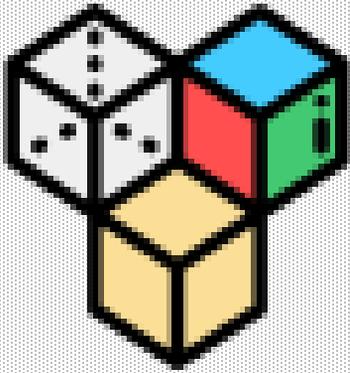




Ardublock





Module 2: 변수와 반복문



Ardublock x 허밍버드 키트

아두블럭의 메인 반복문은 LED를 깜박거리는 동작을 표현 할 때
좋습니다. 다른 반복문을 통해서 서서히 밝아지고
서서히 어두워지는 동작을 표현해봅시다!

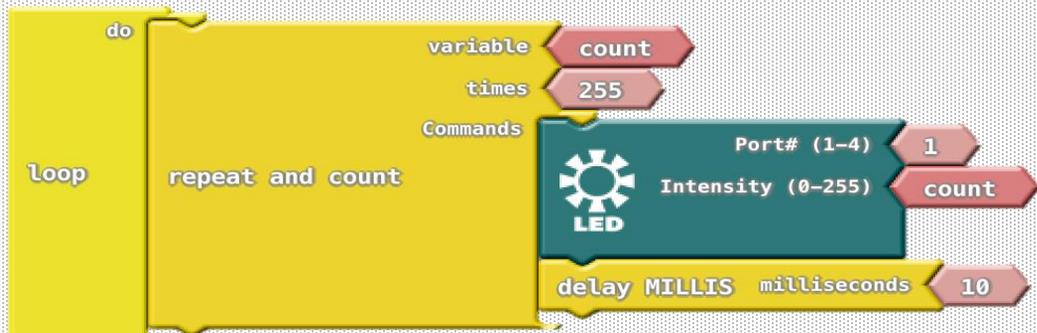


repeat and count 반복블록입니다.

제어메뉴에서 블록을 찾을 수 있습니다.

이 반복블록은 변수를 사용합니다. 변수는 값을 나타내는 이름을
정할 수 있습니다. 변수의 이름을 사용하는 용도에 맞게 정해주세요. 이
제, 진한 분홍색 육각형을 클릭하고 변수 이름을 'count'로
변경해 주세요. repeat and count 블록은 commands의 공간의
블록들을 times 매개변수의 값만큼 반복해줍니다.
여기서 times는 반복횟수를 의미합니다

Repeat and count 블록을 사용한 예제가 아래에 나와있습니다. 이
프로그램을 실행시켜 보세요. LED가 서서히 밝아지는 동작을 확인 할
수 있나요? 블록을 복사하고 싶다면 오른쪽 마우스를
클릭한 다음 **클론 (복제)**를 눌러주세요.



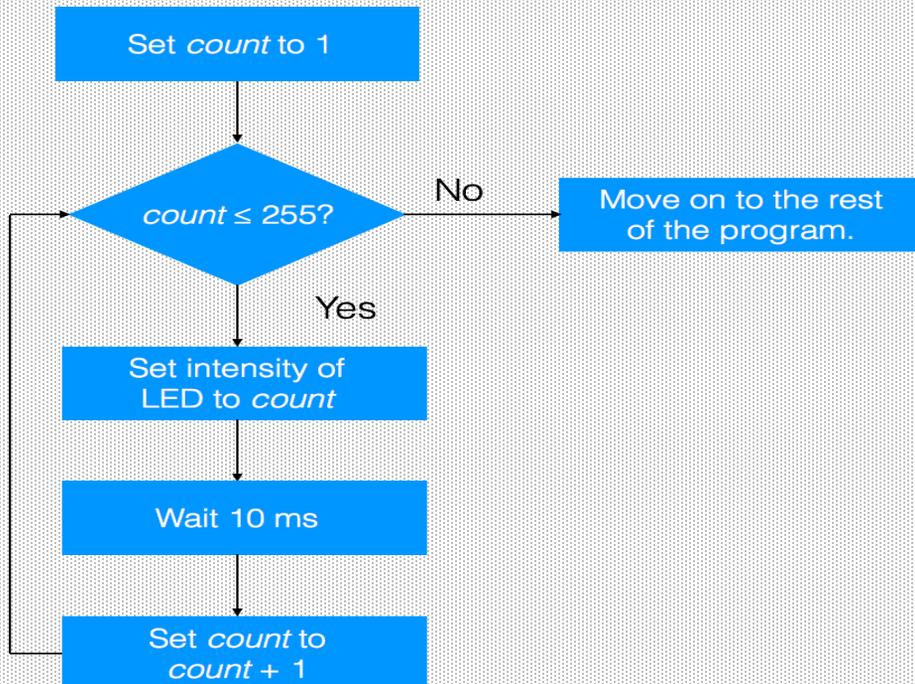
Ardublock x 허밍버드 키트

repeat and count 반복블록은 변수를 사용합니다.

이 변수는 반복문 안의 명령어들이 몇 번 실행 되었는지 알아 보기 위한 변수입니다. 이 반복문의 논리는 아래의 순서도에 나와있습니다.

프로그램이 **repeat and count 반복블록**에 도달하면 변수의 개수를 1로 설정한 다음 count 값이 times의 매개 변수보다 작거나 같은지 확인합니다. 여기서 times의 값은 255입니다. 그럼 commands 공간의 블록들이 실행이 되고 count 값이 1씩 증가합니다. 그런 다음에 다시 count가 255 보다 작거나 같은지 다시 확인을 합니다.

Count가 255보다 작거나 같지 않으면 프로그램은 repeat and count 블록 아래의 블록으로 이동합니다.



Ardublock x 허밍버드 키트

```
for (_ABVAR_1_count= 1; _ABVAR_1_count<= ( 255 ); _ABVAR_1_count++ )
{
  bird.setLED(1,_ABVAR_1_count);
  delay( 10 );
}
```

위에 표시된 블록들은 Arduino 코드를 생성 할 수 있습니다.

변수 이름 개수는 `_ABVAR_1_count`로 변환됩니다.

이 for 루프는 텍스트 기반 프로그래밍에서
매우 일반적인 구조입니다.

맨 위 줄에는 `_ABVAR_1_count`가 1에서 시작하고
`_ABVAR_1_count`가 255보다 작거나 같으면 루프가
반복 되어야 한다고 표시됩니다.

`_ABVAR_1_count ++`는 루프를 통과할 때 마다
`_ABVAR_1_count` 변수가 1 씩 증가해야 함을 의미합니다.
루프가 반복하는 명령은 `{ }` 괄호 안에 있습니다

Ardublock x 허밍버드 키트

Exercise 2.1

두 개의 LED를 연결하고 하나의 LED가 증가할 때 다른 LED 강도가 감소하도록 프로그램을 만들어 봅시다. 연산자 메뉴에서 빼기 블록을 사용해 보세요!



Exercise 2.2

삼색 LED가 빨간색에서 파란색으로 반복적으로 전환되도록 프로그램을 작성하십시오.