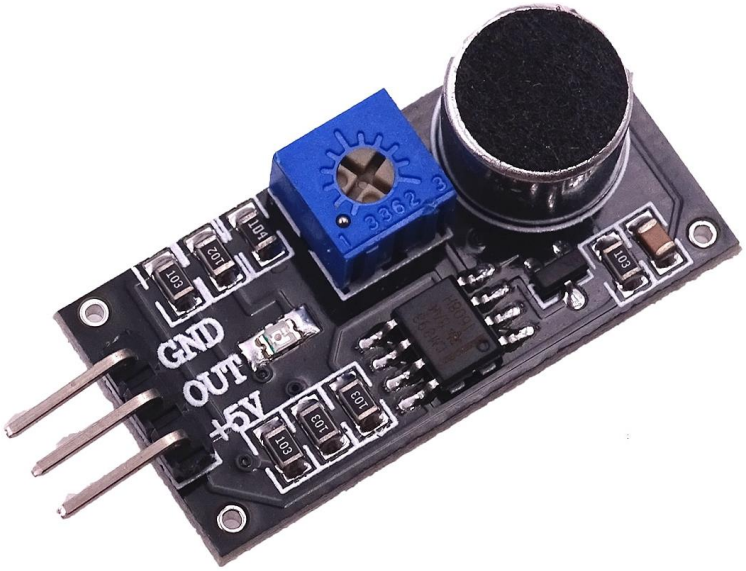
LECTURE 9
Sound Sensor and Application

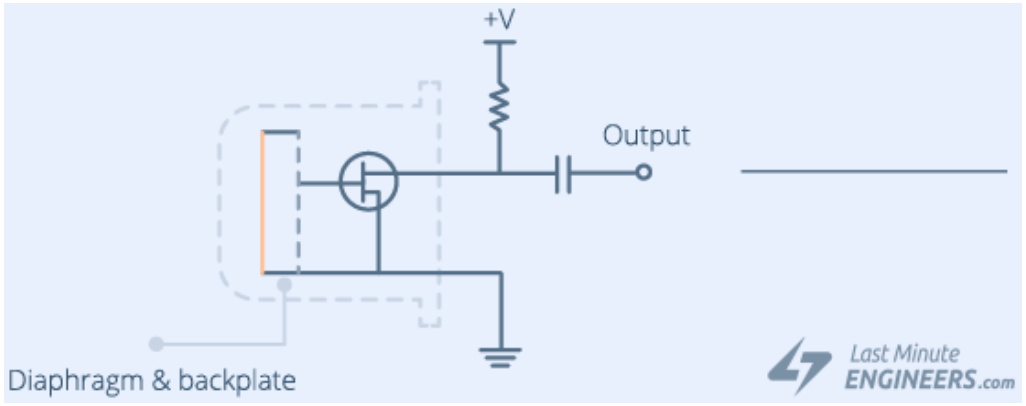
Instructor: Osman Gul, Ph.D. Candidate
Department of Mechanical Engineering
Korea Advanced Institute of Science and Technology (KAIST)

Syllabus:

Week	Date& Time*	Content
1		No Class (Course Add/Drop Period)
2	March 6 (Mon) 20-22	Orientation and Course Overview
3	March 13 (Mon) 20-22	Sensor Fundamentals
4	March 20 (Mon) 20-22	Arduino Fundamentals
5	March 27 (Mon) 20-22	Arduino Programming and Applications
6	April 3 (Mon) 20-22	Force Sensor and Application: Control Multiple LEDs
7	April 10 (Mon) 20-22	Light Sensor and Application: Solar Tracker
8	April 17	No Class (Midterm)
9	April 24 (Mon) 20-22	Humidity and Temperature Sensor and Application: Temperature Controlled Fan
10	May 1 (Mon) 20-22	Gas Sensor and Application: Smoke Detector
11	May 8 (Mon) 20-22	Sound Sensor and Application: Control LED by Clapping
12	May 15 (Mon) 20-22	Accelerometer Sensor and Application: Ping Pong Game
13	May 22 (Mon) 20-22	Ultrasonic Sensor and Application: Flappy Bird Game
14	May 29 (Mon) 20-22	Course Wrap-up
15	June 5	No Class (Final)
16	June 12	No Class (Final)

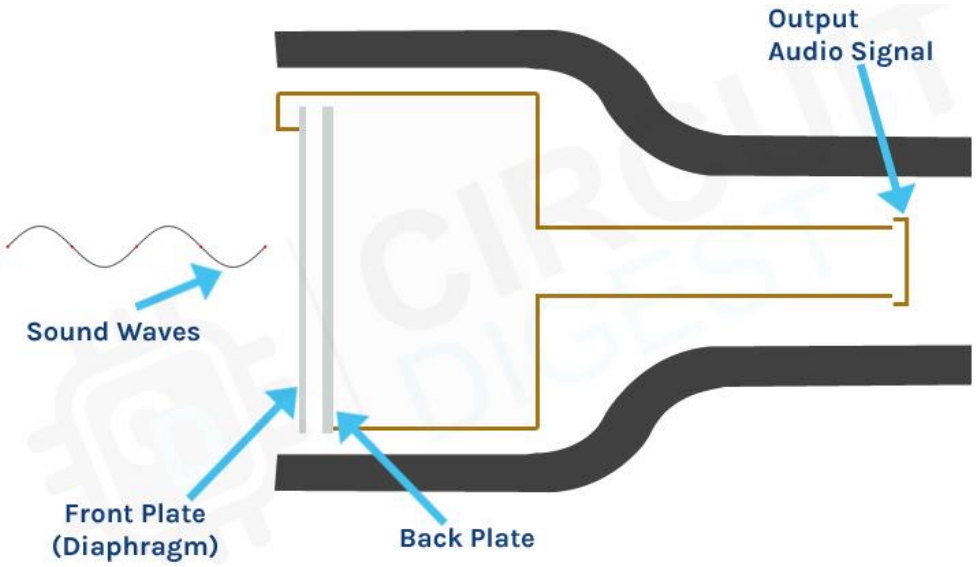
Sound sensor





- The sound sensor is a module that monitors and detects the sound signals like voice, claps, snaps, knocks, etc.
- It is also known as an acoustic sensor or sound detector.
- It contains a microphone, power amplifier, and output actuator.
- The microphone that acts as an input sensor **receives the sound signal and converts it into an electrical signal**. Then this signal is amplified by the power amplifier and its amplitude is detected by the peak detector. The output actuator, like a loudspeaker, converts this amplified electrical signal into a sound signal for listening.
- The sound sensors can detect sound signals in the frequency range of 3kHz to 6kHz and it operates at a DC voltage of 3.3V to 6V.

- The main component of a sound sensor is a microphone. There are many different types of microphones, like Carbon Microphone, Fiber Optic Microphone, Ribbon Microphone, and Laser Microphone, but the sound sensor module we are using has a **condenser microphone**.



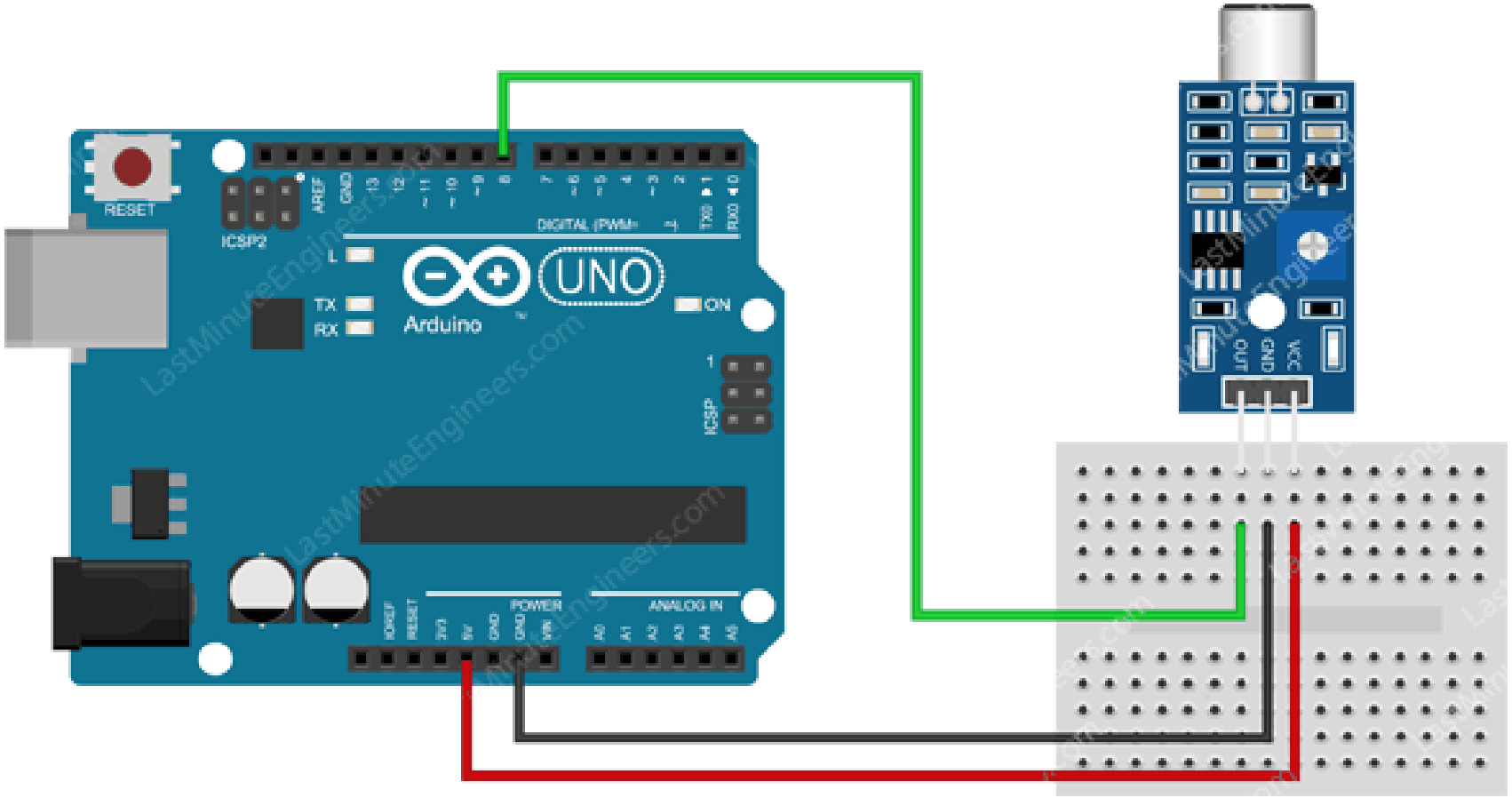
- A condenser microphone consists of two charged metal plates.
- The first plate is called the **diaphragm** and the second plate is the **backplate** of the microphone. These two plates together form **a capacitor**. When a sound wave hits the diaphragm of the microphone the diaphragm starts to vibrate, and the distance between the two plates changes.
- The movement of the diaphragm and the change in spacing produces the electrical signal that corresponds to the sound that's picked up by the microphone and this signal then gets processed by the onboard op-amp.
- This module also has two built-in onboard LEDs, one of which lights up when power is applied to the board and the other one lights up when the incoming audio signal exceeds the threshold value set by the potentiometer.

Sound Sensor Pinout — Last Minute ENGINEERS.com

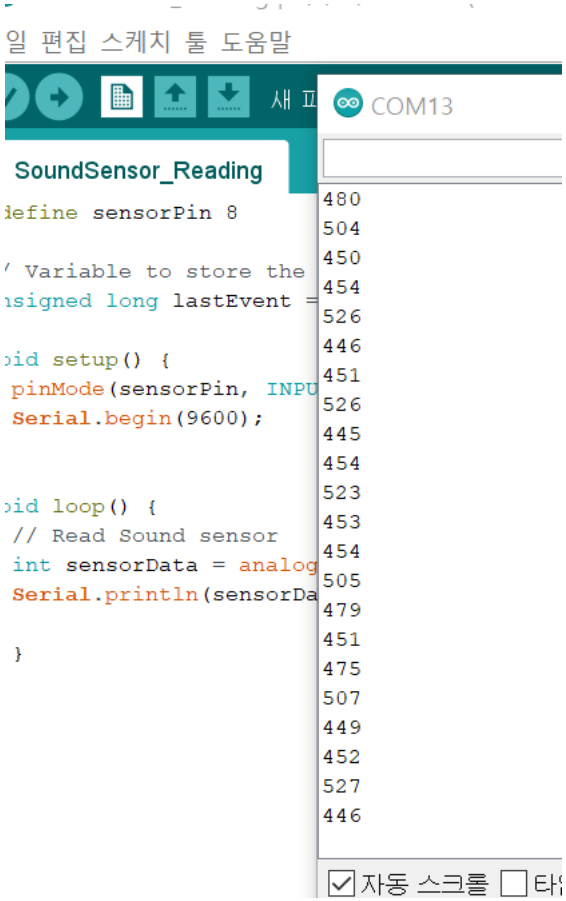
VCC supplies power to the sensor. It is recommended that the sensor be powered from 3.3V to 5V.

GND is the ground pin.

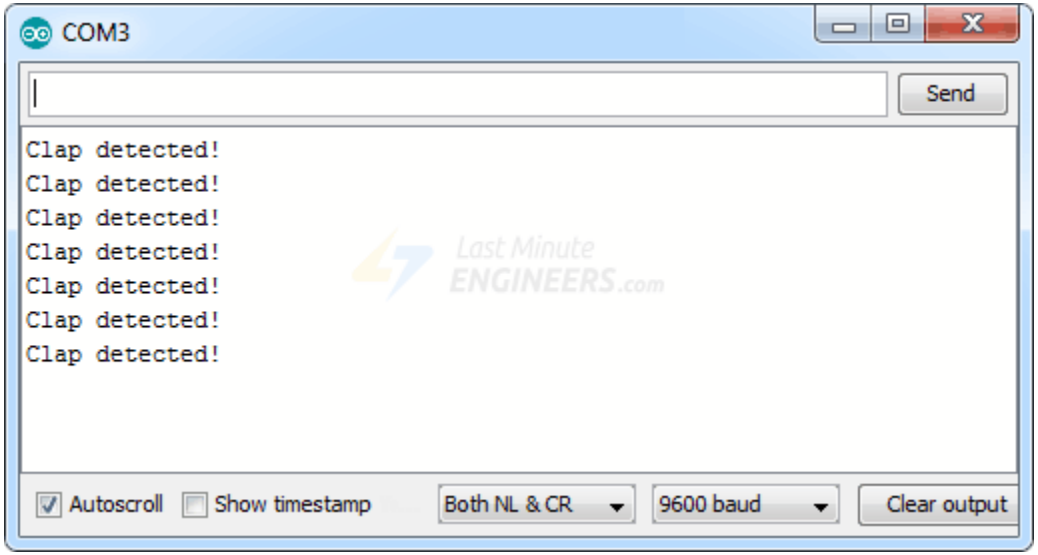
OUT pin outputs HIGH under quiet conditions and LOW when sound is detected. You can connect it to any digital pin on an Arduino or to a 5V relay directly.


RESET
ICSP2
AREF
GND
DIGITAL (PWM~)
TX
RX
L
ON
Arduino
UNO
ICSP
POWER
ANALOG IN
OUT
GND
VCC

```
#define sensorPin 8

// Variable to store the time when last event happened
unsigned long lastEvent = 0;

void setup() {
  pinMode(sensorPin, INPUT);  // Set sensor pin as an INPUT
  Serial.begin(9600);
}

void loop() {
  // Read Sound sensor
  int sensorData = analogRead(sensorPin);
  Serial.println(sensorData);

  }
```



```
#define sensorPin 8

// Variable to store the time when last event happened
unsigned long lastEvent = 0;

void setup() {
	pinMode(sensorPin, INPUT);  // Set sensor pin as an INPUT
	Serial.begin(9600);
}

void loop() {
	// Read Sound sensor
	int sensorData = digitalRead(sensorPin);

	// If pin goes LOW, sound is detected
	if (sensorData == LOW) {

		// If 25ms have passed since last LOW state, it means that
		// the clap is detected and not due to any spurious sounds
		if (millis() - lastEvent > 25) {
			Serial.println("Clap detected!");
		}

		// Remember when last event happened
		lastEvent = millis();
	}
}
```

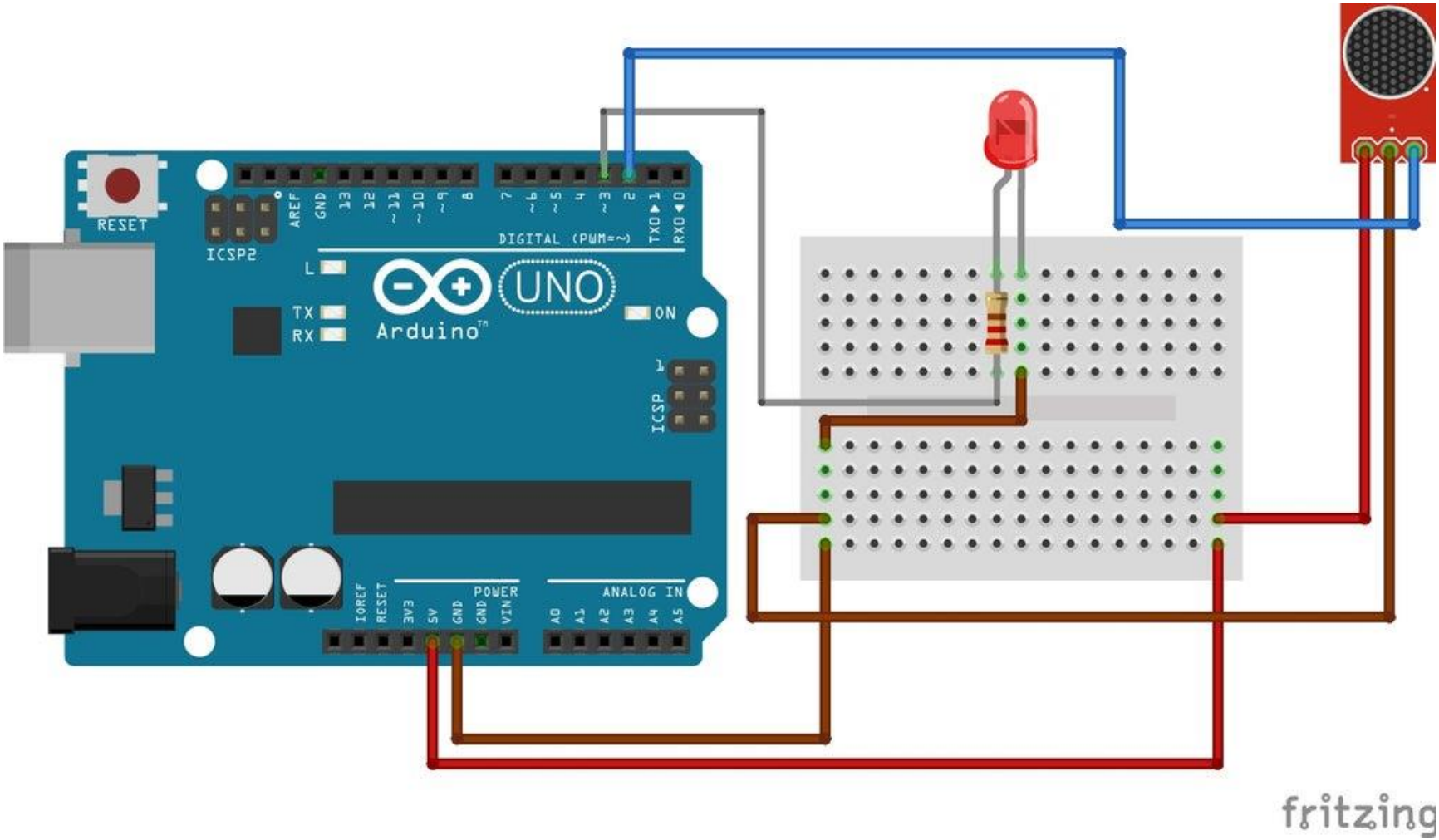

COM3
Send
Clap detected!
Clap detected!
Clap detected!
Clap detected!
Clap detected!
Clap detected!
Clap detected!
Last Minute ENGINEERS.com
Autoscroll
Show timestamp
Both NL & CR
9600 baud
Clear output

```
unsigned long lastEvent = 0;
```

We define a variable called lastEvent that stores when a clap was previously detected. It will help us reduce accidental sound detection.

```
if (sensorData == LOW) {
    if (millis() - lastEvent > 25) {
        Serial.println("Clap detected!");
    }
    lastEvent = millis();
}
```

When the sensor detects a sound loud enough to exceed the threshold value, the output goes LOW. However, we must ensure that the sound is caused by clapping and not by background noise. Therefore, we wait 25 milliseconds after the output goes low. If the output remains LOW for more than 25 milliseconds, the message "Clap detected" is printed on the serial monitor.


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
fritzing

```
int Sensor = A0;
int clap = 0;
long detection_range_start = 0;
long detection_range = 0;
boolean status_lights = false;
void setup() {
pinMode(Sensor, INPUT);
pinMode(10,OUTPUT);
}

void loop() {
int status_sensor = digitalRead(Sensor);
if (status_sensor == 0)
{
if (clap == 0)
{
detection_range_start = detection_range = millis();
clap++;
}
else if (clap > 0 && millis()-detection_range >= 50)
{
detection_range = millis();
clap++;
}
}
if (millis()-detection_range_start >= 400)
{
if (clap == 2)
{
if (!status_lights)
{
status_lights = true;
digitalWrite(10, HIGH);
}
else if (status_lights)
{
status_lights = false;
digitalWrite(10, LOW);
}
}
clap = 0;
}
}
```