# 처음 만드는 아두이노 프로그래밍




한국과학창의재단

# 1 아두이노 IDE 설치하기

## 처음 만드는 아두이노 프로그래밍

아두이노를 하기 위해 필요한 프로그램을 설치하고, 아두이노에 기본적으로 있는 LED를 껐다 켜는 것을 배워봅시다.

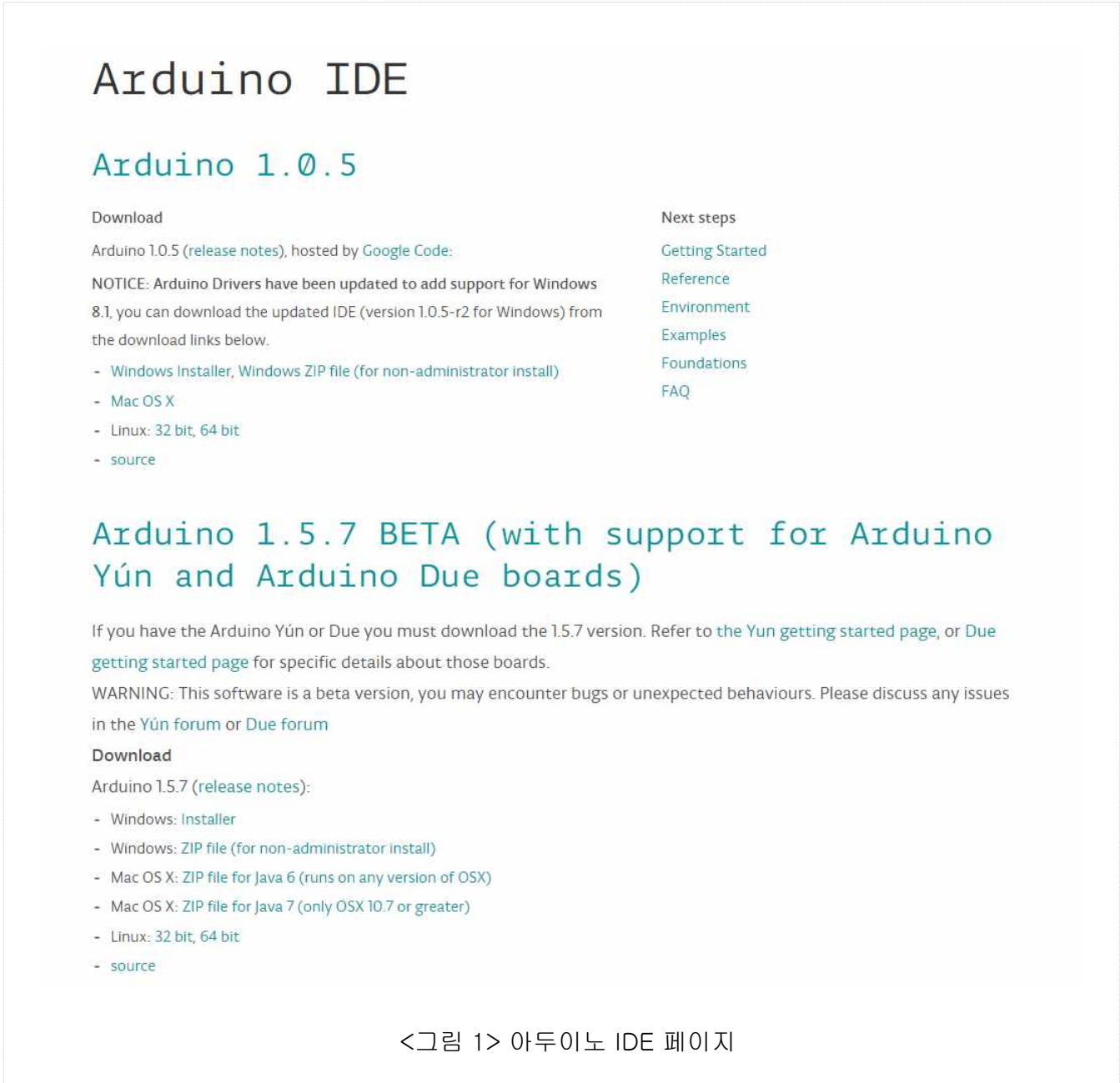
## 아두이노 IDE 설치하기

아두이노를 사용하기 위해서는 아두이노에서 실행할 프로그램이 필요합니다. 그리고 이 프로그램은 아두이노 IDE(통합개발환경)라는 것을 이용해 만들 수 있습니다. 아두이노 IDE를 설치해보도록 하겠습니다.

# 1 아두이노 IDE 설치하기

## 설치파일 다운로드하기

아두이노 IDE를 설치하기 위해서는 아두이노 IDE 페이지로 이동해 설치 파일을 다운로드 해야 합니다.
**아두이노 IDE 페이지** : http://arduino.cc/en/Main/Software




<그림 1> 아두이노 IDE 페이지

# 1 아두이노 IDE 설치하기

아두이노 IDE 페이지로 이동하면 사용할 수 있는 아두이노 버전들이 위와 같이 표시되어 있습니다. BETA는 가장 최신 버전이지만 아직 모든 기능이 안정적으로 돌아간다고 보장을 할 수 없습니다. 따라서 BETA보다 정식 버전을 사용하는 것이 좋습니다. 사용하고자 하는 버전 밑 부분을 보시면 설치파일 링크들이 있습니다. 이 중 자신의 컴퓨터 운영체제에 맞는 설치파일을 다운로드합니다.

## 설치하기

### ▪ 윈도우즈 (Windows)

1. 압축파일(Windows ZIP file)과 인스톨러(Windows Installer) 파일 두 종류가 있습니다. 이 중 인스톨러 파일을 다운로드합니다.
2. 다운로드가 완료되면 인스톨러 파일을 실행합니다.
3. 만약 다른 폴더에 설치하고 싶다면 경로 중간에 한글이 포함되지 않도록 해줍니다. 또한 경로가 너무 길어서도 안 됩니다. 설치를 완료해도 프로그램이 정상적으로 동작하지 않을 수 있습니다.
4. 중간에 드라이버를 설치하겠냐고 묻는다면 확인을 눌러줍니다.

### ▪ 맥 (Mac)

1. 압축파일(Mac OS X)을 다운로드한 후 압축을 풀어줍니다.
2. 압축을 풀면 아두이노 아이콘 파일이 나오는데, 이 파일을 끌어서 응용프로그램이라고 적힌 곳에 옮겨줍니다.

### ▪ 리눅스 (Linux)

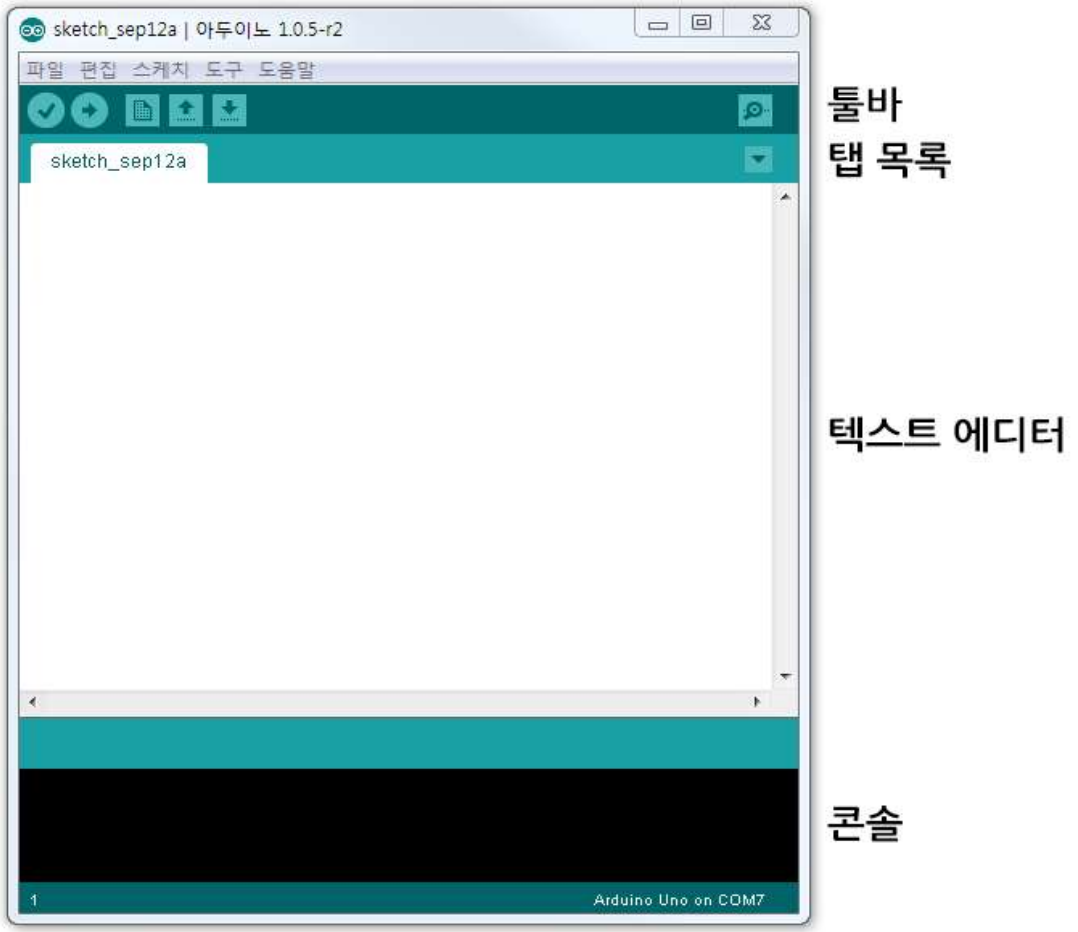
1. 컴퓨터 사양에 맞는 압축파일을 선택하고 다운로드합니다.
2. 터미널 프로그램을 실행시킵니다. 다운로드한 파일 이름이 “arduino-1.0.5-linux32.tgz”라고 가정하고 다음과 같이 입력합니다. 만약 파일명이 다르다면 바꿔서 입력합니다.
   - tar xvfz arduino-1.0.5-linux32.tgz

# 1 아두이노 IDE 설치하기

## | 아두이노 IDE 기본 구성

아두이노 IDE 설치를 완료했다면 실행해봅시다.

아두이노 IDE는 아두이노에서 실행할 프로그램을 작성하는 도구입니다. 아두이노 IDE를 이용해 코드를 입력하고 아두이노에 업로드하면 아두이노가 작동하게 됩니다. 아두이노 IDE에서는 코드를 작성하는 문서의 단위를 스케치라고 부릅니다. 아두이노 IDE를 처음 실행하면 탭 부분에 영어로 sketch라고 적힌 것을 볼 수 있습니다.




<그림 2> 아두이노 IDE 기본구성

# 1 아두이노 IDE 설치하기

아두이노 IDE는 다음과 같이 구성되어 있습니다.

- 툴바 : 아두이노 IDE에서 사용하는 기능의 버튼이 있습니다.
- 탭 목록 : 하나의 스케치는 여러 개의 파일로 이루어질 수 있습니다.
  탭 목록은 스케치에 포함된 파일들을 탭 목록으로 표시해서 보여주는 부분입니다.
- 텍스트 에디터 : 코드를 입력하는 부분입니다.
- 콘솔 : 아두이노 IDE와 관련된 정보가 표시되는 곳입니다. 코드를 작성하고 에러가 나는 경우 콘솔을 이용해 에러가 난 부분을 확인할 수 있습니다.

## 툴바 버튼 설명

**확인 / 컴파일** : 텍스트 에디터에 쓴 코드를 확인하고 컴파일합니다. 만약 코드에 에러가 있으면 도중에 멈추면서 에러가 난 위치가 표시됩니다.

**업로드** : 텍스트 에디터에 쓴 코드를 확인하고 컴파일이라는 것을 한 뒤 아두이노에 업로드 합니다. 컴파일은 여러분이 작성한 코드를 아두이노가 이해할 수 있는 말로 바꾸는 것과 같습니다.

**새로 만들기** : 새 스케치를 만듭니다.

**열기** : 기존에 작업한 스케치를 불러옵니다.

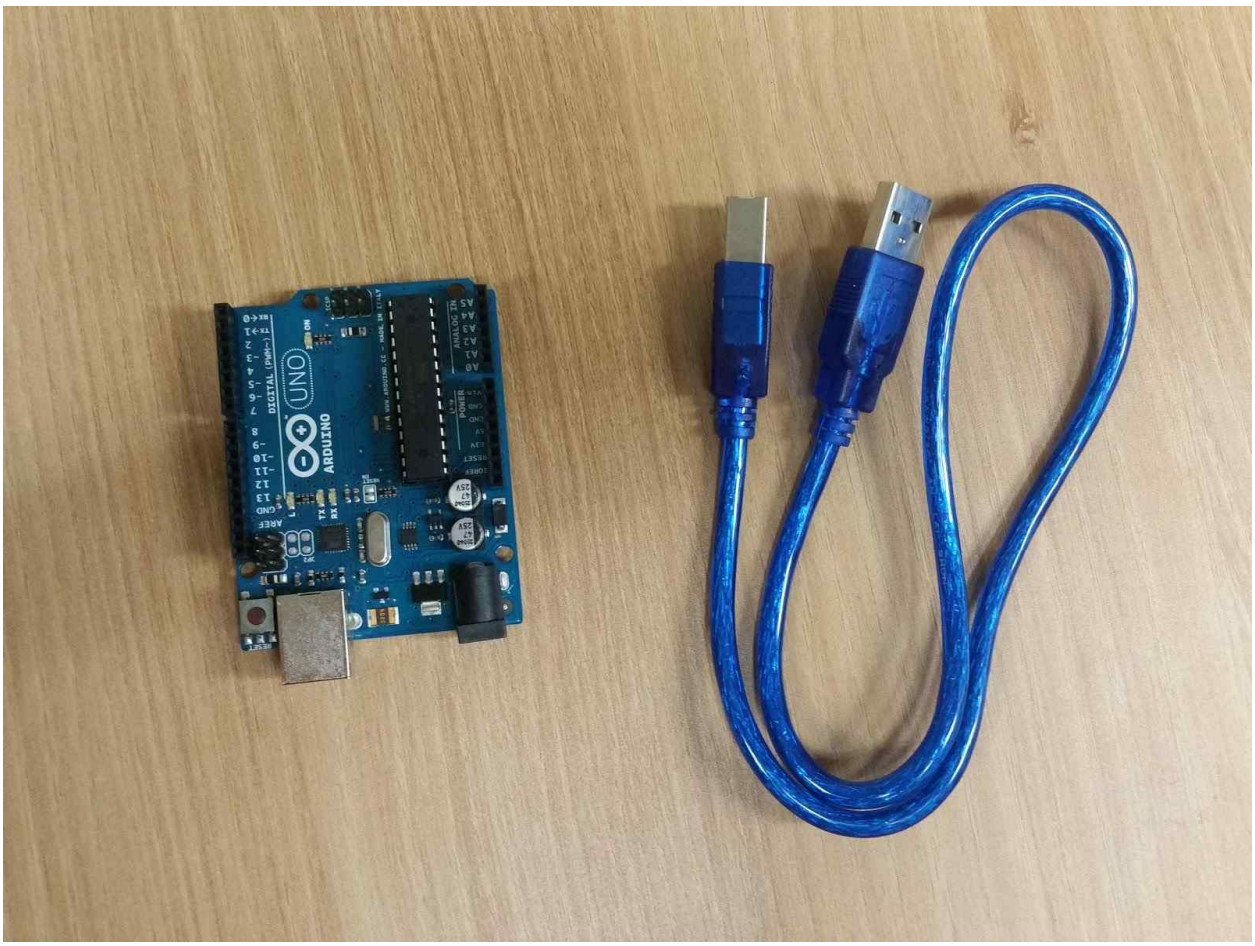
**저장하기** : 현재 보고 있는 스케치를 저장합니다.

**시리얼 모니터** : 아두이노는 컴퓨터와 대화를 할 수 있습니다. 시리얼 모니터는 아두이노와 컴퓨터가 대화할때 사용하는 도구입니다. 버튼을 누르면 시리얼 모니터가 나타납니다.

# 1 아두이노 IDE 설치하기

## | 아두이노 연결하기

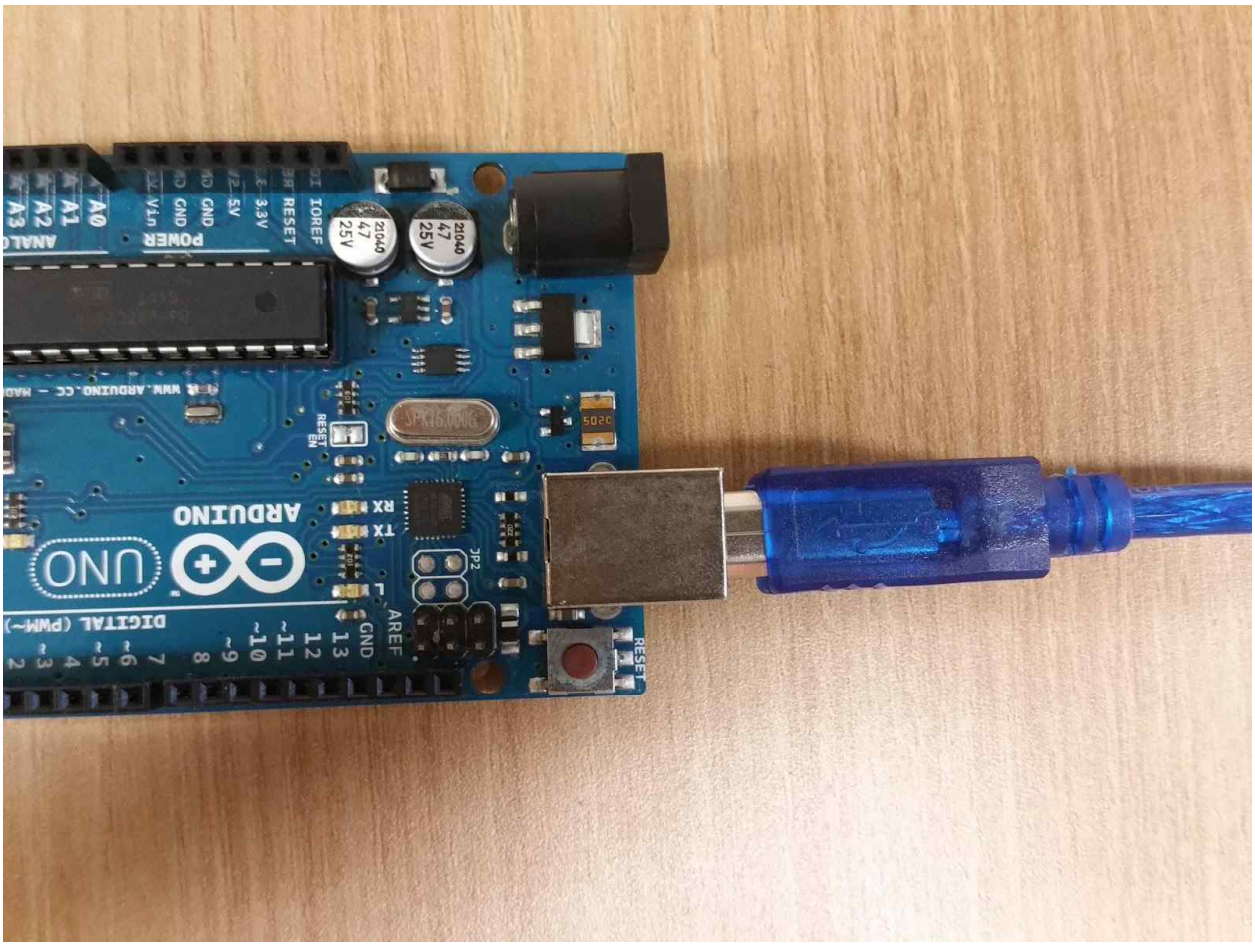
이제 아두이노를 컴퓨터에 연결해봅시다.




<그림 3> 아두이노 연결하기

아두이노를 컴퓨터에 연결하기 위해 USB선을 준비합니다.

# 1 아두이노 IDE 설치하기




<그림 3> 아두이노 연결하기

USB선에 한 쪽을 아두이노에 연결합니다.

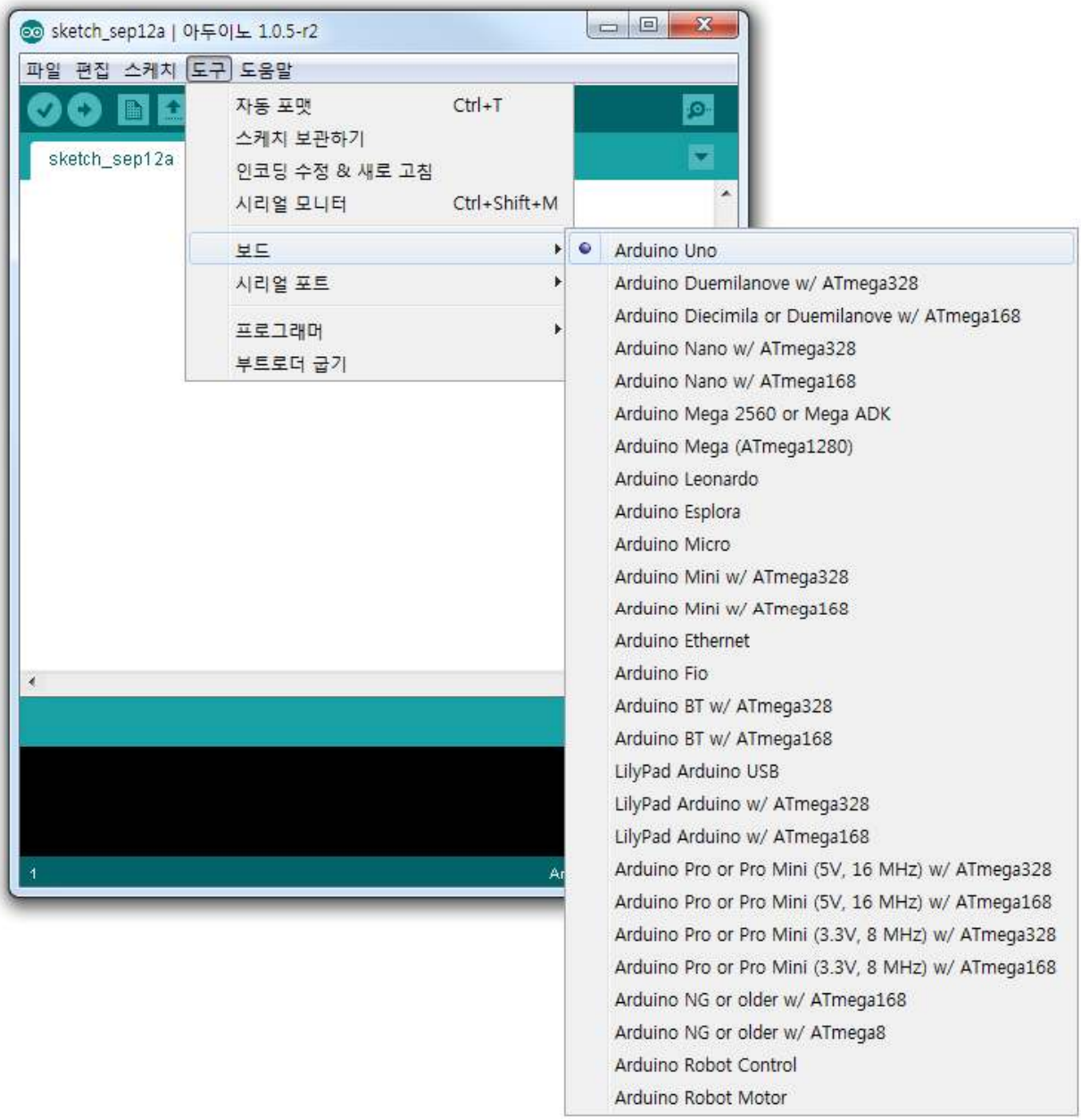
# 1 아두이노 IDE 설치하기


<그림 3> 아두이노 연결하기

USB선의 반대편을 컴퓨터에 연결합니다.

만약 컴퓨터에 처음으로 연결했다면 자동으로 드라이버를 설치한다고 뜨는 것을 볼 수 있습니다.
드라이버 설치가 완료되면 아두이노 IDE를 실행합니다.

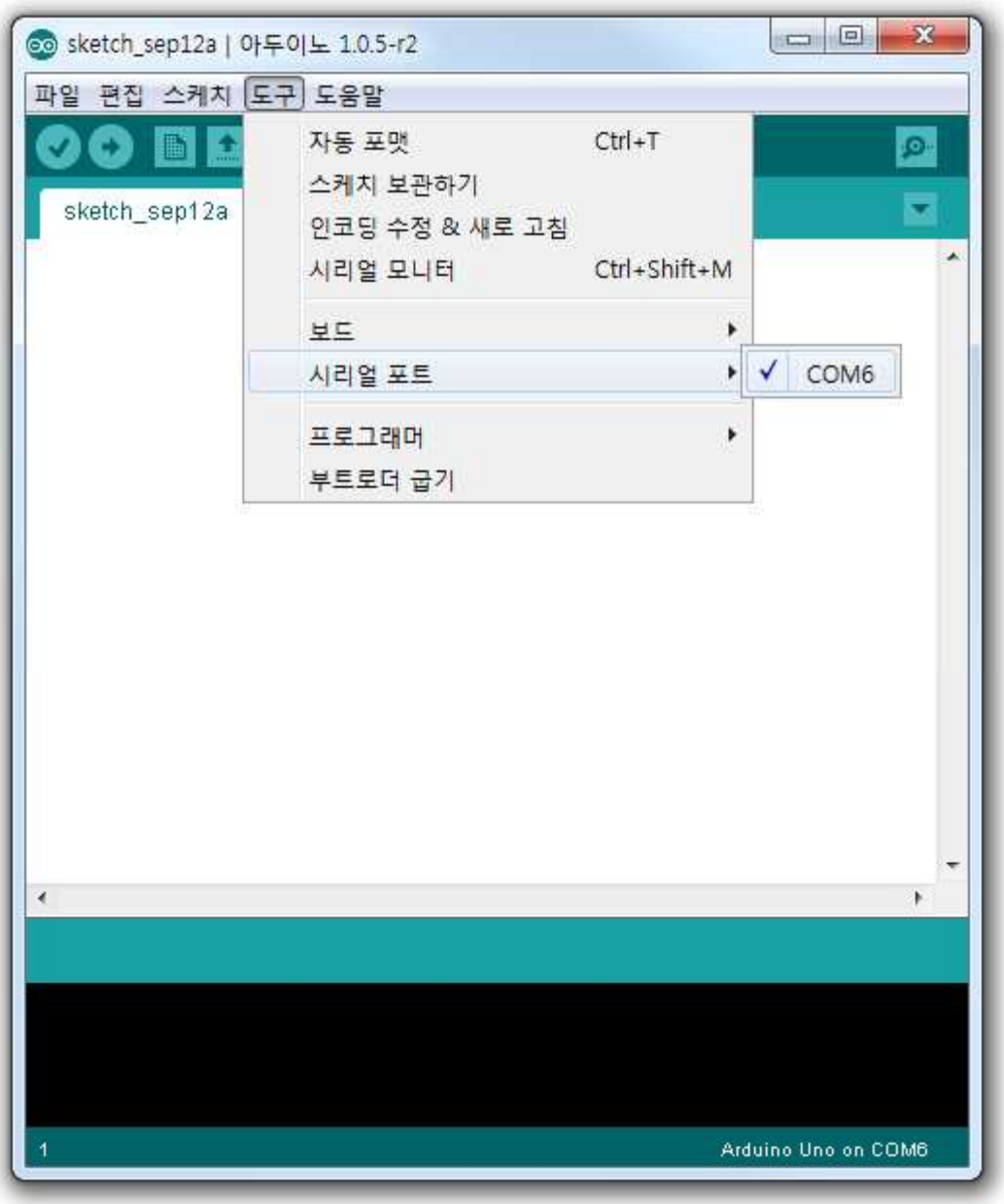
# 1 아두이노 IDE 설치하기




<그림 3> 아두이노 연결하기

아두이노를 사용하려면 아두이노 IDE에서 설정을 해줘야 하는 것이 있습니다. 먼저 사용하는 보드를 선택해야합니다. 메뉴에서 도구-보드를 선택하면 위와 같이 아두이노 보드 목록이 표시됩니다. 여러분이 사용하는 보드는 아두이노 UNO(Arduino UNO)입니다.

# 1 아두이노 IDE 설치하기




<그림 3> 아두이노 연결하기

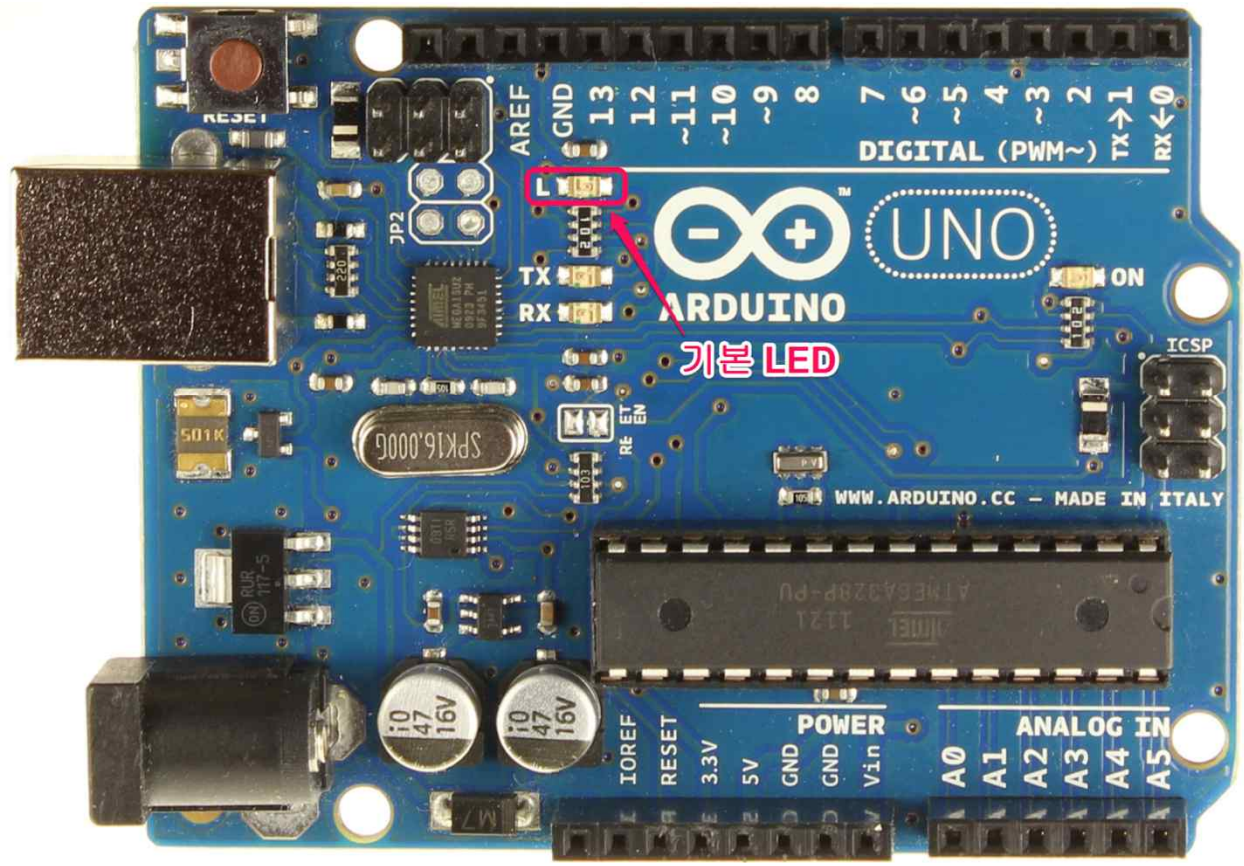
다음으로 시리얼 포트를 선택합니다. 메뉴에서 도구-시리얼 포트를 선택하면 USB로 연결된 장치들이 위와 같이 표시됩니다. 여기서 아두이노에 해당하는 것을 선택합니다. 어떤 것이 아두이노인지 헷갈린다면 먼저 연결하기 전에 시리얼 포트 목록을 보고 연결한 다음에 새로 생긴 것을 선택하면 됩니다. 맥과 리눅스의 경우에는 시리얼 포트가 "/dev/tty.usbmodem"이라고 시작하는 것이 아두이노입니다.

# 2 기본 LED 깜빡이기

## | 기본 LED 깜빡이기

아두이노 연결이 완료됐다면 이제 아두이노에 있는 기본 LED를 껐다 켜는 것을 배워봅시다.

## | 레시피



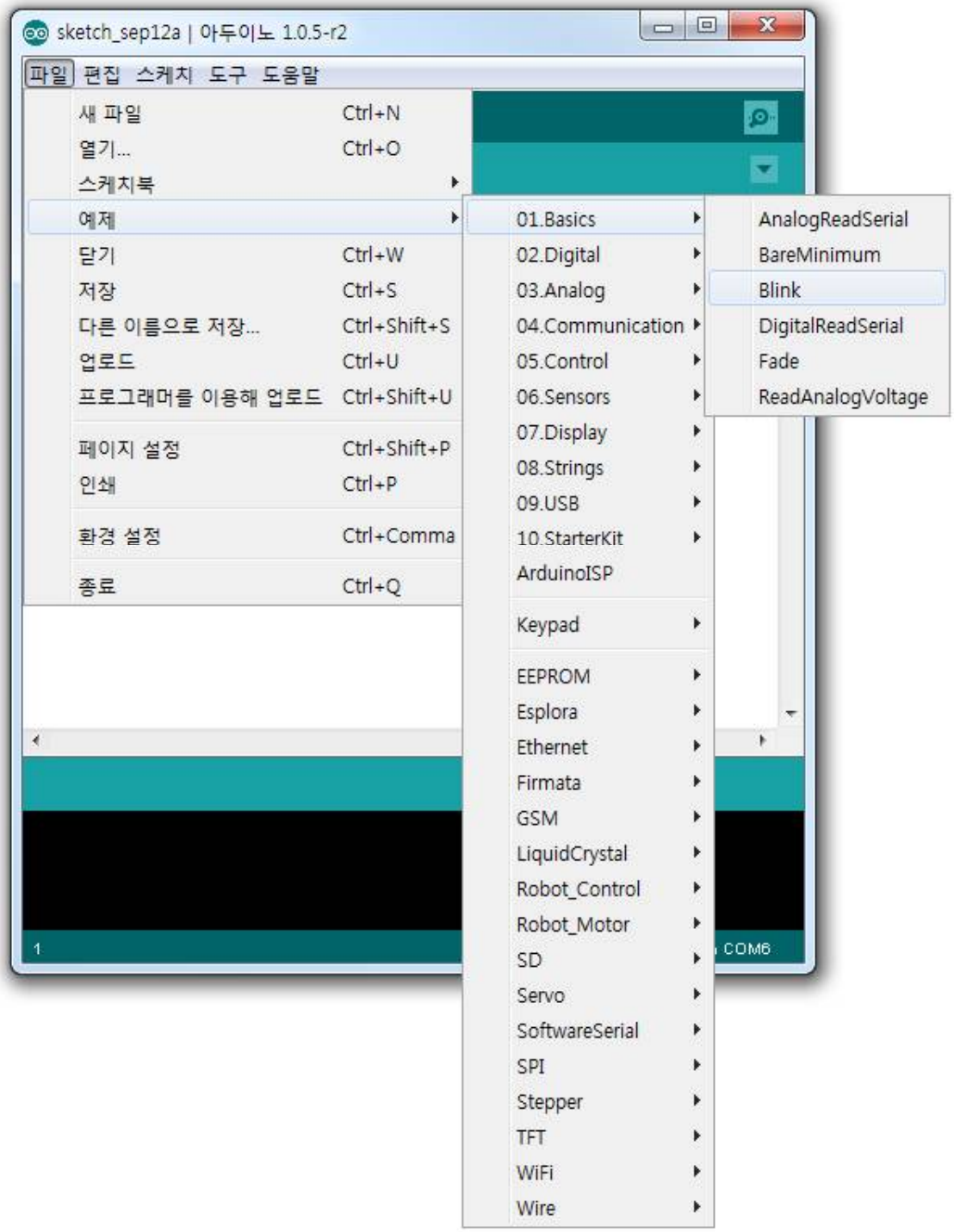

<그림 4> 아두이노 LED 깜빡이기

아두이노 UNO에 보면 그림과 같이 기본 LED가 있습니다.
기본 LED만 사용할때는 다른 부품은 필요하지 않습니다.

# 2 기본 LED 깜빡이기

## | 아두이노 코드 작성하기

코드는 아두이노 IDE에 있는 예제를 사용합니다.




<그림 5> 아두이노 코드 작성하기

# 2 기본 LED 깜빡이기

메뉴에서 파일-예제-01.Basics-Blink를 선택합니다.

<코드 1> Blink 예제

```
/*
  Blink
  Turns on an LED on for one second, then off for
one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino
boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press
reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again
forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on
(HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by
making the voltage LOW
  delay(1000);               // wait for a second
}
```

아두이노 IDE는 C와 C++이라는 프로그래밍 언어를 이용해 코드를 작성합니다. 혹시나 C++를 몰라도 걱정할 필요없습니다. 아두이노 IDE는 C++을 잘 모르는 사람들도 쉽게 프로그램을 작성할 수 있도록 만들어져 있습니다.

코드에 보면 **/* */**, **//**과 같이 표시된 것을 볼 수 있습니다. 이 기호들은 주석 기호를 뜻합니다. 주석이란 사람이 볼때는 글로 표시가 되어있지만 아두이노나 컴퓨터 쪽에서는 보여지지 않는 것을 말합니다. 주로 프로그램에 관련된 중요한 정보나 메모를 기록할때 사용합니다.

# 2 기본 LED 깜빡이기

**//**
이 기호가 왼쪽에 적혀있으면 기호 우측에 있는 글자들이 주석이 됩니다.

**/*  */**
이 기호들은 항상 같이 사용해야 합니다. 이 기호들 사이에 있는 글자들은 주석이 됩니다. 주로 여러줄에 걸쳐서 주석을 작성할때 사용합니다.

코드에 대해 하나 하나 살펴보겠습니다.

**int led = 13;**

중간에 int라고 적힌 것을 볼 수 있습니다. int는 변수의 한 종류를 뜻합니다. 프로그램에서는 숫자나 문자와 같은 다양한 값을 사용하는데, 변수란 바로 이러한 값을 담아놓고 사용하는 일종의 그릇과도 같습니다. 더 나아가 숫자를 담는 변수는 소숫점이 있는 실수를 담는 것과 소숫점이 없는 정수를 담는 것으로 구분됩니다. int는 바로 정수를 담는 변수의 한 종류입니다.

변수를 사용하는 방법은 **[변수형] [변수명]과 같이 입력해서 사용합니다.**

**int led = 13;**

위 문장은 **led라는 이름을 가진 int형 변수를 선언한 것입니다.**

**int led = 13;**

가운데 보면 부등호(=)기호를 볼 수 있습니다.

**A = B**

프로그래밍에서 부등호 기호 하나를 사용한다는 것은 오른쪽에 있는 값을 왼쪽에 집어넣으라는 뜻입니다. 즉, 위 문장으로 보면 B의 값을 A에 넣게 됩니다..

**int led = 13;**

여기서는 **13이라는 값을 led라는 변수에 넣습니다.**

**int led = 13;**

줄의 마지막을 보면 세미콜론(;) 기호를 볼 수 있습니다. 이 기호는 컴퓨터에게 한 줄이 끝났다고 알려주는 것입니다. C++ 에서는 컴퓨터가 스스로 줄의 시작과 끝을 알 수가 없습니다. 따라서 세미콜론 기호를 통해 알려줘야 합니다. 간혹 처음 코드를 작성할때 이 기호를 마지막에 빼먹어서 에러가 나는 경우가 많습니다.

**int led = 13;**

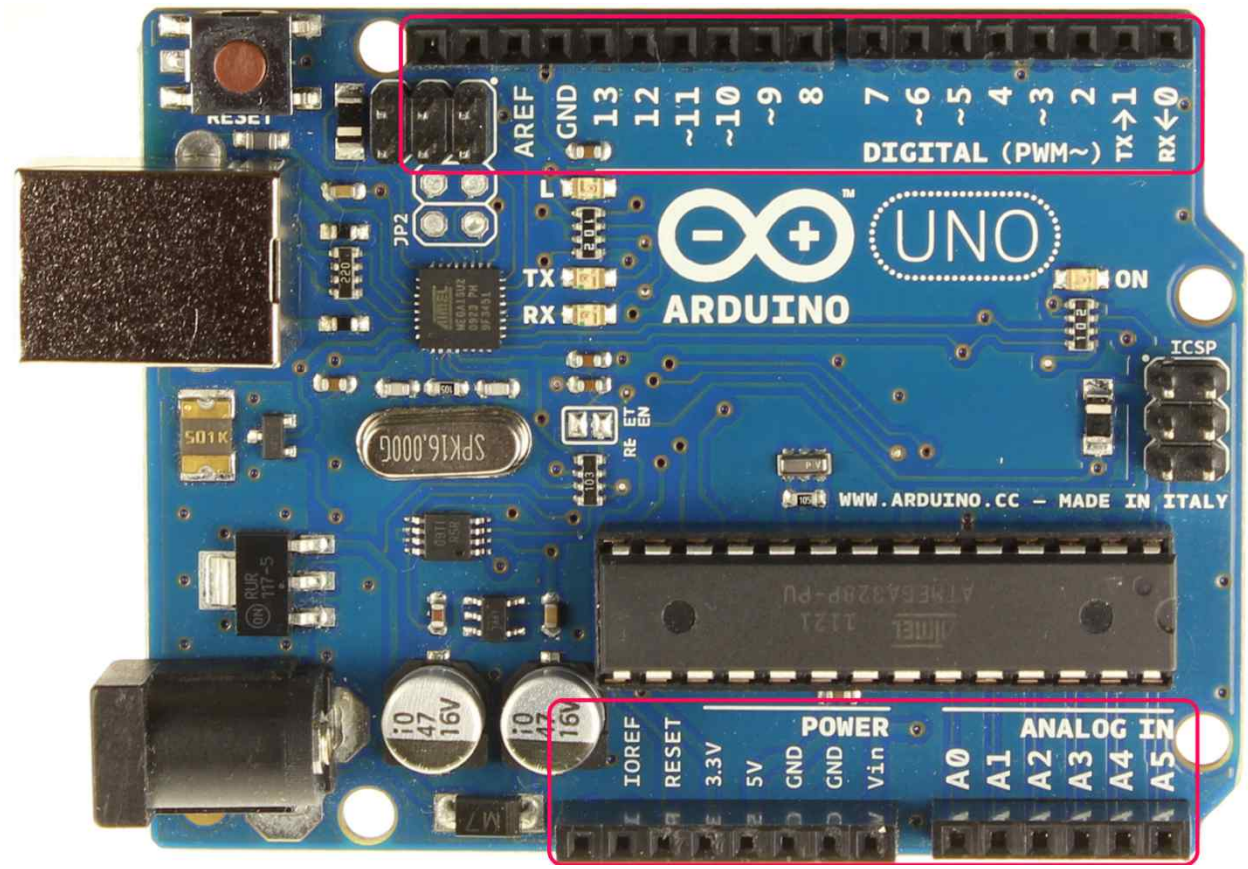
위 코드를 다음과 같이 두 줄로 나누어 볼 수도 있습니다.

**int led;**
**led = 13;**

변수를 먼저 선언하고 나중에 값을 할당하는 것입니다.

**pinMode(핀 번호, 모드)**

setup부분을 보면 pinMode라고 적힌 것을 볼 수 있습니다. pinMode는 아두이노 핀의 모드를 설정하는 명령어입니다. 아두이노 핀의 모드를 설정한다는 것은 아두이노 핀을 신호를 받는 용도로 사용할지 아니면 신호를 보내거나 명령하는 용도로 사용할지 결정하는 것입니다.




<그림 6> 아두이노 pinMode

위에 표시되어있는 것이 아두이노 핀들입니다. 설정하고자 하는 핀 번호를 보드에서 확인하고 그 번호를 pinMode 명령어 첫번째 부분에 입력합니다. 만약 신호를 받는 용도로 사용하고 싶다면 pinMode 명령어 두번째 부분에 대문자로 INPUT을 입력하고, 반대로 신호를 보내는 용도로 사용하는 경우에는 대문자로 OUTPUT이라고 적어줍니다.

# 2 기본 LED 깜빡이기

**pinMode(led, OUTPUT);**

<코드 1>에 위와 같이 적혀있는데, 앞서 배운 변수를 사용한 것이기 때문에 실제로는 13이라고 입력한 것과 같습니다. 바로 13번 핀의 모드를 설정한다는 뜻입니다.

**digitalWrite(핀 번호, 전압);**

다음으로 loop부분을 보면 digitalWrite라고 적힌 것을 볼 수 있습니다. digitalWrite는 해당 핀 번호의 전압을 0V 또는 5V로 설정하는 겁니다. 쉽게 말하면 전원을 연결하거나 끊거나 하는 것과 같습니다. 첫번째 부분에 핀 번호를 입력하고 두번째 부분에 0V인 경우 LOW, 5V인 경우 HIGH라고 입력합니다.

**digitalWrite(led, HIGH);**

위 코드는 13번 핀의 전압을 5V로 설정하라는 뜻으로 전기를 연결한다고 이해하셔도 좋습니다. 13번 핀에 전기가 연결되면 기본 LED의 불이 켜집니다. 바로 기본 LED가 13번 핀에 연결되어있기 때문입니다.

**delay(1000);**

다음으로 delay는 아두이노를 일정시간 동안 멈출때 사용하는 명령어입니다. 명령어 안에 써주는 숫자의 단위는 밀리초로 1000분의 1초를 뜻합니다. 따라서 1000을 입력한다는 것은 1초동안 멈추라고 하는 것과 같습니다.

<코드 2> 설명이 표시된 Blink 코드

```
int led = 13; // led라는 int형 변수를 선언하고, 13이란 값을 집어넣었습니다.

void setup() { // 아두이노가 켜질 때 한번 실행됩니다.
  pinMode(led, OUTPUT); // led 핀의 모드를 출력으로 설정합니다.
}

void loop() { // setup부분이 실행되고 계속 실행됩니다.
  digitalWrite(led, HIGH); // led 핀의 전압을 5V로 설정합니다.
  delay(1000); // 1초 멈춥니다.
  digitalWrite(led, LOW); // led 핀의 전압을 0V로 설정합니다.
  delay(1000); // 1초 멈춥니다.
}
```

앞서 설명한 것을 다시 정리하면 <코드 2>와 같습니다.

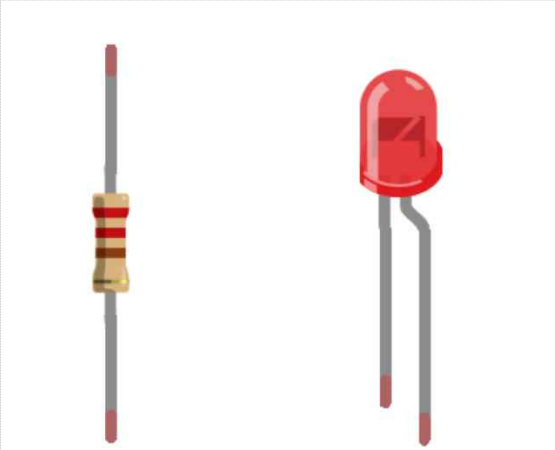
코드를 다 작성했다면 아두이노 UNO를 연결해 업로드합니다. 정상적으로 작동한다면 매 1초 마다 기본 LED가 깜빡입니다.

# 3 외부 LED 깜빡이기

## 외부 LED 깜빡이기

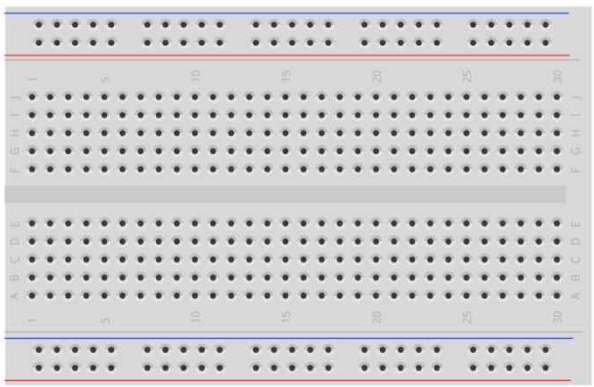
이번에는 기본 LED가 아닌 아두이노에 LED를 직접 연결해 깜빡여보겠습니다.

## 레시피



<그림 7> 220 ohm 저항, LED

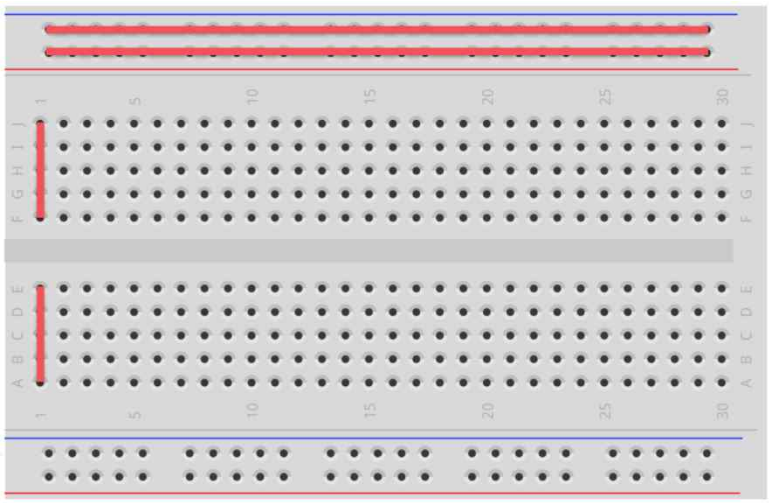
재료는 220 ohm 저항과 LED가 필요합니다. 저항을 사용하는 이유는 혹시나 너무 많은 전류가 LED에 흘러 망가지는 것을 막기위해 사용합니다. 저항은 전류의 일부를 열 에너지로 바꿔주기 때문에  전류가 적게 흐르도록 만듭니다.



<그림 8> 브레드보드

저항과 LED와 같은 부품을 아두이노 UNO에 연결하기 위해서는 위와 같은 브레드보드가 필요합니다.

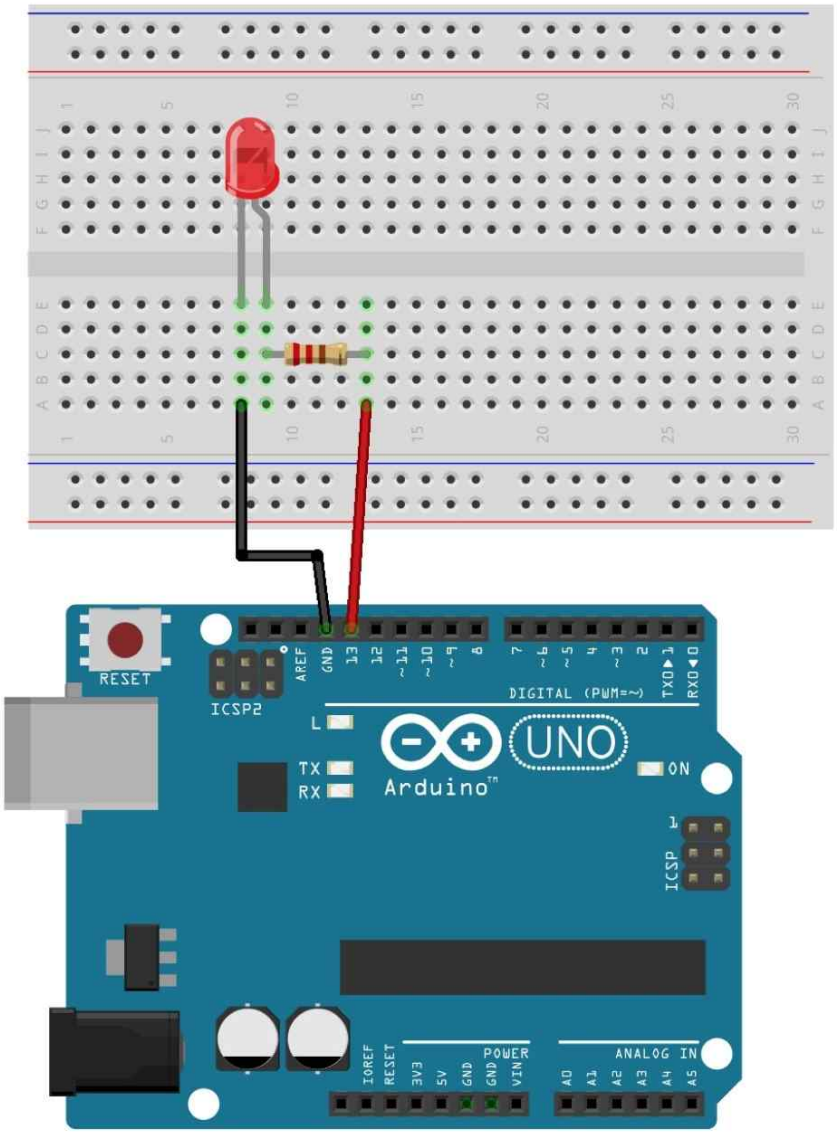
# 3 외부 LED 깜빡이기



<그림 8> 브레드보드

브레드보드는 위에 빨간 줄이 그어진 것과 같이 각 구멍들이 연결되어있습니다.
만약 가로에 긴 줄 중 한 구멍에 전기를 연결하면 나머지 같은 줄에 다른 구멍도 전기가 연결되고, 세로 짧은 줄에 연결하면 같은 세로 줄에 전기가 통하게 됩니다. 세로 줄의 경우 가운데 끊겨 있어서 위에 5개 구멍, 아래 5개 구멍끼리만 연결이 되어있습니다. 긴 브레드보드를 사용하는 경우 가로에 긴 줄이 전부 연결된 경우도 있고 중간에 끊어진 경우도 있기 때문에 확인해서 사용해야 합니다.

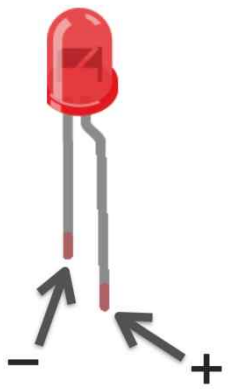
# 3 외부 LED 깜빡이기




<그림 9> 회로도

아두이노는 <회로도>와 같이 연결해줍니다. 13번 핀과 브레드보드를 연결합니다. 만약 13번 핀의 전압이 5V로 설정하면 13번 핀에서 빨간선을 타고 브레드보드 쪽으로 전류가 흐르게 됩니다. 빨간선이 연결된 같은 줄에 220 ohm 저항을 연결합니다. 저항에 반대편 줄에는 LED를 연결합니다.

# 3 외부 LED 깜빡이기




<그림 10> LED

LED는 다리가 긴 쪽이 양극(+)이고, 짧은 쪽이 음극(-)입니다. LED의 양극을 전류가 들어오는 방향에 정확히 꽂아줘야 합니다. 만약 방향을 반대로 꽂으면 LED가 고장날 수 있습니다.

아두이노의 GND 핀과 LED의 음극이 꽂힌 줄을 연결합니다. 만약 LED로 전류가 흐르면 음극으로 전류가 빠져나와 검은 선을 타고 GND핀을 통해 전류가 빠져나가게 됩니다.

### 아두이노 코드 작성하기

아두이노 코드는 앞에서 사용한 <코드 1>을 그대로 사용하면 됩니다. 문제가 없다면 매 초마다 외부 LED가 깜빡일 것입니다.

# 4 도전해보기

## 도전해보기

멈추는 시간을 다르게 설정할 수 있습니다. 더 빠르거나 더 느리게 깜빡여봅니다.

13번 핀이 아닌 다른 핀에 연결해봅니다.

LED의 갯수를 늘려 깜빡여봅니다.

# 5 참조

## 참조

**아두이노 함수 및 상수 설명 페이지** : http://arduino.cc/en/Reference/HomePage