

# TRIBALSTUDIOZ

roboacademy | codeacademy | gameacademy | makeracademy

# Learning Objectives

In this topic you will learn...

## T-Scratch Light Sabre Coding (1/3)

- Declaration of libraries and variables used in the project
- Coding setup() code
- Code the confirmation read-back after uploading
- Coding test state for the T-Sense(button).

# TSabre Coding Sequence

## 5 part bite-size coding

1. Declaration of **essential libraries** and **variables**.
2. Set-up **coding** and update state
3. Main function of an **ON-OFF** button
4. Sound **effects** and clips
5. Alter code to **Add-On** sensors coding

# Declaration of essential libraries

In the declaration space, #include the following:



- software serial → SoftwareSerial.h
- mp3 library → DFPlayer\_Mini\_Mp3.h

```
IntroSabre
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>
```

\*\*note that the fonts are case sensitive and must be followed strictly

# Declaration of Variables (Part 1/3)



Next, in the declaration space below the libraries, declare the following variables:

```
int i;
int vibration = 2;
int tilt = 3;
int button = A1;
int led = 6;
//arbitrary integer
//vibration sensor pin
//tilt sensor pin
//button pin
//led pin
```

Comment / Programming Note

After // are comments  
They will not be executed

# Declaration of Variables (Part 2/3)



**Next, declare the following variables:**

```
int glow;  
int glow_temp;  
int glow_count = 1;  
int state = -1;  
//State == -1 : Saber is Inactive  
//State == 0 : Saber is Active waiting to be ON  
//State == 1 : Saber is ON
```

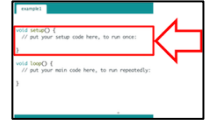
# Declaration of Variables (Part 3/3)



Next, declare the following variables to be used with the sound player:

```
static unsigned long timer = millis();  
SoftwareSerial mySerial(10, 11); // RX, TX  
  
int VibrationTrack = 8;  
  
int TiltTrack = 4;  
int PreviousTilt = 0;
```

Additional declarations  
Used when selecting MP3 Files



# Setup Code (Part 1/2)

In the setup code, we must first define `pinMode()` for all sensors and output.

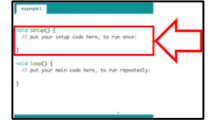
```
void setup() {  
  Serial.begin (9600);  
  mySerial.begin (9600);
```

```
  pinMode(tilt, INPUT);  
  pinMode(vibration, INPUT);  
  pinMode(button, INPUT);  
  pinMode(led, OUTPUT);
```

`pinMode(X,Y)`

X: port connected to sensor  
Y: INPUT/OUTPUT





# Setup Code (Part 2/2)

In the setup code, we also need to startup the MP3 player to play sound files later.

```
mp3_set_serial (mySerial);  
delay(100);  
mp3_set_volume (30);  
delay(500);
```

```
}
```

```
//set Serial for DFPlayer-mini mp3 module  
//delay 1ms to set serial for DFPlayer  
  
//delay for DFPlayer to power up
```

## Setup and delays for MP3 player

This setup time will allow sound files to be played later without delay.

\*\*Comments are good to have for easy referencing at later stage

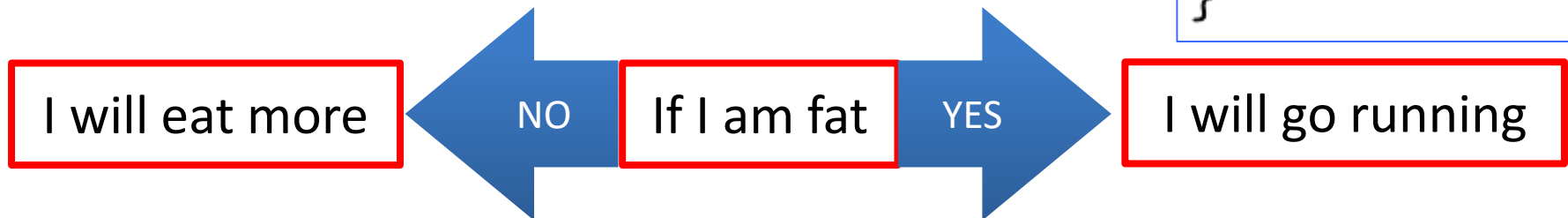


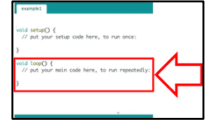
# Conditional coding

## [IF] Logic Programming

- Used to test a logic to determine the outcome/output of the program
- Usually used with sensors
- For example:

```
if(condition){  
    execute;  
}
```





# Main Code (Intro)

To confirm that the MP3 Player is ready to be used, program the next line of code reads back a sound signal

```
void loop() {  
  // "System ready" readback  
  if (state == -1) {  
    mp3_play(16);  
    state = 0;  
  }  
}
```

Remember the brackets  
They must be at the start and end of a function

## Check and update

Check if state is [-1] then  
Update [state -1] to [state 0]

This is the end of the preparation for your main code

**END OF PART 1/3**