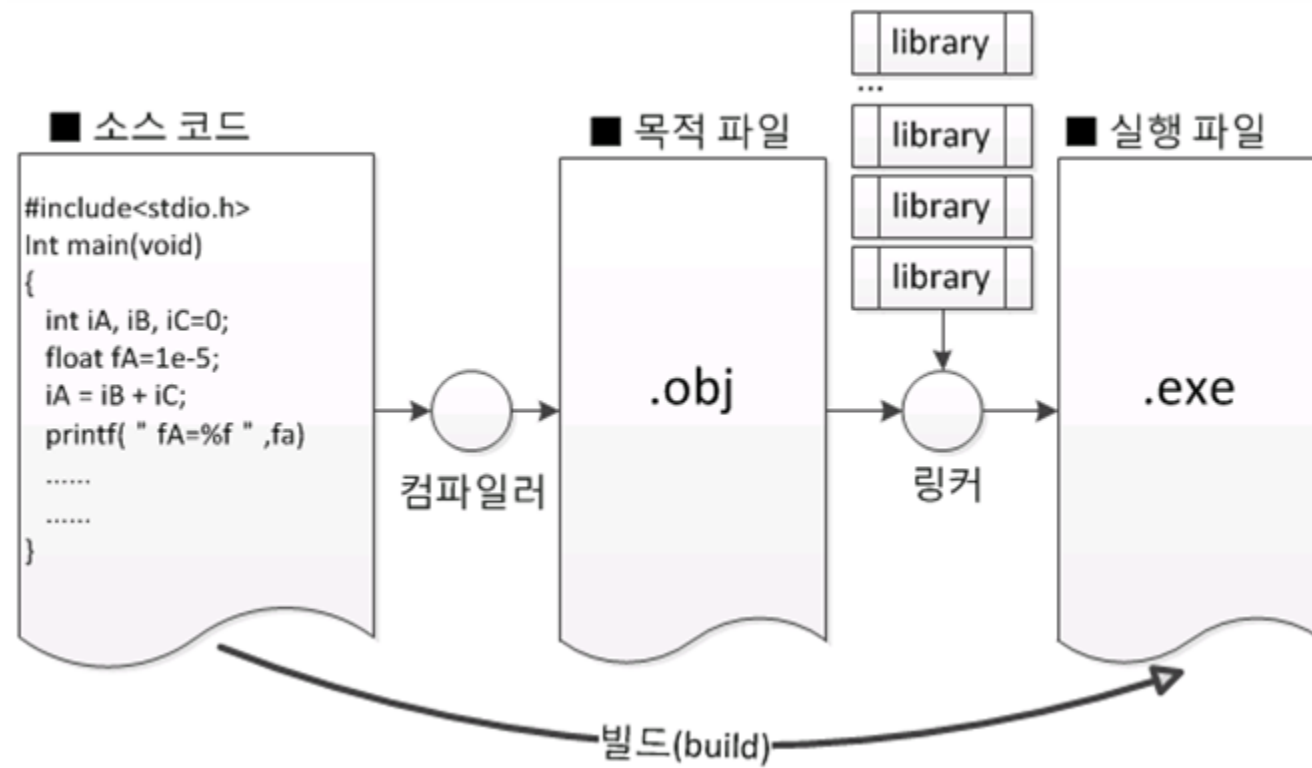


C언어
그리고 영상처리까지

아두이노 SW 프로그래밍 과정



아두이노 프로그래밍 과정

- 소스코드 작성



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 볼록감지 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 볼록이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 볼록이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

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

14 Arduino Gemma on COM4

아두이노 프로그래밍 과정

- 컴파일



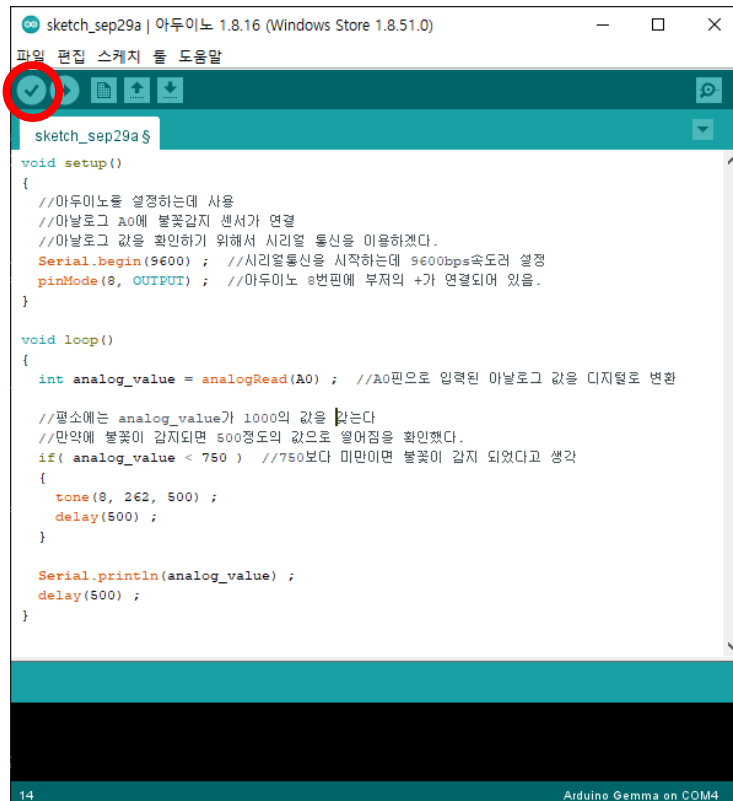
```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

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

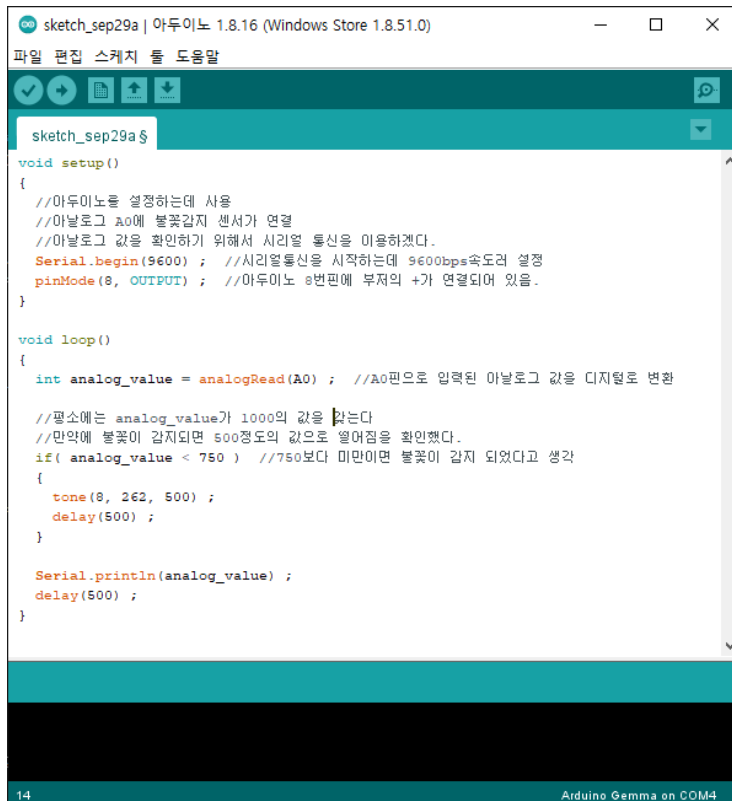
  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

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

14 Arduino Gemma on COM4

아두이노 프로그래밍 과정

• 업로드 & 실행

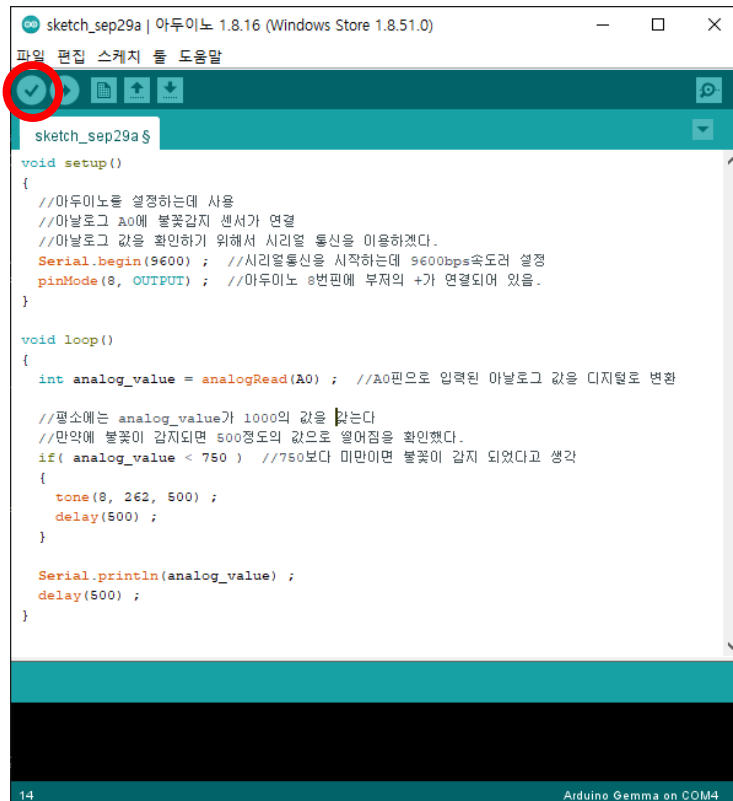


```
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼 통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

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

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 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);
}
```

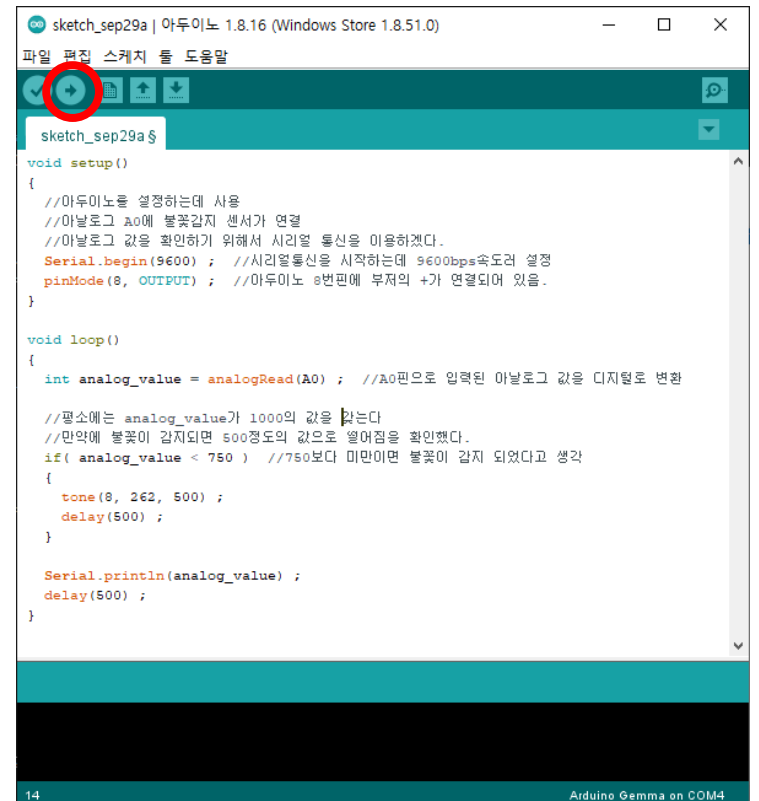


```
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼 통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

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

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 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);
}
```



```
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 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()
```

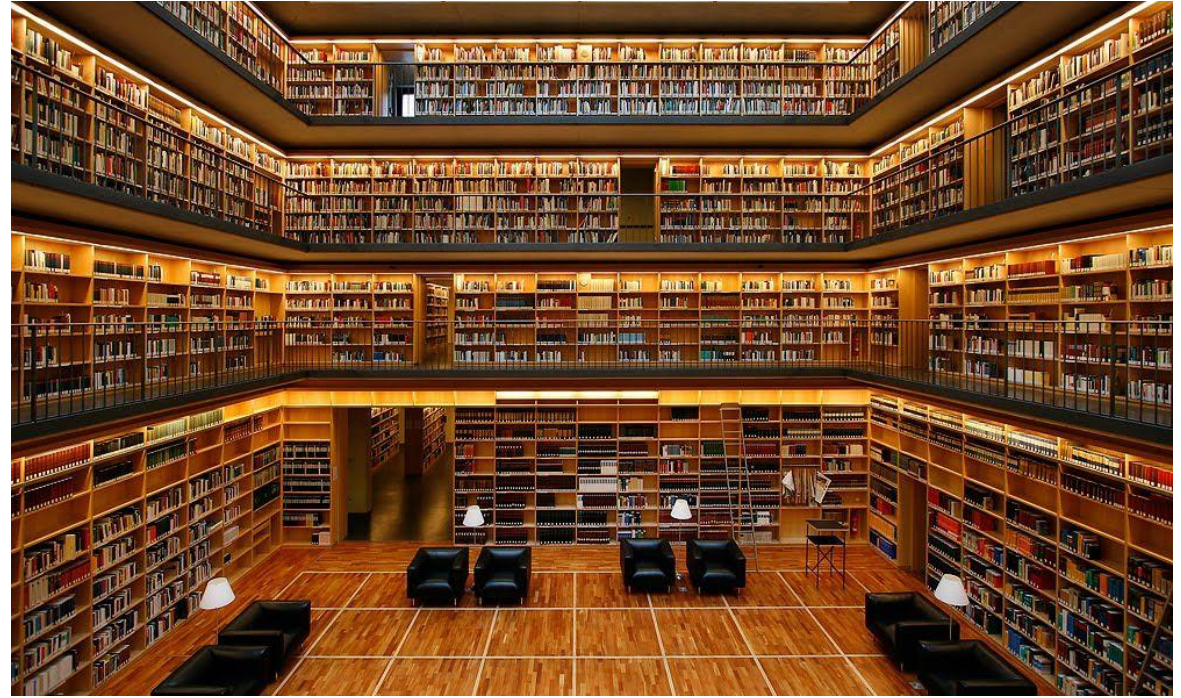
```
{  
}
```

```
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

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

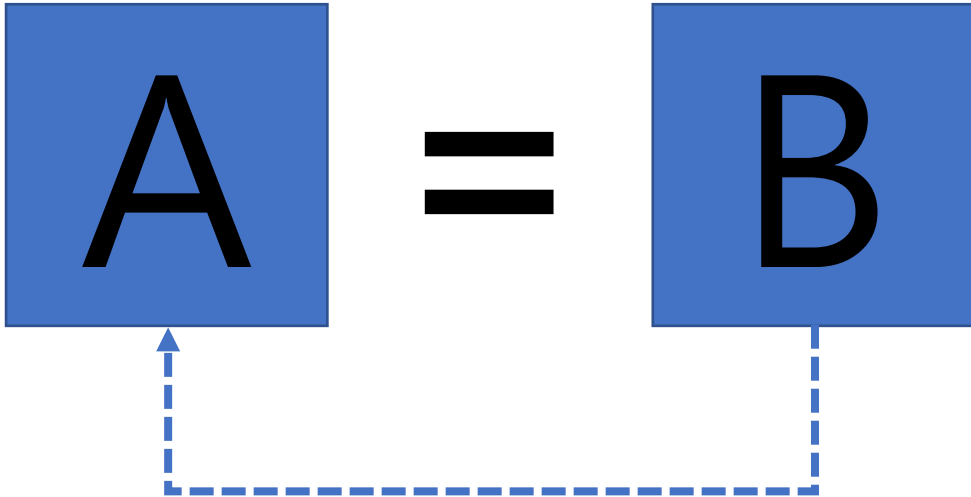
- 더하기(+), 빼기(-), 곱하기(*), 나누기(/), 나머지(%), 증가(++), 감소(--)

- 조건

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

연산

- 대입
 - =



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

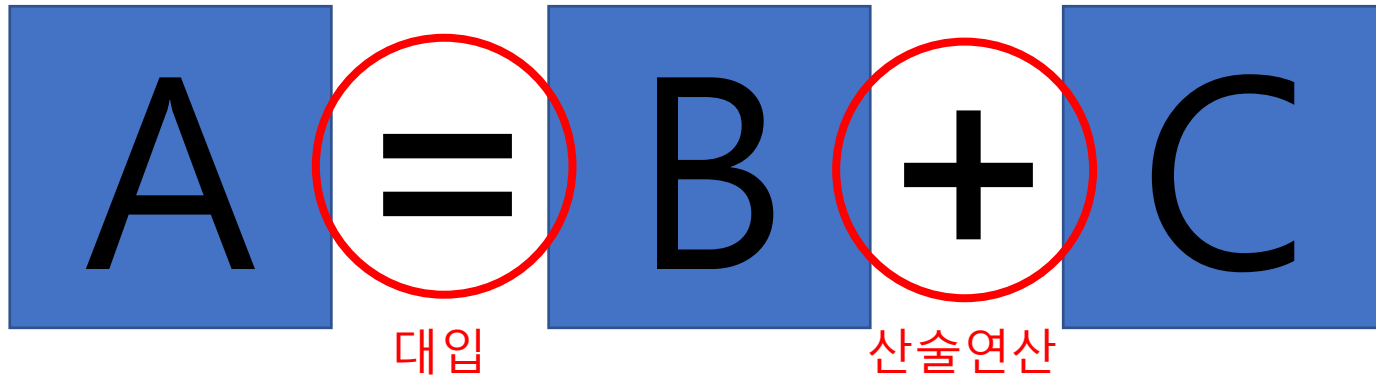
void loop()
{
  int a = 10 ;
  Serial.println(a) ;
  delay(1000);
}
```

예제 : 5-9

연산

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

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) ;

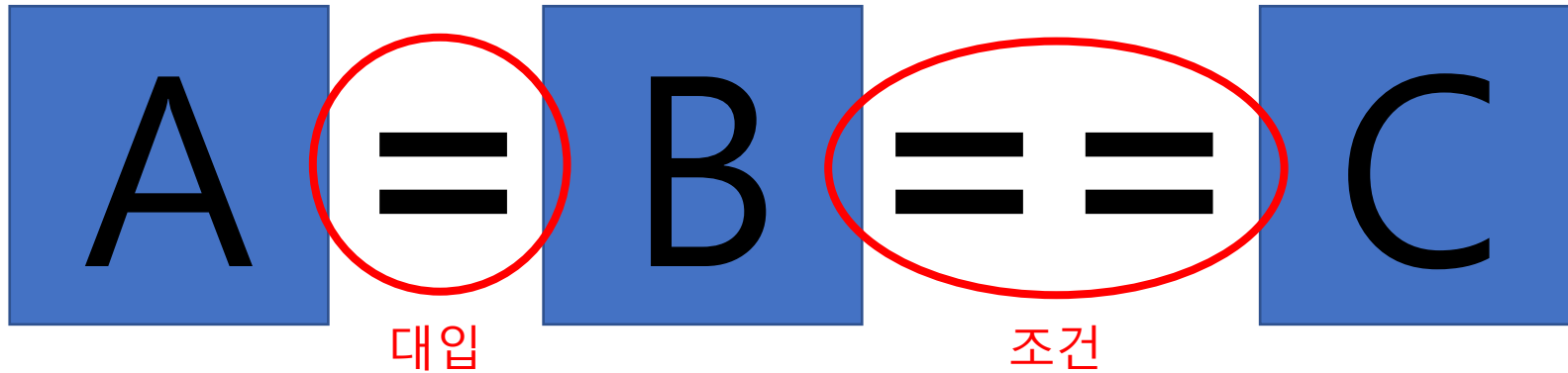
  a = 5 ;
  a-- ;
  Serial.println(a) ;

  delay(1000);
}
```

연산

- 조건

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



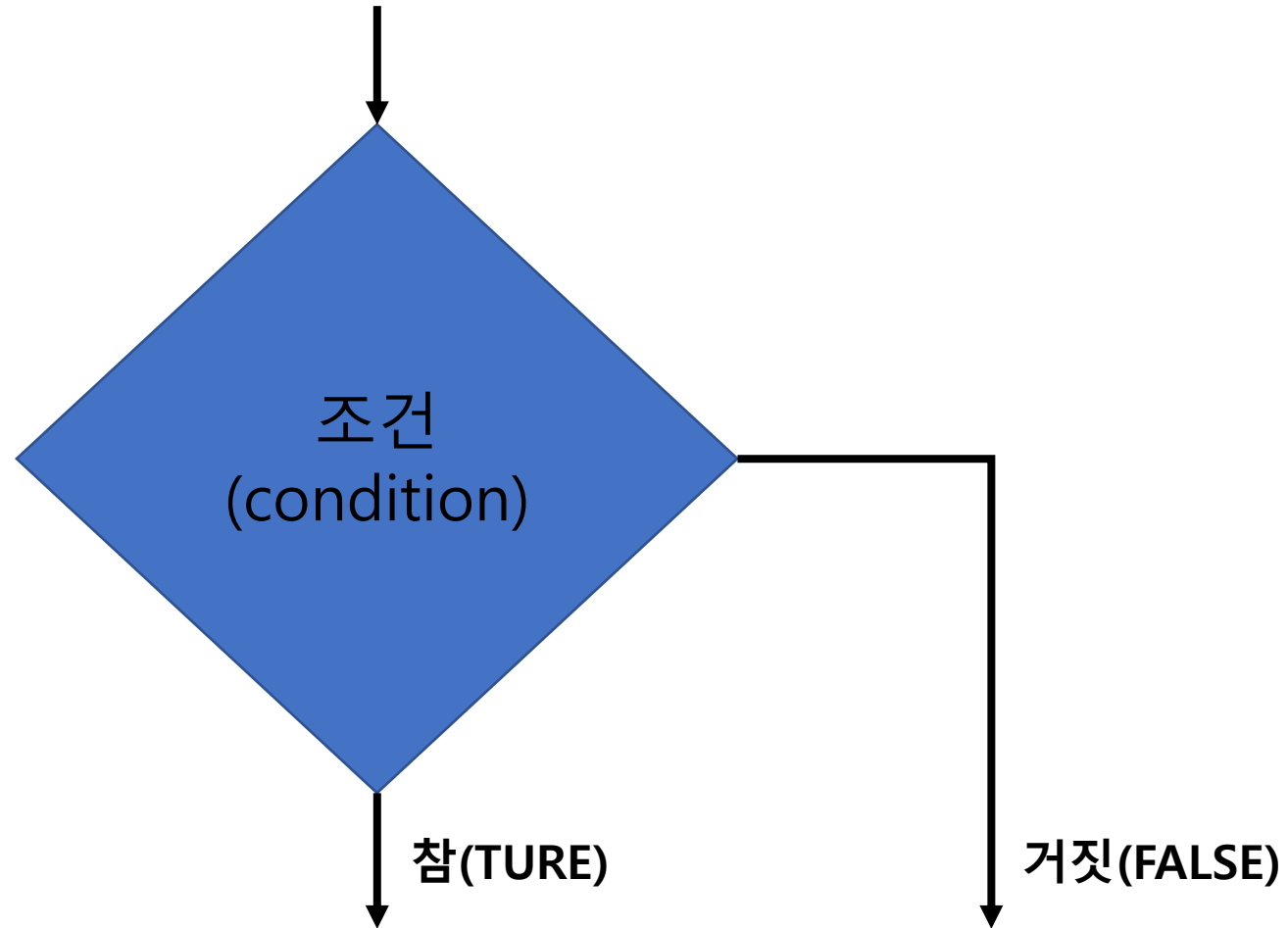
예제 : 5-12

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

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

조건문

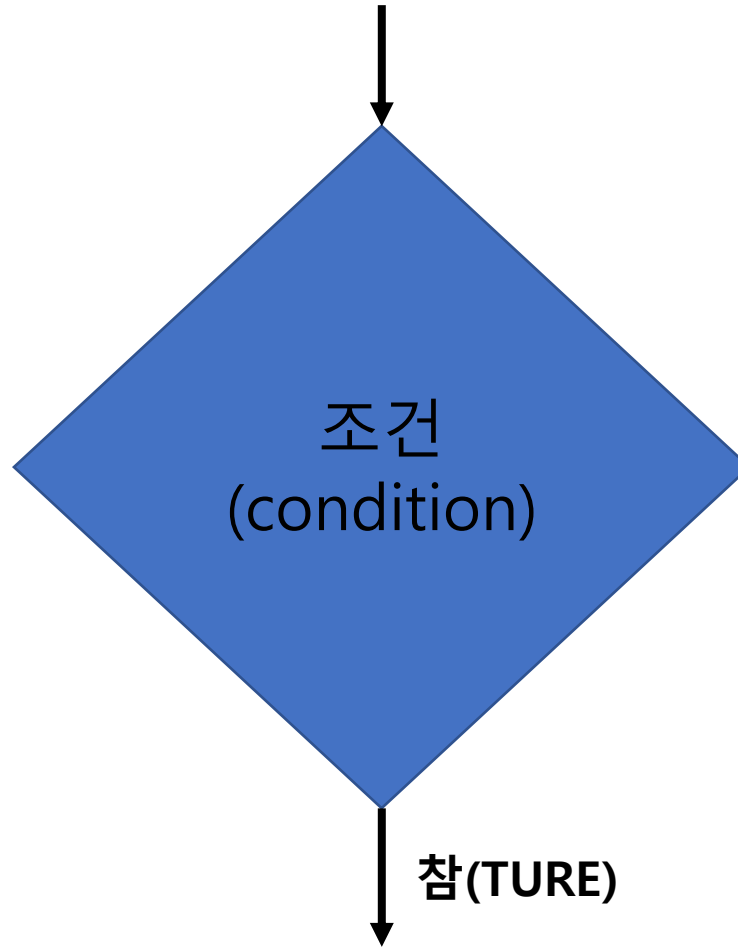
- if
- if~else
- 다중 if



조건문

- if 문

```
if( 조건 )  
{  
    //명령수행  
}
```




조건문

- if 문

if(조건)
{
 //명령수행
}

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



예제 : 5-13

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

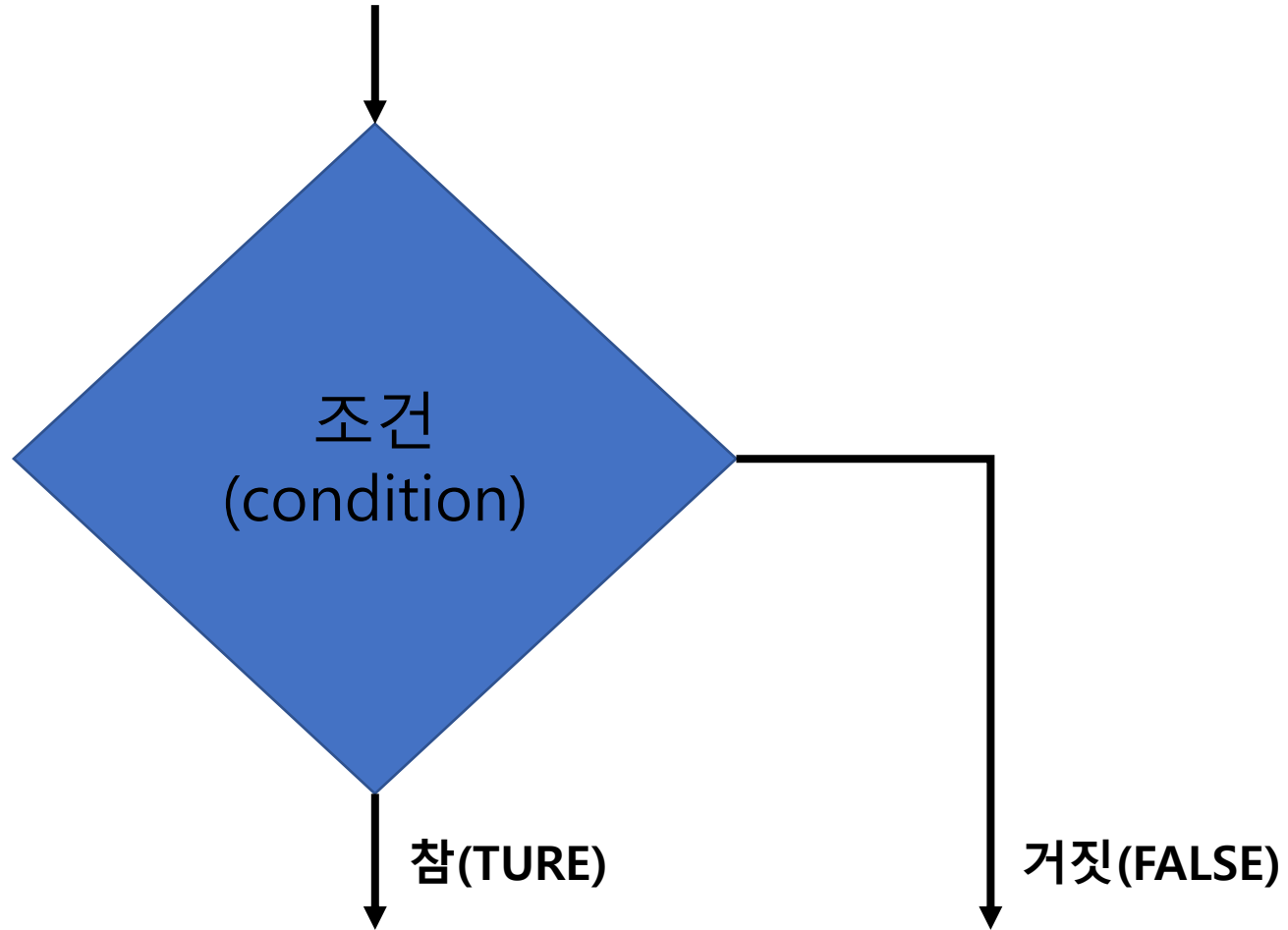
void loop()
{
    int a = 10 ;
    if( a < 10 )
    {
        Serial.println("Run") ;
    }

    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);
}

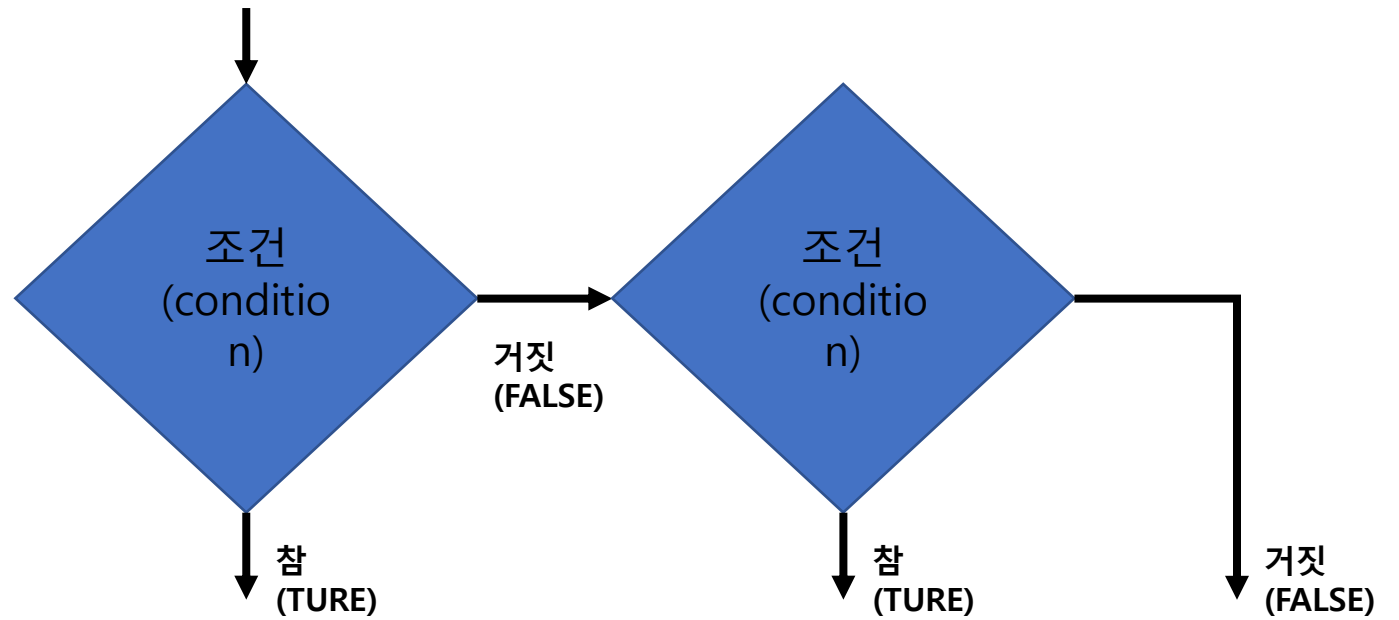
void loop()
{
    int a = 10 ;
    if( a < 10 )
    {
        Serial.println("Run1") ;
    }
    else
    {
        Serial.println("Run2") ;
    }

    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") ;
    }

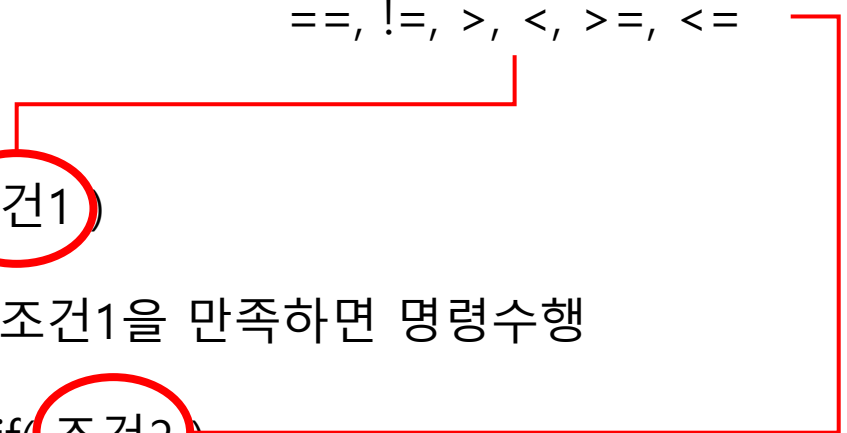
    delay(1000);
}
```

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