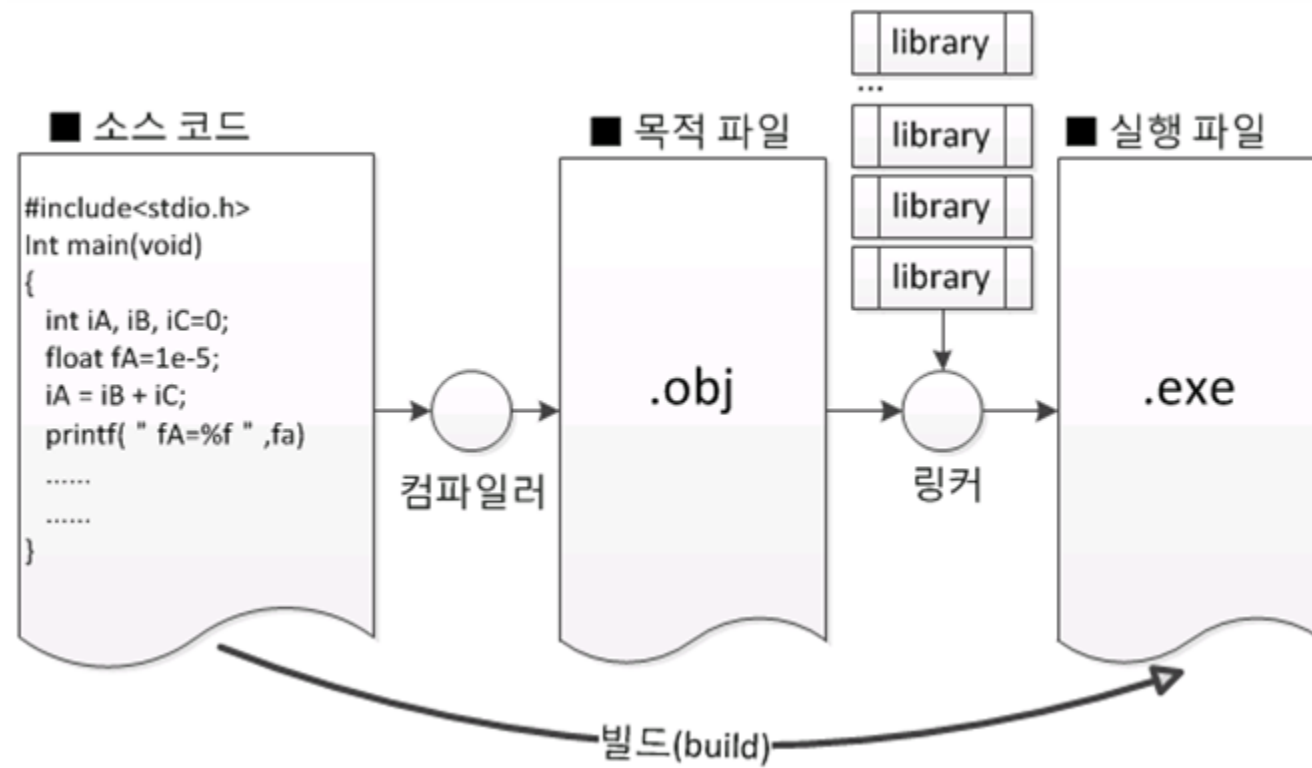




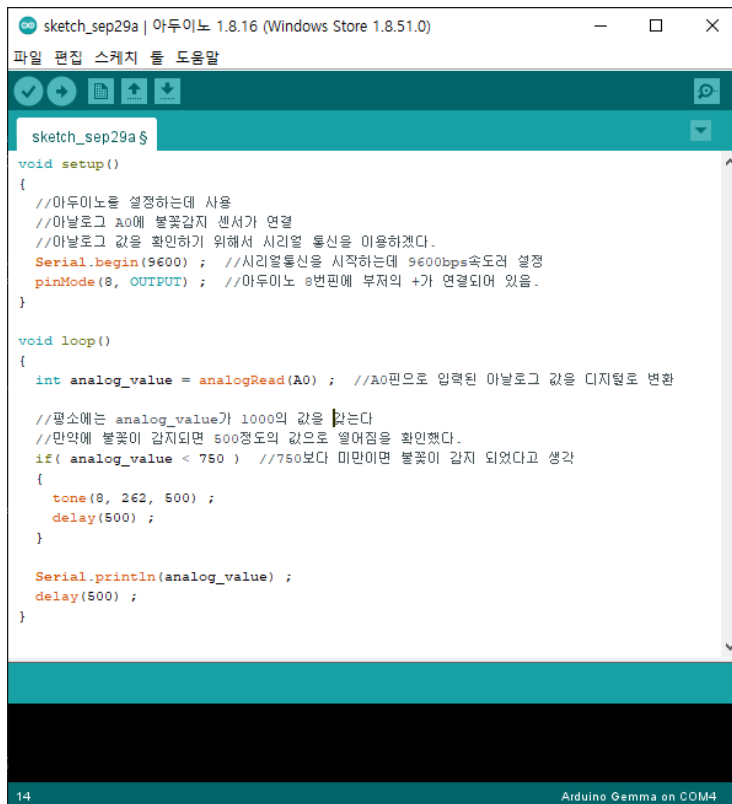
C언어  
그리고 영상처리까지

# 아두이노 SW 프로그래밍 과정



# 아두이노 프로그래밍 과정

- 소스코드 작성



```
sketch_sep29a | 아두이노 1.8.16 (Windows Store 1.8.51.0)
파일 편집 스케치 툴 도움말

sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼 통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}

14 Arduino Gemma on COM4
```

# 아두이노 프로그래밍 과정

- 컴파일



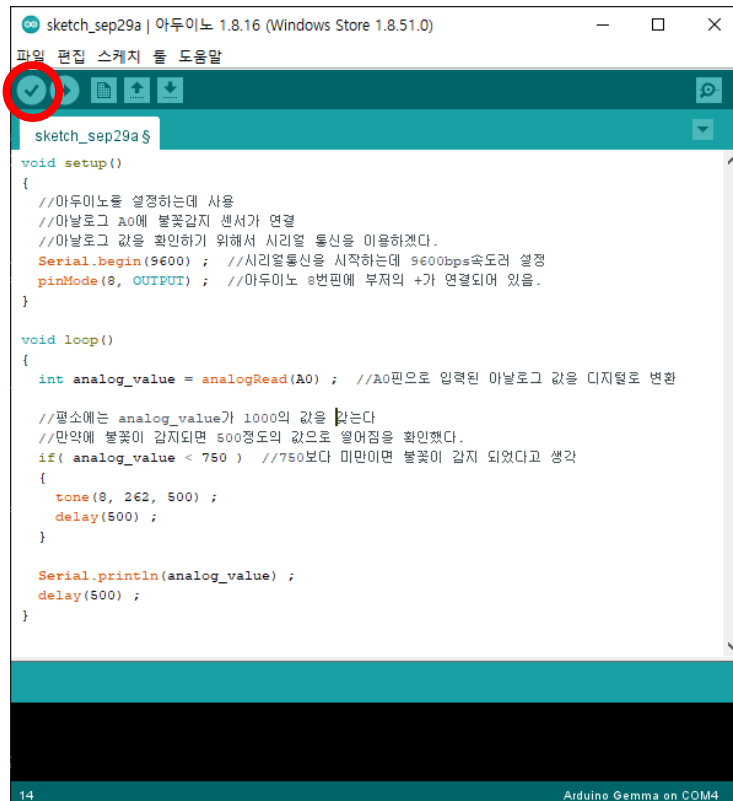
```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

14 Arduino Gemma on COM4



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

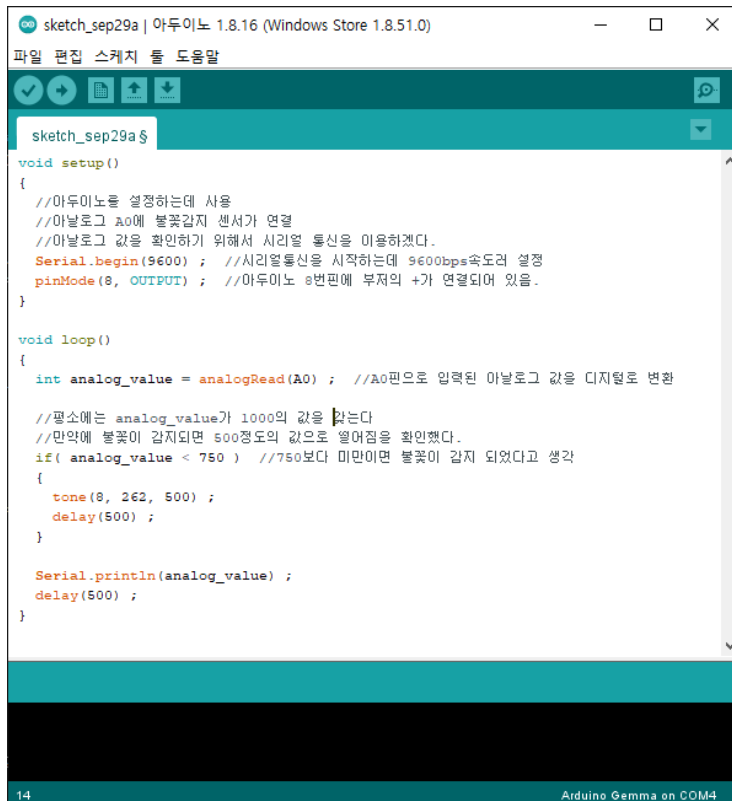
  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

14 Arduino Gemma on COM4

# 아두이노 프로그래밍 과정

## • 업로드 & 실행

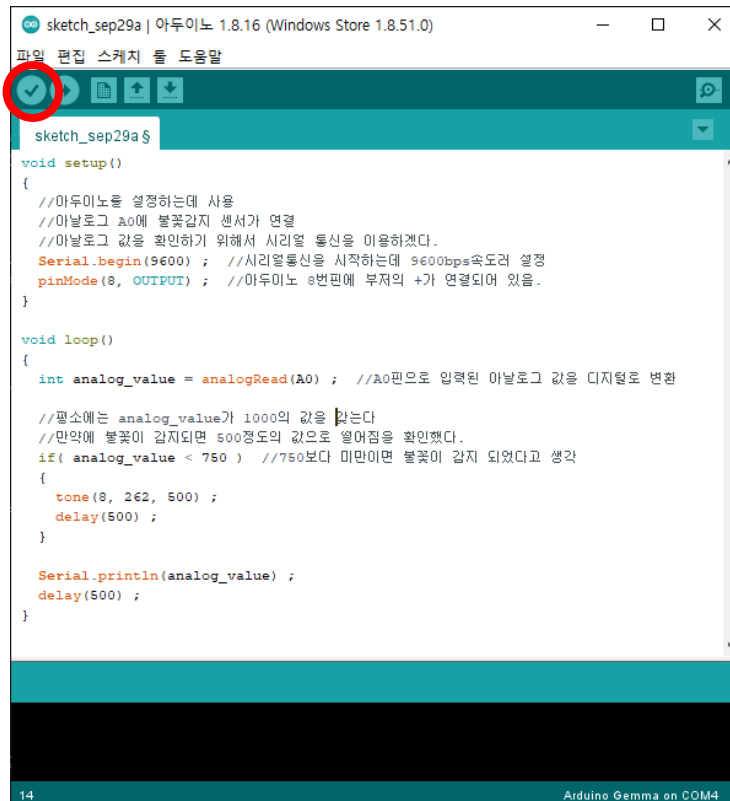


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

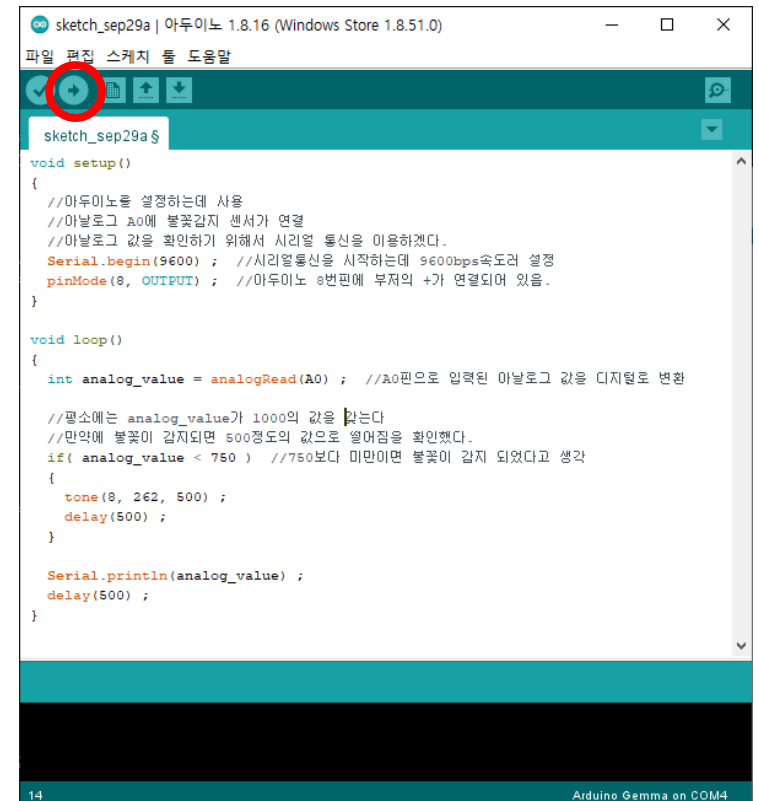


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

# 시리얼 통신

- 아두이노 <-> PC와 정보를 주고 받기 위한 가장 편리한 방법

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Hello, World");
  delay(1000);
}
```

예제 : 5-1

# 아두이노 C언어 기본 함수, 기본 문법

- setup 함수

- 아두이노 보드에 전원이 인가되거나 리셋버튼이 눌리면 처음 실행될때 처음 호출되는 함수로서 아두이노 혹은 주변장치의 초기화용 코드를 작성한다.

```
void setup()
```

```
{  
}
```

```
void loop()
```

```
{  
}
```

- loop 함수

- setup함수 호출 이후에 주기적으로 반복 호출하는 함수로서 아두이노가 처리해야 하는 메인 제어 코드를 작성한다.

```
void setup()
```

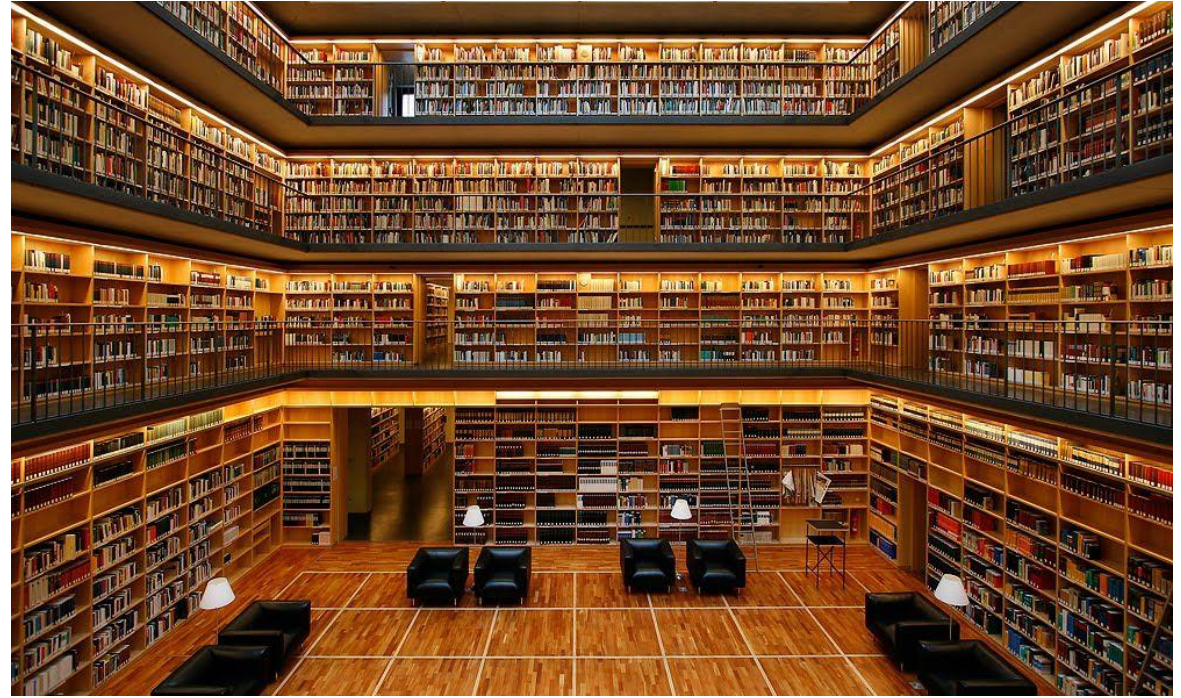
```
{  
}
```

```
void loop()
```

```
{  
}
```

# 변수와 상수

- 변수란?
  - 바구니(메모리)에 임시로 값을 저장
  - 변수는 고유 주소와 값을 갖는다.
  - 변수에는 여러 타입이 있으며 타입에 따라 변수를 담는 바구니(메모리)의 사이즈가 다름



도서관이라는 메모리에 데이터(책)가 저장되어있다.  
원하는 데이터(책)을 어떻게 찾을 수 있을까?



# 변수와 상수

- 변수
  - 정수형(int)
  - 소수점형(float, double)
  - True/False 참거짓형(bool)
  - 문자형(char)
- 상수
  - 변수와 같지만 초기화에 저장된 값을 수정할 수 없다.
- 지역변수
  - {...} 괄호 안에서만 사용하는 변수
- 전역변수
  - 코드 전체에서 사용하는 변수

# 변수와 상수

- 정수형 변수

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int value = 10 ;

  Serial.print("value = ");
  Serial.println(value);

  delay(1000);
}
```

예제 5-2

# 변수와 상수

- 소수점형 변수

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    float value = 1.12 ;

    Serial.print("value = ");
    Serial.println(value);

    delay(1000);
}
```

예제 : 5-3

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    double value = 1.123456 ;

    Serial.print("value = ");
    Serial.println(value);

    delay(1000);
}
```

예제 : 5-4

# 변수와 상수

- True/False 참거짓형 변수

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  bool value = 1 ;      //or 0

  Serial.print("value = ");
  Serial.println(value);

  delay(1000);
}
```

예제 : 5-5

# 변수와 상수

- char 문자형 변수

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    //char value = 41 ;
    char value = 'A' ;

    Serial.print("value = ");
    Serial.println(value);

    delay(1000);
}
```

DEC	HEX	OCT	Char	DEC	HEX	OCT	Char	DEC	HEX	OCT	Char
0	00	000	Ctrl-@ NUL	43	2B	053	+	86	56	126	V
1	01	001	Ctrl-A SOH	44	2C	054	,	87	57	127	W
2	02	002	Ctrl-B STX	45	2D	055	-	88	58	130	X
3	03	003	Ctrl-C ETX	46	2E	056	.	89	59	131	Y
4	04	004	Ctrl-D EOT	47	2F	057	/	90	5A	132	Z
5	05	005	Ctrl-E ENQ	48	30	060	0	91	5B	133	[
6	06	006	Ctrl-F ACK	49	31	061	1	92	5C	134	\
7	07	007	Ctrl-G BEL	50	32	062	2	93	5D	135	]
8	08	010	Ctrl-H BS	51	33	063	3	94	5E	136	^
9	09	011	Ctrl-I HT	52	34	064	4	95	5F	137	_
10	0A	012	Ctrl-J LF	53	35	065	5	96	60	140	`
11	0B	013	Ctrl-K VT	54	36	066	6	97	61	141	a
12	0C	014	Ctrl-L FF	55	37	067	7	98	62	142	b
13	0D	015	Ctrl-M CR	56	38	070	8	99	63	143	c
14	0E	016	Ctrl-N SO	57	39	071	9	100	64	144	d
15	0F	017	Ctrl-O SI	58	3A	072	:	101	65	145	e
16	10	020	Ctrl-P DLE	59	3B	073	;	102	66	146	f
17	11	021	Ctrl-Q DC1	60	3C	074	<	103	67	147	g
18	12	022	Ctrl-R DC2	61	3D	075	=	104	68	150	h
19	13	023	Ctrl-S DC3	62	3E	076	>	105	69	151	i
20	14	024	Ctrl-T DC4	63	3F	077	?	106	6A	152	j
21	15	025	Ctrl-U NAK	64	40	100	@	107	6B	153	k
22	16	026	Ctrl-V SYN	65	41	101	A	108	6C	154	l
23	17	027	Ctrl-W ETB	66	42	102	B	109	6D	155	m
24	18	030	Ctrl-X CAN	67	43	103	C	110	6E	156	n
25	19	031	Ctrl-Y EM	68	44	104	D	111	6F	157	o
26	1A	032	Ctrl-Z SUB	69	45	105	E	112	70	160	p
27	1B	033	Ctrl-[ ESC	70	46	106	F	113	71	161	q
28	1C	034	Ctrl-\ FS	71	47	107	G	114	72	162	r
29	1D	035	Ctrl-] GS	72	48	110	H	115	73	163	s
30	1E	036	Ctrl-^ RS	73	49	111	I	116	74	164	t
31	1F	037	Ctrl_ US	74	4A	112	J	117	75	165	u
32	20	040	Space	75	4B	113	K	118	76	166	v
33	21	041	!	76	4C	114	L	119	77	167	w
34	22	042	"	77	4D	115	M	120	78	170	x
35	23	043	#	78	4E	116	N	121	79	171	y
36	24	044	\$	79	4F	117	O	122	7A	172	z
37	25	045	%	80	50	120	P	123	7B	173	{
38	26	046	&	81	51	121	Q	124	7C	174	
39	27	047	'	82	52	122	R	125	7D	175	}
40	28	050	(	83	53	123	S	126	7E	176	~
41	29	051	)	84	54	124	T	127	7F	177	DEL
42	2A	052	*	85	55	125	U				

# 배열

- 같은 타입(사이즈)의 연속된 변수(메모리)
- 문자와 문자열

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int value[3] ;
  value[0] = 10 ;
  value[1] = 12 ;
  value[2] = 31 ;

  Serial.print("value = ");
  Serial.println(value[0]);

  delay(1000);
}
```

예제 : 5-6

# 함수

- 기능을 하나의 블록으로 만들어서 필요시에 호출하여 사용
- 코드의 반복을 줄일 수 있음.

```
리턴형 함수이름(입력인자1, 입력인자2, ....)
{
}
```

```
void FunctionName(void)
{
}
```

```
void FunctionName(int a)
{
}
```

```
int FunctionName(int a)
{
    int b= a+1 ;
    return b ;
}
```

# 함수

- 내 이름을 출력하는 함수를 만들자

```
void PrintMyName(void)
{
    Serial.println("Juhong Park") ;
}
```

```
void PrintMyName(void)
{
    Serial.println("Juhong Park") ;
}

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    PrintMyName() ;
    delay(1000);
}
```



# 함수

- 2개의 정수를 입력 받아 덧셈하여(+) 그 결과값을 return하는 함수를 만들자.

```
int Add(int a, int b)
{
    int c = 0 ;
    c = a+b ;
    return c ;
}
```

```
int Add(int a, int b)
{
    int c = 0 ;
    c = a + b ;
    return c ;
}

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int add_value = Add(1, 2) ;
    Serial.println(add_value) ;
    delay(1000);
}
```

예제 : 5-8

# 함수

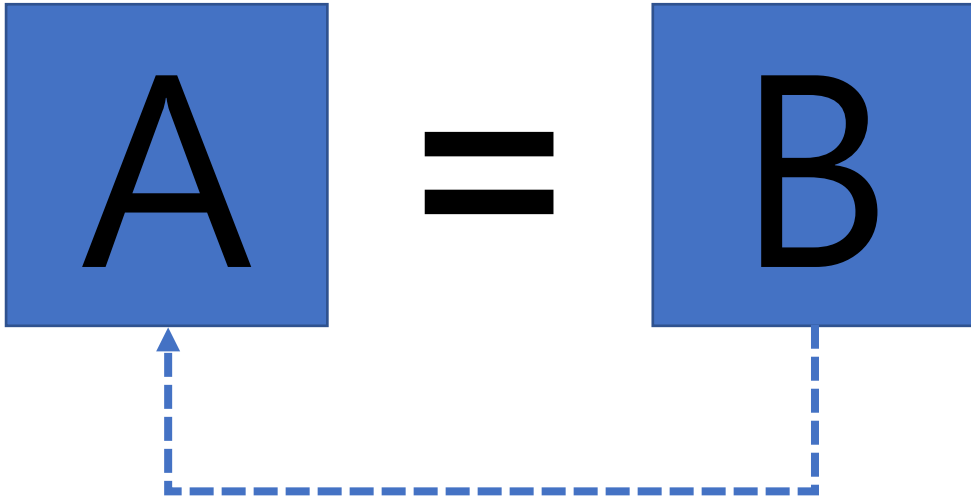
- 테스트: 2개의 소수점을 입력 받아 뺄셈하여(-) 그 결과값을 return하는 함수를 만들고 테스트 코드를 완성하시오.
- 4칙연산을 하는 함수들을 만들고 테스트 코드를 완성하시오.
  - 이름 : Add ,Sub, Mul, Div

# 연산

- 대입
  - =
- 산술연산
  - 더하기(+), 빼기(-), 곱하기(\*), 나누기(/), 증가(++), 감소(--)
- 조건
  - ==, !=, >, <, >=, <=

# 연산

- 대입
  - =



```
void setup()
{
  Serial.begin(9600);
}

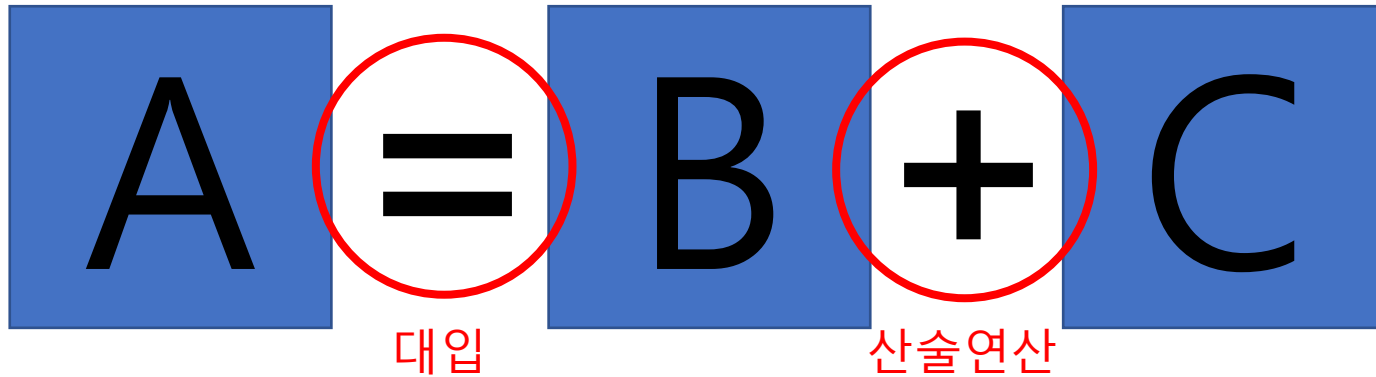
void loop()
{
  int a = 10 ;
  Serial.println(a) ;
  delay(1000);
}
```

예제 : 5-9

# 연산

- 산술연산
  - 더하기(+), 빼기(-), 곱하기(\*), 나누기(/)

예제 : 5-10



```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int a = 0 ;
  int b = 10 ;
  a = b + 1 ;
  Serial.println(a) ;
  delay(1000);
}
```

# 연산

- 산술연산
  - 증가(++), 감소(--)



예제 : 5-11

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int a = 0 ;
  a++ ;
  Serial.println(a) ;

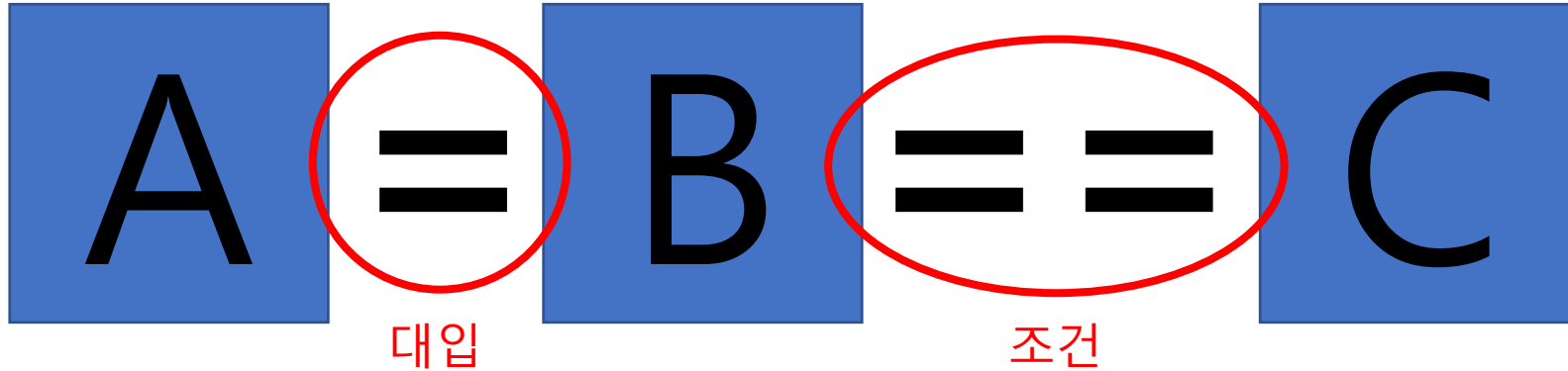
  a = 5 ;
  a-- ;
  Serial.println(a) ;

  delay(1000);
}
```

# 연산

- 조건

- `==`, `!=`, `>`, `<`, `>=`, `<=`



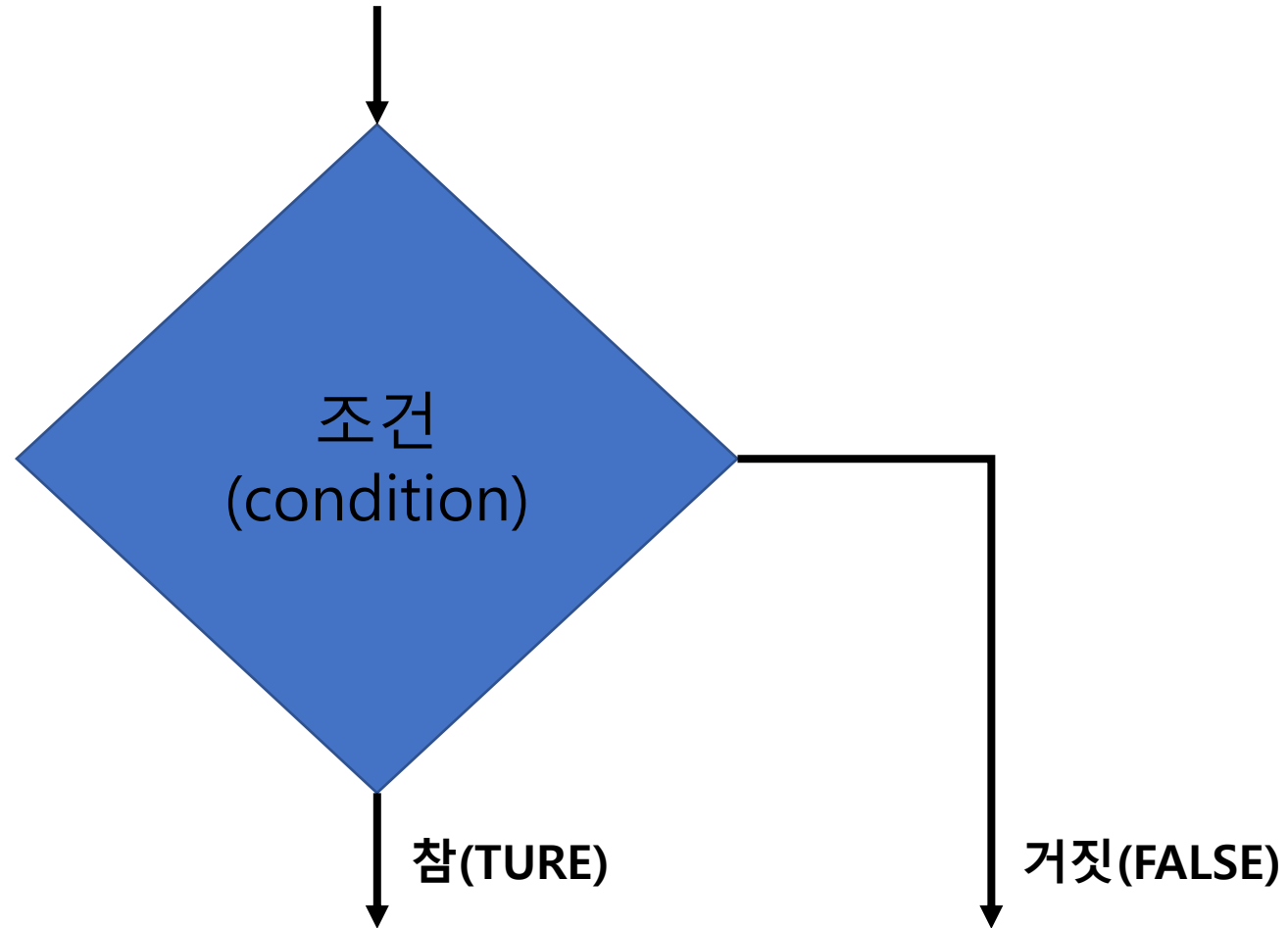
예제 : 5-12

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 0 ;
    int b = 10 ;
    bool c = a == b ;
    Serial.println(c) ;
    delay(1000);
}
```

# 조건문

- if
- if~else
- 다중 if

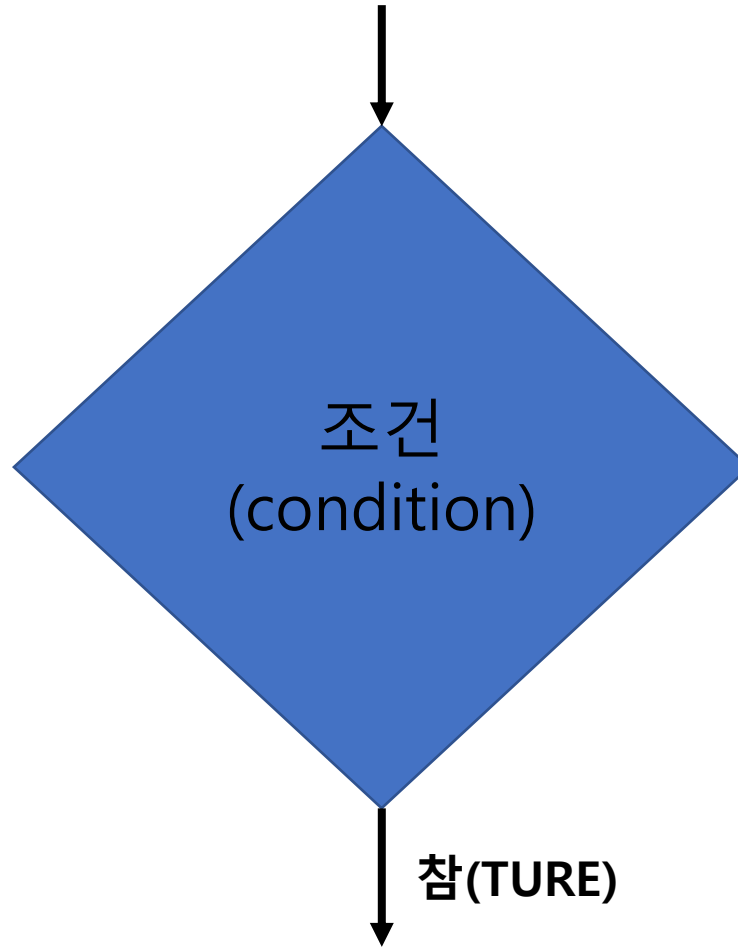




# 조건문

- if 문

```
if( 조건 )  
{  
    //명령수행  
}
```




# 조건문

- if 문

==, !=, >, <, >=, <=

if(조건)  
{  
    //명령수행  
}



예제 : 5-13

```
void setup()
{
    Serial.begin(9600);
}

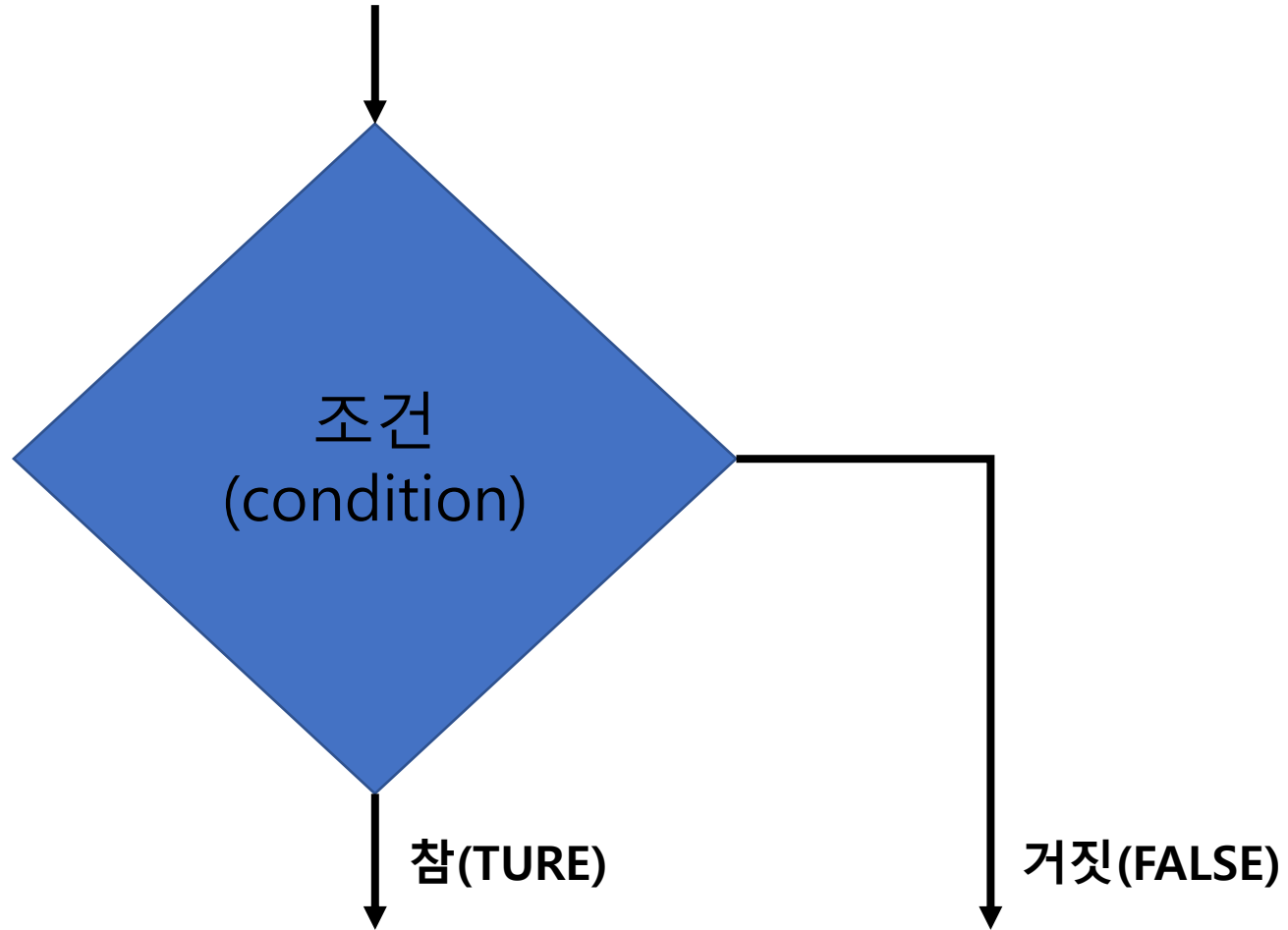
void loop()
{
    int a = 10 ;
    if( a < 10 )
    {
        Serial.println("Run") ;
    }

    delay(1000);
}
```

# 조건문

- if ~ else 문

```
if( 조건 )  
{  
    //명령수행  
}  
else  
{  
    //그렇지 않으면  
}
```




# 조건문

- if ~ else 문

==, !=, >, <, >=, <=

```
if(조건)
{
    //명령수행
}
else
{
    //그렇지 않으면
}
```



예제 : 5-14

```
void setup()
{
    Serial.begin(9600);
}

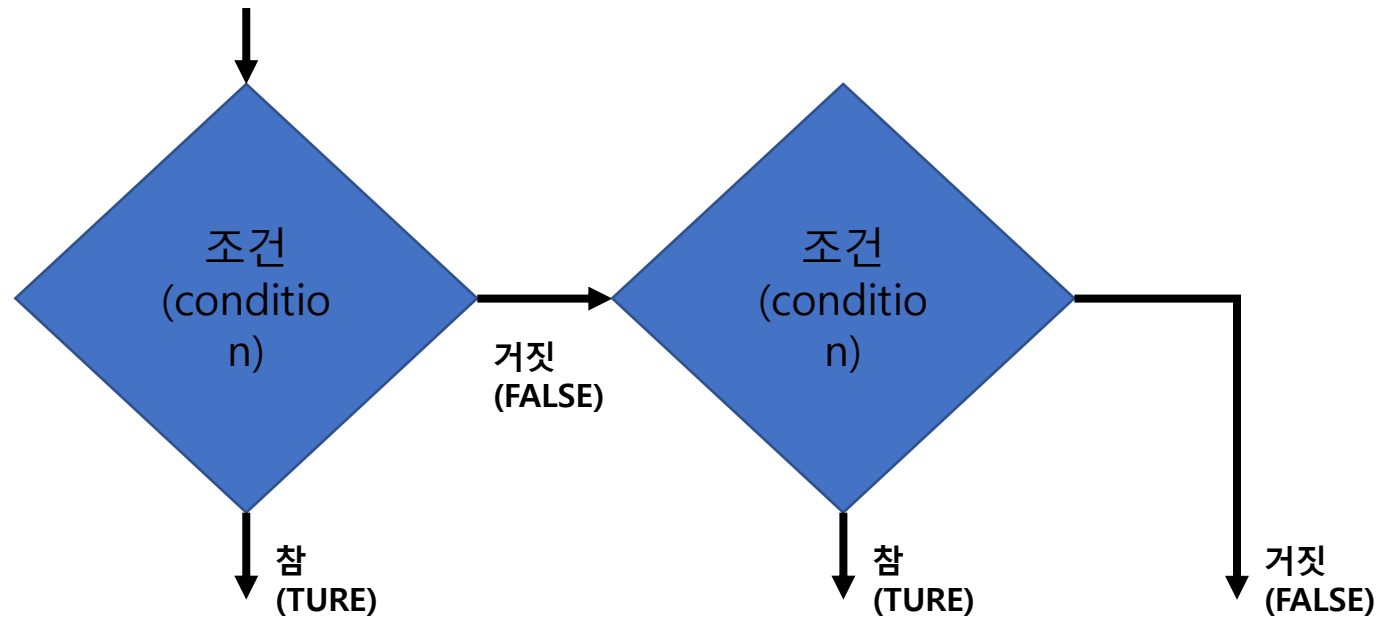
void loop()
{
    int a = 10 ;
    if( a < 10 )
    {
        Serial.println("Run1") ;
    }
    else
    {
        Serial.println("Run2") ;
    }

    delay(1000);
}
```

# 조건문

- 다중 if 문

```
if( 조건1 )  
{  
    //조건1을 만족하면 명령수행  
}  
else if( 조건2 )  
{  
    //조건2를 만족하면 명령수행  
}
```

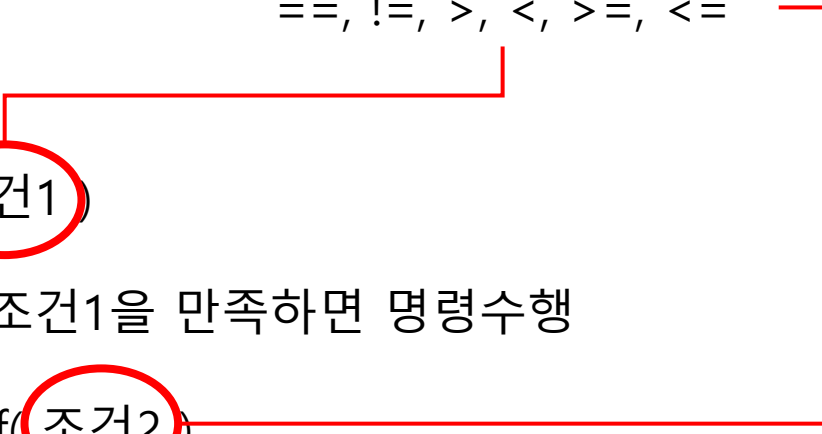


# 조건문

- 다중 if 문

==, !=, >, <, >=, <=

```
if(조건1)
{
    //조건1을 만족하면 명령수행
}
else if(조건2)
{
    //조건2를 만족하면 명령수행
}
```



예제 : 5-15

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 3 ;
    if( a < 10 )
    {
        Serial.println("Run1") ;
    }
    else if( a < 20 )
    {
        Serial.println("Run2") ;
    }

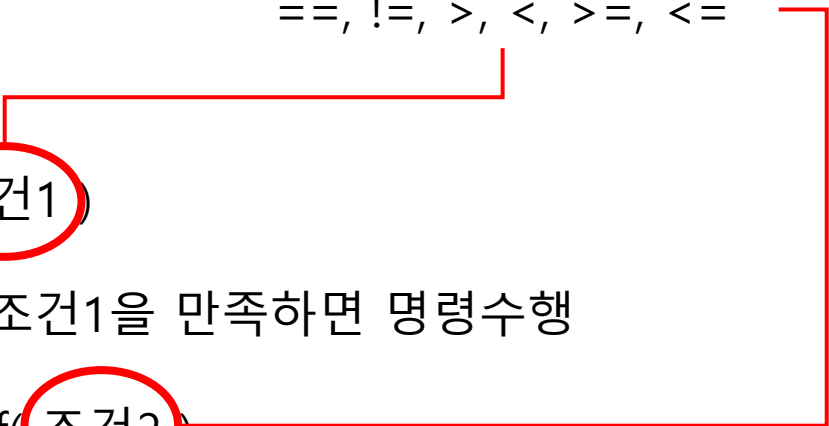
    delay(1000);
}
```

# 조건문

## • 다중 if 문

==, !=, >, <, >=, <=

```
if(조건1)
{
    //조건1을 만족하면 명령수행
}
else if(조건2)
{
    //조건2를 만족하면 명령수행
}
else
{
    //조건1과 2를 모두 만족하지 않으면 명령수행
}
```



예제 : 5-16

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 20 ;
    if( a < 10 )
    {
        Serial.println("if Run") ;
    }
    else if( a < 20 )
    {
        Serial.println("else if Run") ;
    }
    else
    {
        Serial.println("else Run") ;
    }

    delay(1000);
}
```

# 반복문

- for
- while
- do~while
- break와 continue



# 반복문

- for

```
for( 조건초기화, 조건확인, 조건변경)
{
    //명령수행
}
```

예제 : 5-17

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    for(int i=0 ; i<5 ; i++ )
    {
        Serial.println("");
        Serial.print("*");
    }
    delay(1000);
}
```

# 반복문

- while 문

```
while( 조건 )  
{  
    //조건이 TRUE라면 반복하여 명령수행  
}
```

예제 : 5-18

```
void setup()  
{  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    int i = 0 ;  
    while(i<5)  
    {  
        Serial.print("i = ") ;  
        Serial.println(i) ;  
  
        i = i + 1 ;  
    }  
    delay(1000);  
}
```

# 반복문

- do ~ while 문

```
do
{
    //조건이 TRUE라면 반복하여 명령수행
} while( 조건 );
```

예제 : 5-19

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int i = 0 ;
    do
    {
        Serial.print("i = ") ;
        Serial.println(i) ;

        i = i + 1 ;
    }while(i<5) ;

    delay(1000);
}
```

# 반복문

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int i = 5 ;
    while(int i<5)
    {
        Serial.print("i = ") ;
        Serial.println(i) ;

        i = i + 1 ;
    }
    delay(1000);
}
```

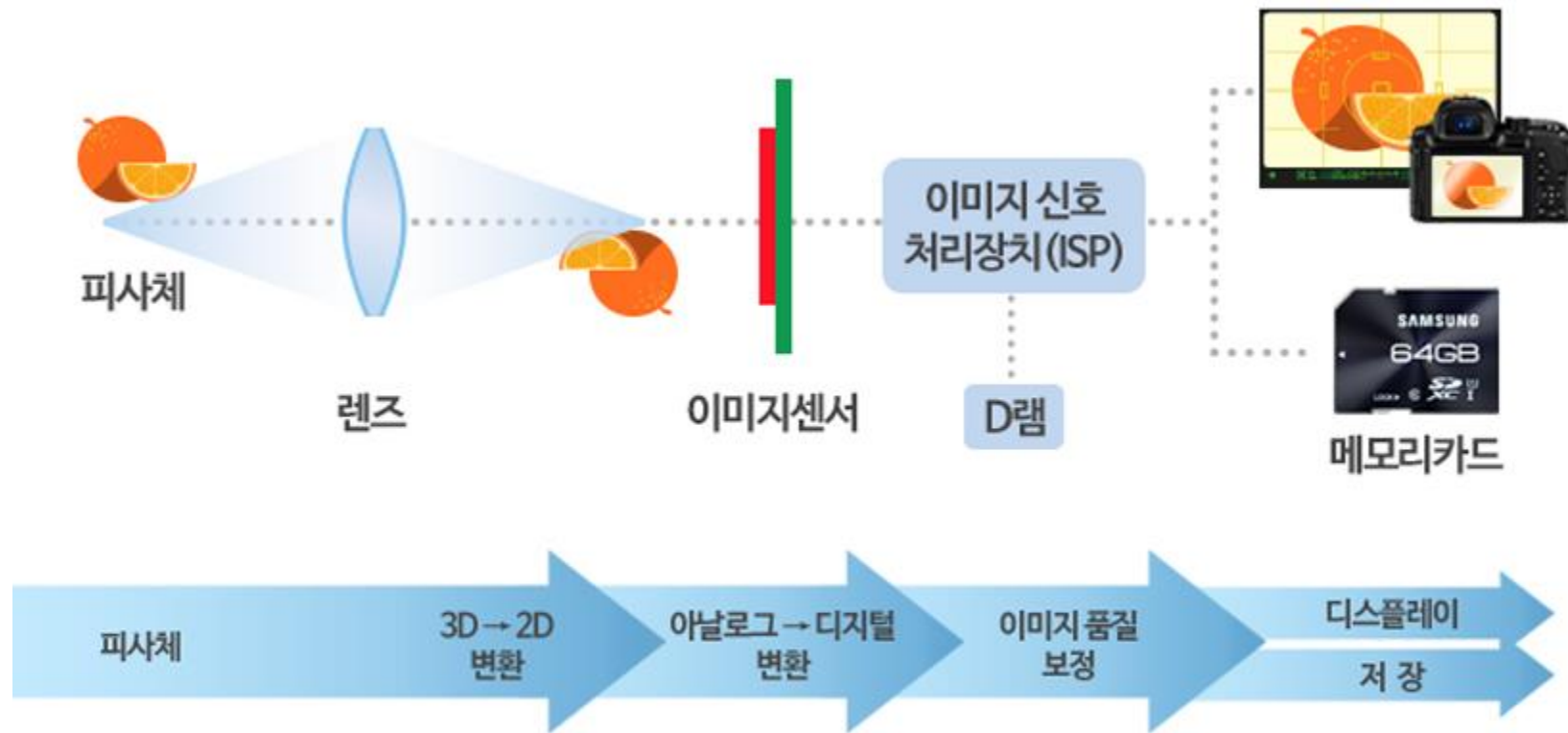
```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int i = 5 ;
    do
    {
        Serial.print("i = ") ;
        Serial.println(i) ;

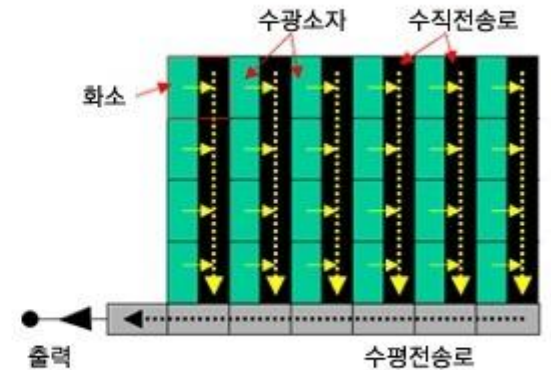
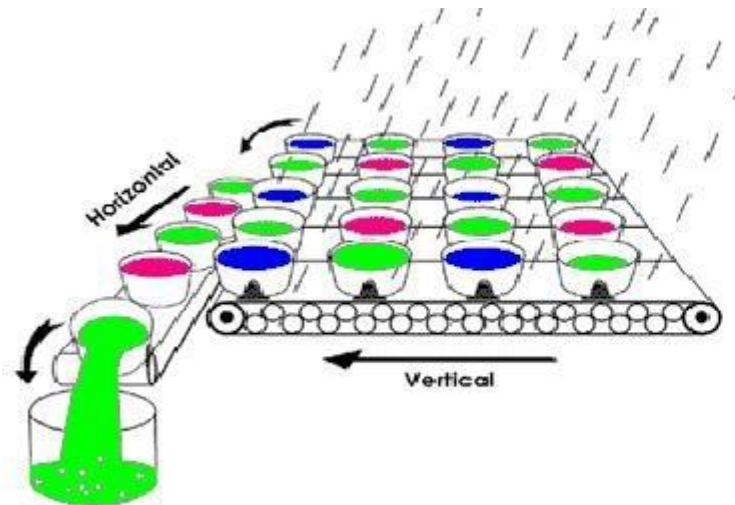
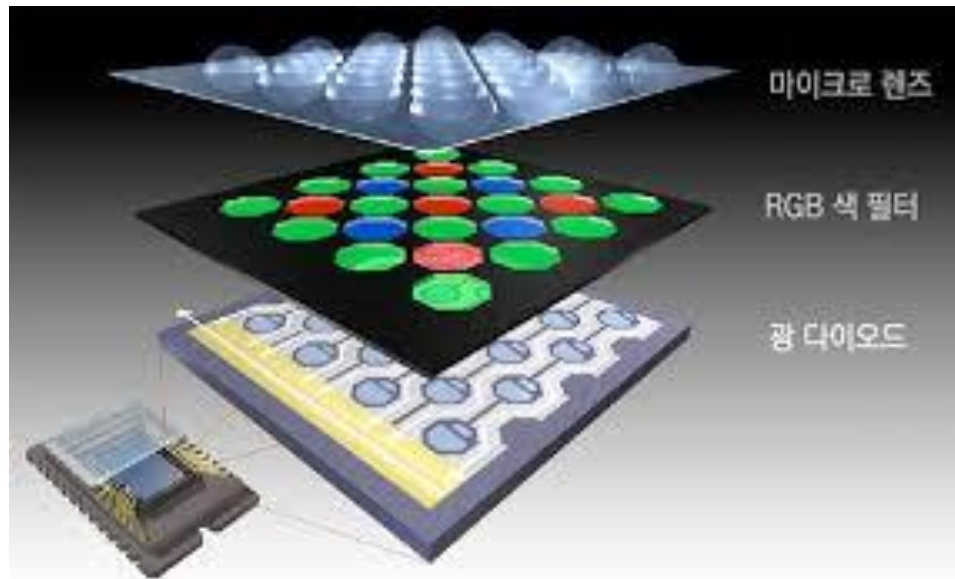
        i = i + 1 ;
    }while(int i<5) ;

    delay(1000);
}
```

# 배열과 for문을 이용한 pixel(이미지)참조



# 배열과 for문을 이용한 pixel(이미지)참조



# 배열과 for문을 이용한 pixel(이미지)참조

- Pixel(화소)와 데이터, 그리고 메모리

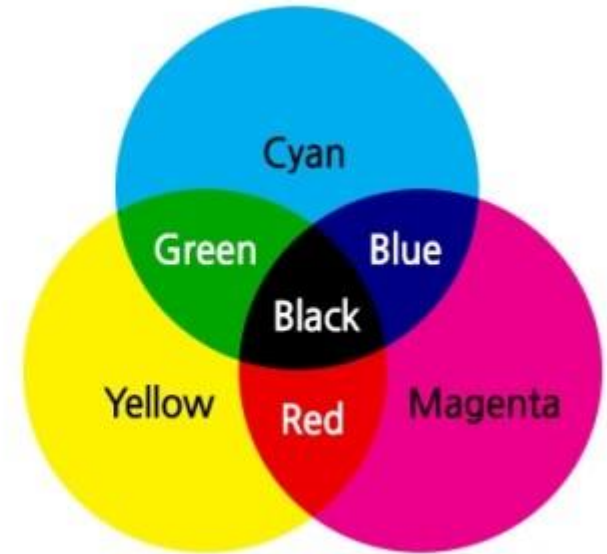
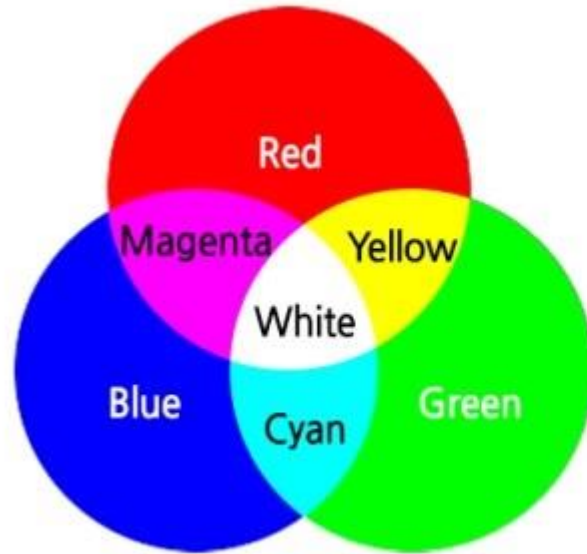


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

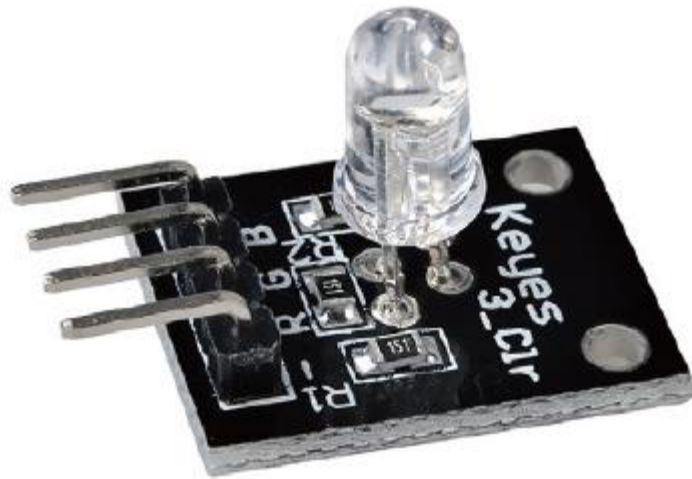
# 이미지 정보(RGB888)

- RGB888
  - Red : 1 Byte
  - Green : 1 Byte
  - Blue : 1 Byte

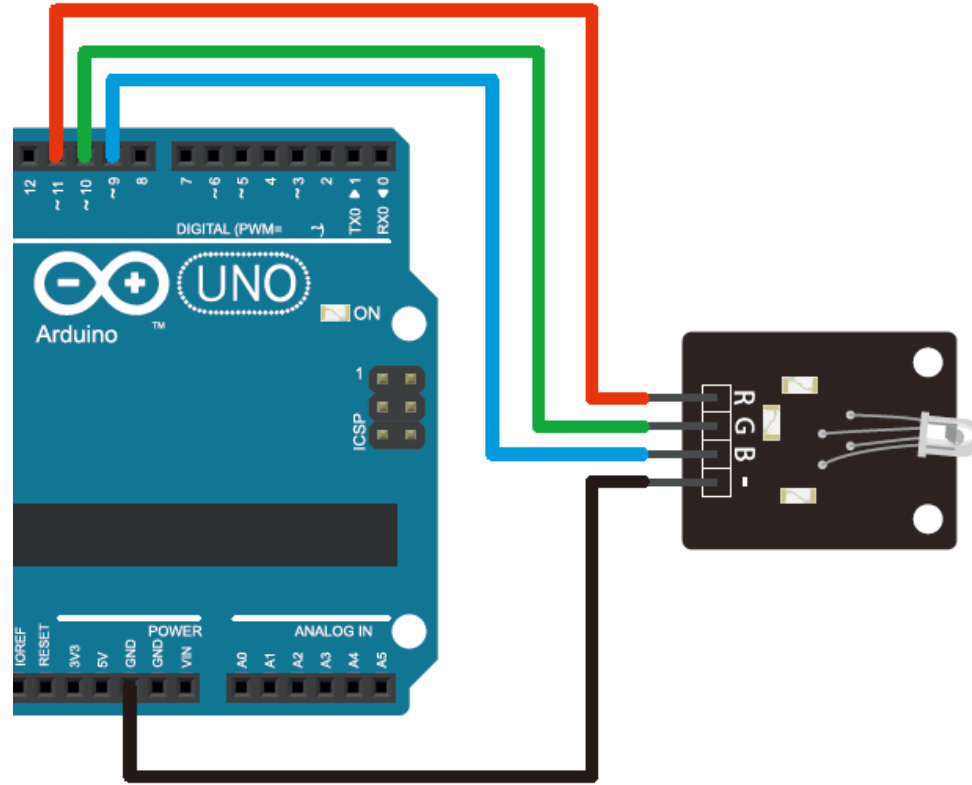




# 3색 LED를 이용하여 RGB 표현



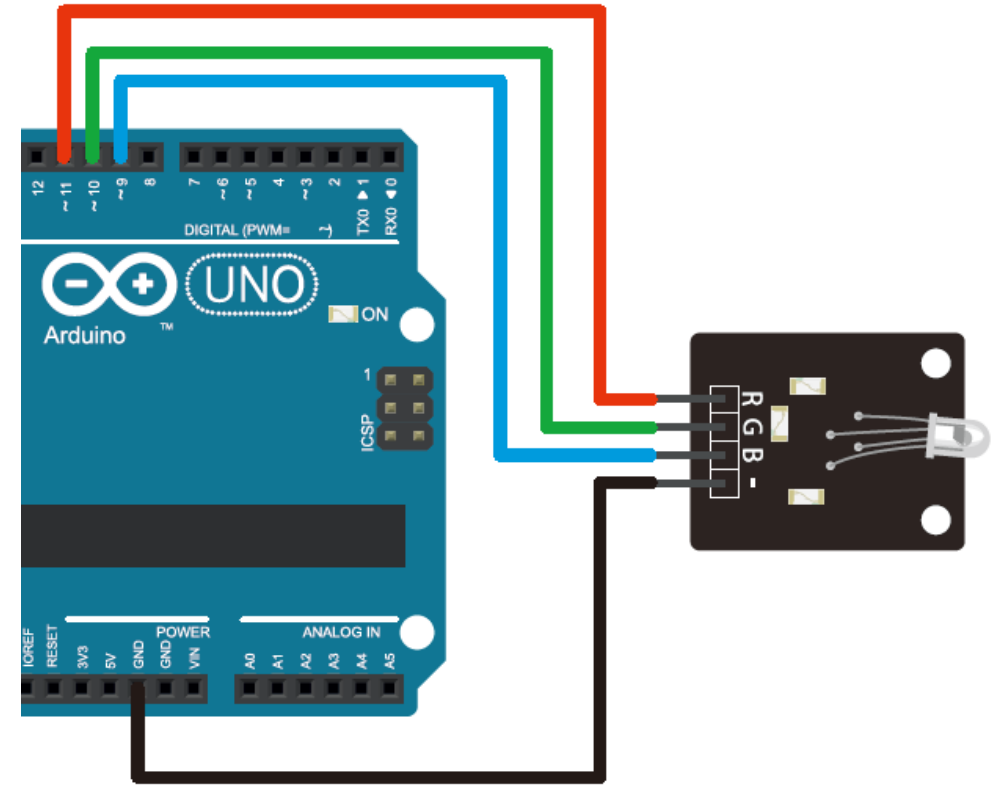
# 3색 LED를 이용하여 RGB 표현



# 3색 LED를 이용하여 RGB 표현

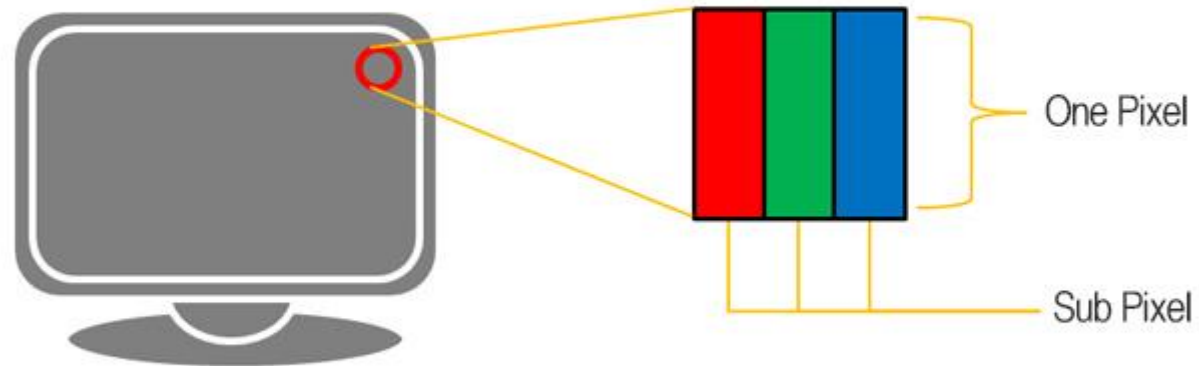
예제 5-20

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(11, OUTPUT);  
  pinMode(10, OUTPUT);  
  pinMode(9, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  analogWrite(11, 255);  
  analogWrite(10, 128);  
  analogWrite(9, 0);  
}
```



# 이미지 정보 출력(모니터)

---



# 이미지 정보(RGB888)를 메모리에 저장

[illegible]

# 이미지에서 특정 색상 찾기

