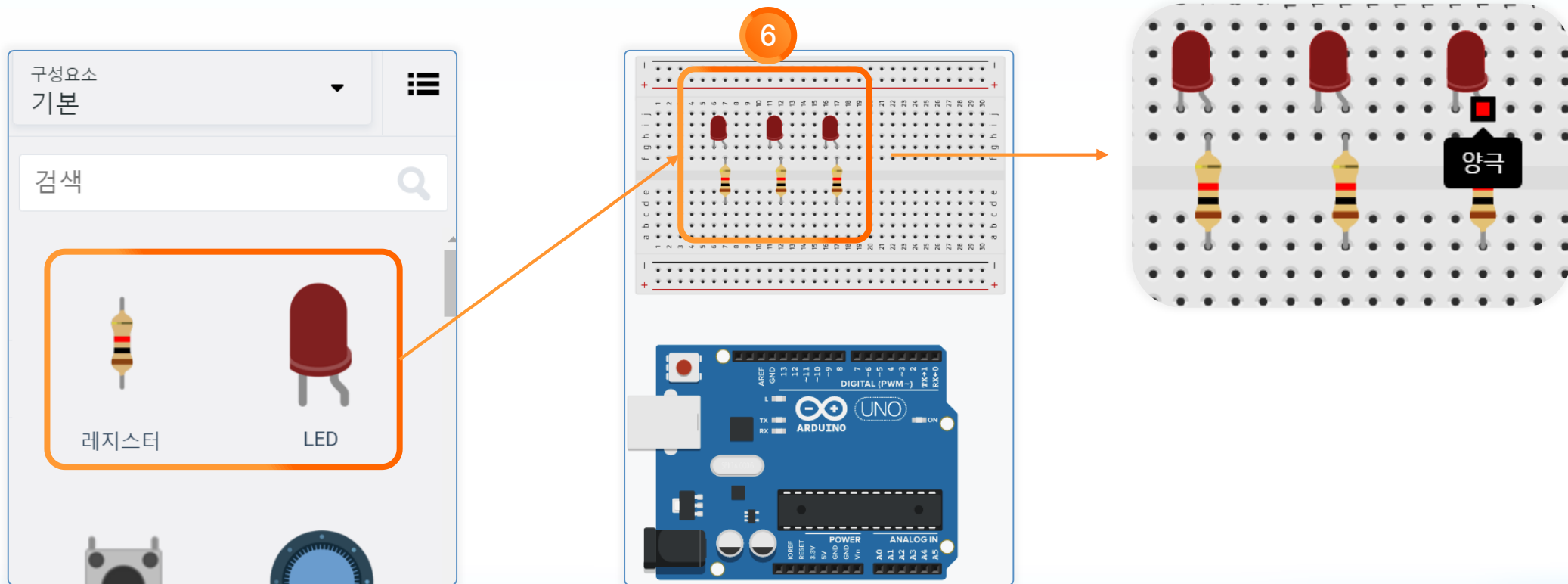
[사전준비 - 하드웨어 연결]

- ④ [LED] 선택 후 브레드보드 위로 드래그
- ⑤ [레지스터] 선택 후 드래그하여 LED의 “양극”과 연결



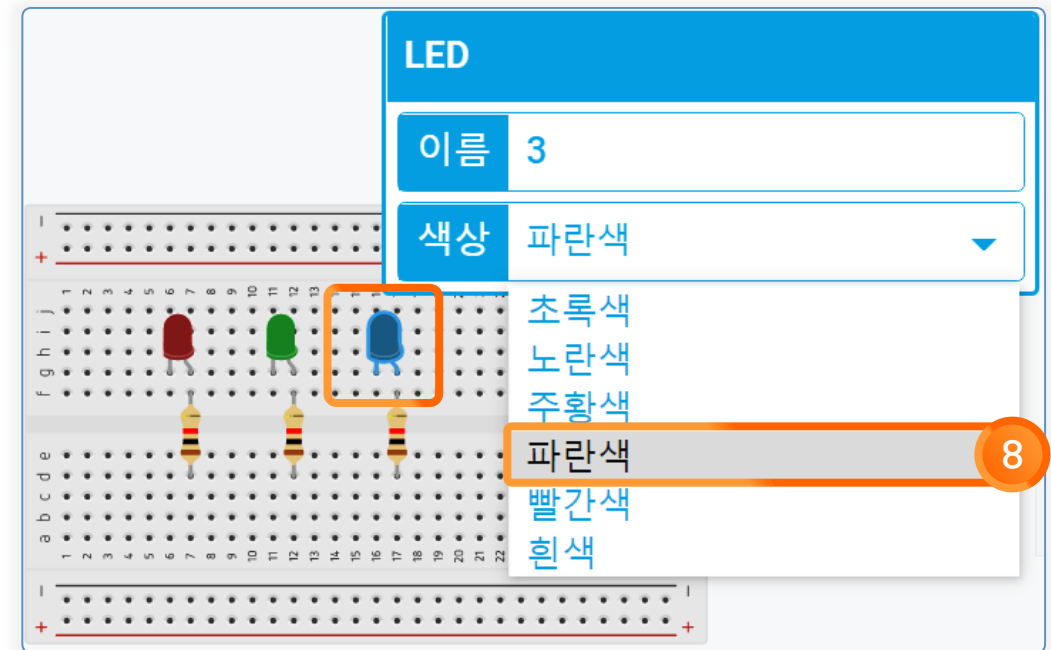
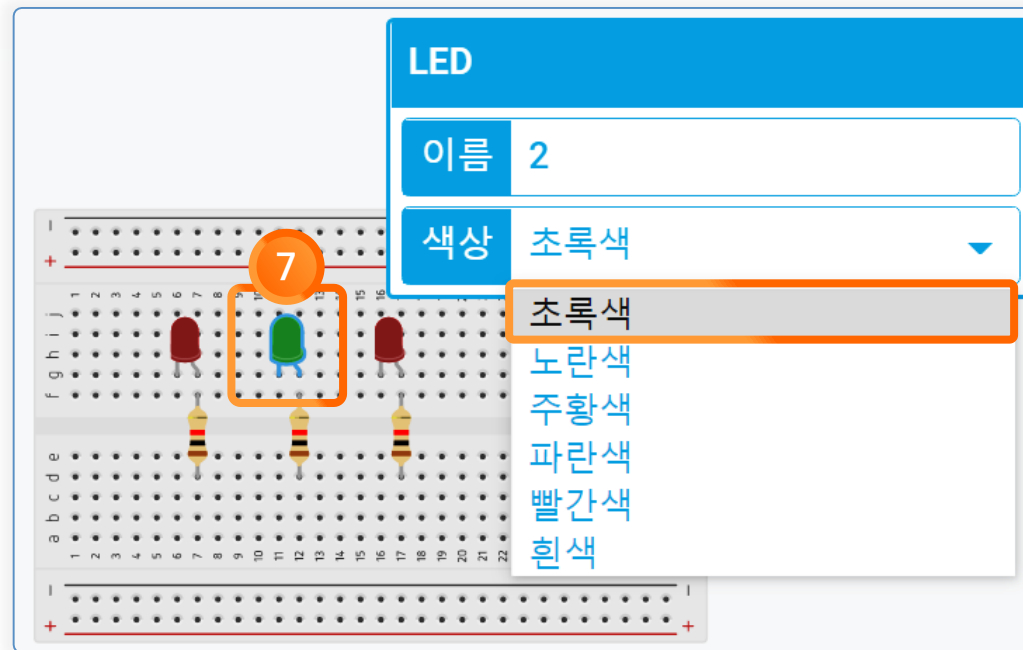
[사전준비 - 하드웨어 연결]

- ⑥ 계속해서 [LED]와 [레지스터(저항)]을 추가하여 전체 3개의 LED와 레지스터를 브레드 보드에 삽입



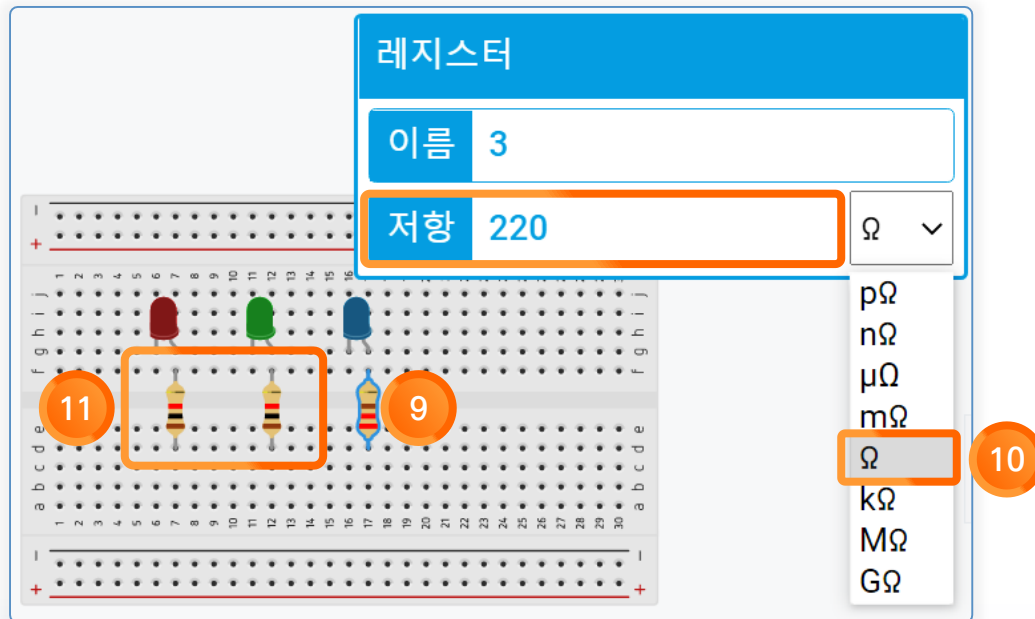
[사전준비 - LED 색상 변경]

- ⑦ 브레드 보드에 있는 두 번째 LED를 클릭하여 색상을 초록색(GREEN)으로 수정
- ⑧ 나머지 LED도 파란색(BLUE)으로 수정



[사전준비 - 저항 값 변경]

- ⑨ 브레드 보드의 오른쪽 첫 번째 레지스터(저항) 클릭
- ⑩ 단위를 k Ω 에서 “ Ω ”(옴)으로 변경 후 저항 값을 “220”으로 수정
- ⑪ 나머지 저항도 같은 방법으로 모두 수정

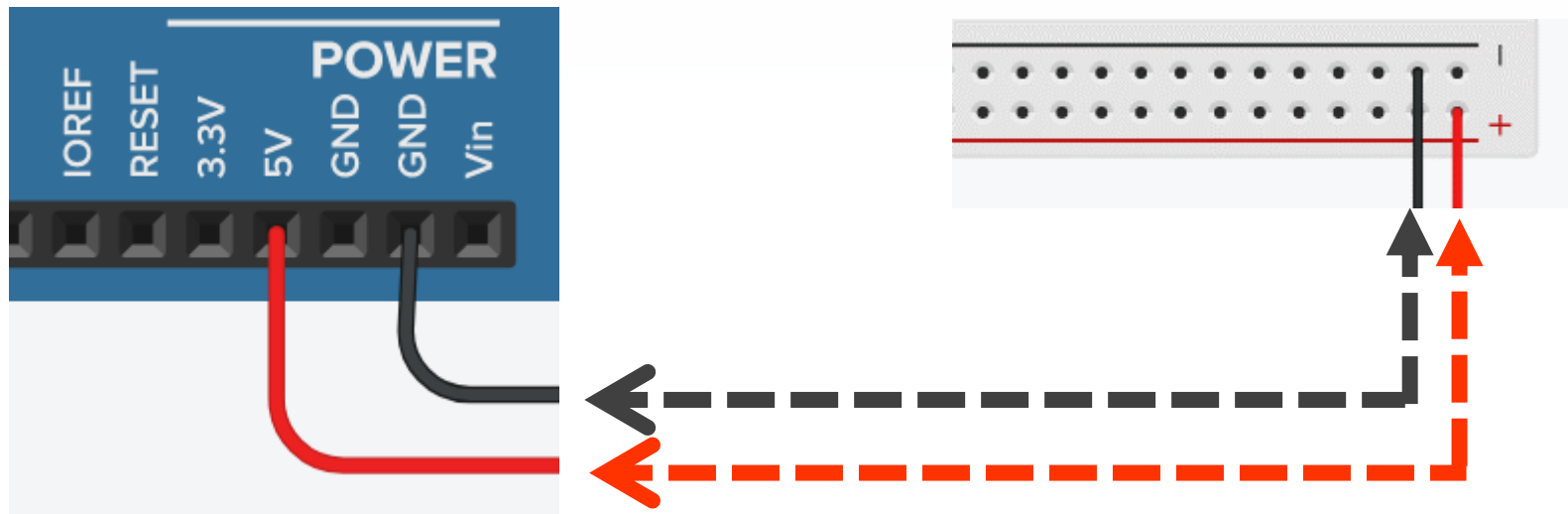


- 저항(Resistor) 값이 크면(k Ω 이상)
LED의 불빛이 흐리거나 켜지지 않음.
권장 사항은 220 Ω (옴)



[사전준비 - 배선하기]

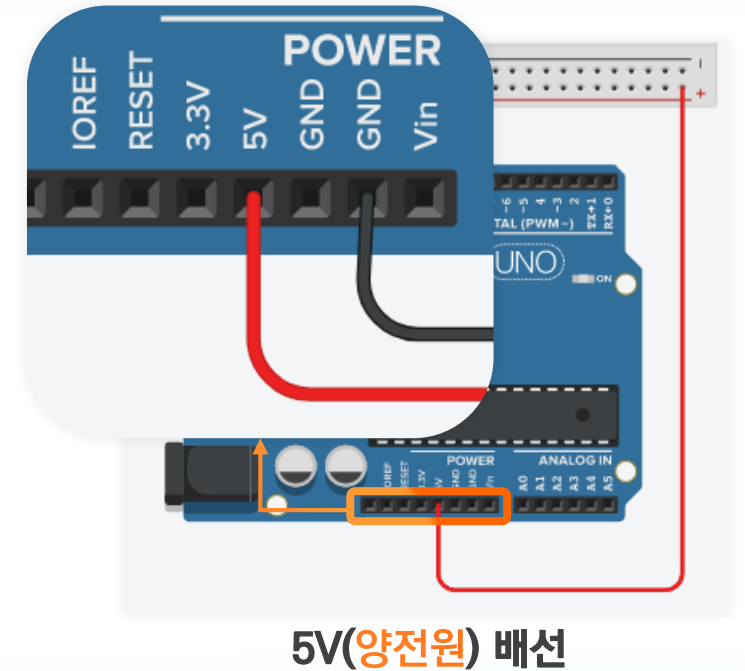
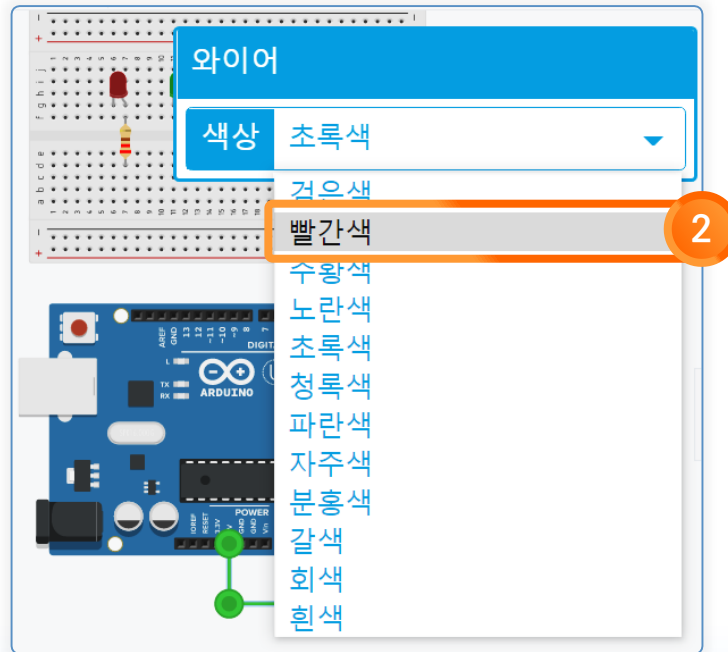
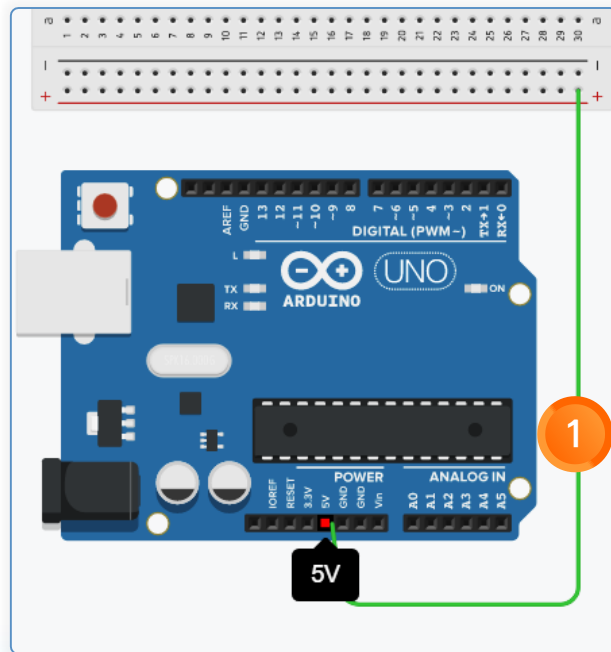
아두이노의 전원을 브레드 보드에서 공용으로 사용하기 위해 아두이노의 전원 단자와 브레드 보드의 “+,-” 표시 가로줄과 마우스를 드래그하여 와이어를 연결



와이어(배선)
: 양(+5V)전원은 빨간색, 음전원(-,GND)은 검은색 권장

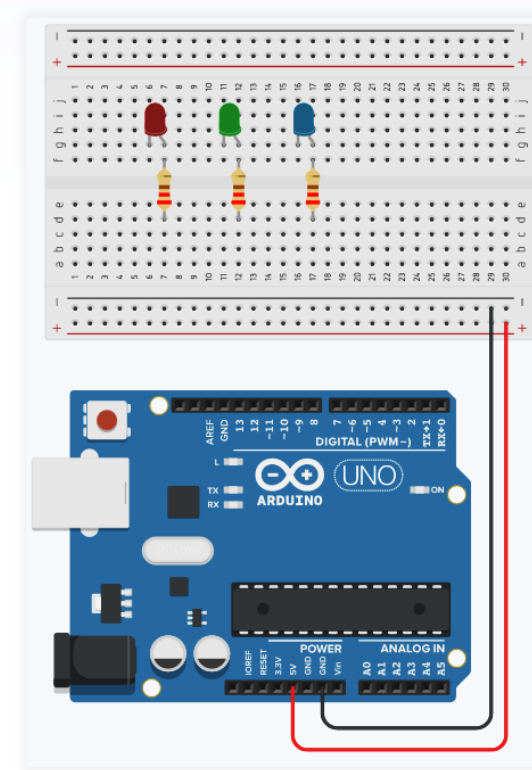
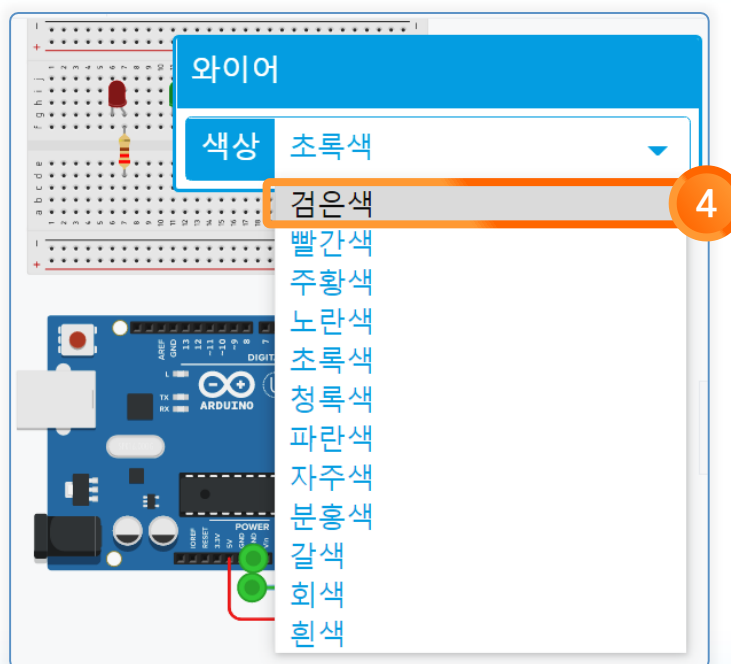
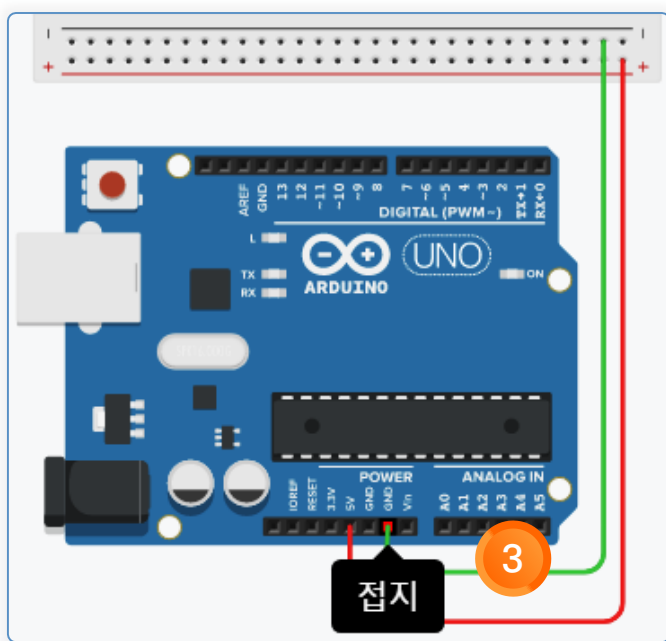
[사전준비 - 배선하기]

- 1 아두이노의 “5V” 단자와 브레드 보드의 “+” 표시 가로줄과 드래그하여 와이어를 연결
- 2 와이어 색상은 양극(+)전원이므로 빨간색으로 색상 변경



[사전준비 - 배선하기]

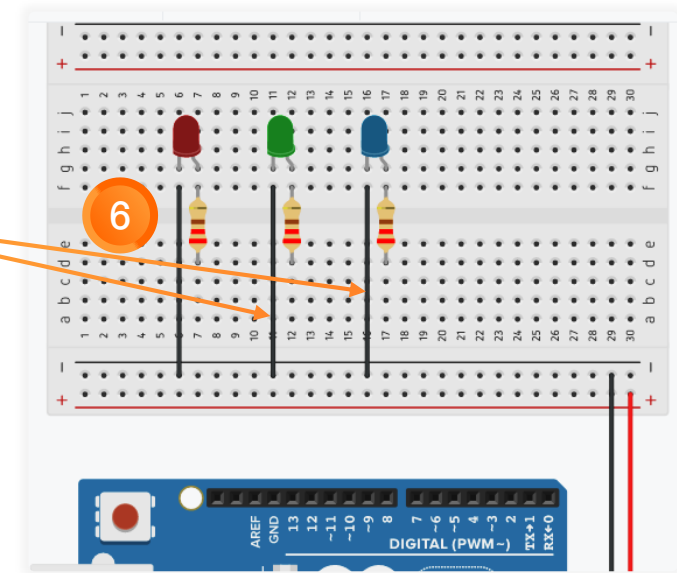
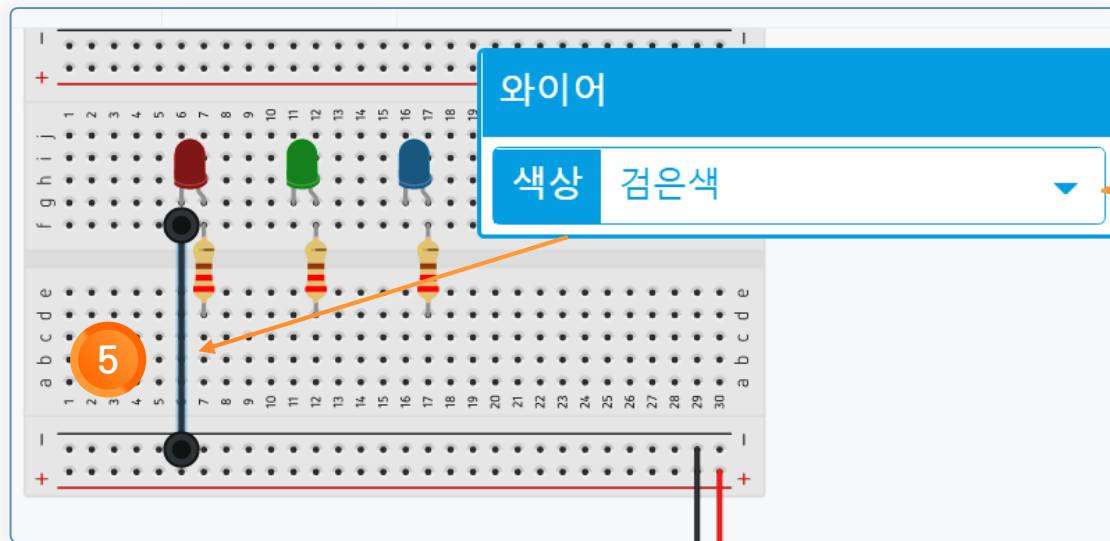
- ③ 브레드 보드의 “-” 표시 가로줄과 마우스를 드래그하여 아두이노 “GND(접지)” 단자와 와이어를 연결
- ④ 와이어 색상은 음극(-)이므로 검은색으로 색상 변경



전원 배선 완성

[사전준비 - 배선하기]

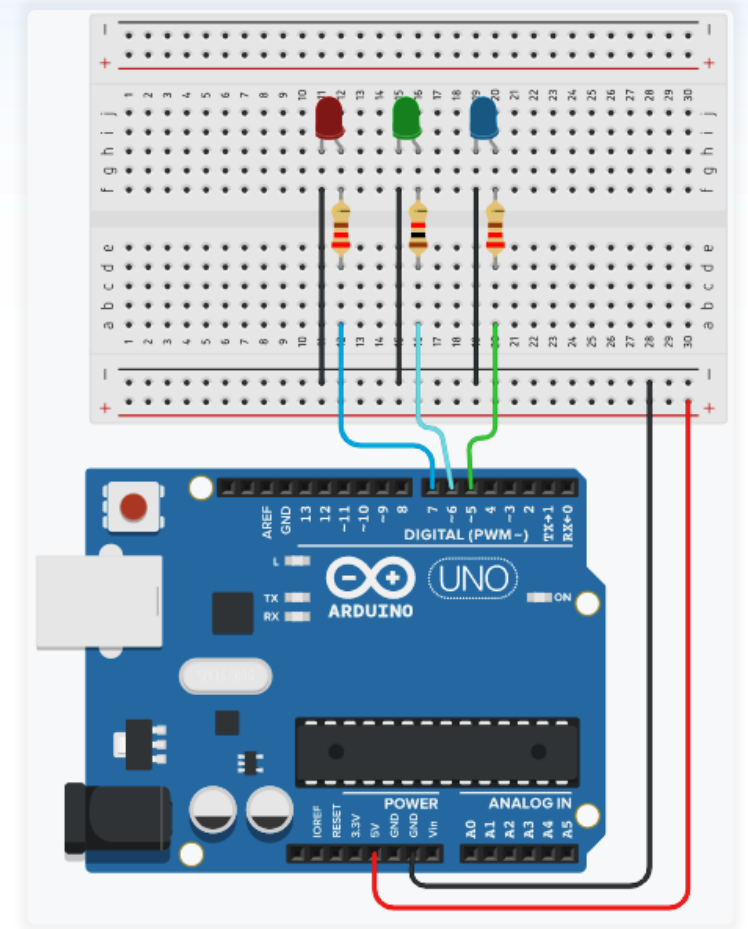
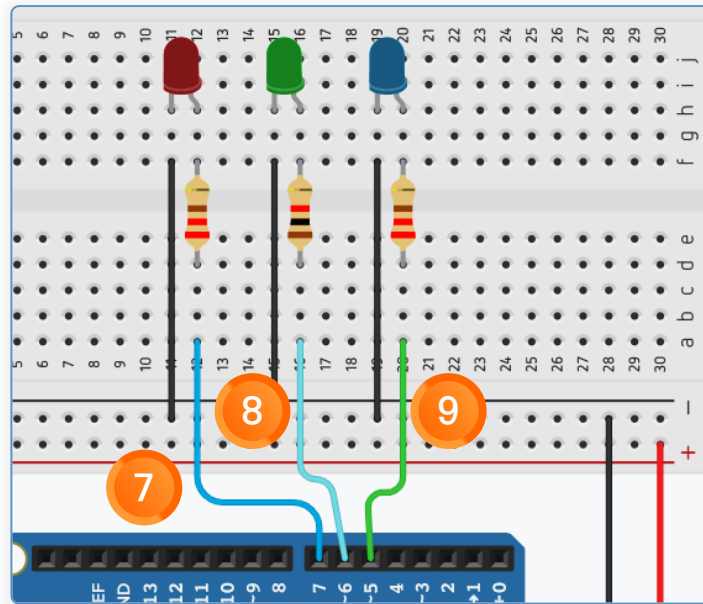
- ⑤ **빨간색** LED의 음극과 브레드 보드의 “-”표시 단자를 연결하고 와이어 색상은 검은색으로 수정
- ⑥ 마찬가지로 나머지 **초록색 LED**, **파란색 LED**의 음극 단자와 브레드 보드의 “-” 공통 단자를 연결하고 와이어 색상을 검은색으로 수정



LED 접지선 연결

[사전준비 - 배선하기]

- ⑦ 빨간색 LED와 연결된 저항의 반대쪽을 아두이노의 7번 포트에 연결
- ⑧ 초록색 LED와 연결된 저항의 반대쪽을 아두이노의 6번 포트에 연결
- ⑨ 파란색 LED와 연결된 저항의 반대쪽을 아두이노의 5번 포트에 연결

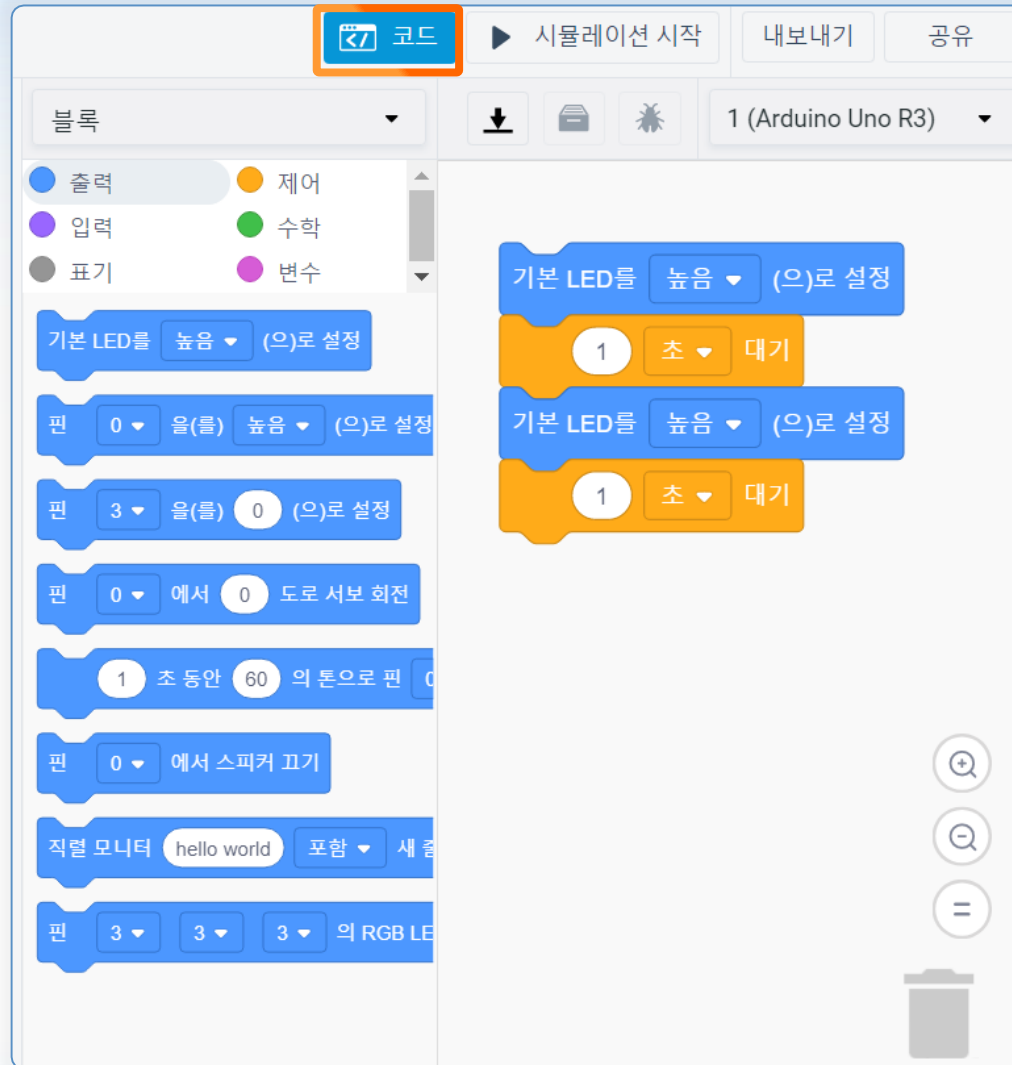


배선 연결 완성



1. 스크립트 작성

- 화면 상단의 코드 클릭





1. 스크립트 작성

- 기본으로 제공되는 블록을 휴지통으로 드래그하여 삭제





1. 스크립트 작성

- 출력 블록 중 **핀 0** 을(를) **높음** (으)로 설정 블록을 작업 영역으로 드래그





1. 스크립트 작성

- 핀 번호를 5로 지정, **높음**으로 설정





1. 스크립트 작성

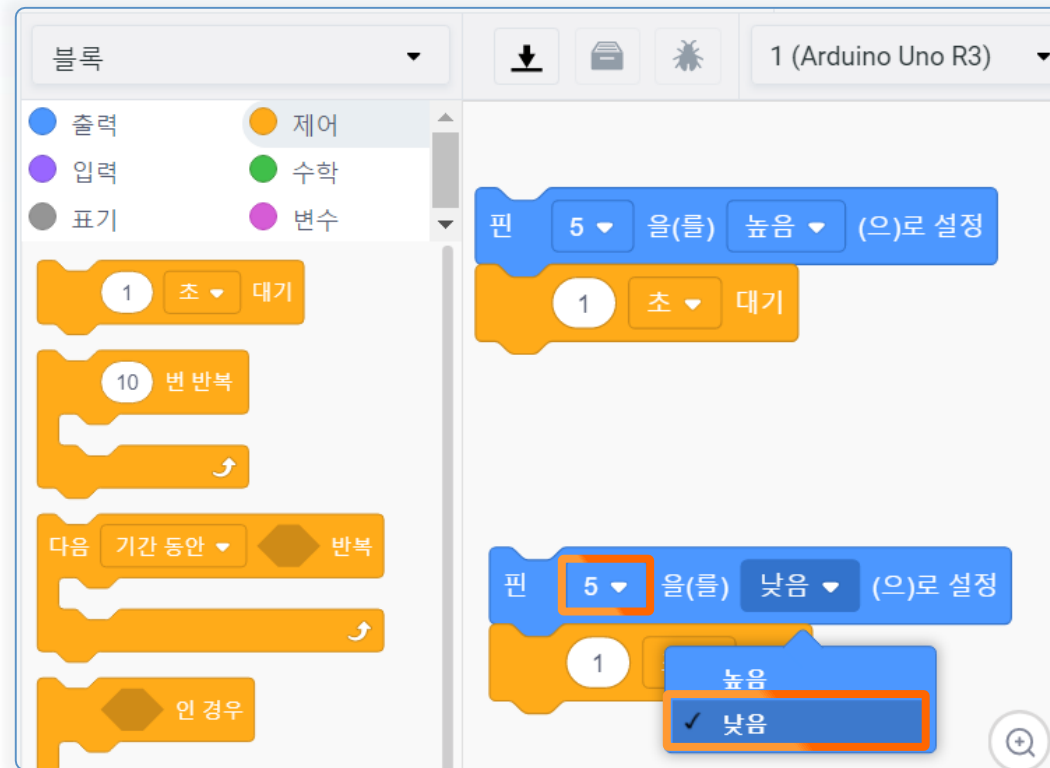
- 제어 블록 중 **1 초 대기**를 드래그 하고 **핀 5 을(를) 높음 (으)로 설정** 블록과 결합






1. 스크립트 작성

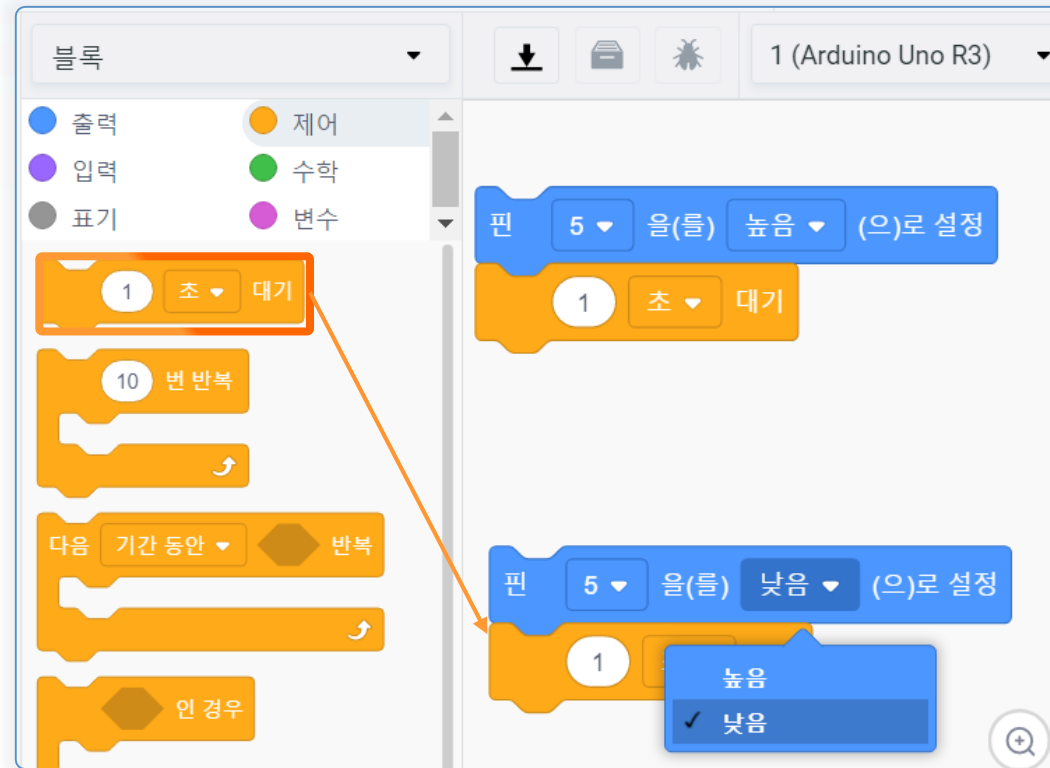
- 핀 0 을(를) 높음 (으)로 설정 블록을 작업 영역으로 드래그. 핀 번호를 5로 지정, 높음을 [낮음]으로 지정





1. 스크립트 작성

- 를 드래그하여 1의 블록과 결합





1. 스크립트 작성

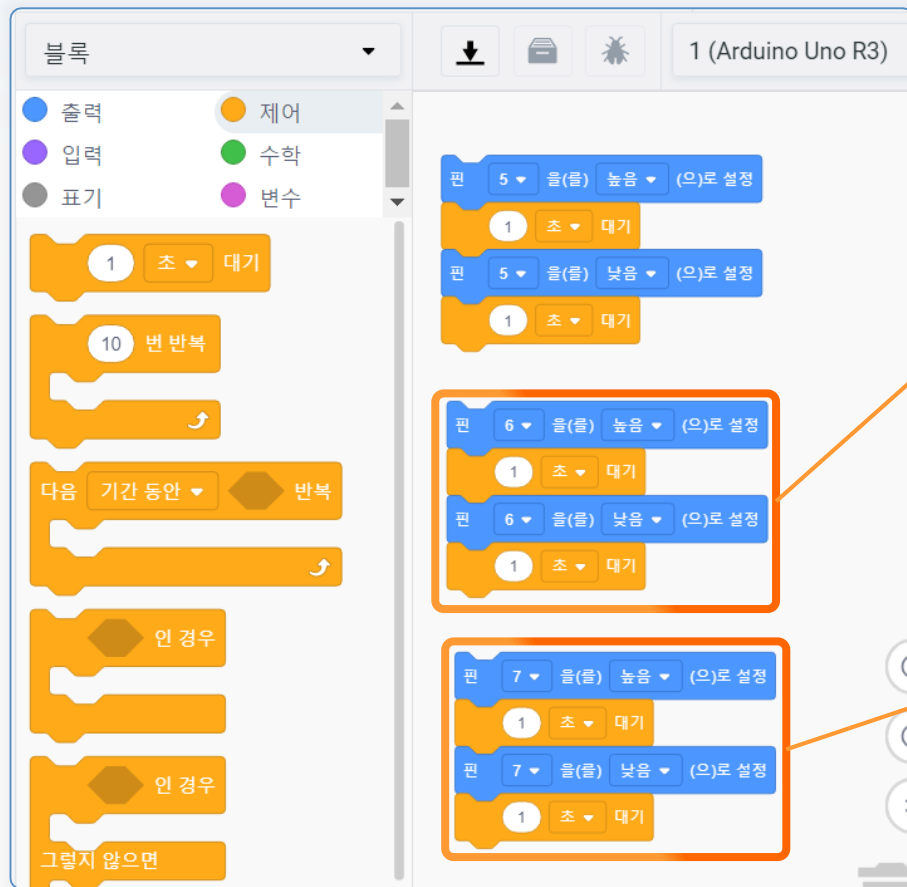
- 5번 핀에 대한 블록을 모두 결합





1. 스크립트 작성

- 동일한 방법으로 반복하여 6, 7번 포트에 대한 출력 및 제어 블록을 추가하여 프로그래밍 완성



6번 핀 LED에 대한
블록코딩



7번 핀 LED에 대한
블록코딩



1. 스크립트 작성

- 하드웨어 구성 및 전체 스크립트 완성 화면(블록 기반 언어 - 스크래치)

The screenshot displays the Scratch IDE interface with an Arduino Uno R3 hardware setup on the left. The script area on the right contains a sequence of blocks for controlling three LEDs. The script is organized into three distinct sections, each corresponding to a different LED (pins 5, 6, and 7). Each section consists of a '핀' (Pin) block to set the pin mode to '출력' (Output), followed by a '높음' (High) or '낮음' (Low) block to set the pin state, and a '1 초' (1 second) delay block. The script is as follows:

```

핀 5 을(를) 높음 (으)로 설정
1 초 대기
핀 5 을(를) 낮음 (으)로 설정
1 초 대기

핀 6 을(를) 높음 (으)로 설정
1 초 대기
핀 6 을(를) 낮음 (으)로 설정
1 초 대기

핀 7 을(를) 높음 (으)로 설정
1 초 대기
핀 7 을(를) 낮음 (으)로 설정
1 초 대기
    
```



1. 스크립트 작성

- 하드웨어 구성 및 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

The screenshot displays the Tinkercad IDE interface. On the left, a breadboard circuit is shown with an Arduino Uno R3 connected to three LEDs. The central workspace contains a block-based script for the Arduino. The script includes the following blocks:

- 핀 5 을(를) 높음 (으)로 설정** (Set pin 5 to HIGH)
- 1 초 대기** (Wait 1 second)
- 핀 5 을(를) 낮음 (으)로 설정** (Set pin 5 to LOW)
- 1 초 대기** (Wait 1 second)
- 핀 6 을(를) 높음 (으)로 설정** (Set pin 6 to HIGH)
- 1 초 대기** (Wait 1 second)
- 핀 6 을(를) 낮음 (으)로 설정** (Set pin 6 to LOW)
- 1 초 대기** (Wait 1 second)
- 핀 7 을(를) 높음 (으)로 설정** (Set pin 7 to HIGH)
- 1 초 대기** (Wait 1 second)

On the right, the code editor shows the corresponding C++ code for the Arduino Uno R3:

```

1 void setup()
2 {
3   pinMode(5, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(7, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(5, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(5, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14
15  digitalWrite(6, HIGH);
16  delay(1000); // Wait for 1000 millisecond(s)
17  digitalWrite(6, LOW);
18  delay(1000); // Wait for 1000 millisecond(s)
19
20  digitalWrite(7, HIGH);
21  delay(1000); // Wait for 1000 millisecond(s)
22  digitalWrite(7, LOW);
23  delay(1000); // Wait for 1000 millisecond(s)
24 }
    
```



1. 스크립트 작성

- 하드웨어 구성 및 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

```

1 void setup()
2 {
3   pinMode(5, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(7, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(5, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(5, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14
15  digitalWrite(6, HIGH);
16  delay(1000); // Wait for 1000 millisecond(s)
17  digitalWrite(6, LOW);
18  delay(1000); // Wait for 1000 millisecond(s)
19
20  digitalWrite(7, HIGH);
21  delay(1000); // Wait for 1000 millisecond(s)
22  digitalWrite(7, LOW);
23  delay(1000); // Wait for 1000 millisecond(s)
24 }

```

[설명]

setup()
pinMode(5, OUTPUT);

- ☞ 초기 값 설정
- ☞ 아두이노 5(6,7)번 핀은 출력모드

loop()
digitalWrite(5, HIGH);
delay(1000);

- ☞ 프로그램 반복
- ☞ 5(6,7)번 포트를 1초 동안 **HIGH**상태 출력

digitalWrite(5, LOW);
delay(1000);

- ☞ 5(6,7)번 포트를 1초 동안 **LOW**상태 출력
- ☞ 1초간 지연.(1000은 1,000ms, 즉 1초

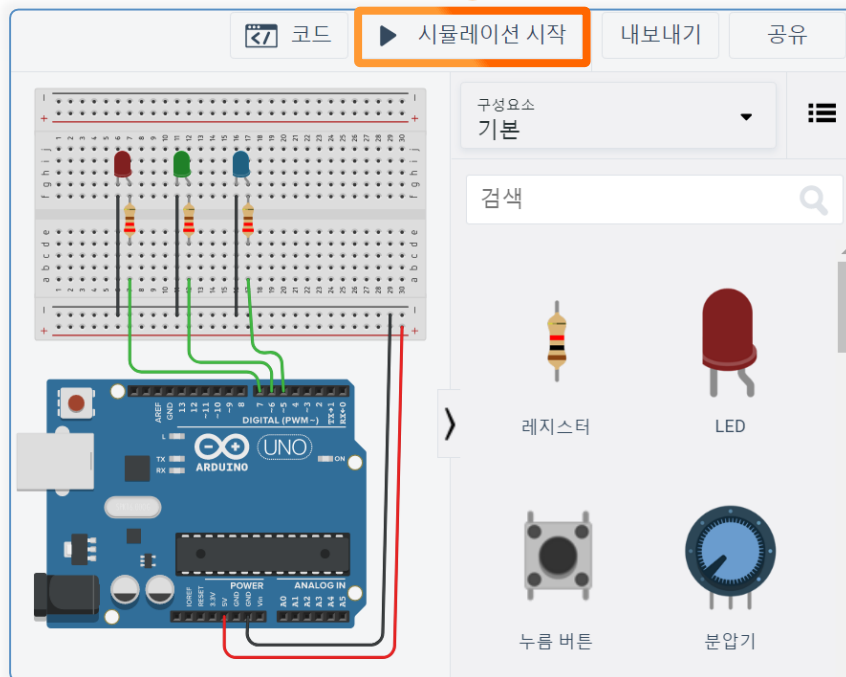
주석문 : // 로 시작하는 문장. 코딩에 대한 참고 문구. 실행되지 않음



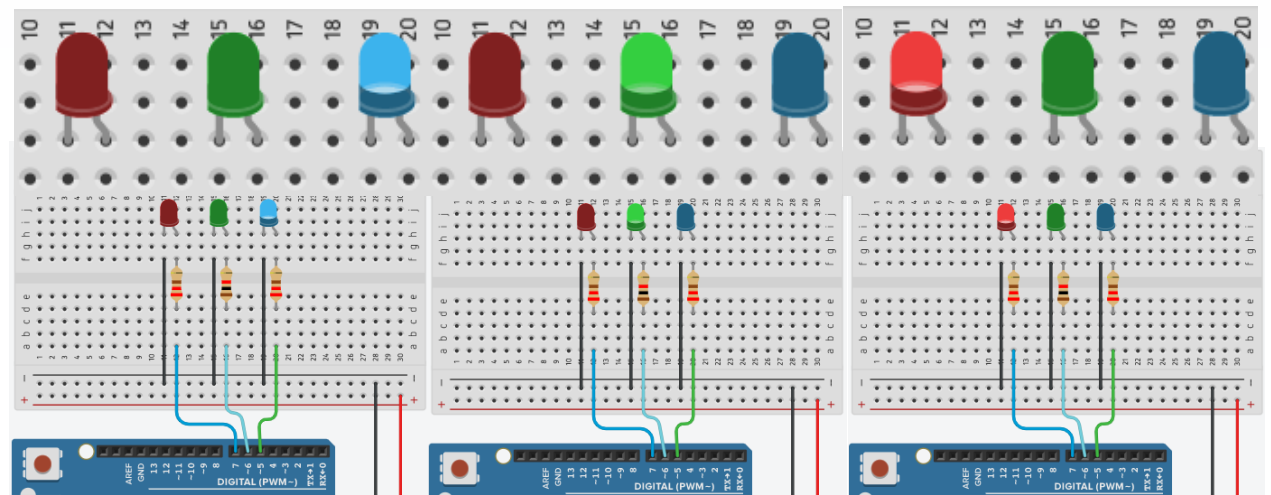
2. 스크립트 실행

- [시뮬레이션 시작] 메뉴 클릭하여 결과 확인
 - ▶ 3개의 LED가 1초 간격으로 각각 켜졌다(ON), 꺼지는(OFF) 반복 동작 확인

1



2



5번 포트에
연결된 파란색 LED
ON/OFF(1초 동안)

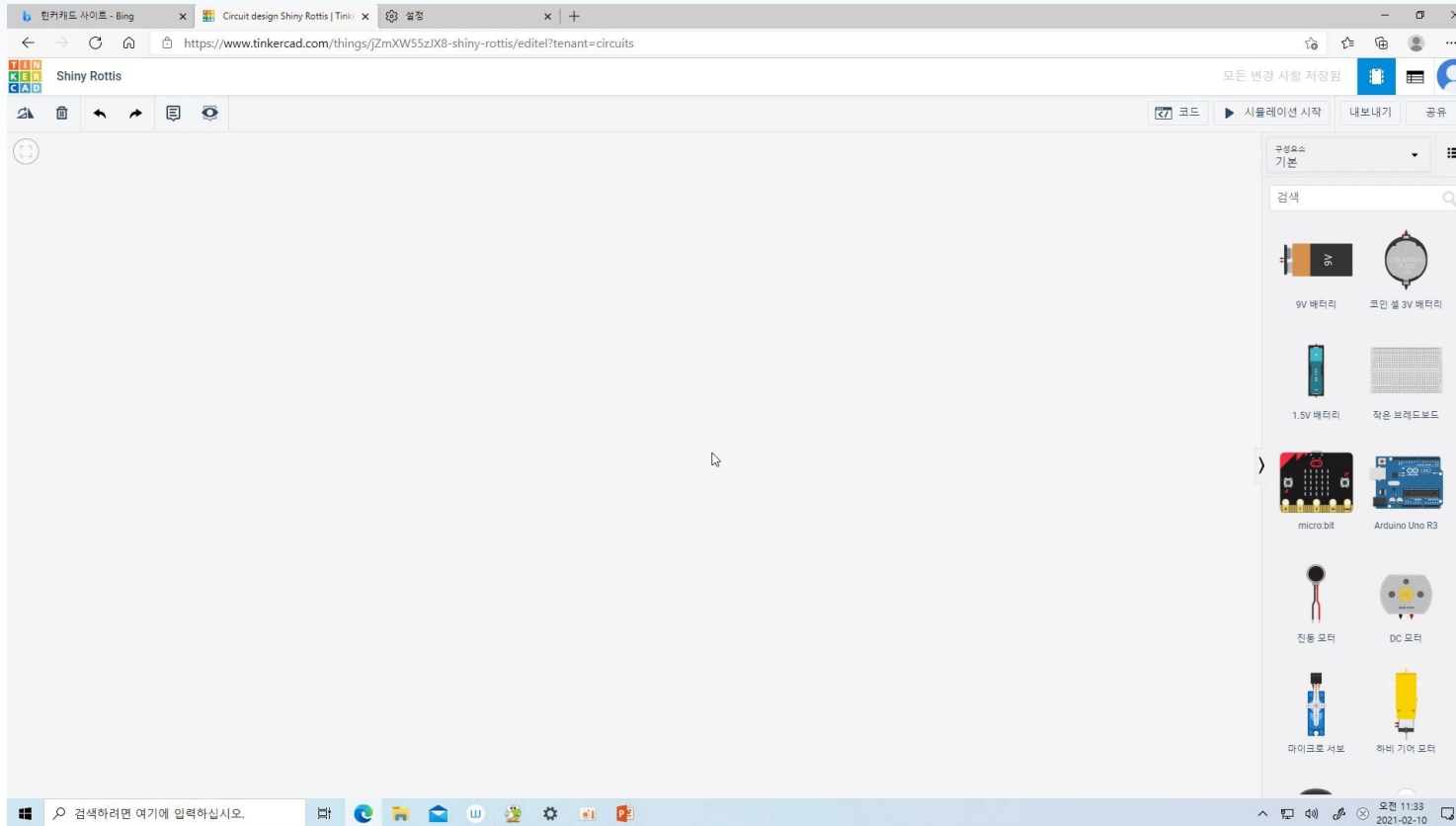
6번 포트에
연결된 녹색 LED
ON/OFF(1초 동안)

7번 포트에
연결된 빨간색 LED
ON/OFF(1초 동안)



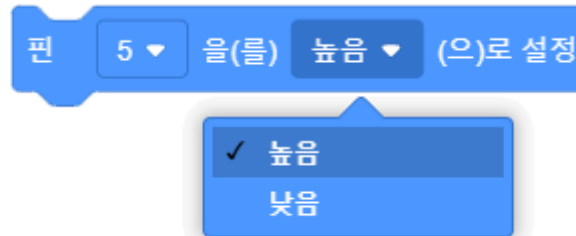
2. 스크립트 실행

- 실습영상(틴커캐드 서킷을 이용한 LED 제어 시뮬레이터)




학습정리

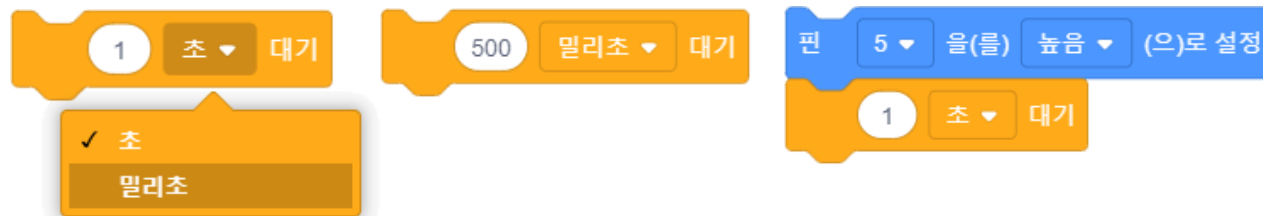
- ✓ 브레드보드(bread board)는 수 많은 홀(hole)들 중 공통 단자로 사용 할 수 있는데 가로로 연결된 홀들과 5개씩 세로로 연결된 홀들은 내부적으로 서로 연결되어 있다.
- ✓ LED On/Off를 위한 스크래치 블록 설정
 - ▶ 출력 - **핀** 0 **을(를)** **높음** **(으)로 설정** 블록에서 출력 단자 번호와 상태 (높음,낮음) 지정. “높음”은 전원(5V) 공급. 낮음은 전원을 차단(Off)



학습정리

✓ LED On/Off 시간 제어를 위한 스크래치 블록 설정

- ▶ 제어 -  블록에서 **대기 시간**을 지정.
대기시간은 해당 코딩의 수행 시간. 초 또는 밀리초로 설정. 1000밀리초는 1초



✓ On/Off제어 : 설정된 시간 간격으로 5번 핀에 연결된 LED가 켜졌다(On), 꺼졌다(Off)를 반복



1초 간격으로 On/ Off



0.5초(500밀리초) 간격으로 On/ Off

본 수업자료는 저작권법 제 25조 2항에 따라
학교 수업을 목적으로 이용되었으므로,
본 수업자료를 외부에 공개, 게시하는 것을 금지하며,
이를 위반하는 경우 저작권 침해로서 관련법에 따라 처벌될 수 있습니다.