



드론으로 배우는
프로그래밍 교실

Ch3. 아두이노 기초1 - 1



| | |
|-------------------------|----|
| 01 아두이노와 친해지기 | 01 |
| 아두이노 설치하기 | 02 |
| 마이크로 컨트롤러란? | 03 |
| 아두이노란? | 04 |
| 02 아두이노 스케치 | 06 |
| 아두이노 스케치란? | 07 |
| 스케치 구성 | 08 |
| 03 아두이노 프로그래밍 기초 | 11 |
| 함수란? | 12 |
| 프로그래밍 기본 상식 | 14 |
| 함수 작성 해보기 | 15 |
| 곱셈 함수 작성 해보기 | 16 |



드론으로 배우는
프로그래밍 교실

초판발행 2016년 9월 23일
지은이 최정애 | 펴낸이 최정애
펴낸곳 WHIT | 주소 안산시 한양대학교로55 창업보육센터 B01
전화 010-5125-2139

Published by WHIT. Printed in Korea
Copyright © 2016 최정애 & WHIT

이 책의 저작권은 최정애와 WHIT에 있습니다.
저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

01 아두이노와 친해지기

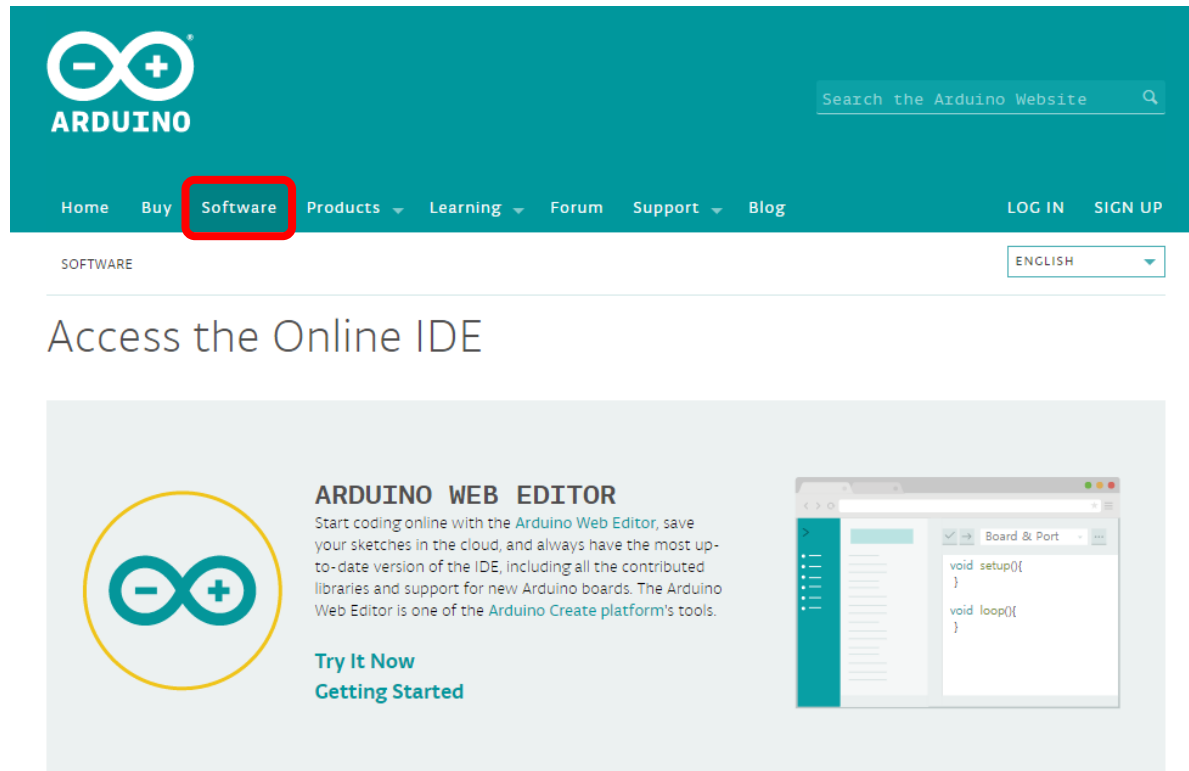


아두이노는 하드웨어를 다루기 쉽게 해주는 관련 도구 및 개발 환경입니다. 아두이노 이전에는 하드웨어에 대해 쉽게 접근하기 어려웠는데, 아두이노의 등장으로 손쉽게 하드웨어 제품을 만들 수 있게 되었습니다. 납땜할 필요 없이 핀을 이용하여 쉽게 하드웨어를 제작하고, 레지스터 설정이 필요 없이 쉽게 프로그래밍이 가능합니다. 아두이노와 함께 DIY세계로 빠져 봅시다.

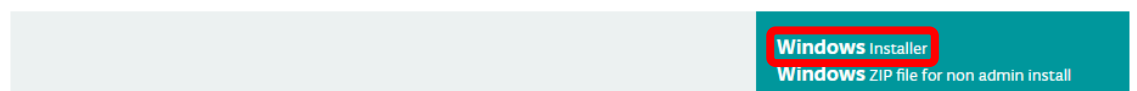
아두이노 스케치 설치하기

아두이노
스케치
설치

- 1 <https://www.arduino.cc/> 사이트에 접속합니다.
- 2 Software 탭의 Windows Installer를 클릭합니다.



Download the Arduino IDE



<그림1-1> 아두이노 홈페이지

- 3 **JUST DOWNLOAD** 를 클릭합니다.
- 4 다운받은 프로그램을 실행합니다.
- 5 License Agreement에서 **I Agree** 를 클릭합니다.
- 6 Installation Options에서 **Next >** 를 클릭합니다.
- 7 Installation Folder에서 **Install** 를 클릭합니다.

꿀TIP

드라이버 설치

설치 도중 나오는
드라이버는 다 설치
및 허용 해야합니다.

마이크로 컨트롤러란?

마이크로 컨트롤러

마이크로 컨트롤러는 보통 검은색의 작고 네모난 칩으로, 컴퓨터와 같은 역할을 합니다. 그림과 같이 생겼으며, 초소형 컴퓨터 역할을 하여 보통 작은 전자제품에 들어가 정해진 작업을 반복합니다.



<그림1-2> 마이크로 컨트롤러(atmega32u4)

활용분야

마이크로 컨트롤러는 생활 및 산업 전 영역에서 사용되고 있으며, 무궁무진한 응용이 가능합니다. 여러분의 집에 있는 세탁기, 라디오, 선풍기 등 다양한 전자제품에 사용됩니다.



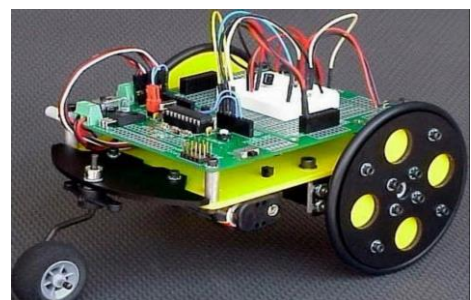
<그림1-3> 의료용 기구'



<그림1-4> 선풍기



<그림1-5> 리모콘



<그림1-6> 라인 트레이서

아두이노란?

아두이노
정의

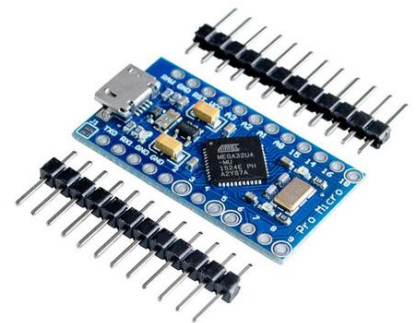
아두이노는 오픈 소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성 된 보드(상품)와 관련 개발 도구 및 환경을 말합니다.

쉽게 말해 마이크로 컨트롤러를 간편히 사용할 수 있게 제공되는 키트입니다.

아래 사진은 아두이노의 한 종류인 아두이노 우노와 아두이노 프로 마이크로입니다.



<그림1-7> 아두이노 우노



<그림1-8> 아두이노 프로 마이크로

링크 : <https://www.youtube.com/watch?v=Q4YeUspAjuU>

아두이노
DIY

아두이노는 마이크로컨트롤러를 간편히 제어할 수 있는 환경을 제공해주어 많은 사람들에게 환영을 받았습니다.

아두이노 DIY도 굉장히 많이 있어서, 여러분 또한 여러분의 아이디어를 현실로 만들어 낼 수 있을 겁니다.

꿀TIP

DIY란?

Do It Yourself의 줄임말로, 스스로 만들어 내는 것을 의미합니다.



링크 : https://www.youtube.com/watch?v=t_Za7G38YFU

<그림1-9> 아두이노 RC카



아두이노 활용

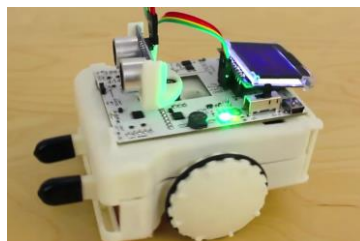
아두이노 스마트 전등



<그림1-11> 아두이노 스마트 전등

출처 : <https://www.youtube.com/watch?v=PHPU-xoFYgA>

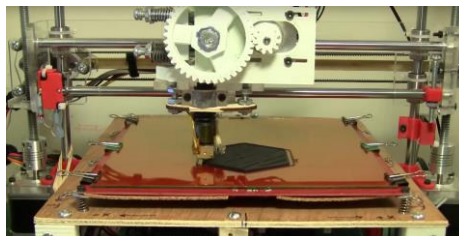
아두이노 로봇



<그림1-12> 아두이노 로봇

출처 : https://www.youtube.com/watch?time_continue=42&v=2igPl-MTfTQ

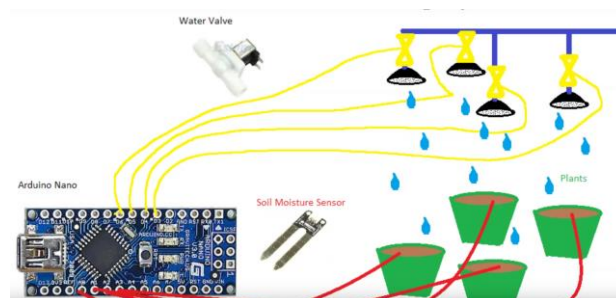
아두이노 3D 프린터



<그림1-13> 아두이노 3D 프린터

출처 : https://www.youtube.com/watch?time_continue=127&v=pyFZKc356eQ

아두이노 농업 자동화



<그림1-14> 아두이노 농업 자동화

출처 : <https://www.youtube.com/watch?v=z0WRY56i8RA>

02 아두이노 스케치



아두이노 스케치는 아두이노에 들어갈 소프트웨어를 구성할 수 있는 IDE입니다. IDE란 Integrated Development Environment의 약자로 통합 개발 환경라는 뜻입니다. 통합 개발 환경이란 개발을 하는데 필요한 것들을 지원해주는 것입니다.

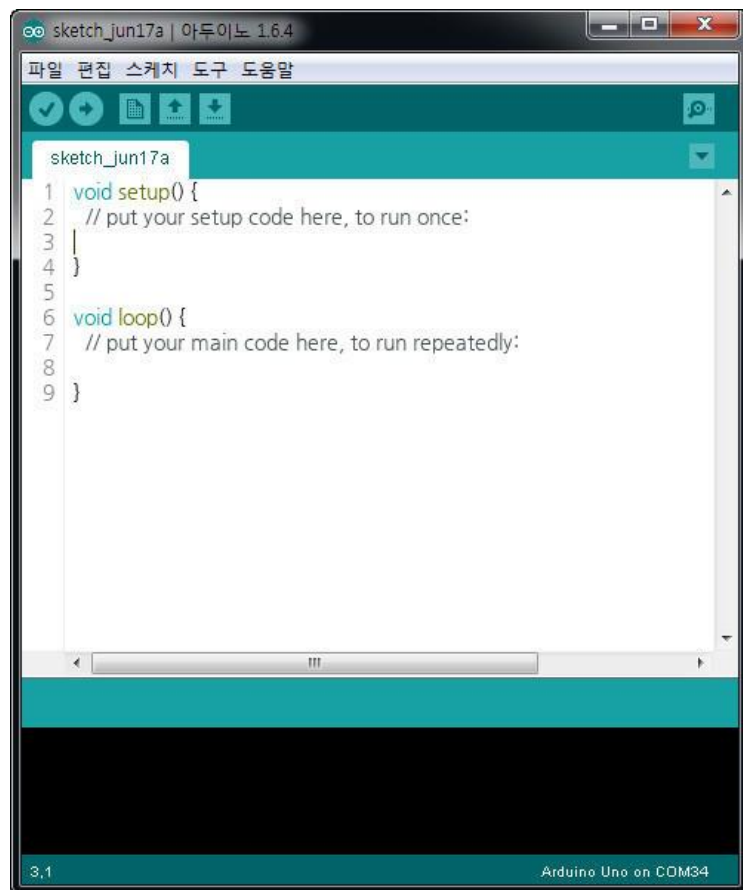
아두이노 스케치를 통해서 여러분들은 프로그래밍을 하게 되니까, 스케치를 잘 다룰 줄 알아야 합니다.

아두이노 스케치란?

아두이노
스케치

아두이노 스케치는 아두이노에 프로그래밍을 하기 위한 도구입니다. 메모장이 메모를 하기 위한 도구인 것처럼 말입니다.

메모장에 메모를 하듯 스케치에 프로그래밍을 하게 되는데, 이 때 이 스케치를 IDE(Integrated Development Environment: 통합 개발 환경)이라고 합니다.

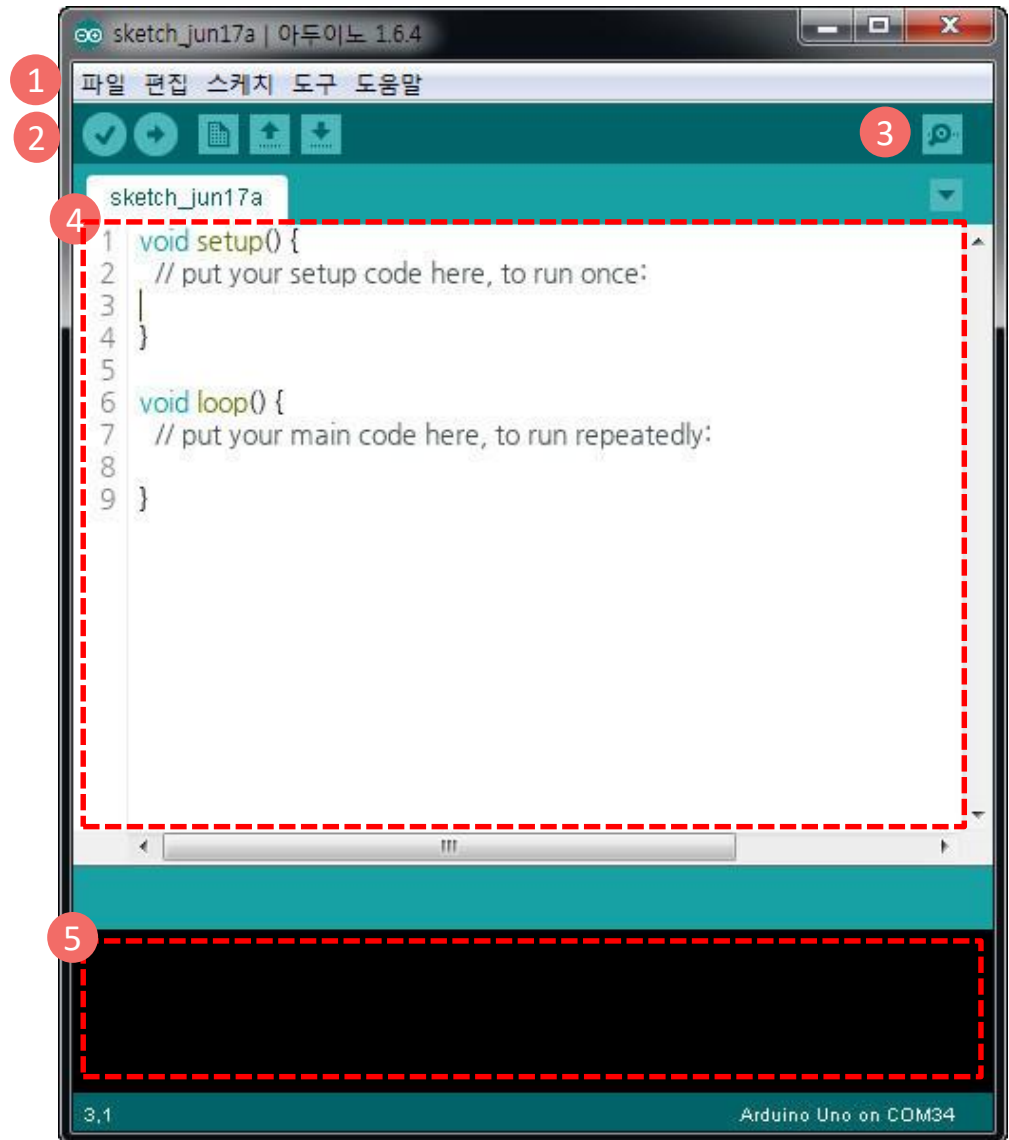


<그림2-1> 아두이노 스케치

프로그래밍
업로드

앞으로 여러분은 아두이노 스케치를 이용해 프로그래밍을 하게 될 것입니다. 여기서 만들어진 프로그램이 실제 아두이노 프로 마이크로에 들어가게 될 것입니다. 이렇게 프로그램을 아두이노 보드에 집어 넣는 것을 업로드라고 합니다.

스케치 구성



<그림2-2> 스케치 구성

- 1 메뉴 바 : 파일을 불러오거나 저장하고, 설정 등을 할 수 있습니다.
- 2 툴 바 : 코드를 컴파일하고 보드에 업로드 합니다.
- 3 시리얼 모니터 : 아두이노와 메시지를 주고 받을 수 있습니다.
- 4 텍스트 에디터 창 : 프로그램을 만들 수 있습니다.
- 5 콘솔 창 : 업로드 결과 및 에러 등을 확인할 수 있습니다.

메뉴 바

메뉴바에는 File, Edit, Sketch, Tools, Help가 있습니다.

- File : 새 파일 작성, 열기, 저장, 닫기 및 텍스트 되돌리기 등의 기능이 있습니다. 중요 기능은 예제 불러오기인데, 각종 예제 파일들을 가져올 수 있어서 유용합니다.
- Edit : 텍스트 에디터 창에서 글을 잘못 썼을 경우 되돌리기, 주석 등의 편집 관련 기능을 제공합니다.
- Sketch : 컴파일 및 업로드 등의 기능을 제공합니다.
- Tools : 보드 선택, 포트 선택 등의 기능을 제공합니다.
- Help : 도움을 얻을 수 있습니다.

툴 바

툴 바에서는 컴파일, 업로드, 새 파일, 열기, 저장 등의 기능을 아이콘으로 제공하여 편리하게 기능을 이용할 수 있게 되어 있습니다. 보통 컴파일이나 업로드의 경우 메뉴바가 아닌 툴 바를 이용합니다.

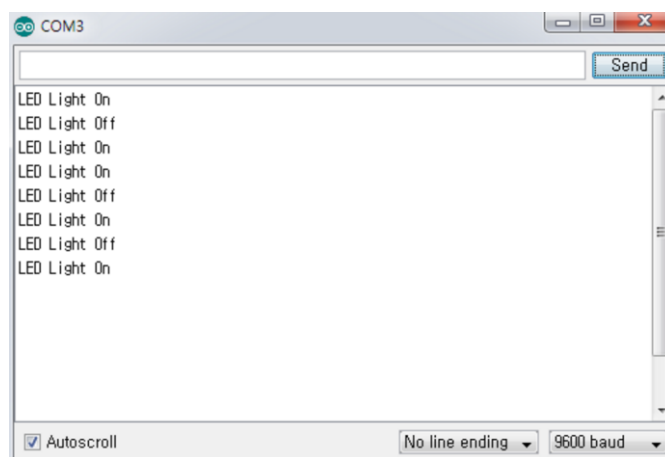


<그림2-3> 툴 바

시리얼 모니터



시리얼 모니터는 아두이노와 컴퓨터 간 통신을 하면서 서로 주고 받는 데이터를 눈으로 확인할 수 있는 대화의 창입니다. 보통 아두이노에 있는 데이터가 정확한 값이 맞는지 체크할 때 쓰입니다.



<그림2-4> 시리얼 모니터

꿀TIP

컴파일이란?

컴파일은 작성한 프로그램의 정상 동작을 위해 에러가 있는지 확인하는 과정입니다.

텍스트 에디터 창



<그림2-5> 텍스트 에디터 창

꿀TIP

줄 번호 표시

파일 - 환경설정에서
줄 번호 표시 네모박스에
V체크를 합니다.

☒ 줄 번호 표시

텍스트 에디터 창에서는 소스코드를 작성하고 수정할 수 있습니다.
좌측에는 줄 번호가 매겨져 있는데, 코드의 위치를 파악하기 쉽게 해 줍니다.
들여쓰기 등의 기능을 지원 해 줍니다.

콘솔 창



<그림2-6> 콘솔 창

콘솔 창에서는 스케치를 컴파일한 결과나 에러 등을 볼 수 있습니다.
컴파일이나 업로드 등에서 발생하는 문제에 대해 알려줌으로 해결 방안을 찾을 수 있습니다.

03 아두이노 프로그래밍 기초



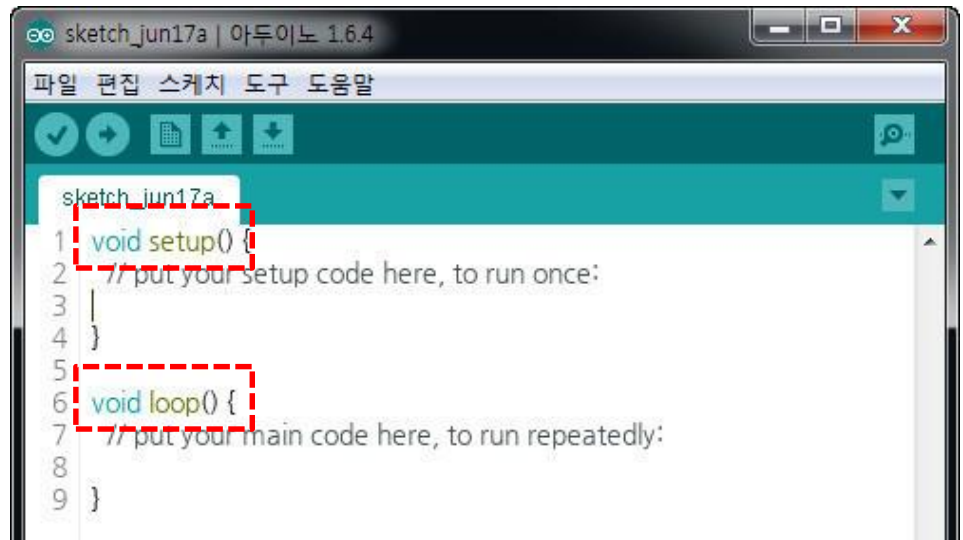
아두이노의 기본 함수는 프로그램의 본격적인 시작 전 설정을 해 주는 `setup()`과 반복적인 일을 수행하는 `loop()`로 구성되어 있습니다.

아두이노의 기초 함수에 대해 알아보고, 함수가 무엇인지, 함수는 어떻게 구성되는지 배워 봅시다.

프로그래밍의 기초적인 상식으로 중괄호, 세미콜론, 블록 주석, 문장 주석에 대해 알아 봅시다.

함수란?

아두이노 기초 함수

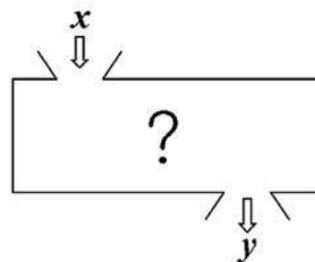


<그림3-1> setup()과 loop()

Setup()과 loop()는 아두이노의 기초 함수입니다.

함수란?

함수는 입력에 대해 어떤 처리를 거쳐 나오는 출력의 과정을 의미합니다.



<그림3-2> 함수의 입력, 처리, 출력

함수는 코드의 뭉텅이로 이름을 가지고 있고, 명령어로 구성된 블록입니다. 함수는 자신의 이름이 불렸을 때 실행됩니다.

이미 만들어져 있는 함수도 있고, 우리가 함수를 만들 수도 있습니다. 함수는 반복되는 작업을 수행할 수 있고, 코드를 깔끔하게 볼 수 있게 해 줍니다.

함수 구조

예를 들어 덧셈을 하는 함수가 있다면 입력으로 두 숫자를 받은 후 더하기 처리를 하여 결과를 출력합니다.

함수 이름
add

정수 a, b 입력

정수 출력

```
int add(int a, int b){  
    return a+b;  
}
```

a, b의 합 출력

<그림3-3> 함수의 구조

setup()과 loop() 앞에 있는 void는 아무것도 출력하지 않는다는 뜻입니다.

함수 사용

실제로 함수를 사용할 경우 함수의 이름과 입력 값을 괄호안에 적으면 됩니다.

```
sketch_feb02a  
1 void setup() {  
2   // put your setup code here, to run once:  
3   add(4, 5);  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10  
11 int add(int a, int b){  
12   return a+b;  
13 }  
14
```

<그림3-4> 함수의 실제 사용

기타

```

1 void setup() {
2   // put your setup code here, to run once:
3 }

```

<그림3-5> 프로그래밍 기본 상식

- ◆ { } 는 중괄호로 함수의 범위를 지정해 줍니다. 만약, 중괄호가 없을 경우 어디까지가 함수의 끝인지 알 수 없게 됩니다.
중괄호는 { 로 시작하여 }로 끝이 나는데, 항상 쌍을 이룹니다. 만약 중괄호가 한 짝 밖에 없다면 에러가 나게 됩니다.
- ◆ // 는 문장 주석입니다. 한 줄의 문장을 주석으로 처리하여 명령어처럼 실행되지 않게 합니다.
예를 들어
// 이렇게 되어 있으면 이 문장은 실행되지 않습니다.
- ◆ /* ... */ 는 블록 주석입니다. /*으로 시작하여 */로 끝나는데, 이 범위안에 있는 코드는 실행되지 않습니다. 코드에 대한 해석이나 의견을 적을 때 사용하게 됩니다.
/* 이렇게 블록 주석으로 쌓여 있으면
문장이 명령어처럼 실행되지 않습니다. */
- ◆ ; 는 세미콜론입니다. 세미콜론은 명령어 문장의 끝을 나타냅니다. 예를 들어 변수 x를 지정하고 x에 5를 집어 넣는 명령을 할 경우
int x = 5;
와 같이 문장을 작성하게 되는데, 세미콜론을 빼먹는다면 문장의 끝이 어딘지 모르게 돼 에러가 납니다.

뱀셈 함수
작성

다음 뱀셈 함수를 작성 해 봅시다.

ch3_1_3_sub

```

1 void setup() {
2     // put your setup code here, to run once:
3     sub(15, 9);
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8 }
9
10 int sub(int a, int b){
11     return a-b;
12 }

```

<그림3-6> 뱀셈 함수



작성이 완료 되면 좌측 상단의 컴파일 버튼을 눌러 틀린 부분이 있는지 검사합니다.

이상이 없으면 컴파일 완료가 컴파일창에 나타납니다.

컴파일 완료.

스케치는 프로그램 저장 공간 221995 바이트(51%)를 사용, 최대 434160 바이트, 전역 변수는 동적 메모리 31568바이트(38%)를 사용, 50352바이트의 지역변수가 남음

ESP8266 Module 80 MHz 40MHz DIO 115200 512K (64K SPIFFS) ok Disabled None on COM3

<그림3-7> 컴파일 완료

곱셈 함수
작성

Q 세개의 숫자를 받아서 곱하는 함수는 어떻게 만들까요?

ch3_1_3_sub_mul

```

1 void setup() {
2     // put your setup code here, to run once:
3     sub(15, 9);
4     mul(4, 6, 5);
5 }
6
7 void loop() {
8     // put your main code here, to run repeatedly:
9 }
10
11 int sub(int a, int b){
12     return a-b;
13 }
14
15
16
17

```

꿀TIP

sub, mul

sub는 subtraction
mul은 multiplication
의 줄임말입니다.
코딩에선 줄임말을 많이
사용합니다.

<그림3-8> 곱셈 함수 작성 해보기

곱셈 함수

A 세개의 숫자를 받아서 곱하는 함수 정답

ch3_1_3_sub_mul

```
1 void setup() {  
2   // put your setup code here, to run once:  
3   sub(15, 9);  
4   mul(4, 6, 5);  
5 }  
6  
7 void loop() {  
8   // put your main code here, to run repeatedly:  
9 }  
10  
11 int sub(int a, int b){  
12   return a-b;  
13 }  
14  
15 int mul(int a, int b, int c){  
16   return a*b*c;  
17 }
```

<그림3-9> 곱셈 함수

Q 나눗셈 함수도 생각해 봅시다!



WHIT