

개발 환경 구축

아두이노 종류

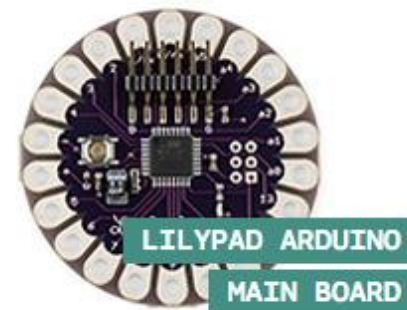
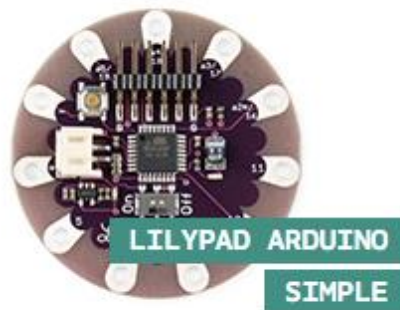
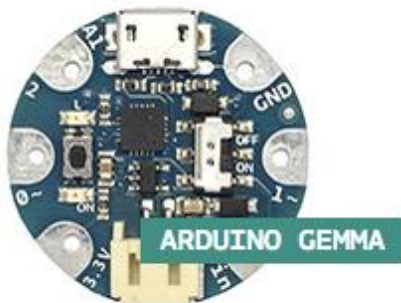
- 성능, 용도별 다양한 제품 군이 형성
- 가장 많이 쓰이는 건 UNO 버전

ENTRY LEVEL	UNO LEONARDO 101 ESPLORA MICRO NANO MINI MKR2UNO ADAPTER STARTER KIT LCD SCREEN
ENHANCED FEATURES	MEGA ZERO DUE MEGA ADK MO MO PRO MKR ZERO MOTOR SHIELD USB HOST SHIELD PROTO SHIELD MKR PROTO SHIELD 4 RELAYS SHIELD MEGA PROTO SHIELD MKR RELAY PROTO SHIELD ISP USB2SERIAL MICRO USB2SERIAL CONVERTER
INTERNET OF THINGS	YÚN ETHERNET TIAN INDUSTRIAL 101 LEONARDO ETH MKR FOX 1200 MKR WAN 1300 MKR CSM 1400 MKR1000 YUN MINI YÚN SHIELD WIRELESS SD SHIELD WIRELESS PROTO SHIELD ETHERNET SHIELD V2 CSM SHIELD V2 MKR IoT BUNDLE
EDUCATION	CTC 101
WEARABLE	GEMMA LILYPAD ARDUINO USB LILYPAD ARDUINO MAIN BOARD LILYPAD ARDUINO SIMPLE LILYPAD ARDUINO SIMPLE SNAP
3D PRINTING	MATERIA 101

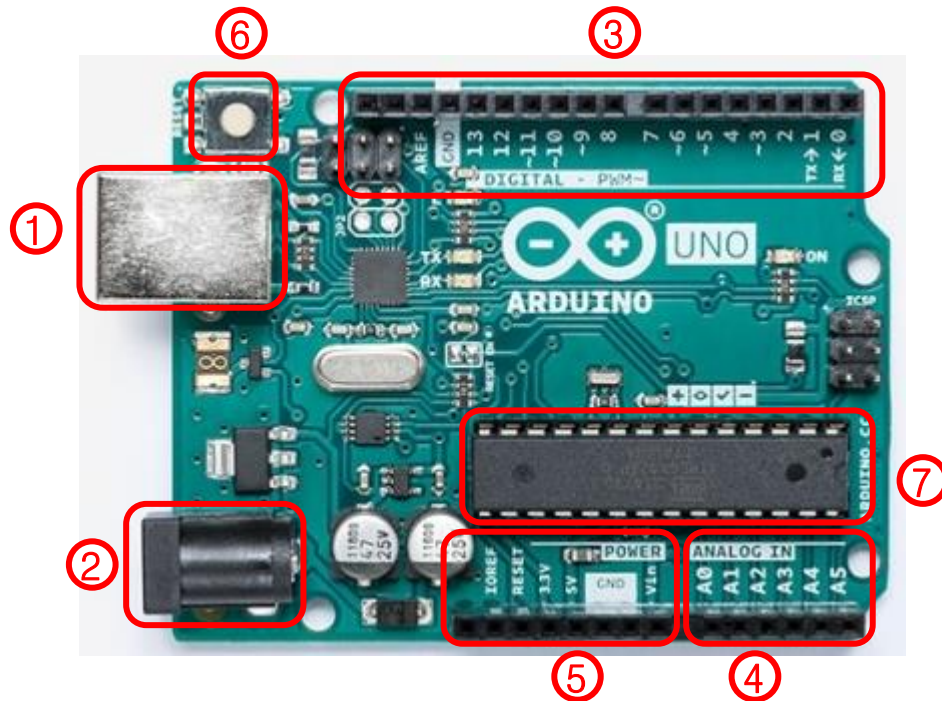
아두이노 종류



아두이노 종류



아두이노 Uno보드



- ① USB 커넥터
- ② 전원 커넥터
- ③ 디지털 핀
- ④ 아날로그 핀
- ⑤ 전원 핀
- ⑥ 리셋 스위치
- ⑦ MCU

아두이노 Uno보드

항목	값
MCU	ATmega328
동작 전압	5V
외부 입력 전압	7~12V
디지털 핀	14개
아날로그 핀	6개
최대 디지털 출력 전류	40mA
메모리	32KB
클럭 속도	16MHz

아두이노 개발 환경

■ IDE

- ❖ 통합개발환경(Integrated Development Environment)
- ❖ 에디터부터 컴파일 환경 및 다운로드 기능 까지 제공하는 통합환경

■ 스케치(Sketch)

- ❖ 아두이노 IDE(프로그램 형태)와 웹 에디터(웹페이지 이용)
- ❖ Arduino CC 에서 제공하는 IDE
- ❖ 모니터링 기능과 라이브러리 확장 기능 제공

실습 - 아두이노 설치하기 (1)

- 다운로드 사이트:
<https://www.arduino.cc/en/software>

Downloads

OS에 맞는 IDE 다운로드



Arduino IDE 2.0.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

1

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits

macOS

Release N

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **70,506,624** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

2

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

실습 - 아두이노 설치하기 (2)

3 Arduino IDE 설치

사용권 계약
Arduino IDE(을)를 설치하시기 전에 사용권 계약 내용을 살펴보시기 바랍니다.

사용권 계약 동의 사항의 나머지 부분을 보시려면 [Page Down] 키를 눌러 주세요.

Terms of Service

The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted.

내용에 동의하셨다면 '동의함'을 눌러 주세요, Arduino IDE(을)를 설치하기 위해서는 반드시 내용에 동의하셔야 합니다.

Arduino IDE 2,0,4

동의함 취소

4 Arduino IDE 설치

설치 옵션 선택
이 프로그램의 사용자는 누구인가요?

이 프로그램을 모든 사용자가 사용할 수 있도록 할 것인지 아니면 자신만 사용할 수 있도록 할 것인지 선택하십시오.

☐ 이 컴퓨터를 사용하는 모든 사람(모든 사용자)

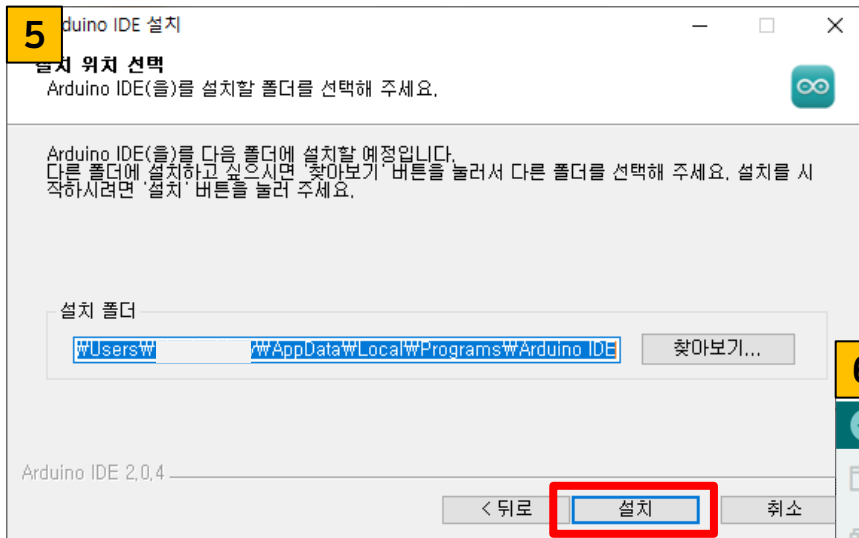
☒ 전용

현재 사용자만 새로 설치합니다.

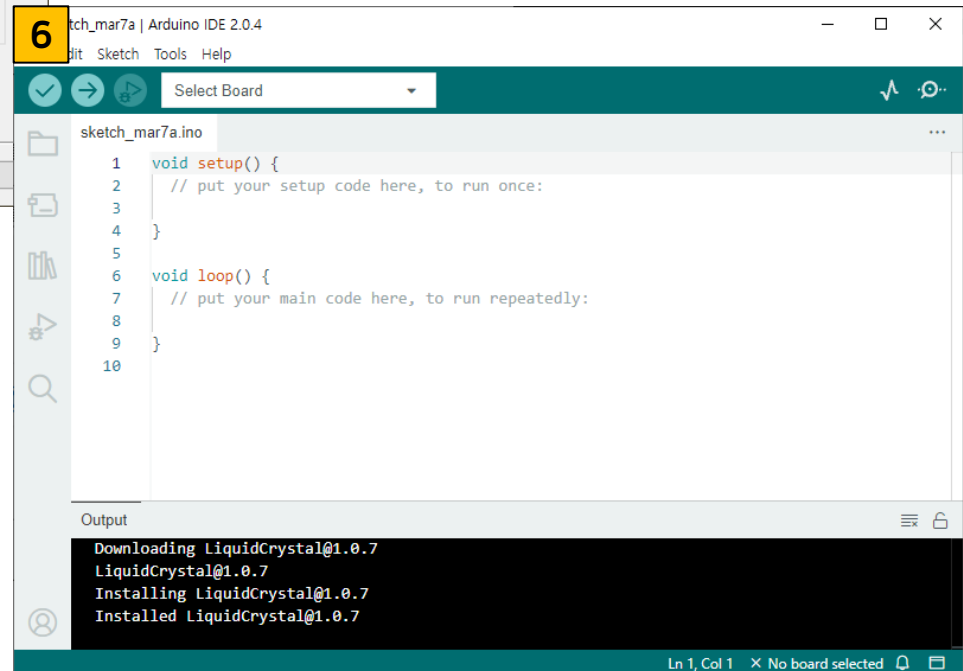
Arduino IDE 2,0,4

< 뒤로 다음 > 취소

실습 - 아두이노 설치하기 (3)

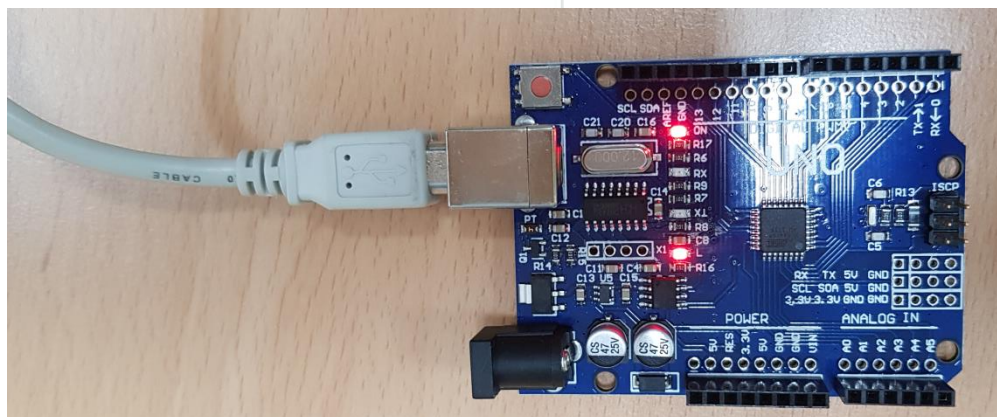
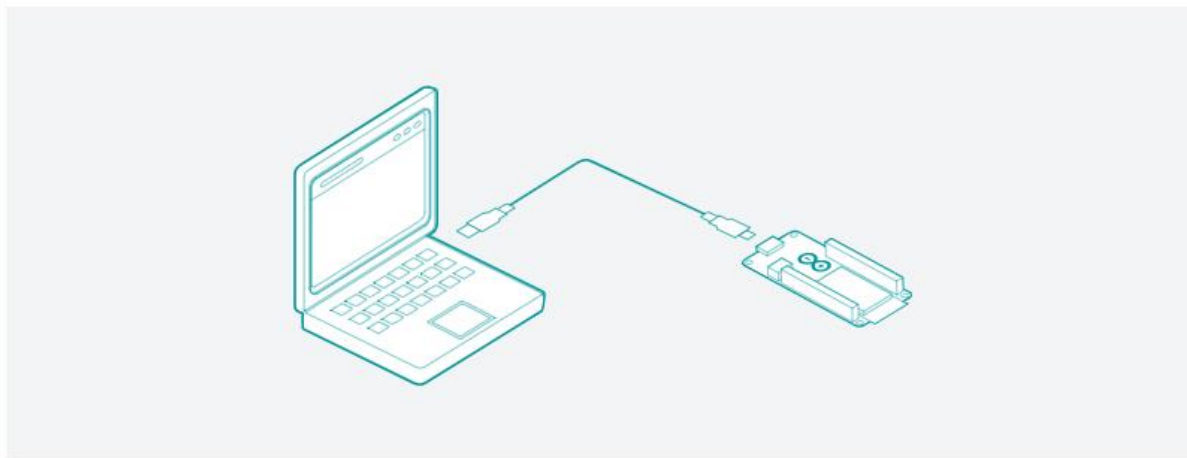


↓ 설치 완료 후 IDE 실행 화면



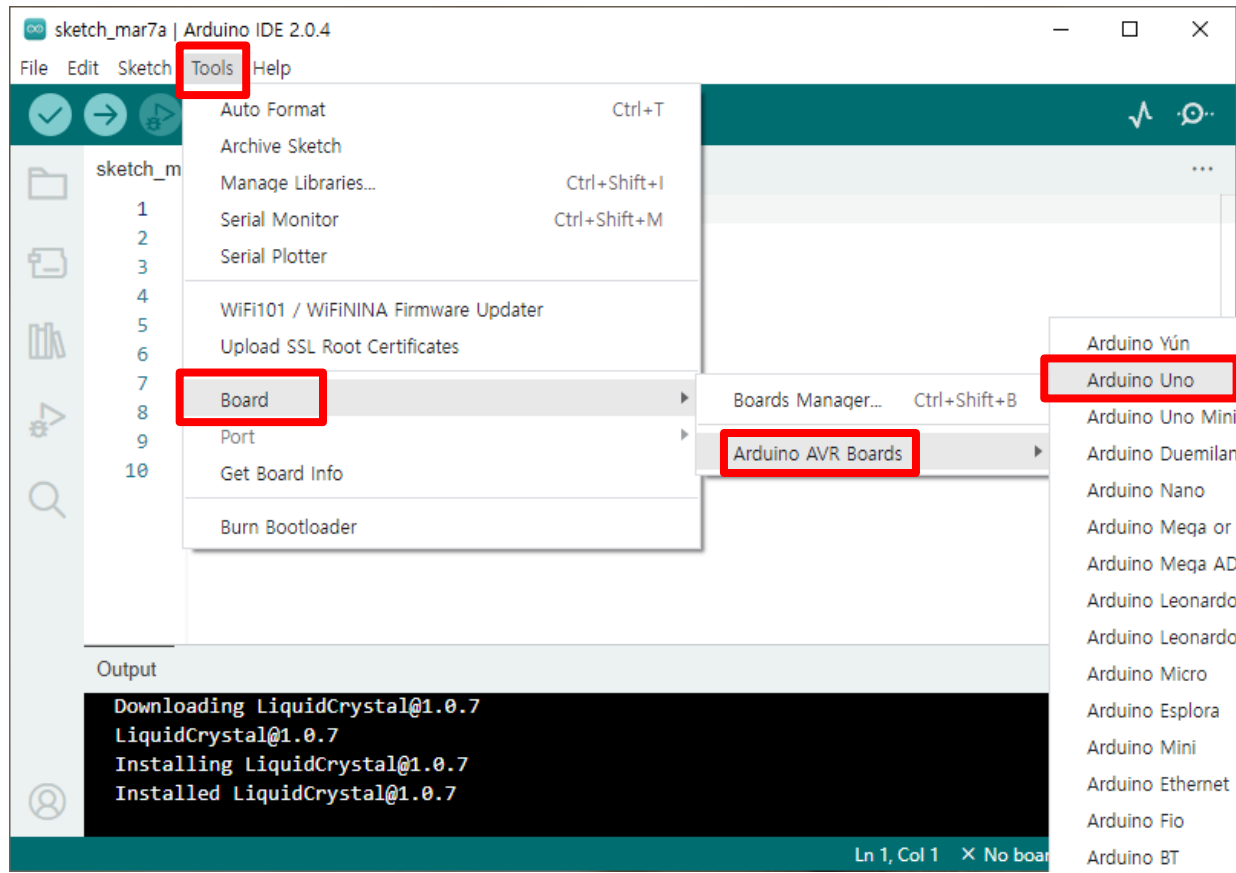
실습 - 아두이노 연결하기

- PC와 아두이노를 연결한다.



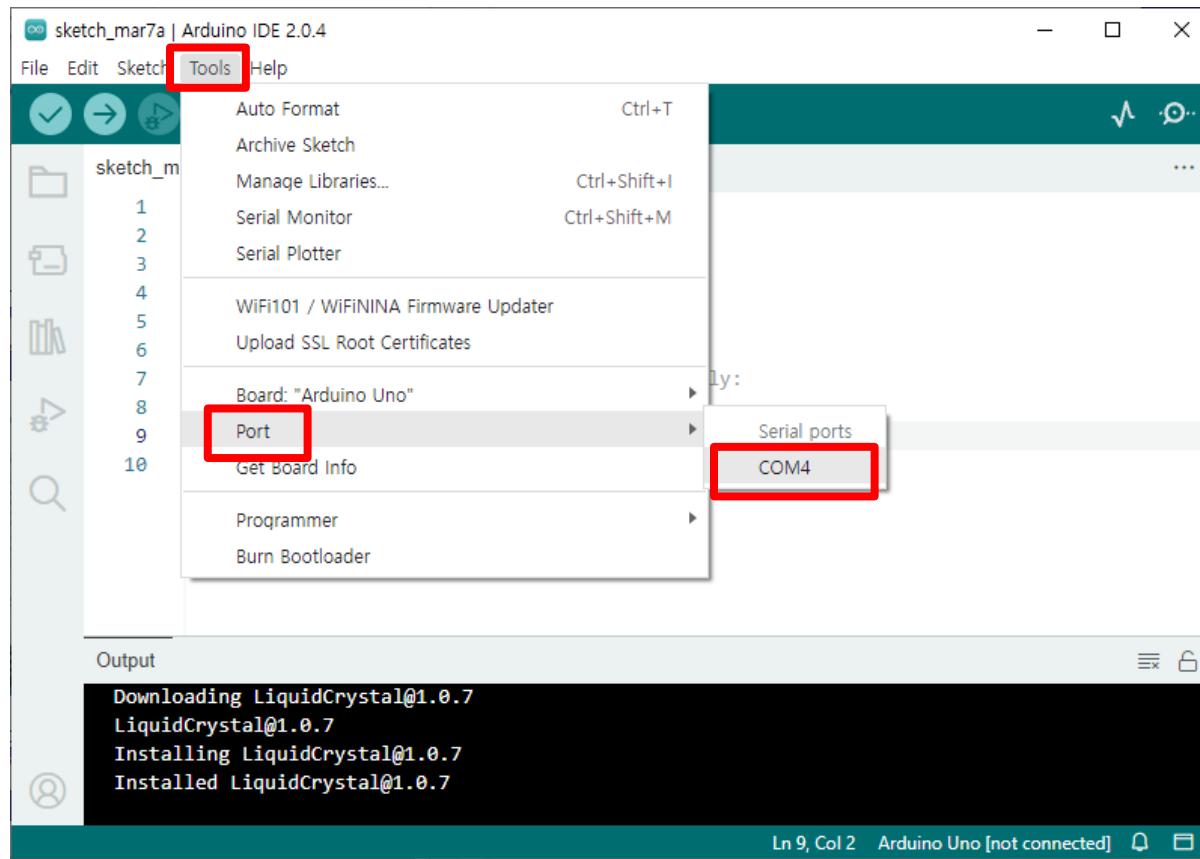
실습 - 아두이노 보드 설정

Tools -> Board -> Arduino AVR Boards -> Arduino Uno 선택



실습 - 아두이노 포트 지정

Tools -> Port -> COM4 선택(숫자는 다를 수 있습니다.)

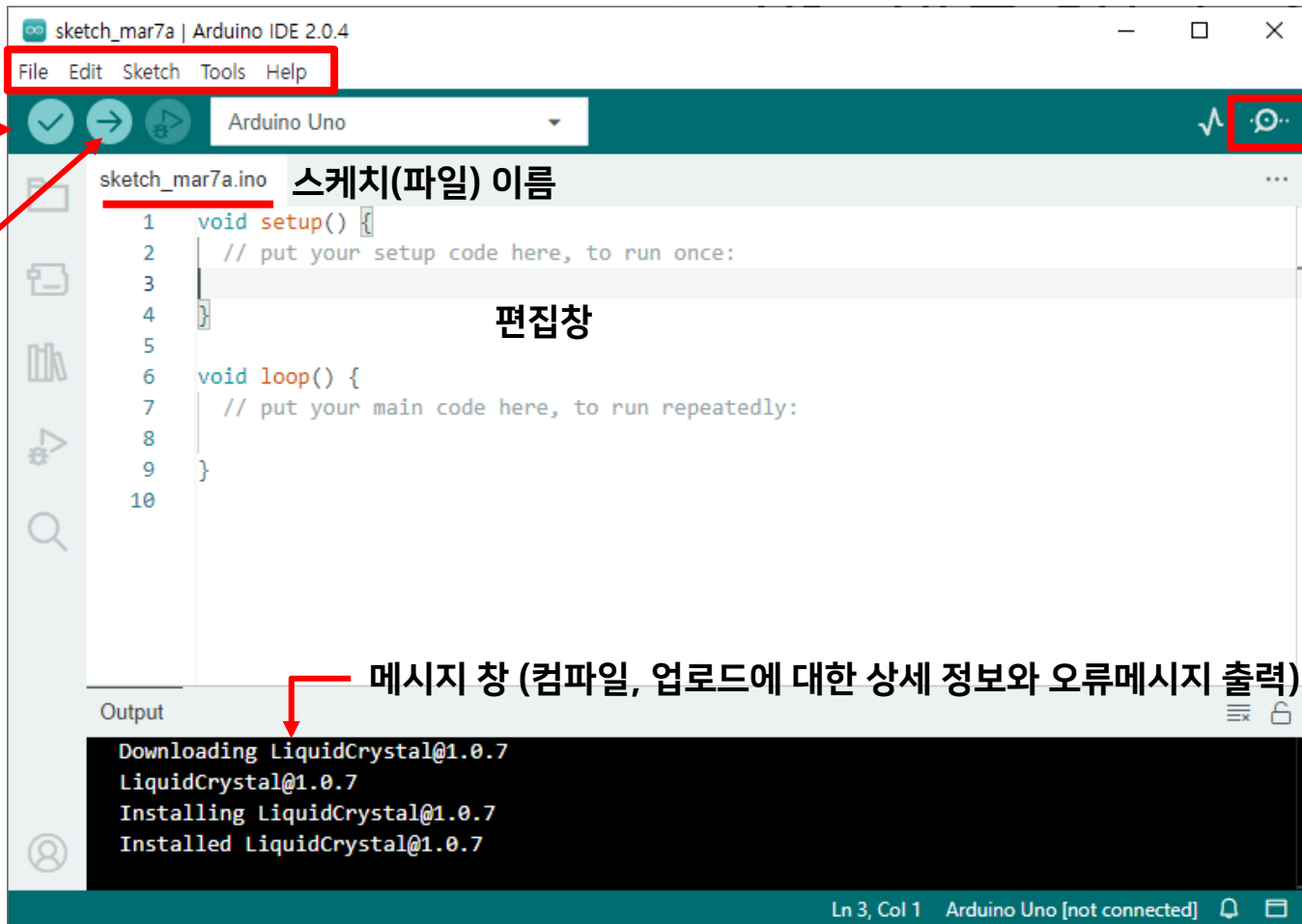


아두이노 IDE 기본 사용법

메뉴 바

확인
(컴파일)

업로드



시리얼 모니터

메시지 창 (컴파일, 업로드에 대한 상세 정보와 오류메시지 출력)

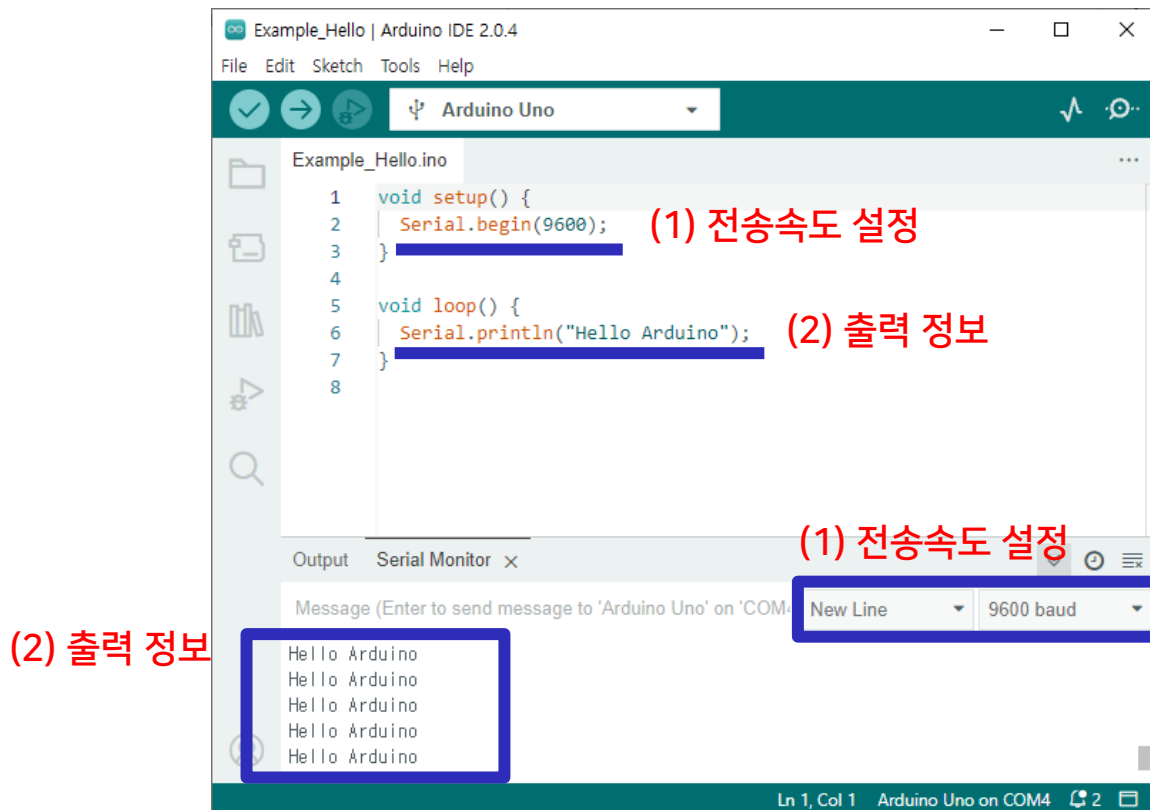
실습 - 아두이노 업로드 (1)

■ Hello World 를 작성해보자

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.write("Hello!");  
}
```


시리얼모니터 사용 예

- 아두이노에서 출력하는 문자열을 보여 준다.
- 디버깅 또는 통신 용도로도 사용할 수 있다.



시리얼모니터 사용 예

■ print 와 println 차이

Example_Hello | Arduino IDE 2.0.4

```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.print("Hello Arduino");
7 }
8

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM4') New Line 9600 baud

Hello ArduinoHello ArduinoHello ArduinoHello ArduinoHello ArduinoHello ArduinoHello ArduinoHello ArduinoHello ArduinoHello Arduino

Ln 6, Col 30 Arduino Uno on COM4

Example_Hello | Arduino IDE 2.0.4

```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.println("Hello Arduino");
7 }
8

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM4') New Line 9600 baud

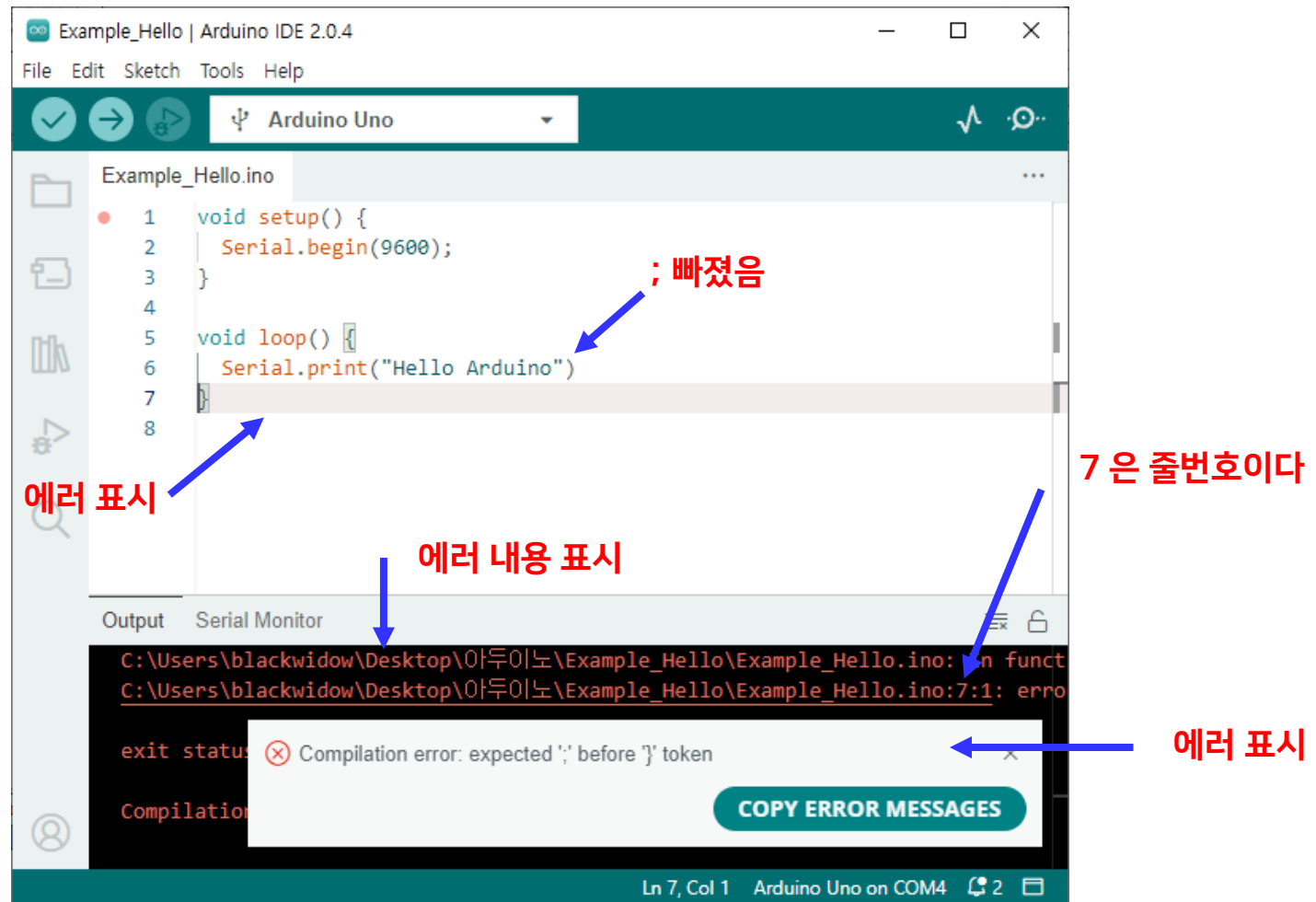
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino
Hello Arduino

Ln 1, Col 1 Arduino Uno on COM4

에러가 발생 했을 때

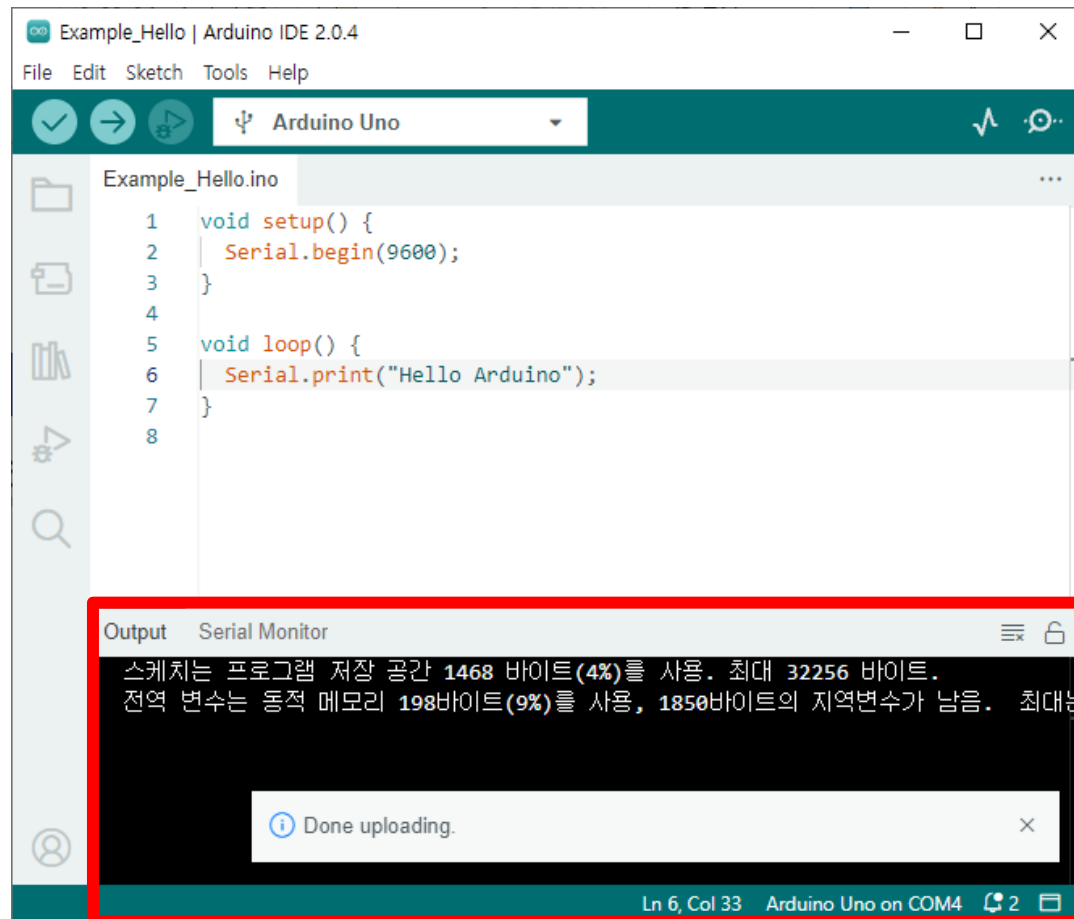
- 에러가 발생한 부분이 표시된다.
- 에러가 아래의 라인에 영향을 미칠 경우 다음 줄에 표시 된다.
- 하단의 창에 더 자세한 내용이 나온다.
- ; 이 빠졌다.

에러가 발생 했을 때



정상적으로 실행될 경우

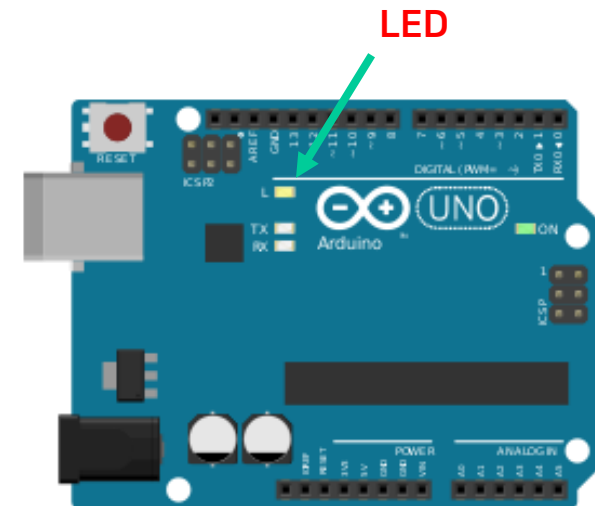
- 하단에서 Done uploading을 볼 수 있다.



컴파일 내용

실습 - Blink Example (1)

- LED 를 점멸 시키는 아두이노 예제를 구동 시켜보자
- 보드위의 LED 는 D13 에 연결되어 있다.
- 업로드 버튼을 눌러 실행할 수 있다.

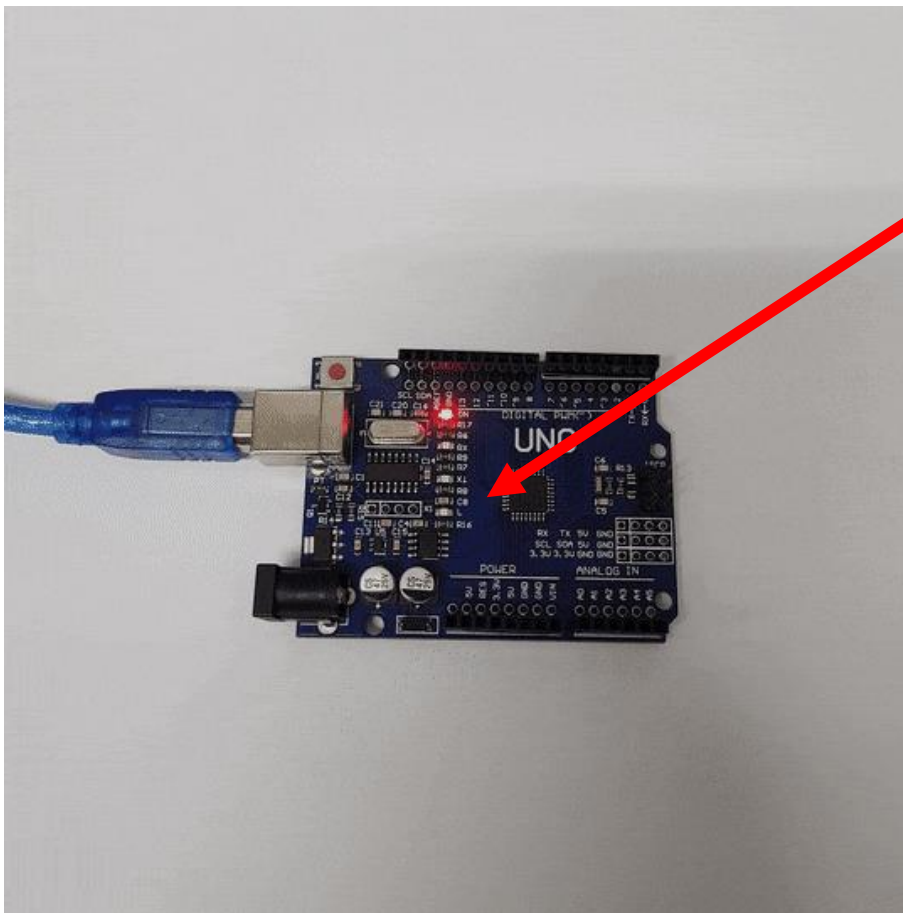


실습 - Blink Example (2)

■ File -> New Sketch -> 코드 작성 -> 컴파일 -> 업로드

```
1  #define LED_PIN LED_BUILTIN
2  void setup() {
3      // put your setup code here, to run once:
4      pinMode(LED_PIN, OUTPUT);
5  }
6
7  void loop() {
8      // put your main code here, to run repeatedly:
9      digitalWrite(LED_PIN, HIGH); // LED 켜짐
10     delay(1500); // 켜진 상태를 1.5초(1500밀리초) 유지
11     digitalWrite(LED_PIN, LOW); // LED 꺼짐
12     delay(500); // 꺼진 상태를 0.5초(500밀리초) 유지
13 }
```

실습 - Blink Example (3)



보드의 LED가 점멸

Blink Example 실습 과정

- 아두이노는 컴파일 언어를 사용하고 있어 실행전에 반드시 프로그램 코드 컴파일을 해야 된다.
- 컴파일된 코드는 아두이노에 업로드 되어진다.

02. 아두이노 프로그램 구조 이해하기

아두이노 함수 구조

```
void setup() {  
    .....  
}
```

하드웨어 및
초기 설정

```
void loop() {  
    .....  
}
```

반복적 동작 설정

setup()

- 핀 모드 설정, 시리얼 통신 속도 설정 등 주로 하드웨어 상태 설정에 사용 되는 함수

- 장치 부팅 후 처음 한번만 동작한다

```
void setup() {  
    .....  
}
```

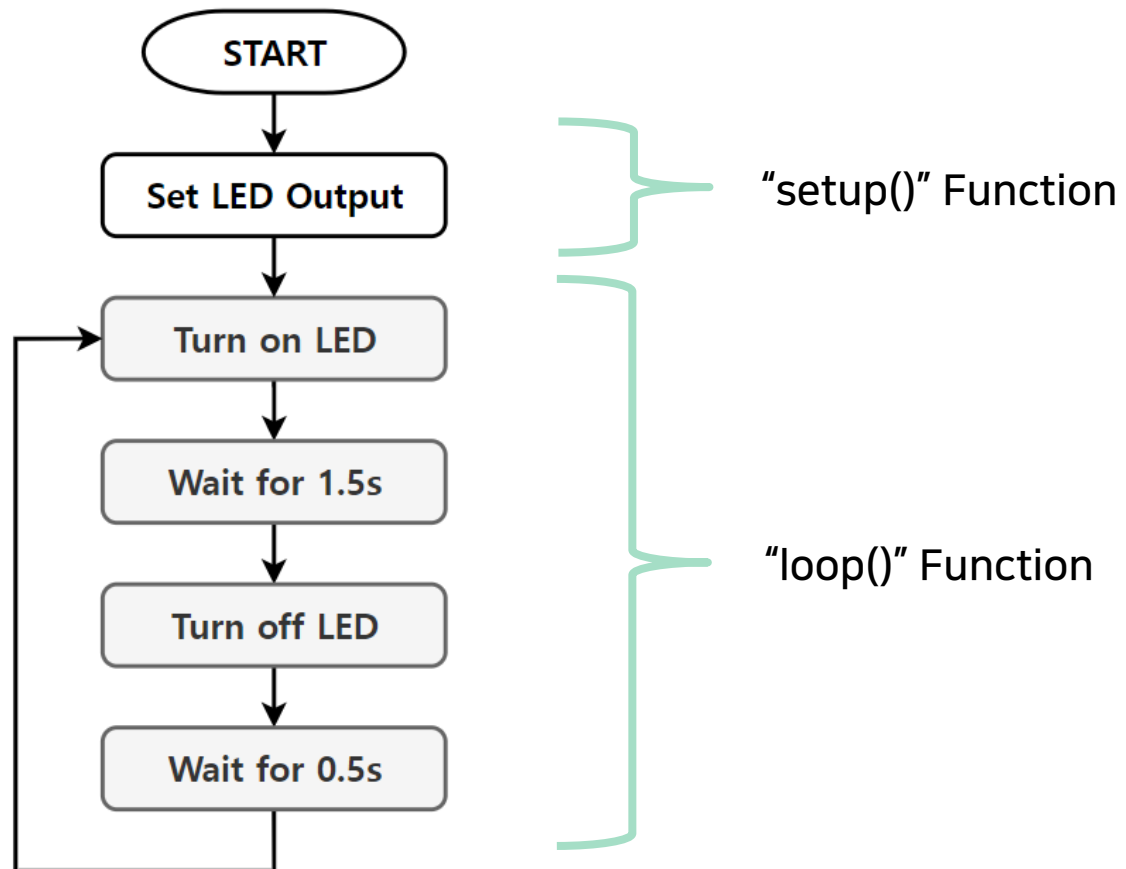
- 모든 설정이 반드시 setup()에서만 해야 되는 것은 아니다

void loop()

- while과 같이 무한반복 되는 함수이다
- 실제 동작코드를 넣어주면 된다
- 아두이노에서 처리되어져야 할 동작을 1회 정의하여 반복 실행되어지도록 한다
- 대부분의 제품들은 켜진 이후로 꺼지지 않고 계속 반복적인 동작을 한다.
이 반복적 동작을 정의하기 위한 함수이다.

setup과 loop 함수 이해

■ Blink 예제



함수 소개

- pinMode(pin, mode)
- digitalWrite(pin, value)
- delay(ms)

함수 소개 - pinMode

■ pinMode(pin, mode)

❖ 핀의 입출력 모드를 설정하는 함수

❖ pin : 설정할 핀의 번호

❖ mode : 설정할 모드

INPUT : 입력 모드

OUTPUT : 출력 모드

INPUT_PULLUP : 풀업 입력 모드, 잘 안 씀

❖ ex) 13번 핀의 출력을 OUTPUT으로 설정하려면 pinMode(13, OUTPUT);

함수 소개 - digitalWrite

■ digitalWrite(pin, value)

- ❖ 핀의 출력을 정하는 함수, OUTPUT 모드여야 한다
- ❖ pin : 출력 설정 할 핀의 번호
- ❖ value : 출력할 값
 - HIGH or 1 : HIGH 로 출력
 - LOW or 0 : LOW 로 출력
- ❖ ex) OUTPUT 출력인 13번 핀을 HIGH 상태로 바꿀 경우
digitalWrite(13, HIGH);

함수 소개 - delay

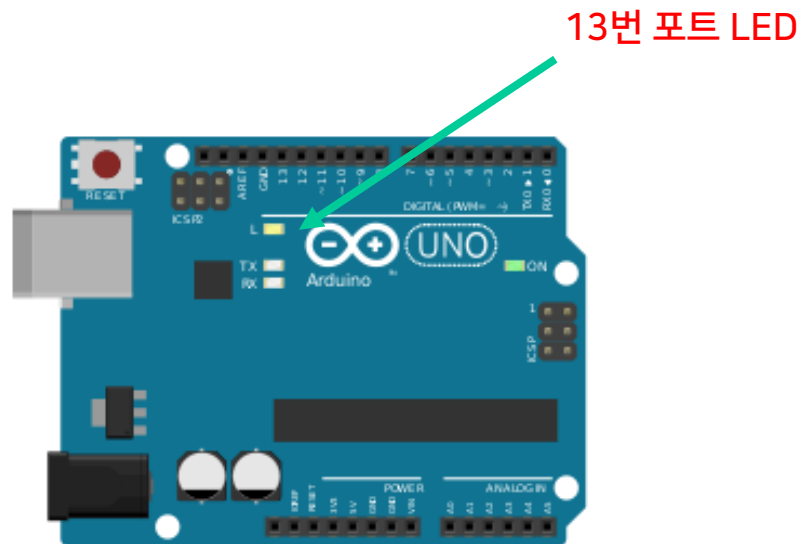
■ delay(ms)

- ❖ ms 만큼 프로그램을 일시 정지하는 함수
- ❖ ms : 일시 정지할 시간, millisecond
- ❖ ex) 1초간 프로그램 일시 정지
`delay(1000);`

실습

■ 목표

- ❖ delay() 와 digitalWrite() 를 1번씩만 써서 LED 가 깜빡이는 예제를 만들어보자



해답

```
int value;

void setup() {
    value = HIGH;
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_BUILTIN, value);
    value = !value;
    delay(1000);
}
```

digitalWrite()는 한번 사용되어졌다.
하지만 loop()에 진입 시
상태는 반전된다.

고찰

- loop()는 LED를 On/Off 동작의 반복이다. LED의 상태는 loop()의 호출에 따라 달라질 뿐이다
- LED_BUILTIN는 이미 선언되어진 값이며, port 번호 13이다. 13번 포트는 외부로 나와 있으니 사용할 때 주의하자

문제

■ value 변수를 loop() 안에 넣으면 어떻게 될까?

❖ 현상과 이유를 써주세요

■ LED 를 13번 포트가 아니라 10번 포트에 연결하고 싶다. 어디를 수정해야 하는가?

❖ 현재 코드를 쓰고 변경 코드를 적어주세요

■ Hello Arduino 문자열을 출력하고 싶다. 단 한번만 나오도록 수정해 보자. 어디에 추가를 해야 되는 가?

❖ 해당 함수와 주변 코드를 같이 적어주세요

03. 다양한 변수 다루기

- String 클래스

String 클래스

- 초보자가 C++로 문자열을 다루기는 힘들다. 아두이노에서는 쉽게 문자열을 다루기 위해 String 클래스를 만들었다.
- String 클래스는 문자열을 다루는 프로그램 묶음이다

String 오브젝트 만들기

- String 클래스는 필요할 때마다 오브젝트를 만들어서 사용하면 된다. 오브젝트를 만들 때 초깃값을 지정할 수 있다.

```
String 오브젝트1;  
String 오브젝트2 = "초깃값";
```

– 예시

```
String fullName;  
String name = "Gildong";  
String family = "Hong";  
  
fullName = family + " " + name; // Hong Gildong  
fullName = String("Hong") + " " + "Gildong"; // hong gildong  
fullName = "Hong" + " " + "Gildong"; // 오류
```

String 오브젝트 만들기

- String 오브젝트에 하나의 문자열은 직접 할당할 수 있다.
- String 오브젝트는 몇 개든 +로 연결할 수 있으며. 연결할 대상 중 적어도 하나는 String 오브젝트 속성을 가져야 한다.
- String은 클래스고 fullName, name, family는 오브젝트이며 오브젝트를 다루는 다양한 함수(메소드)를 사용할 수 있다.

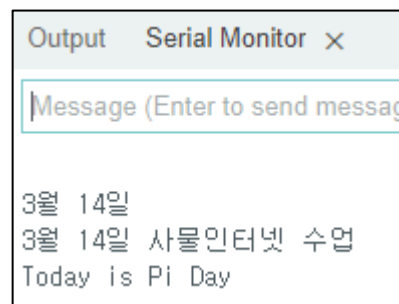
실습 - String 오브젝트 만들기

String.ino

```

1  void setup() {
2      Serial.begin(9600);
3      Serial.println();
4
5      String today; // String 오브젝트;
6      String month = "3월"; // 초깃값 지정
7      String day = "14일";
8
9      today = month + " " + day;
10     Serial.println(today);
11
12     today = month + " " + "14일" + " " + String("사물인터넷 수업");
13     Serial.println(today);
14
15     // today = "Today" + "is" + "White Day"; => 오류
16     // 적어도 하나는 String 오브젝트여야 한다.
17     today = String("Today is") + " " + "Pi Day";
18     Serial.println(today);
19 }
20
21 void loop() {
22
23 }
```

- 실행 결과



다른 변수형을 스트링으로 변환하기

■ String()

상수나 변수를 String으로 감싸도 스트링으로 변환된다.

```
String(문자 또는 문자열)  
String(숫자)  
String(변수)  
String(float형 변수 또는 상수, 소수점 이하 자릿수)
```

실습 - 스트링 변환 예제

String-concat.ino

```

1 void setup() {
2   Serial.begin(9600);
3   Serial.println();
4
5   int h = 9;
6   String hStr = String(h);    // "9"
7
8   int m = 8;
9   String mStr = String(m);    // "8"
10  if (mStr.length() == 1) {
11    mStr = '0' + mStr;        // "08"
12  }
13
14  int s = 35;
15  String sStr = String(s);     // "35"
16
17  float t = 27.56;
18  String tStr = String(t, 1);  // "27.6"
19
20  String longStr = hStr + ":" + mStr + ":" + sStr + "," + tStr;
21  Serial.println(longStr);     // "9:08:35,27.6"
22 }
23
24 void loop() {
25
26

```

시리얼 모니터와 Serial.begin() 의
통신속도를 맞춰준다!

- 실행 결과

Output Serial Monitor x

Message (Enter to send message)

9:08:35,27.6

실습 - 스트링 변환 예제 설명

```

1 void setup() {
2   Serial.begin(9600);
3   Serial.println();
4
5   int h = 9;
6   String hStr = String(h);    // "9"
7
8   int m = 8;
9   String mStr = String(m);    // "8"
10  if (mStr.length() == 1) {
11    | mStr = '0' + mStr;        // "08"
12  }
13
14  int s = 35;
15  String sStr = String(s);     // "35"
16
17  float t = 27.56;
18  String tStr = String(t, 1);  // "27.6"
19
20  String longStr = hStr + ":" + mStr + ":"
21  | | | | | | | + sStr + "," + tStr;
22  Serial.println(longStr);     // "9:08:35,27.6"
23 }

```

❶ 8-12행 : 스트링에 .length()를 붙이면 길이를 알려준다.

길이 = 스트링.length();

❷ 17-18행 : String()의 두 번째 인수로 소수점 이하 자릿수를 지정할 수 있다. 생략하면 소수점 2자리로 반올림한다.

스트링 = String(값, 소수점 이하 자릿수);

스트링에서 값 골라내기

■ indexOf()

```
위치 = 스트링.indexOf(찾을 문자열); // 처음부터 찾을  
위치 = 스트링.indexOf(찾을 문자열, 찾기 시작하는 위치);
```

스트링에서 인수1의 위치를 알려준다. 인수2는 실행 위치로 생략하면 처음부터 실행한다.
인수1이 문자열의 첫 번째 글자면 0, 찾을 수 없는 경우엔 -1을 반환한다.

■ substring()

```
잘라낸 스트링 = 스트링.substring(시작 위치); // 끝까지 자름  
잘라낸 스트링 = 스트링.substring(시작 위치, 이 숫자 앞까지 자름);
```

substring()은 정해진 범위의 문자열을 잘라 돌려준다.
인수1에서 인수2 앞 글자까지 잘라낸다. 인수2가 생략되면 마지막까지 자른다.

스트링을 숫자로 변환하기

■ toInt(), toFloat()

```
스트링.toInt();  
스트링.toFloat();
```

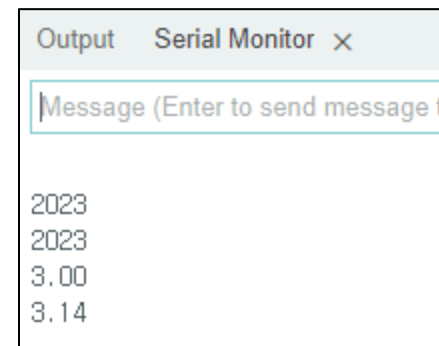
toInt()와 toFloat()는 스트링을 int형과 float형의 값으로 반환, 온전한 숫자가 아니라면 0을 반환한다.

실습 - 스트링을 숫자로 변환하기

```

string-toInt-toFloat.ino
1  void setup() {
2    Serial.begin(9600);
3    Serial.println();
4
5    String year = "2023";
6    int iyear = year.toInt();
7
8    String year2 = "2023년";
9    int itoday2 = year.toInt();
10
11   String month = "3월";
12   float fmonth = month.toFloat();
13
14   String pie = "3.14159265359";
15   float fpie = pie.toFloat();
16
17   Serial.println(iyear);
18   Serial.println(iyear);
19   Serial.println(fmonth);
20   Serial.println(fpie);
21 }
22
23 void loop() {
24
25 }
  
```

- 실행 결과



실습 - 스트링에서 값을 골라내고 숫자로 바꾸기

- 앞의 설명을 참고하여 코드의 빈칸을 채워 결과를 출력한다.

String-substring.ino

```

1 void setup() {
2   Serial.begin(9600);
3   Serial.println();
4
5   String longStr = "9:08:35,27.6";
6
7   int p1 = longStr.indexOf(":");
8   int p2 = longStr.indexOf(":", p1+1);
9   int p3 = longStr.indexOf(",");
10
11  String hStr = longStr.substring(0, p1);
12  String mStr = longStr.substring(p1+1, p2);
13  String sStr = longStr.substring(p2+1, p3);
14  String tStr = longStr.substring(p3+1);
15
16  int h = hStr.   ;
17  int m = mStr.   ;
18  int s = sStr.   ;
19  float t = tStr.   ;
20
21  Serial.println(h);
22  Serial.println(m);
23  Serial.println(s);
24  Serial.println(t);
25 }
26
27 void loop() {
28
29

```

-실행 결과

```

9
8
35
27.60

```

04. 시리얼모니터 이해

시리얼 모니터로 출력하기

- 아두이노 보드에서 Serial 오브젝트를 이용하여 PC로 데이터를 전송하면, 아두이노 IDE에 포함되어있는 시리얼 모니터 프로그램은 수신된 내용을 화면에 표시한다.

메소드	의미
begin(송수신 비트 수)	초기화하면서 초당 송수신 비트 수를 지정함
print(출력할 값)	인수를 눈에 보이는 글자로 출력함
println(출력할 값)	인수를 눈에 보이는 글자로 출력하고 줄을 바꿈
write(숫자 값)	메모리상의 내용을 1바이트로 출력함

실습 - 시리얼 모니터로 출력하기

■ begin()

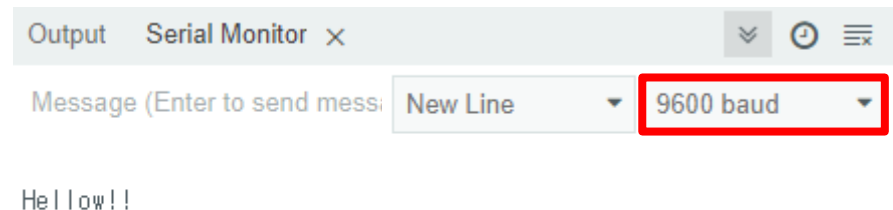
```
Serial.begin(통신속도);
```

Serial 오브젝트를 사용하기 위해서는 통신속도인 '보 레이트(baud rate)'를 지정해야 한다.

serial-hellow.ino

```
1 void setup() {  
2     Serial.begin(9600);  
3     Serial.println();  
4     Serial.println("Hello!!");  
5  
6 }  
7  
8 void loop() {  
9  
10 }
```

- 실행 결과



시리얼 모니터로 출력하기

■ print()

```
Serial.print(val);  
Serial.print(val, format);
```

```
Serial.println(val);  
Serial.println(val, format);
```

Serial.print()는 값을 문자, 문자열 또는 10진수 숫자 형식으로 출력하지만, Format 으로 DEC, BIN, OCT, HEX 를 지정하면 값을 각각 10, 2, 8, 16진수로 출력한다.

실습 - 시리얼 모니터로 출력하기

■ print()

serial-format.ino

```
1 void setup() {
2   Serial.begin(9600);
3   Serial.println();
4
5   char c = 'a';
6   Serial.println(c);           // a
7   Serial.println(c, DEC);      // 97
8   Serial.println(c, BIN);      // 0110001(출력 시 앞의 0은 생략)
9   Serial.println(c, HEX);      // 61
10
11   float f = 35.657;
12   Serial.println(f);           // 35.66
13   Serial.println(f, 1);        // 35.7
14   Serial.println(f, 0);        // 36
15
16 }
17
18 void loop() {
19
20 }
```

- 실행 결과

```
a
97
1100001
61
35.66
35.7
36
```


시리얼 모니터로 출력하기

■ write()

```
Serial.write(byte형 val);  
Serial.write(문자|문자열);
```

문자, 문자열을 출력하는 경우엔 Serial.print() 와 Serial.write()의 결과가 같다.
숫자를 출력하는 경우에는 차이가 있다.

- Serial.print() : 눈에 보이는 숫자로 바뀌어서 출력
- Serial.write() : 메모리 내용을 그대로 출력. ex) 97은 아스키 코드 'a'

Serial.write()를 사용하여 숫자를 전송하면 데이터양이 줄어들어 통신속도를 올릴 수 있지만 다루기가 까다로워 필요할 때만 사용하는 것이 좋다.

실습 - 시리얼 모니터로 출력하기

■ write()

serial-write.ino

```

1  void setup() {
2    Serial.begin(9600);
3    Serial.println();
4
5    Serial.println("-----문자");
6    char c = 'a';
7    Serial.println(c);
8    Serial.write(c);
9
10   Serial.println("\n-----문자열");
11   char s[] = "abc";
12   Serial.println(s);
13   Serial.write(s);
14
15   Serial.println("\n-----byte");
16   byte b = 97;
17   Serial.println(b);
18   Serial.write(b);
19   Serial.println();
20 }
21
22 void loop() {
23
24 }
  
```

- 실행 결과

```

-----문자
a
a
-----문자열
abc
abc
-----byte
97
a
  
```

시리얼 모니터에서 입력받기

■ Serial 입력 메소드

메소드	의미
begin(송수신 비트 수)	초기화하면서 초당 송수신 비트 수를 지정
available()	입력 버퍼에 도착한 문자의 개수를 반환
read()	입력 버퍼에서 한 글자를 읽어서 반환
parseInt()	입력 버퍼에서 정수가 되는 부분까지 읽은 후 정수로 바꿔서 반환
parseFloat()	입력 버퍼에서 소수 점 있는 수가 되는 부분까지 읽은 후 float로 바꿔서 반환
readStringUntil(끝 표시 문자)	끝 문자로 표시된 문자가 나오기 전까지 읽어서 지정한 배열에 삽입

실습 - 시리얼 모니터에서 입력받기

■ 입력한 내용을 그대로 출력하기

serial-read-echo.ino

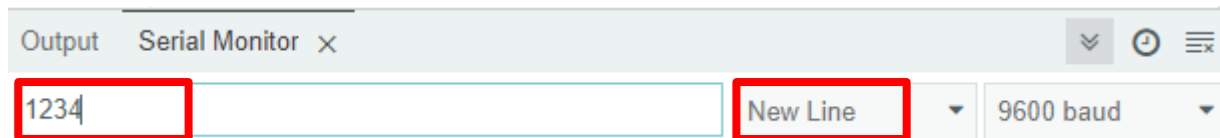
```
1 void setup() {  
2   Serial.begin(9600);  
3   Serial.println();  
4 }  
5  
6 void loop() {  
7   while(Serial.available()) {  
8     char c = Serial.read();  
9     Serial.print(c);  
10  }  
11 }
```

시리얼 모니터에서 데이터를 입력하면 보드의 수신 버퍼에 도착하게 됨. 이 때 Serial.available()을 통해 도착한 문자 수를 알 수 있다.

문자 수가 0이 아니면 Serial.read() 를 호출하여 한 글자씩 읽어 올 수 있음.

실습 - 시리얼 모니터에서 입력받기

■ 입력한 내용을 그대로 출력하기



시리얼 모니터에서 '추가 입력 문자' 를 'New Line' 으로 바꾸고
입력란에 '1234' 를 입력한 후 엔터키를 누른다.

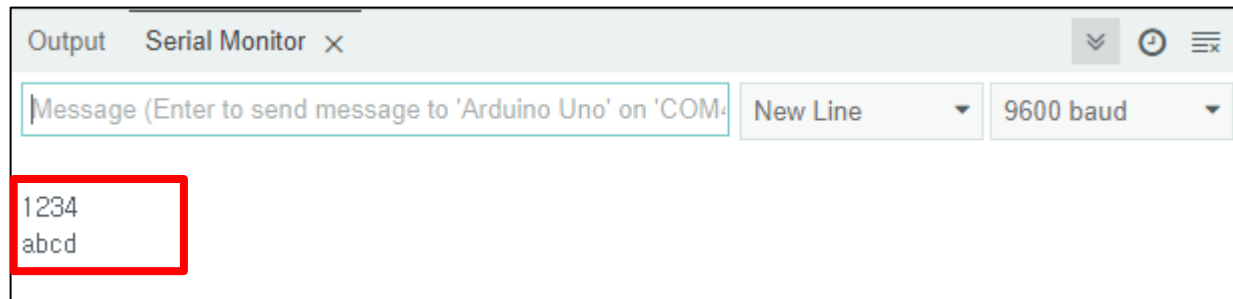


1234

입력란에 'abcd' 를 입력한 후 엔터키를 누른다.

실습 - 시리얼 모니터에서 입력받기

■ 입력한 내용을 그대로 출력하기



'1234' 를 입력한 후 전송을 하면 '1234'와 함께 '새 줄' 을 나타내는 문자 (아스키코드 12)가 연달아 전송된다. 그래서 'abcd'는 다음 줄에 출력된다.

추가 입력 문자	추가로 입력되는 문자
No Line Ending	추가 입력 문자 없음
New Line	"\n", 아스키코드 12
Carriage Return	"\r", 아스키코드 15
Both NL & CR	"\r\n"

시리얼 모니터에서 입력받기

■ int, float, 문자열 입력받기

- key 와 val

시리얼 통신으로 정보를 주고받을 땐 데이터 형식을 먼저 정해야 한다.
 이때 데이터 이름(key)과 값(val)을 한 쌍으로 주고받으면 편하다.

- key, val 데이터 형식 설계

데이터	key	val 의 데이터 형식	val 의 변수 이름	입력 예
개수	c	int	cnt	c125 또는 c 125
온도	t	float	temp	t26.6 또는 t 26.6
이름	n	String, 문자 끝에 ;	name	ngildong;

실습 - 시리얼 모니터에서 입력받기

■ int, float, 문자열 입력받기

serial-read-key.ino

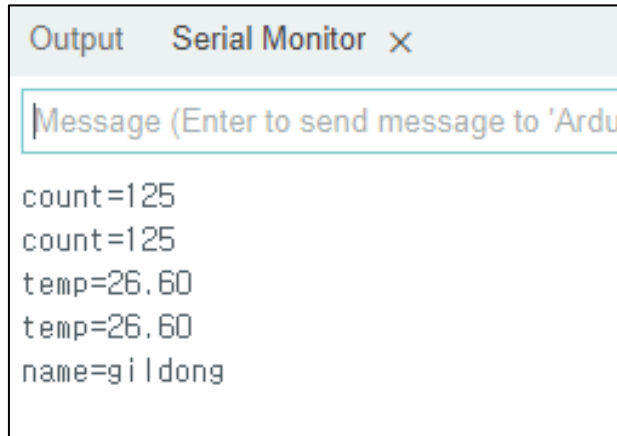
```
1  int cnt;
2  float temp;
3  String name;
4
5  void setup() {
6    Serial.begin(9600);
7    Serial.println();
8  }
9
```

```
10 void loop() {
11   while(Serial.available()) {
12     char c = Serial.read();
13     if (c == 'c') {
14       cnt = Serial.parseInt();
15       Serial.print("count=");
16       Serial.println(cnt);
17     }
18     else if (c == 't') {
19       temp = Serial.parseFloat();
20       Serial.print("temp=");
21       Serial.println(temp);
22     }
23     else if (c == 'n') {
24       name = Serial.readStringUntil(';');
25       Serial.print("name=");
26       Serial.println(name.c_str());
27     }
28   }
29 }
```


실습 - 시리얼 모니터에서 입력받기

■ int, float, 문자열 입력받기

- 63p 표의 '입력 예'에 따라 시리얼 모니터에 입력한다.
- 결과



```
Output Serial Monitor ×
Message (Enter to send message to 'Ardu...
count=125
count=125
temp=26.60
temp=26.60
name=gildong
```

c 125 입력 -> count=125

먼저 한 글자를 읽어서 key 를 알아내고,
val의 데이터 형식에 맞는 메소드를 선택해서 읽는다.

실습 - 시리얼 모니터에서 입력받기

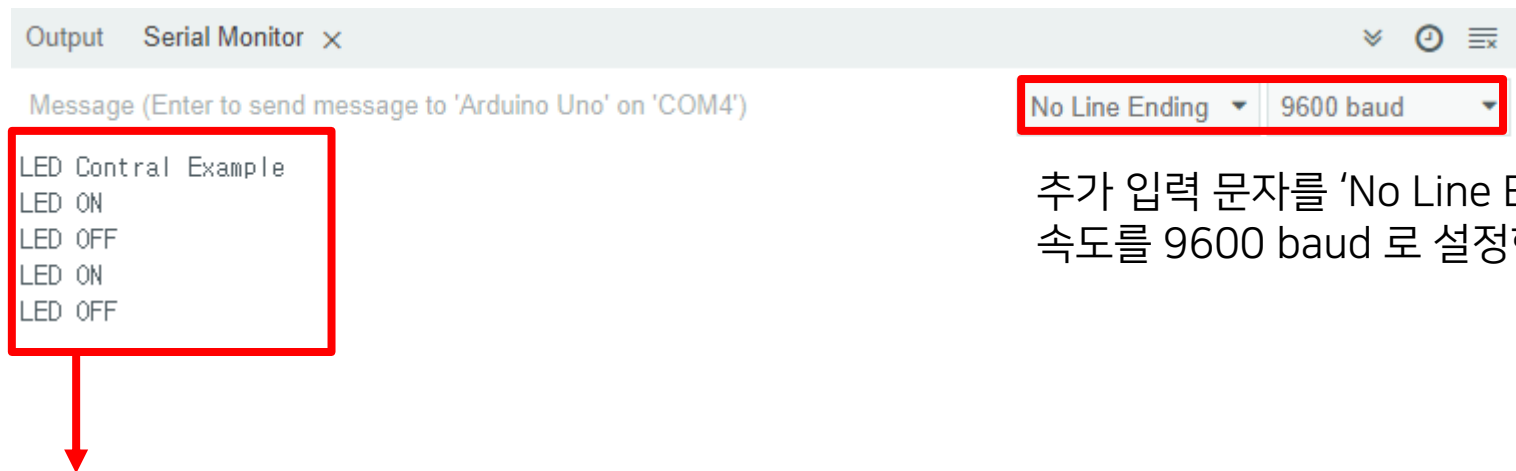
■ String class 를 이용하여 "On" 과 "Off"로 제어하기

serial-led-onoff.ino

```
1  #define LED_PIN LED_BUILTIN
2  #define LED_ON  HIGH
3  #define LED_OFF LOW
4
5  String inputVal;
6  int ledVal;
7
8  void setup() {
9      Serial.begin(9600);
10     Serial.println("LED Control Example");
11     pinMode(LED_PIN, OUTPUT);
12 }
13
14 void loop() {
15     inputVal = Serial.readString();
16
17     if (inputVal == "on") {
18         Serial.println("LED ON");
19         ledVal = LED_ON;
20     }
21     else if (inputVal == "off") {
22         Serial.println("LED OFF");
23         ledVal = LED_OFF;
24     }
25     digitalWrite(LED_PIN, ledVal);
26 }
```

실습 - 시리얼 모니터에서 입력받기

■ String class 를 이용하여 "On" 과 "Off"로 제어하기



추가 입력 문자를 'No Line Ending',
속도를 9600 baud 로 설정한다.

시리얼 모니터에 on 과 off 를 입력하여 LED가 제어되는 것을 확인한다.

실습 - 시리얼 모니터로 Blink Time 조절하기

■ LED 모드와 시간을 입력받기 (Serial 입력 메서드 활용)

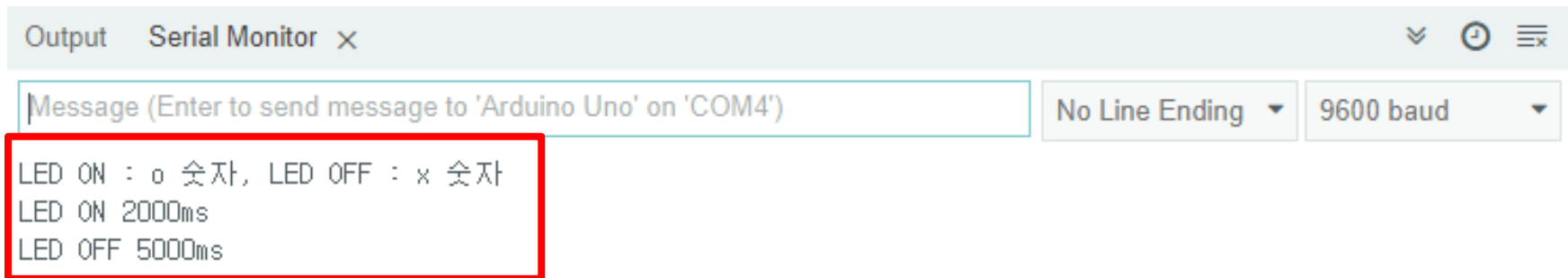
serial-led-onoff2.ino

```
1  #define LED_PIN LED_BUILTIN
2
3  String ledState;
4  char mode;
5  int onTime = 1000;
6  int offTime = 1000;
7
8  void setup() {
9      Serial.begin(9600);
10     Serial.println("LED ON : o 숫자, LED OFF : x 숫자");
11     pinMode(LED_PIN, OUTPUT);
12 }
13
```

```
14 void loop() {
15     while(Serial.available()){
16         mode = Serial.read();
17         if (mode == 'o') {
18             onTime = Serial.parseInt();
19             ledState = "LED ON " + String(onTime) + "ms";
20             Serial.println(ledState);
21         }
22
23         else if (mode == 'x') {
24             offTime = Serial.parseInt();
25             ledState = "LED OFF " + String(offTime) + "ms";
26             Serial.println(ledState);
27         }
28     }
29     digitalWrite(LED_PIN, HIGH);
30     delay(onTime);
31     digitalWrite(LED_PIN, LOW);
32     delay(offTime);
33 }
```

실습 - 시리얼 모니터로 Blink Time 조절하기

■ LED 모드와 시간을 입력받기 (Serial 입력 메서드 활용)



ON/OFF 기본값을 1000으로 설정하여 업로드를 하면 1초 간격으로 LED가 점등한다.
시리얼 모니터에 모드와 시간을 입력하여 delay 시간을 조절한다.

Ex)

시리얼 모니터에 o 2000 입력 -> 2초간 LED가 켜짐

시리얼 모니터에 x 5000 입력 -> 5초간 LED가 꺼짐