

마이크로 로봇을 이용한 소프트웨어 교육

한국기술교육대학교

라 효 진

아이스 브레이킹

소개

라효진



 Ratataca



구독



좋아요



알람설정

학력

한국기술교육대학교 컴퓨터공학부

경력

1. 메이식스 창업
2. IoT 플랫폼 구축 프로그램 개발
3. 중소기업 IT 시스템 구축 프리랜서

이메일

gywls18@koreatech.ac.kr



한국기술교육대학교
KOREA UNIVERSITY OF TECHNOLOGY & EDUCATION

목표

1. 프로그래밍이 가능한 마이크로 로봇을 이용하여 코딩을 통해 로봇을 제어한다.
2. 로봇 제어에 필요한 프로그래밍 문법과 구조를 이해한다.
3. 마이크로 로봇 내 각종 센서들의 동작 방법을 학습한다.
4. 제시한 주제에 대해 센서 및 모터를 포함한 로봇의 동작을 스스로 코딩한다.

컴퓨터 구성 요소 및 동작 원리 소개

어떤 게 하드웨어이고 소프트웨어 일까?



컴퓨터 구성 요소 및 동작 원리 소개

어떤 게 하드웨어이고 소프트웨어 일까?



하드웨어 : 아이패드

소프트웨어 : YouTube

컴퓨터 구성 요소 및 동작 원리 소개

어떤 게 하드웨어이고 소프트웨어 일까?



하드웨어 : 만질 수 있는 것



소프트웨어 : 만질 수 없는 것

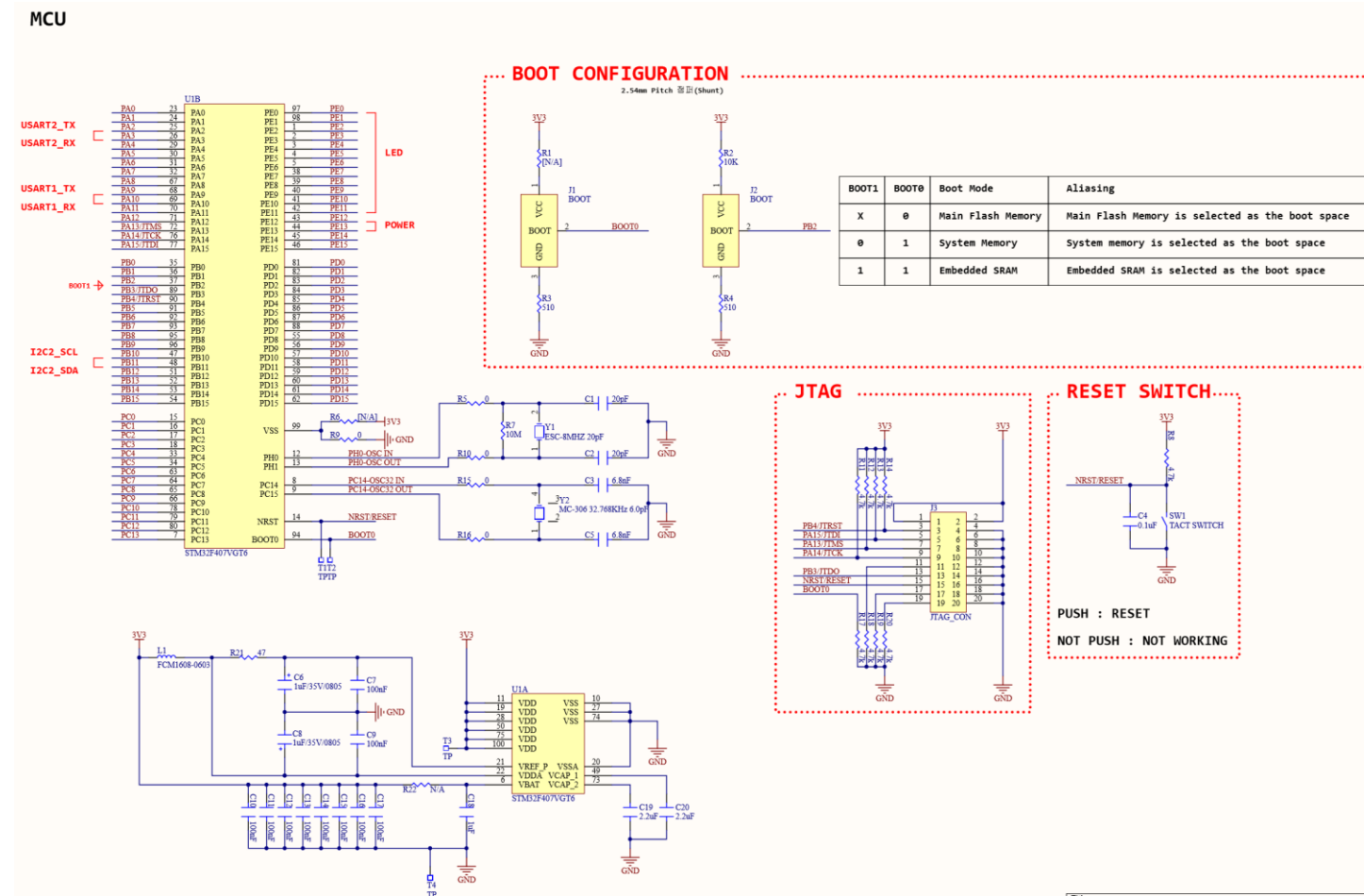
컴퓨터 구성 요소 및 동작 원리 소개

인공지능 스피커는 하드웨어 일까요? 소프트웨어 일까요?

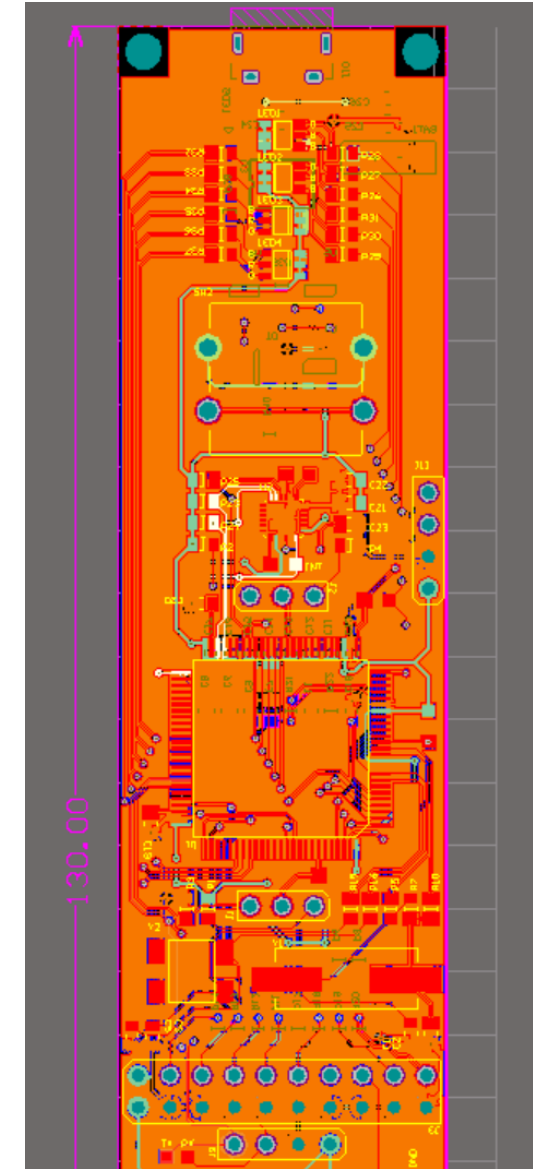


컴퓨터 구성 요소 및 동작 원리 소개

하드웨어 개발



전자회로 설계



전자기판 설계

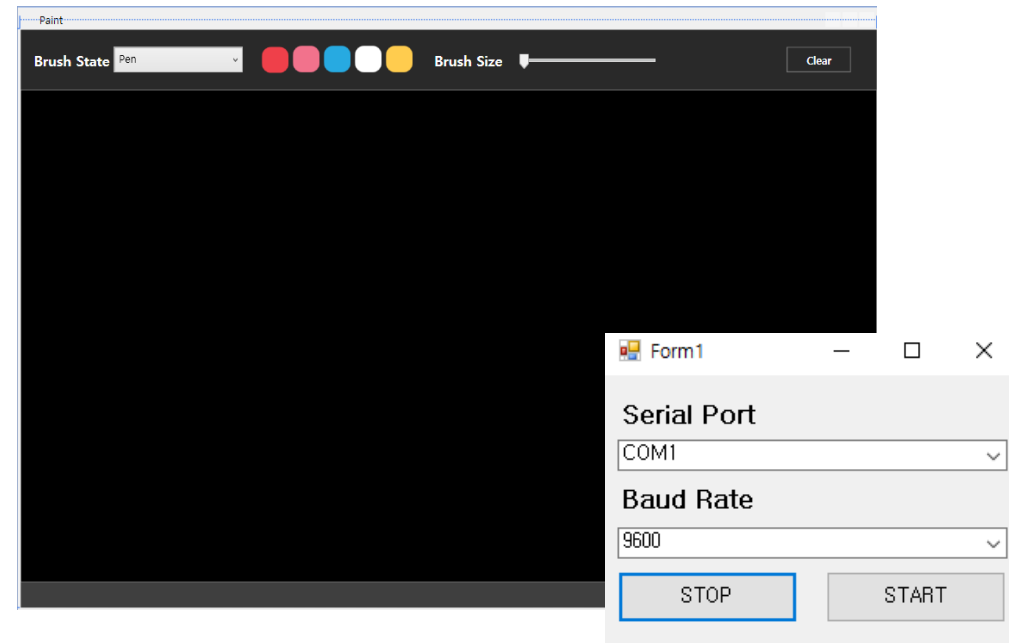
컴퓨터 구성 요소 및 동작 원리 소개

https://youtu.be/F_xNoEZ0Cg4

소프트웨어 개발

```
////////////////////////////////////  
// ICM Sensor  
ICM20648_VALUE value = NAM_ICM20648_Get_Data();  
int sensitivity = 300;  
  
mouse_state = get_mouse_state(mouse_state, value);  
mouse_state.gyroX = value.gyroX / sensitivity / 1.1 * -1;  
mouse_state.gyroZ = value.gyroZ / sensitivity * -1;  
  
mouse_state.button_front = NAM_Button_Front_Is_Push();  
mouse_state.button_center = NAM_Button_Center_Is_Push();  
mouse_state.button_back = NAM_Button_Back_Is_Push();  
  
printf("%d\t%d\t%d\t%d\t%d\t%d\r\n", mouse_state.velocity_x, mouse_state.velocity_y,  
  
PACKET_DATA[0] = PACKET_START;  
PACKET_DATA[1] = PACKET_LEN;  
  
uint8_t which_sign = 0; // 0 : +, 1 : -  
if(mouse_state.velocity_x <= 0)  
{  
    which_sign = 1;  
    mouse_state.velocity_x *= -1;  
}  
PACKET_DATA[2] = which_sign;  
PACKET_DATA[3] = mouse_state.velocity_x;  
  
which_sign = 0;  
if(mouse_state.velocity_y <= 0)  
{  
    which_sign = 1;  
    mouse_state.velocity_y *= -1;  
}  
  
PACKET_DATA[4] = which_sign;  
PACKET_DATA[5] = mouse_state.velocity_y;
```

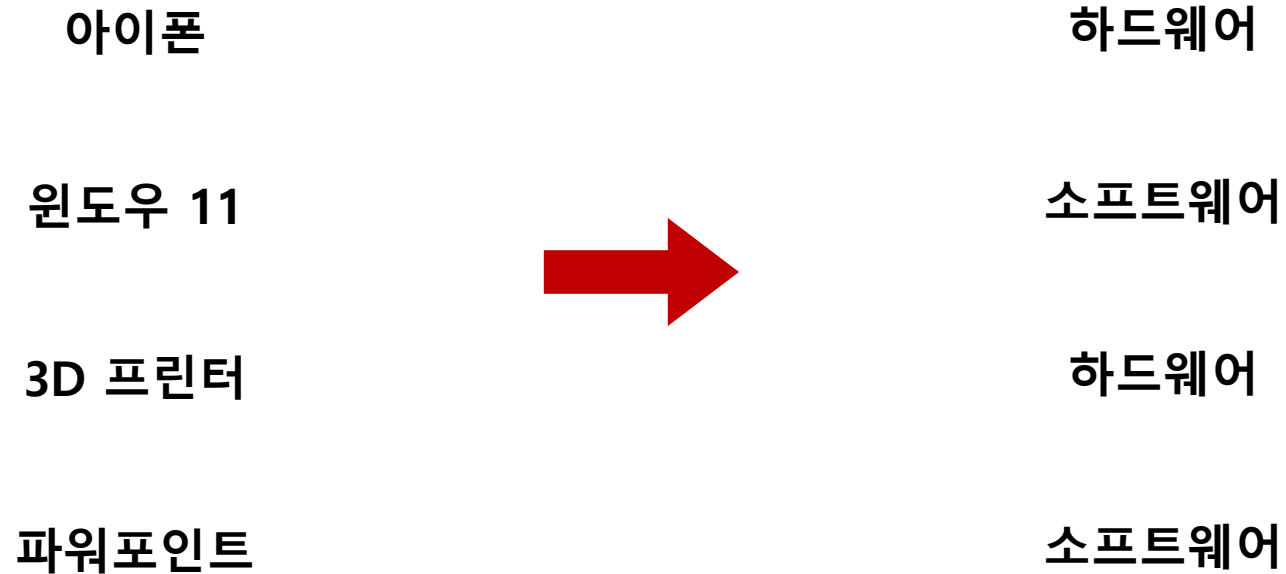
에어팬슬 코드



에어팬슬 프로그램

컴퓨터 구성 요소 및 동작 원리 소개

하드웨어를 개발한다? 소프트웨어를 개발한다?



컴퓨터 구성 요소 및 동작 원리 소개

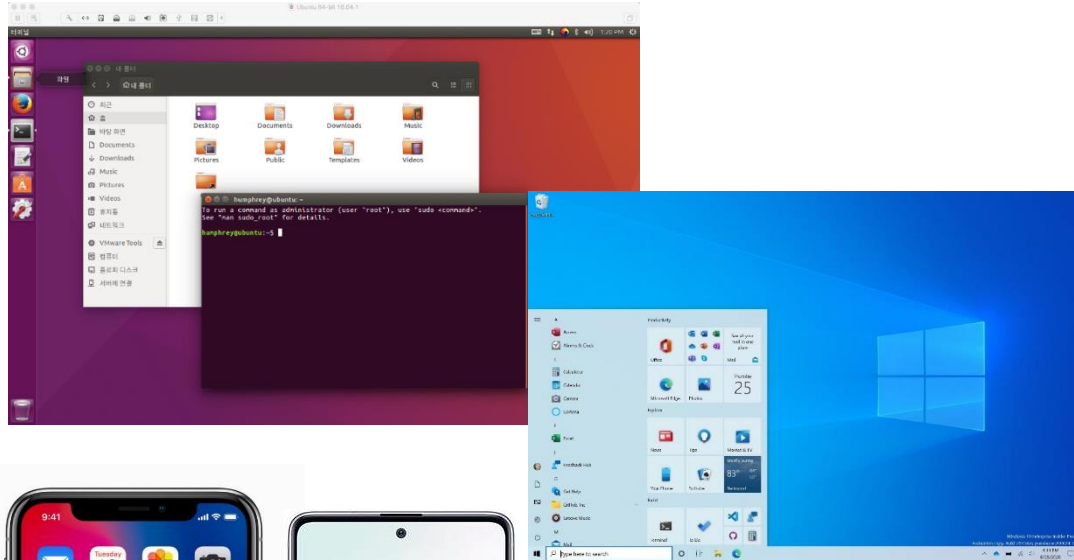
프로그래밍 분야

- 소프트웨어 개발자
- 웹 개발자
- 게임 개발자
- 앱 개발자
- 화이트 해커
- 네트워크 관리자

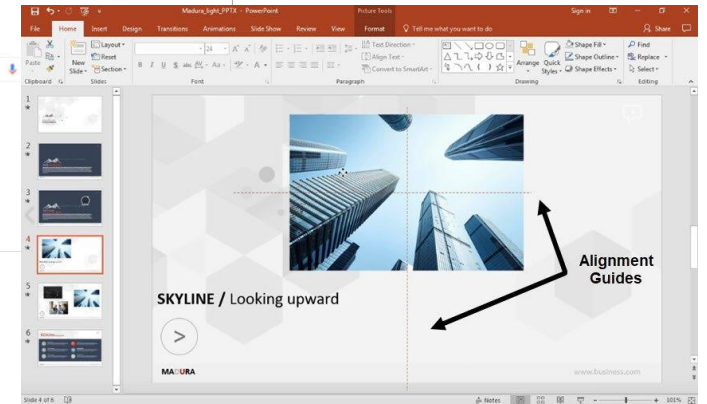
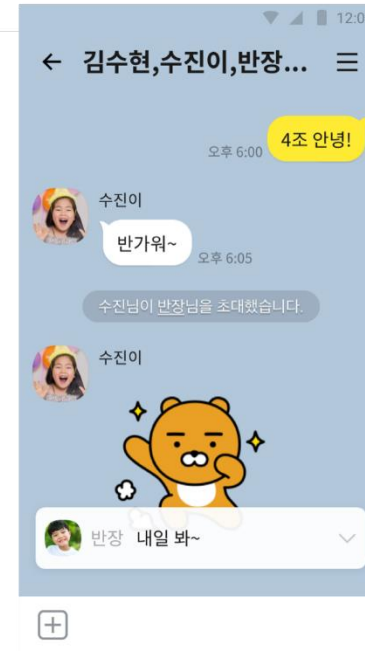


컴퓨터 구성 요소 및 동작 원리 소개

이 둘의 차이는 무엇일까요?



VS



컴퓨터 구성 요소 및 동작 원리 소개

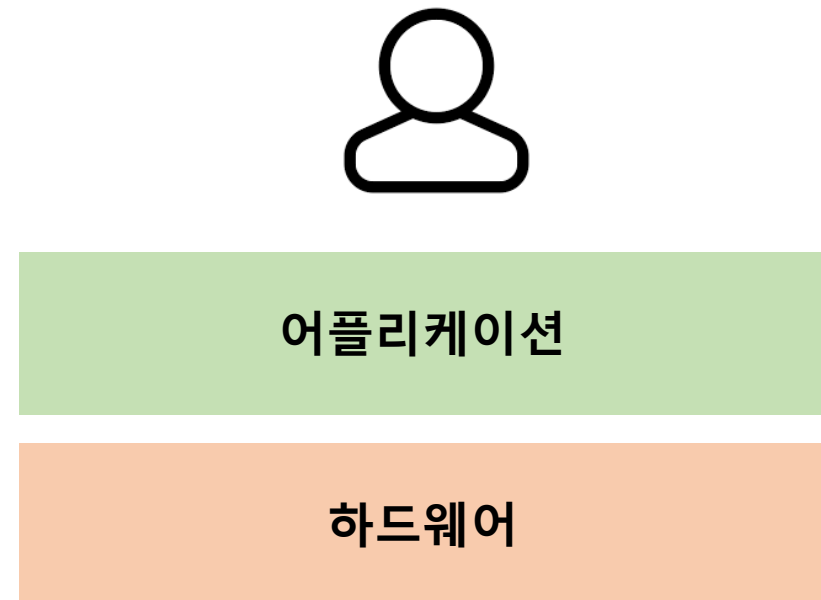
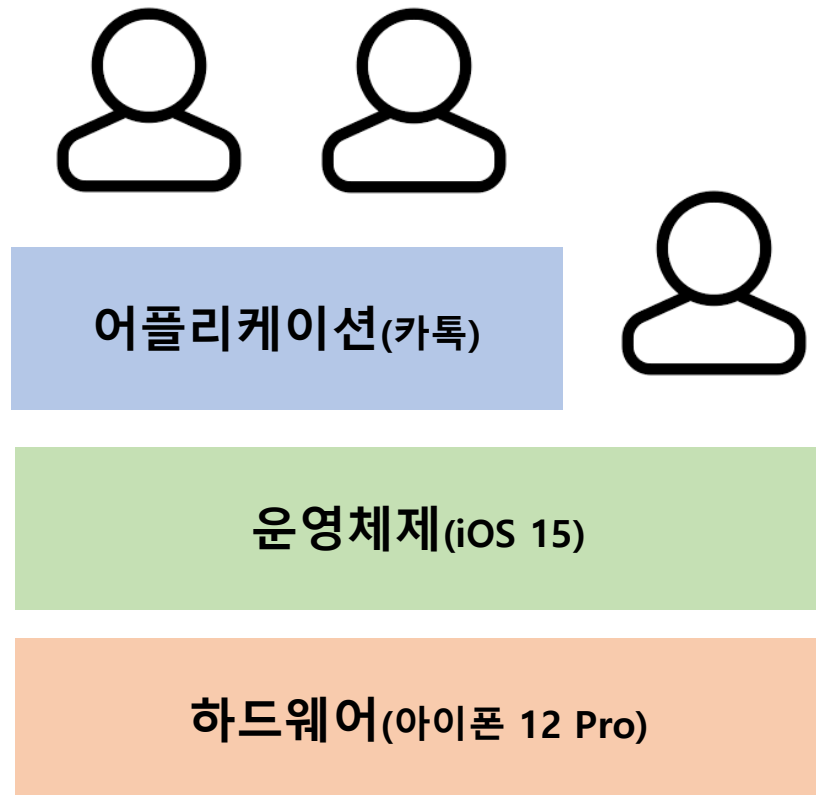
이 둘의 차이는 무엇일까요?

운영체제(OS)

VS

어플리케이션

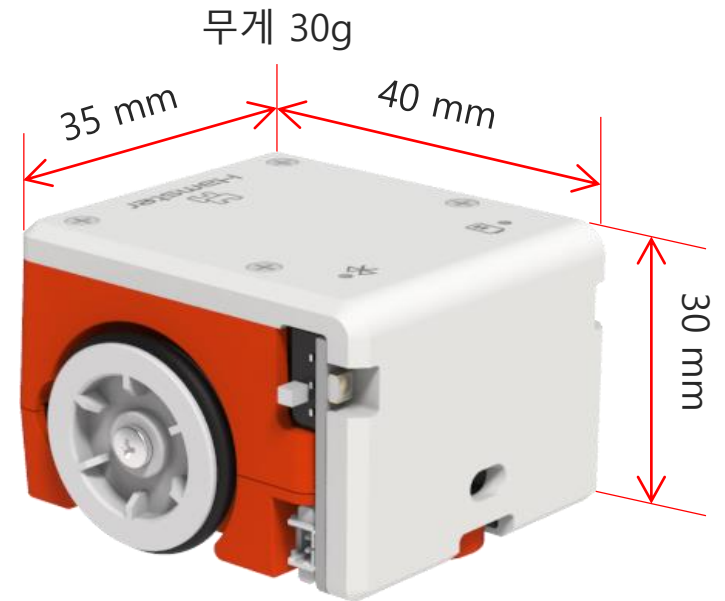
컴퓨터 구성 요소 및 동작 원리 소개



마이크로 로봇 소개

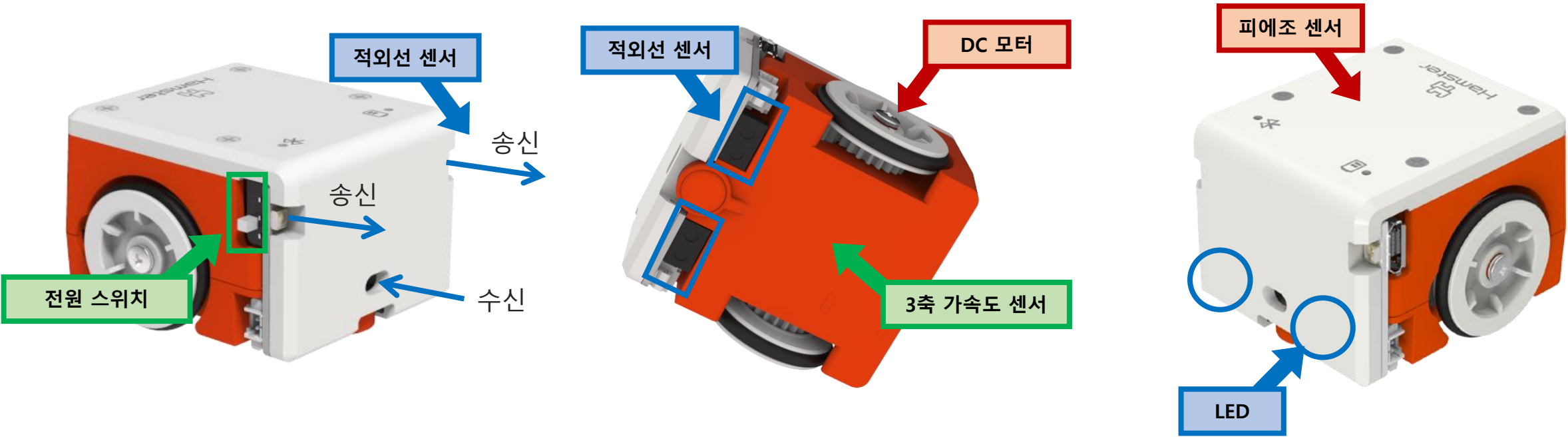
햄스터 로봇이란?

- 작은 소프트웨어 교육용 로봇
- 본체와 USB 동글이 서로 블루투스 연결을 하여 명령어를 전달
- 기본 구성 외 다른 부품으로 다양하게 활용이 가능



마이크로 로봇 소개

햄스터 로봇의 센서



마이크로 로봇 소개

햄스터 로봇의 센서



리튬이온 충전배터리



DC 모터 × 2



바닥(라인) 센서 × 2



3축 가속도 센서



내부 온도 센서



외부 확장 단자 × 2 : 센서 및 모터 추가 연결



블루투스 4.0 BLE



전방 거리 센서 × 2



7가지 색의 LED × 2



조도 센서



피에조 스피커

마이크로 로봇 소개

햄스터 로봇의 센서



리튬이온 충전배터리



DC 모터 × 2



바닥(라인) 센서 × 2



3축 가속도 센서



내부 온도 센서



외부 확장 단자 × 2 : 센서 및 모터 추가 연결



블루투스 4.0 BLE



전방 거리 센서 × 2



7가지 색의 LED × 2



조도 센서



피에조 스피커

마이크로 로봇 소개

햄스터 로봇의 센서



리튬이온 충전배터리



DC 모터 × 2



바닥(라인) 센서 × 2



3축 가속도 센서



내부 온도 센서



외부 확장 단자 × 2 : 센서 및 모터 추가 연결



블루투스 4.0 BLE



전방 거리 센서 × 2



7가지 색의 LED × 2



조도 센서



피에조 스피커

마이크로 로봇 소개

햄스터 로봇의 센서



DC 모터 : 전진, 후진, 회전, 오른쪽, 왼쪽

7가지 색의 LED : 7가지 색으로 왼쪽, 오른쪽 2개의 LED 램프

피에조 스피커 : 소리를 낼 수 있는 스피커

바닥 센서 : 바닥 감지용 적외선 센서

전방 거리 센서 : 전방 물체 감지용 적외선 센서(1 ~ 30cm)

조도 센서 : 주변의 밝기를 측정하는 센서

내부 온도 센서 : 내부 온도를 측정하는 센서

3축 가속도 센서 : 로봇의 움직임 감지를 할 수 있는 센서

마이크로 로봇 소개

햄스터 로봇 구성품

햄스터 로봇 충전을 해주세요!



충전 케이블

햄스터 로봇

USB동글

마이크로 로봇 소개

프로그래밍 언어 동작 원리



마이크로 로봇 소개

프로그래밍 언어 동작 원리



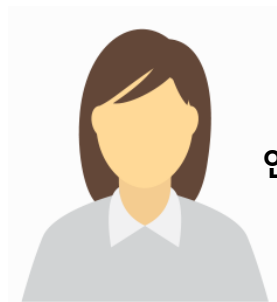
한국인



통역사



미국인



사람

언어



기계

기계어

010010110...

마이크로 로봇 소개

프로그래밍 언어 동작 원리



한국인



통역사



미국인



사람

언어

인터프리터



기계

기계어

010010110...

마이크로 로봇 소개

프로그래밍 언어 동작 원리



한국인



통역사



미국인



사람

언어

인터프리터



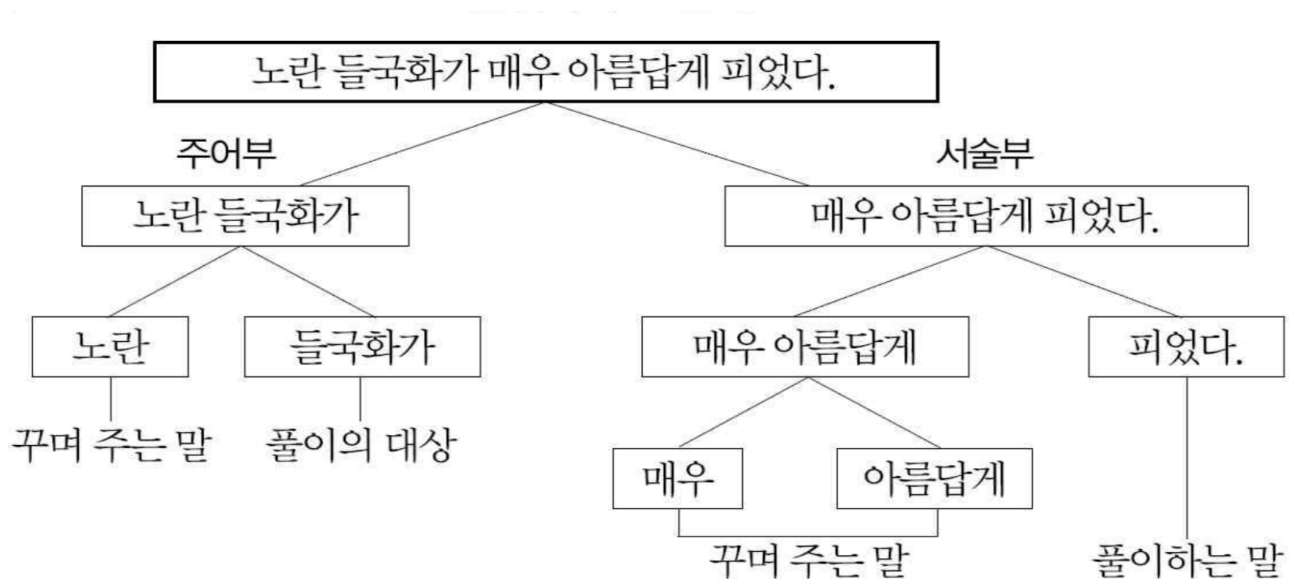
기계

기계어

010010110...

마이크로 로봇 소개

프로그래밍 언어 동작 원리 - 언어의 문법



1형식: 주어 + 동사

명사, 대명사 be동사, 조동사, 일반동사

2형식: 주어 + 동사 + 보어

명사, 형용사, 대명사

3형식: 주어 + 동사 + 목적어

명사, 대명사

4형식: 주어 + 동사 + 간접 목적어 + 직접 목적어

5형식: 주어 + 동사 + 목적어 + 보어

마이크로 로봇 소개

프로그래밍 언어 동작 원리 - 프로그램 언어의 문법

```
def add5(x):  
    return x+5
```

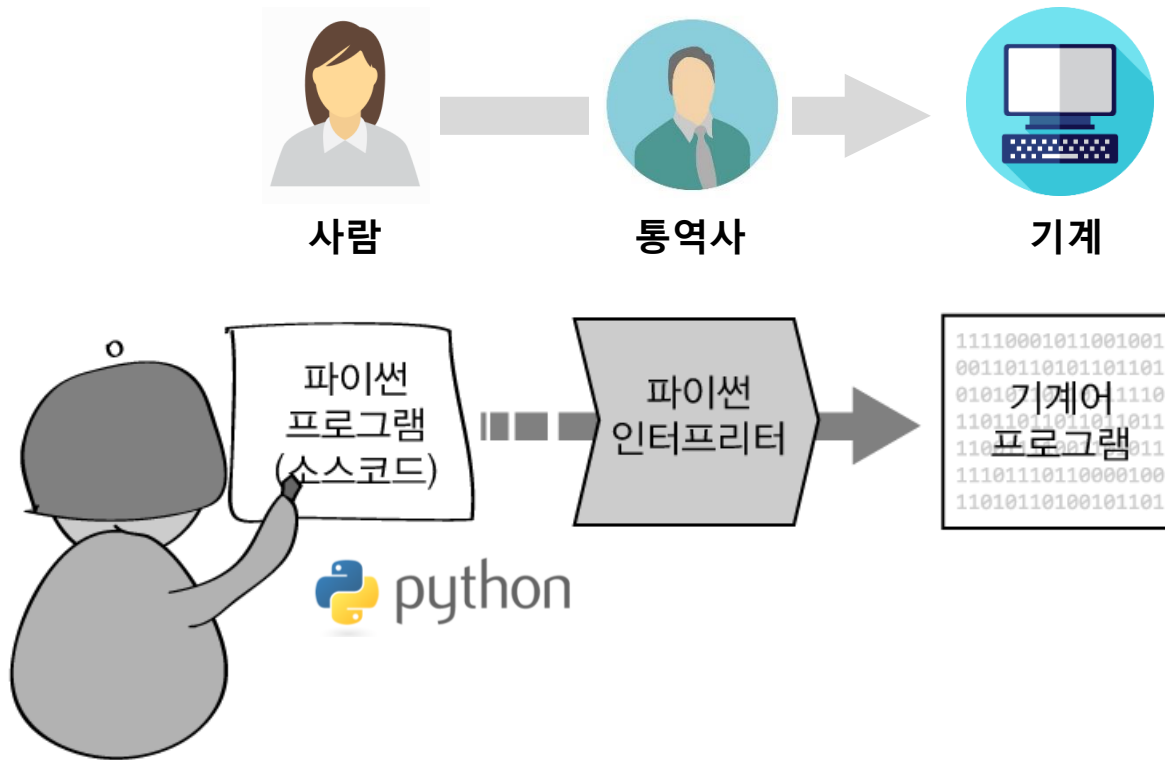
```
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '    %s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s'];' % ast[1]  
        else:  
            print '"]'  
    else:  
        print '"]';'  
    children = []
```

```
#include <stdio.h>  
void sub( int n )  
{  
    printf(" n = %d, &n = %d \n", n, &n );  
}  
  
int main(void)  
{  
    int  a = 20, b = 30, c = 40;  
    puts(" 스택의 이해 ");  
  
    sub( a );  
    sub( b );  
    sub( c );  
  
    puts("주소를 잘 보세요.");  
  
    return 0;  
}
```

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Example</title>  
        <link rel="stylesheet" href="st</head>  
    <body>  
        <h1>  
            <a href="/">Header</a>  
        </h1>  
        <nav>  
            <a href="one/">One</a>  
            <a href="two/">Two</a>  
            <a href="three/">Three</a>  
        </nav>
```

마이크로 로봇 소개

마이크로 로봇 동작 원리



기계어(0100...)



마이크로 로봇 소개

마이크로 로봇 동작 원리



실습 프로그램 설치

설치 순서

1. USB 동글 드라이버 설치(cp2102)

-> 컴퓨터와 햄스터 로봇이 통신하기 위한 통신 모듈

2. Mu Editor 설치

-> python 코딩 환경

3. MuEditor에 roboid 패키지 설치

-> 햄스터 로봇을 python으로 제어하기 위한 라이브러리 다운

실습 프로그램 설치

설치 순서

1. USB 동글 드라이버 설치(cp2102)

-> 컴퓨터와 햄스터 로봇이 통신하기 위한 통신 모듈

2. Mu Editor 설치

-> python 코딩 환경

3. MuEditor에 roboid 패키지 설치

-> 햄스터 로봇을 python으로 제어하기 위한 라이브러리 다운

실습 프로그램 설치

디바이스 드라이버 설치

1. 칩 제조사 홈페이지 접속, Window 전용 디바이스 드라이버 설치

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

OVERVIEW **DOWNLOADS** TECH DOCS COMMUNITY & SUPPORT

Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

Software Downloads

Software (11)

Software • 11

CP210x Universal Windows Driver	v10.1.10 1/13/2021
CP210x VCP Mac OSX Driver	v6.0.1 4/1/2021
CP210x VCP Windows	v6.7.6 9/4/2020
CP210x Windows Drivers	v6.7.6 9/4/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6 9/4/2020

Legacy OS Software Versions

Driver Package download links and support information

Serial Enumeration Driver

What is the serial enumeration driver and why would I need it?

1. 디바이스 드라이버 다운로드

실습 프로그램 설치










디바이스 드라이버 설치

2. 다운 받은 ZIP 파일 압축 풀기

 CP210x_Windows_Drivers	2021-07-13 오후 3:36	압축(ZIP) 파일	7,001KB
 CP210x_Windows_Drivers	2021-07-13 오후 3:47	파일 폴더	

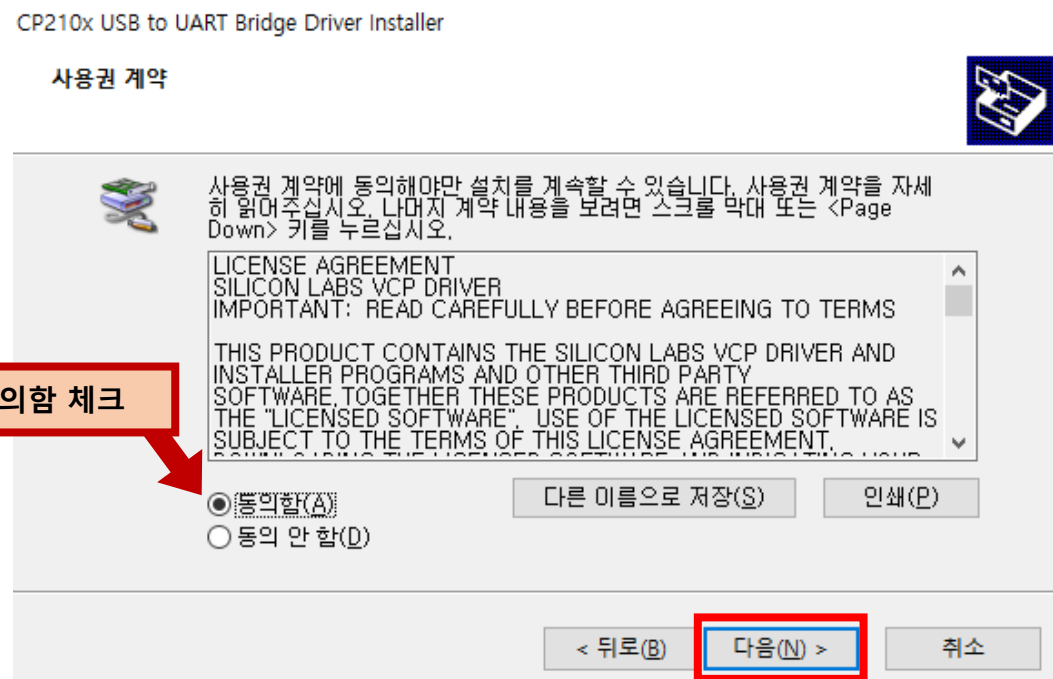
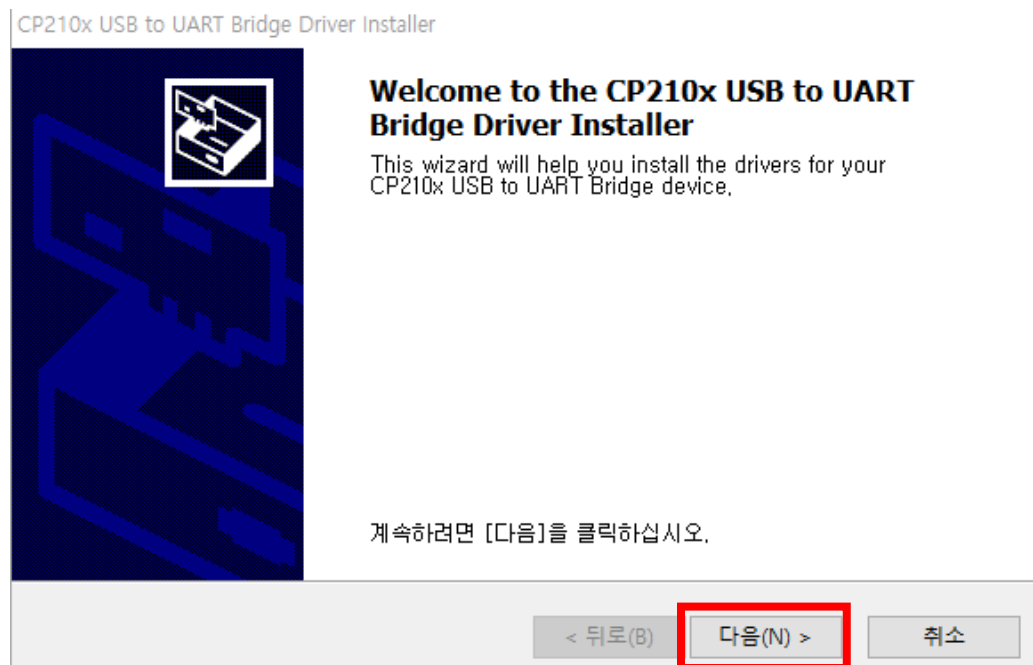
3. 압축 푼 폴더에 있는 exe 파일을 클릭하여 설치

드라이버 실행 파일 클릭

 x64	2018-06-16 오전 12:13	파일 폴더	
 x86	2018-06-16 오전 12:13	파일 폴더	
 CP210xVCPInstaller_x64	2017-09-28 오전 2:58	응용 프로그램	1,026KB
 CP210xVCPInstaller_x86	2017-09-28 오전 2:58	응용 프로그램	903KB
 dpinst	2017-09-28 오전 2:45	XML 문서	12KB
 SLAB_License_Agreement_VCP_Windows	2017-09-28 오전 2:46	텍스트 문서	9KB
 slabvcp	2018-06-02 오전 5:35	보안 카탈로그	11KB
 slabvcp	2018-06-02 오전 5:35	설치 정보	8KB
 v6-7-6-driver-release-notes	2018-06-16 오전 3:51	텍스트 문서	16KB

실습 프로그램 설치

디바이스 드라이버 설치



실습 프로그램 설치

디바이스 드라이버 설치



실습 프로그램 설치

설치 순서

1. USB 동글 드라이버 설치(cp2102)

-> 컴퓨터와 햄스터 로봇이 통신하기 위한 통신 모듈

2. Mu Editor 설치

-> python 코딩 환경

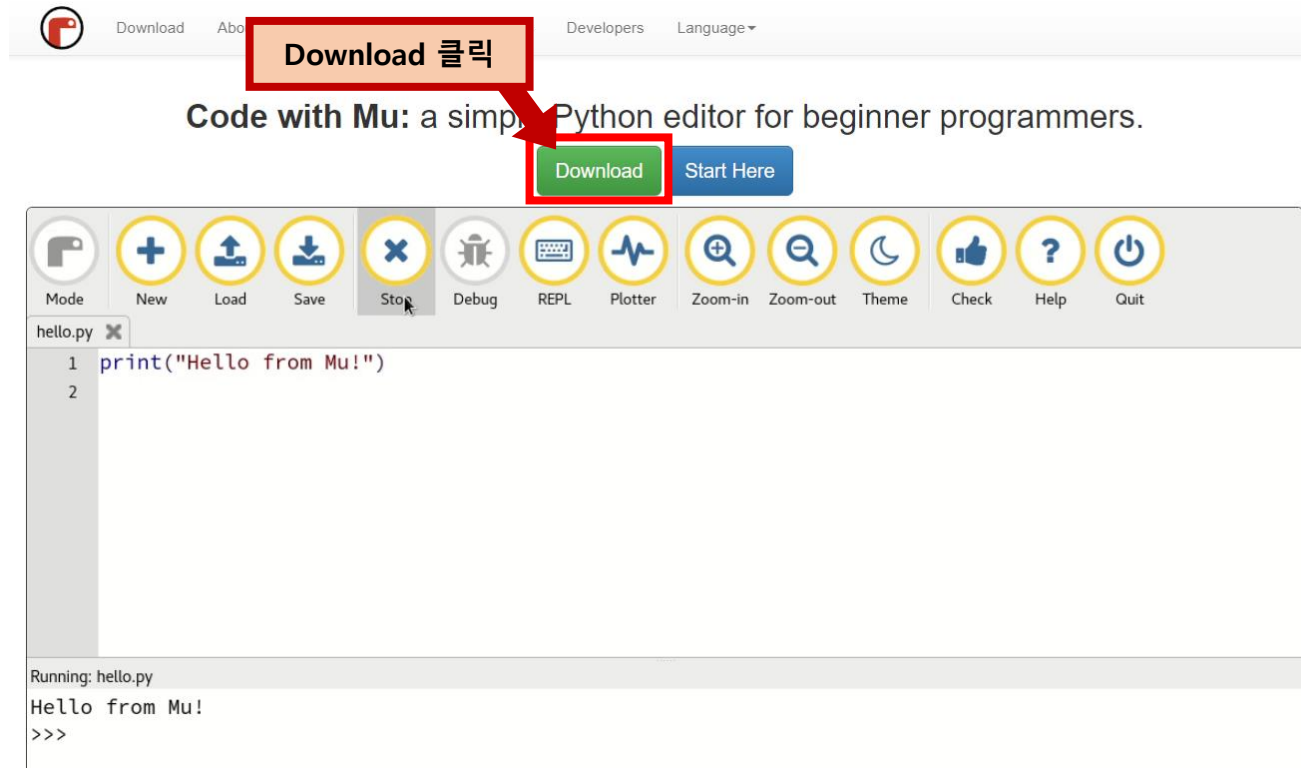
3. MuEditor에 roboid 패키지 설치

-> 햄스터 로봇을 python으로 제어하기 위한 라이브러리 다운

실습 프로그램 설치

IDE 설치

1. <https://codewith.mu/> 접속
2. Download 버튼 클릭



실습 프로그램 설치

IDE 설치

3. Windows installer 64-bit 클릭

4. 내 PC "/>

다운받은 설치 파일 실행

Mu-Editor-Win64-1.1.0b5
CP210x_Windows_Drivers
CP210x_Windows_Drivers

39

실습 프로그램 설치

IDE 설치

5. 추가정보 클릭

6. 실행 버튼 클릭

추가 정보 클릭

Windows의 PC 보호

Microsoft Defender SmartScreen에서 인식할 수 없는 앱의 시작을 차단했습니다. 이 앱을 실행하면 PC가 위험에 노출될 수 있습니다.

추가 정보

실행 안 함

Windows의 PC 보호

Microsoft Defender SmartScreen에서 인식할 수 없는 앱의 시작을 차단했습니다. 이 앱을 실행하면 PC가 위험에 노출될 수 있습니다.

앱: Mu-Editor-Win64-1.1.0b5.msi

게시자: 알 수 없는 게시자

실행 클릭

실행

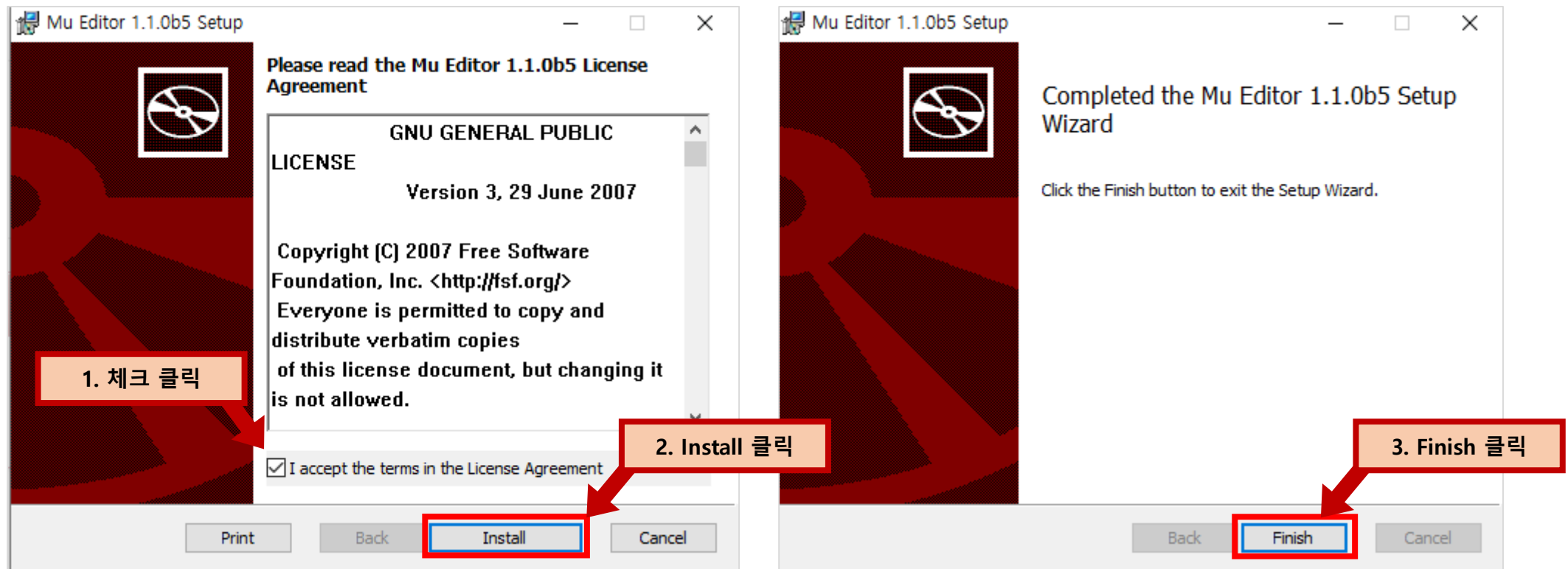
실행 안 함

실습 프로그램 설치

IDE 설치

7. 라이선스 동의 후 설치 버튼 클릭

8. 설치 완료



실습 프로그램 설치

Mu IDE 실행

1. 윈도우 창에 Mu Editor를 검색한다.
2. Mu Editor IDE 프로그램 실행

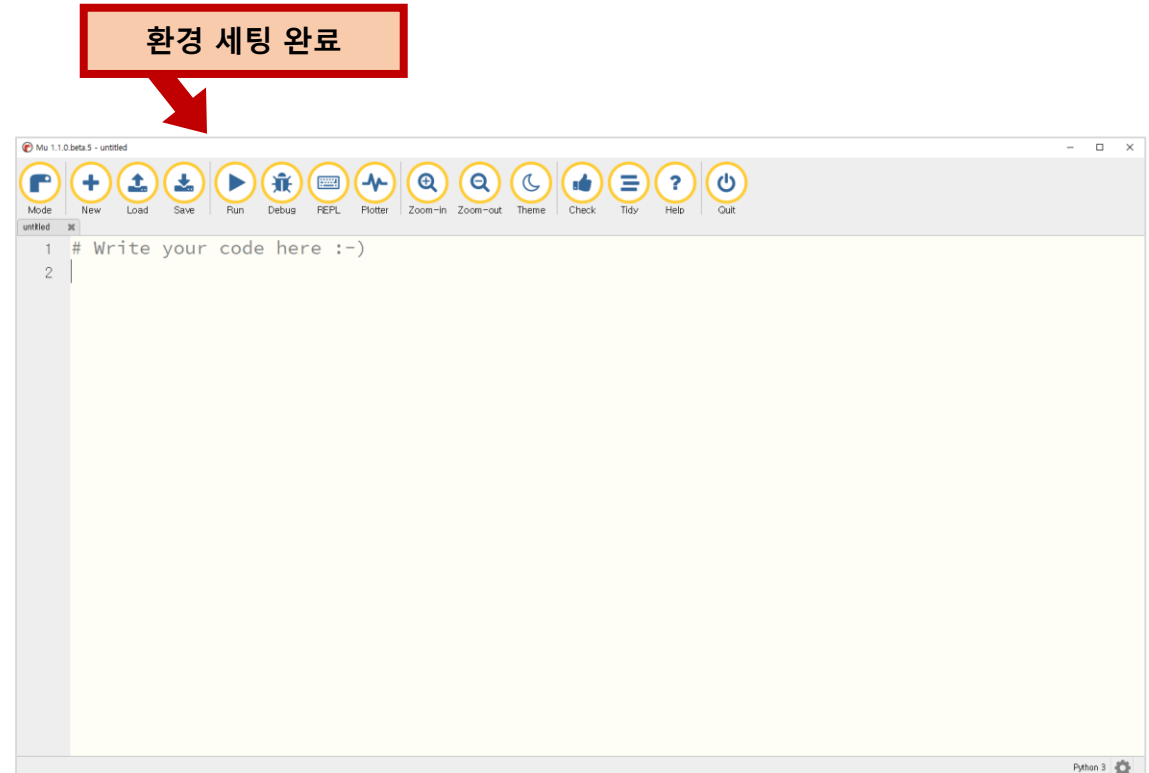
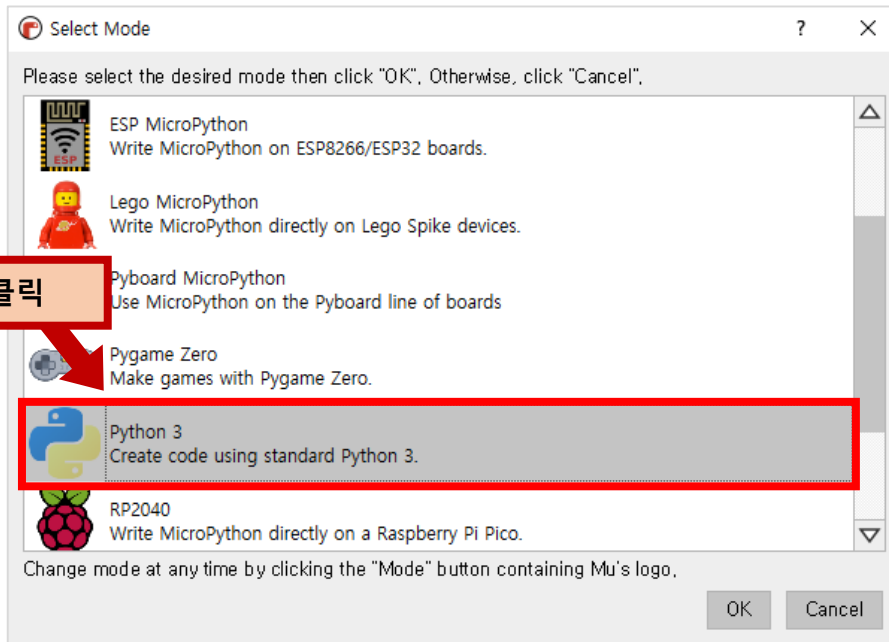


실습 프로그램 설치

Mu IDE 실행

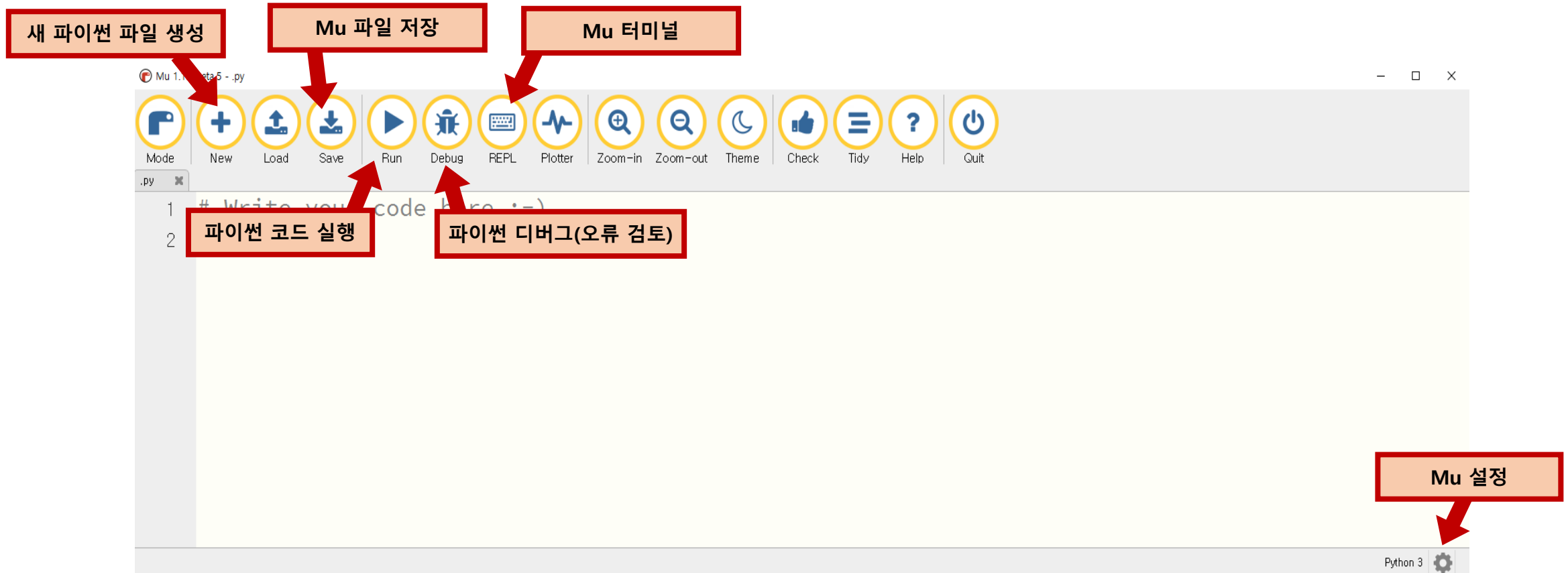
3. Python 3 Mode 선택

4. Mu Editor 세팅 완료



실습 프로그램 설치

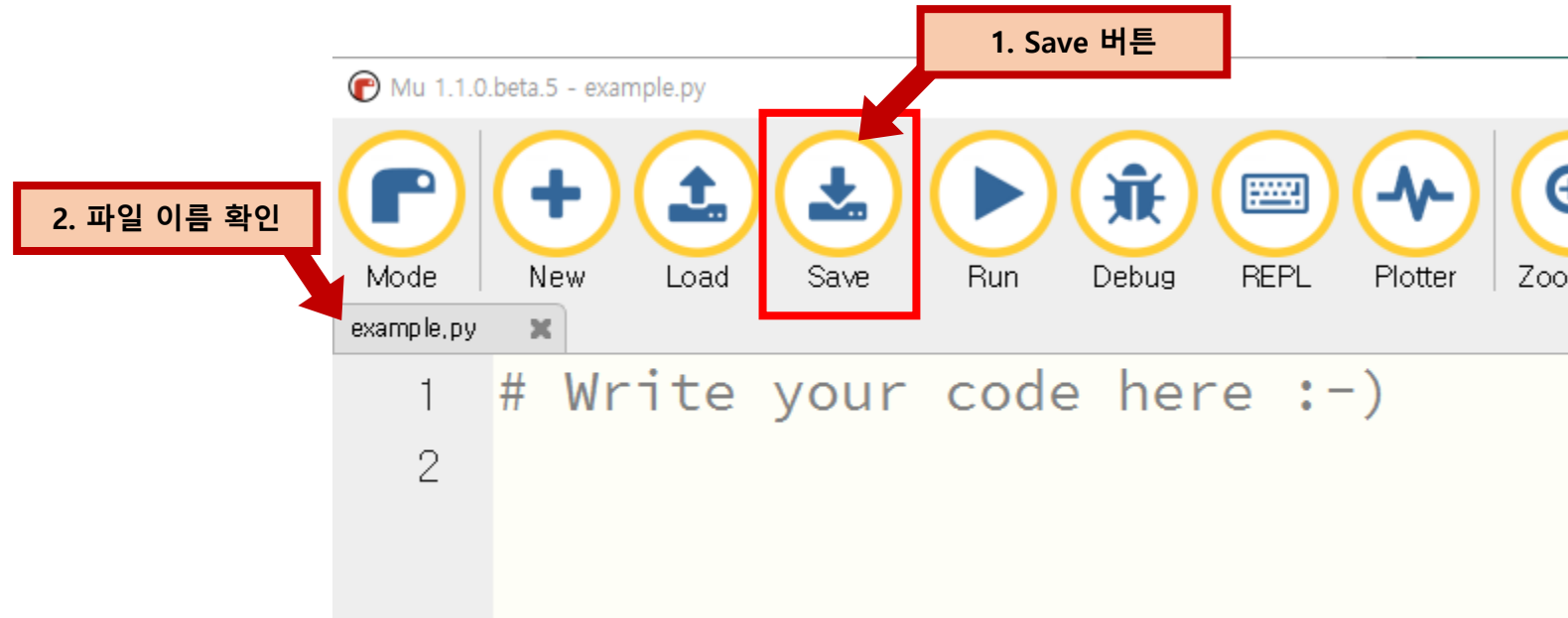
Mu IDE 기능 설명



실습 프로그램 설치

example.py 파일 생성

1. 내 폴더를 열어, D 드라이브에 myMu 폴더 생성
2. 해당 경로에 Mu에서 Save 버튼 클릭하여 example.py 파일 생성



실습 프로그램 확인

python 테스트

1. print("Hello") 입력
2. Run 버튼을 클릭하여 실행
3. 결과값 확인



실습 프로그램 설치

설치 순서

1. USB 동글 드라이버 설치(cp2102)

-> 컴퓨터와 햄스터 로봇이 통신하기 위한 통신 모듈

2. Mu Editor 설치

-> python 코딩 환경

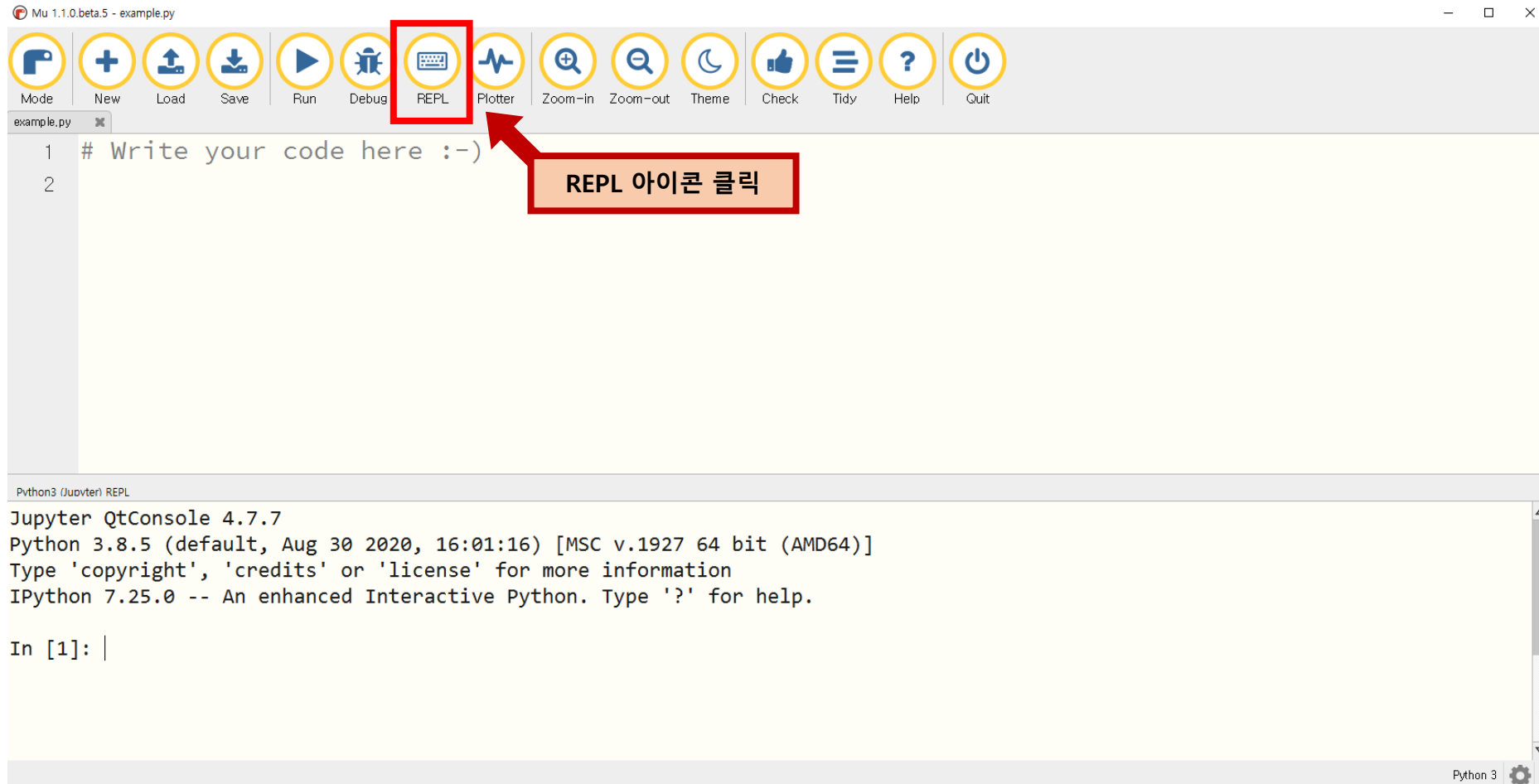
3. MuEditor에 roboid 패키지 설치

-> 햄스터 로봇을 python으로 제어하기 위한 라이브러리 다운

실습 프로그램 설치

라이브러리 설치

1. REPL 아이콘 클릭 하여 터미널 창 열기



실습 프로그램 설치

라이브러리 설치

2. 터미널 명령어 입력 \$ pip install -U roboid

명령어 입력

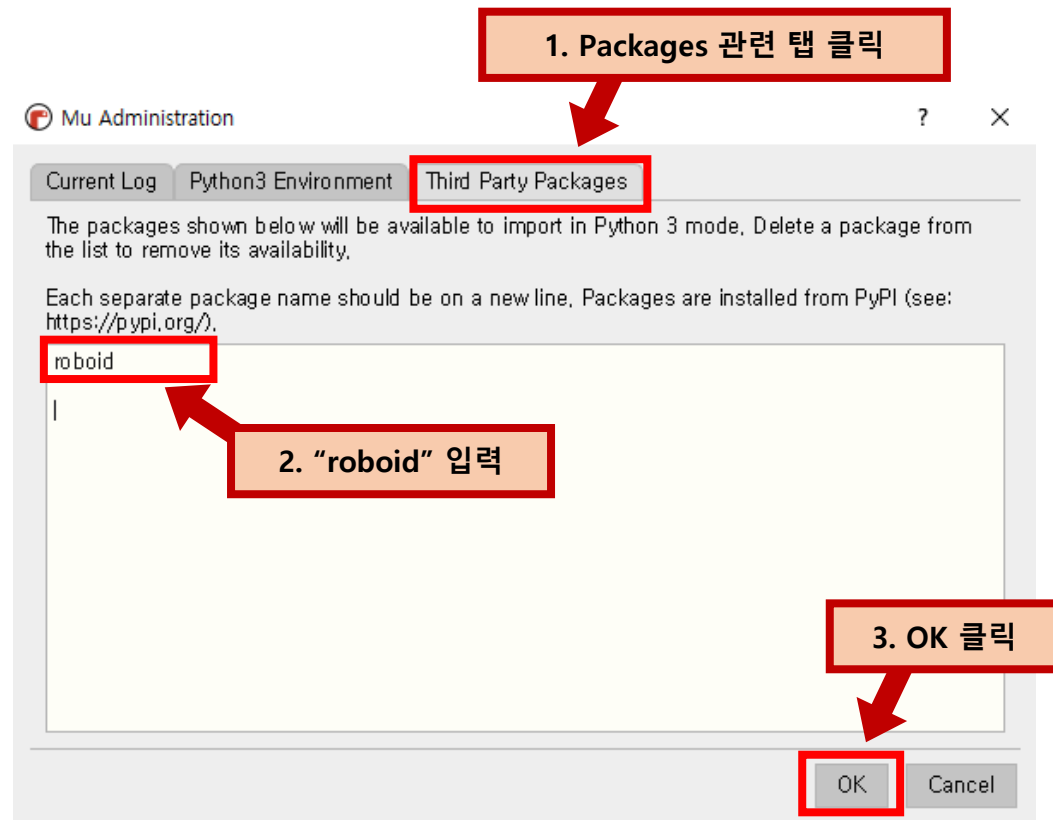
```
Python3 (Jupyter) REPL
Jupyter QtConsole 4.7.7
Python 3.8.5 (default, Aug 30 2020, 16:01:16) [MSC v.1927 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
Python 7.25.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: pip install -U roboid
Collecting roboid
  Using cached roboid-1.5.14-py3-none-any.whl (64 kB)
```

실습 프로그램 설치

라이브러리 설치 다른 방법

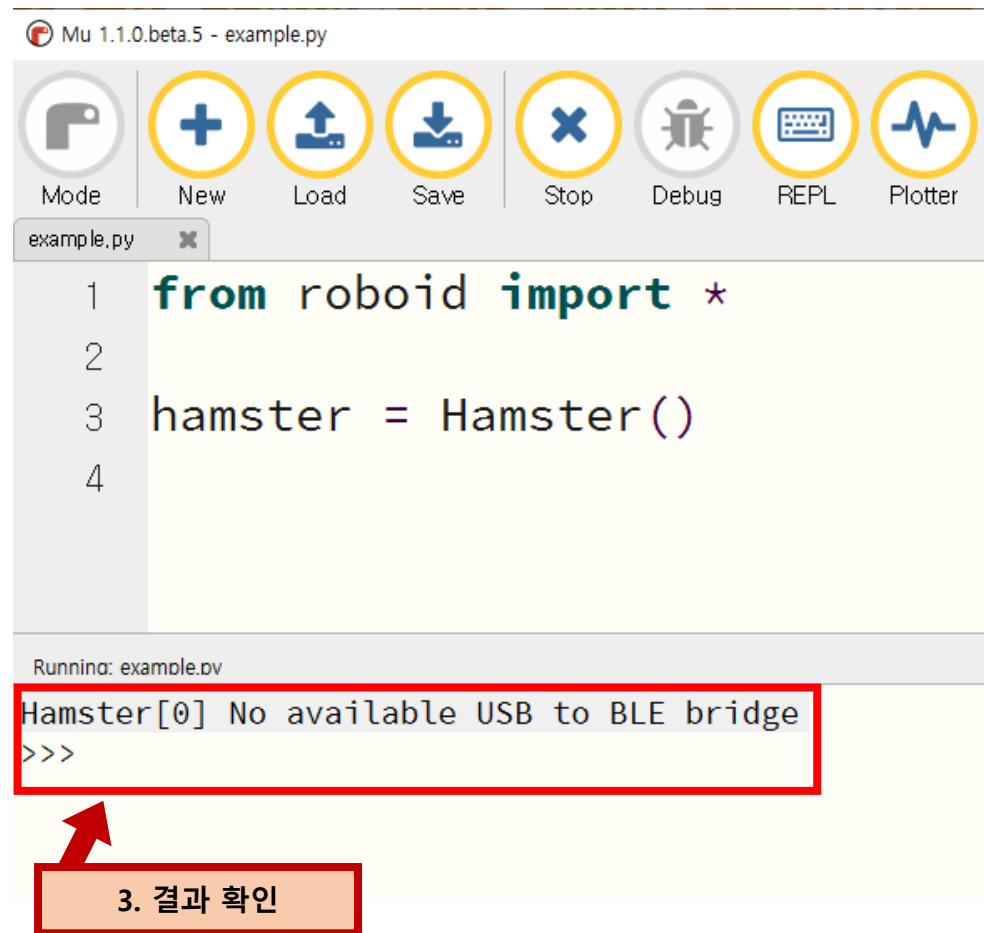
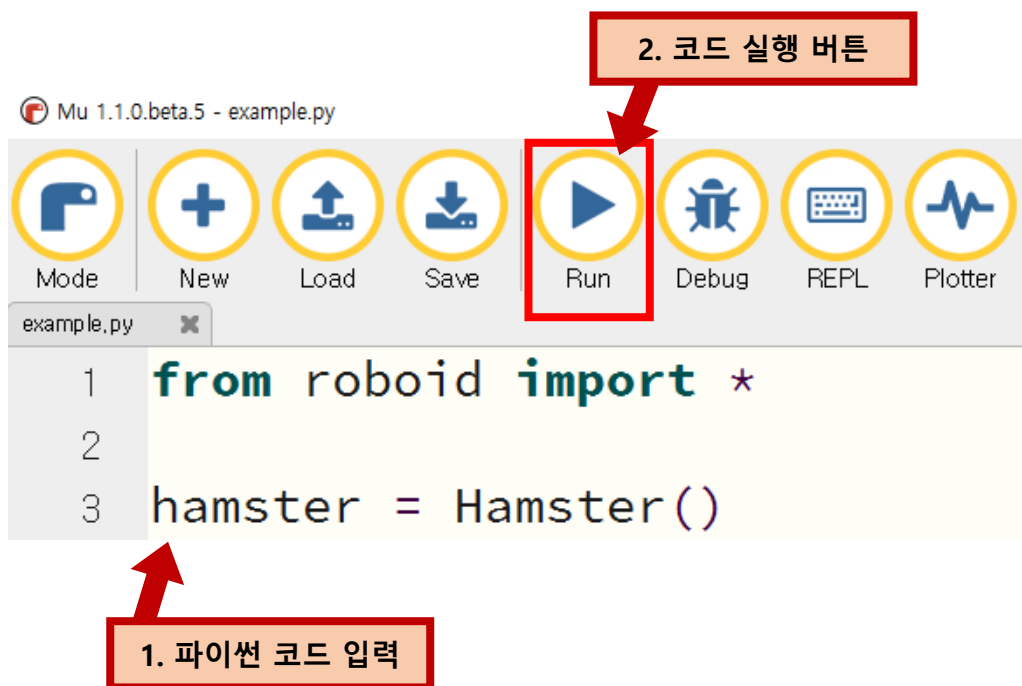
1. 설정 아이콘 클릭



실습 프로그램 확인

햄스터 모듈 테스트

1. 아래의 코드를 입력
2. Run 버튼을 클릭하여 실행
3. 결과 확인



: 드라이버와 패키지가 모두 잘 설치 됨
(햄스터 로봇과 연결 가능한 상태)

실습 프로그램 설치 완료



설치 완료.

파이썬

파이썬(python)이란?

- 1991 귀도 반 로섬 개발자에 의해 만들어진 인터프리터 프로그래밍 언어

파이썬의 특징

- 고급 프로그래밍 언어
- 인터프리터 언어
- 탭으로 구분
- 윈도우, 리눅스 등 다양한 OS 환경에서 사용 가능

파이썬을 이용한 서비스

- 카카오, 구글, NASA 등...



파이썬 코딩 방법 2가지

1. 터미널 창에 직접

```
C:\Users\Wj\anglab>python
Python 3.7.0 (default, Jun 28 2018, 08:04:48)
Type "help", "copyright", "credits" or "license()"
>>> a = 5
>>> a
5
>>> b = 10
>>> b
10
>>> a + b
15
>>>
```

2. “.py”파일을 만들어 코드를 작성한 후 실행

```
1 from CNLP.Engine.Tokenizer.RegexTokenizer import RegexTokenizer
2
3 tokenizer = RegexTokenizer()
4 print(tokenizer.tokenize("안녕111나222는 대한3민국4학생555이야!!"))
5 print(tokenizer.tokenize("27일 '놀라운 토요일'에서 멤버들은 아우내 순대국밥을 먹고 지드래곤의 '크레용' 가사 맞추기를 했다."))
```

```
D:\Projects\NLP\CNLP\CNLP\Engine\Tokenizer>python RegexTokenizer.py
```

['안녕', '111', '나', '222', '는', '대한', '3', '민국', '4', '학생', '555', '이야', '!', ' ', '!!']

['27', '일', '"', '놀라운', '토요일', '"', '에서', '멤버들은', '아우내', '순대국밥을', '걸고', '지드래곤의', '"', ' ', '크레용', '"가사", '맞추기를', '했다', ' ', ' ']

파이썬의 자료형

변수(Variable)

- 데이터의 형태(정수, 실수, 문자)를 구분하며, 데이터의 범위나 부호 사용 여부 등을 결정 짓는 것

없음	None	0 byte
불	bool/boolean	1 byte
정수	int(integer)	4 byte
실수	float, double	float – 4, double – 8
단일 문자	char(character)	1 byte
문자열	string	
리스트	list	
사전	dictionary	
튜플	tuple	

변수와 상수



변수(Variable)

- 한가지 값으로 고정되지 않고, 여러가지 값으로 변할 수 있는 공간

상수(Constant)

- 한가지 값으로 고정되며, 수정되지 않음

변수와 상수

실습

1. Mu의 빈 “.py” 를 실행한다.
2. 아래의 창에 >>> 옆에 커서를 둔다.
3. 해당 터미널에 다음과 같이 입력한다.

```
>>> a = 7
```

```
>>> a
```

```
>>> b = 10
```

```
>>> a + b
```

```
>>> test = "hello"
```

```
>>> test
```

표준 출력 함수

print

- cmd창에 우리가 입력한 자료형을 출력할 때 사용

사용방법

```
print("Hello World")
```

```
print("I am Ratataca!")
```

```
>>> print("Hello")
Hello
>>> print("I am Ratataca!")
I am Ratataca!
>>>
```

표준 입력 함수

input

- 키보드를 이용하여 변수에 사용자가 값을 입력할 때 사용

사용방법

```
a = input("Enter the string : ")
```

주로 문자 입력

```
print(a)
```

```
>>> a = input("Enter the string : ")
Enter the string : hello! I am Ratataca
>>> print(a)
hello! I am Ratataca
>>>
```

```
b = int(input("Enter the Value : "))
```

주로 숫자 입력

```
print(b)
```

```
>>> b= int(input("Enter the Value : "))
Enter the Value : 192
>>> print(b)
192
```

표준 입력 함수

input

- 키보드를 이용하여 변수에 사용자가 값을 입력할 때 사용

사용방법

```
a = input("Enter the string : ")
```

주로 문자 입력

```
print(a)
```

```
>>> a = input("Enter the string : ")
Enter the string : hello! I am Ratataca
>>> print(a)
hello! I am Ratataca
>>>
```

```
b = int(input("Enter the Value : "))
```

주로 숫자 입력

```
print(b)
```

```
>>> b= int(input("Enter the Value : "))
Enter the Value : 192
>>> print(b)
192
```

조건문

조건문 : 수행할 경우를 조건에 따라 분류

만약에 유명한 연예인이 나에게 사귀자고 한다면?

조건문

조건문 : 수행할 경우를 조건에 따라 분류

사권다?

안 사권다?

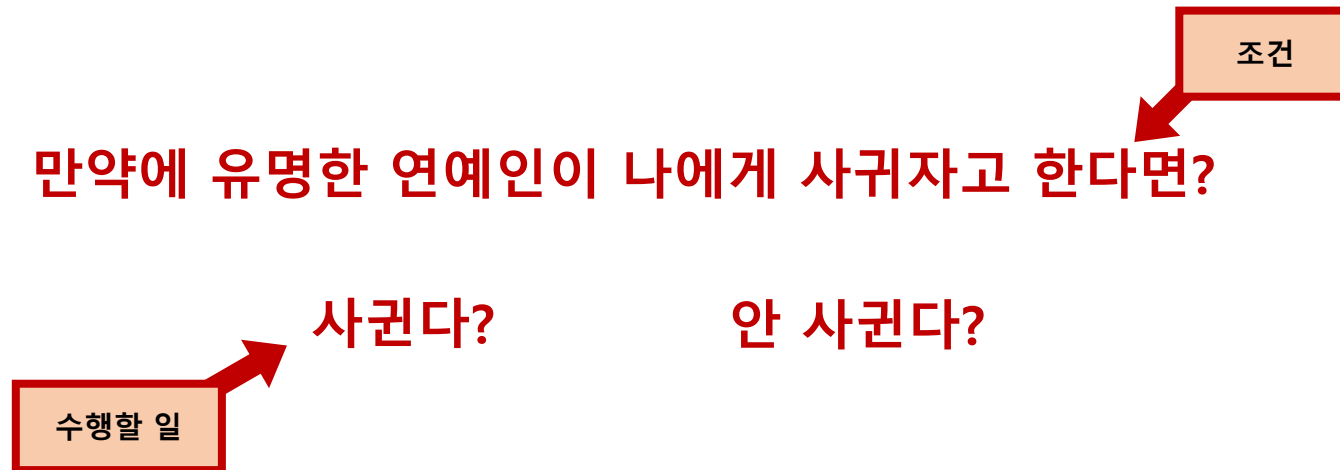
조건문

조건문 : 수행할 경우를 조건에 따라 분류



조건문

조건문 : 수행할 경우를 조건에 따라 분류



조건문

조건문 : 수행할 경우를 조건에 따라 분류

한국어 : 만약에 유명한 연예인이 나에게 사귀자고 한다면?

파이썬 : if 유명한 연예인이 나에게 사귀자고 한다면:

조건문

조건문 : 수행할 경우를 조건에 따라 분류

한국어 : 만약에 유명한 연예인이 나에게 사귀자고 한다면?

파이썬 : if 유명한 연예인이 나에게 사귀자고 한다면:

조건

조건문

조건문 : 수행할 경우를 조건에 따라 분류

한국어 : 만약에 유명한 연예인이 나에게 사귀자고 한다면?

사귀다 ← 수행할 일

파이썬 : if 유명한 연예인이 나에게 사귀자고 한다면:

사귀다 ← 수행할 작업


조건문

조건문 : 수행할 경우를 조건에 따라 분류

한국어 : 만약에 유명한 연예인이 나에게 사귀자고 한다면?

사귀다 ← 수행할 일

파이썬 : if 유명한 연예인이 나에게 사귀자고 한다면:

 사귀다 ← 수행할 작업

수행할 작업은 컴퓨터가 이해할 수 있게 들여쓰기 하기! (탭 1번)

조건문

조건문

if - 조건이 참인지 거짓인지 구분, 조건이 참이면 if 문 내 코드를 수행

```
she_like_me = True  
if she_like_me == True:  
    print("I love you too")
```

```
3 she_like_me = True  
4  
5 if she_like_me == True:  
6     print("I love you too")
```

조건문

조건문 : 수행할 경우를 조건에 따라 분류

만약에 내가 좋아하는 사람에게 고백을 했는데 성공했다면?

사귀다?

못 사귀다?

조건문

조건문 : 수행할 경우를 조건에 따라 분류



조건문

조건문 : 수행할 경우를 조건에 따라 분류

그게 아니라면? $\pi\pi$

조건문

조건문 : 수행할 경우를 조건에 따라 분류

사~~기~~다?

못 사권다?

조건문

조건문 : 수행할 경우를 조건에 따라 분류

한국어 : 만약에 내가 좋아하는 사람에게 고백했는데 성공했다면?

사귀다 ← 성공 했을 때 수행할 일

그게 아니라면? $\pi\pi$

못 사귀다 ← 실패 했을 때 수행할 일

파이썬 : if 내가 좋아하는 사람에게 고백했는데 성공했다면 :

사귀다 ← 성공 했을 때 수행할 일

else:

못 사귀다 ← 실패 했을 때 수행할 일

조건문

조건문

if - 조건이 참인지 거짓인지 구분, 조건이 참이면 if 문 내 코드를 수행

else - 조건이 거짓인 경우, else 문 내 코드를 수행

```
would_you_marry_me = True
if would_you_marry_me == True:
    print("Nice")
else:
    print("I'm so sad ☹️")
```

```
3 would_you_marry_me = False
4
5 if would_you_marry_me == True:
6     print("Nice")
7 else:
8     print("I'm so sad")
```

반복문

반복문 : 동일한 작업을 여러 번 수행

아침 먹고 땡

반복문

반복문 : 동일한 작업을 여러 번 수행

~~하고 나서려는데 화려한 조명... 똥 밟...~~

반복문

반복문 : 동일한 작업을 여러 번 수행

공부...

반복문

반복문 : 동일한 작업을 여러 번 수행

점심 먹고 공부

반복문

반복문 : 동일한 작업을 여러 번 수행

저녁 먹고 공부

반복문

반복문 : 동일한 작업을 여러 번 수행

한국어 : 아침 먹고 공부, 점심 먹고 공부, 저녁 먹고 공부... 계속 공부...

파이썬 : while True:

조건이 참인 동안 계속 반복

공부 ← 수행할 작업

반복문

반복문

while - 조건이 참인 경우, **while** 문 내 코드를 계속 반복하여 수행

while True:

 print("Studying...")

3 **while True:**

4 **print**("Studying...")

반복문

반복문 : 동일한 작업을 여러 번 수행

고 1때 공부...

반복문

반복문 : 동일한 작업을 여러 번 수행

고 2때 공부...

반복문

반복문 : 동일한 작업을 여러 번 수행

고 3때 공부...

반복문

반복문 : 동일한 작업을 여러 번 수행

대학 합격하면?

반복문

반복문 : 동일한 작업을 여러 번 수행

공부 왜 해??

반복문

반복문 : 동일한 작업을 여러 번 수행

한국어 : **고1때** 공부, **고 2때** 공부, **고 3때** 공부

파이썬 : for grade in **range(1, 3):**

조건이 참인 동안 계속 반복

공부 ← 수행할 작업

range() - 범위 표현 함수

range(시작점, 끝점)

range(시작점, 끝점, 간격)

예시)

range(1, 3) - 1, 2, 3

range(1, 10, 2) - 1, 3, 5, 7, 9

range(3, 1, -1) - 3, 2, 1

반복문

반복문

for - 주어진 범위 내에 **for** 문 내 코드를 반복하여 수행

```
for grade in range(1, 3):  
    print("Studying...")
```

```
print("I'm not studying!")
```

```
3 for grade in range(1, 3):  
4     print("Studying...")  
5  
6 print("I'm not studying!")
```

마이크로 로봇 연결

햄스터 로봇과 컴퓨터 연결

1. 자신의 햄스터 로봇과 USB 동글에 표시(이름, 사인, 특수 모양 등)



2. USB 동글을 컴퓨터 USB 포트에 연결

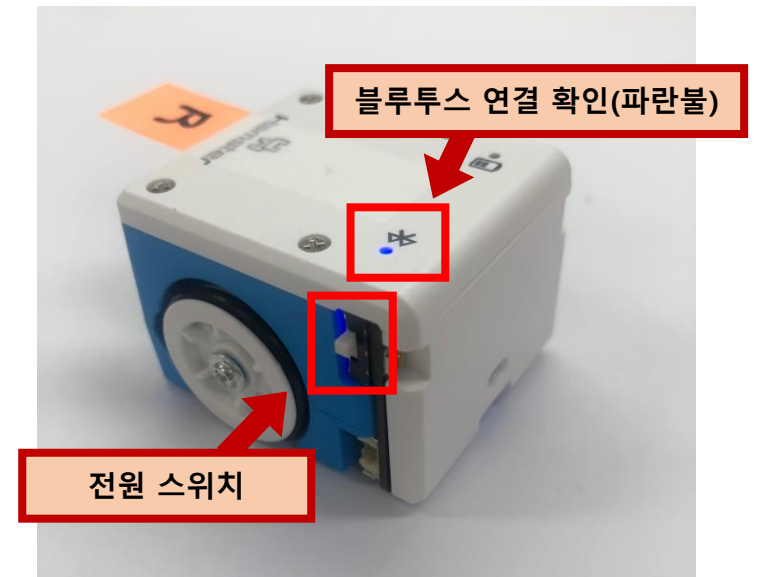
3. USB 동글에 파란 불이 들어오면(천천히 깜빡) 정상 작동 중인 상태



마이크로 로봇 연결

햄스터 로봇과 컴퓨터 연결

4. 햄스터 로봇의 전원 버튼 ON,
5. 햄스터 로봇과 USB 동글이 15cm 이내로 가깝게 둬
6. 뽁- 소리가 나고 파란불이 지속적으로 들어면 정상 연결됨



마이크로 로봇 연결

컴퓨터와 연결 후 동작 테스트

7. 해당 코드를 다시 실행한다.

```
1 from roboid import *  
2  
3 hamster = Hamster()  
4
```

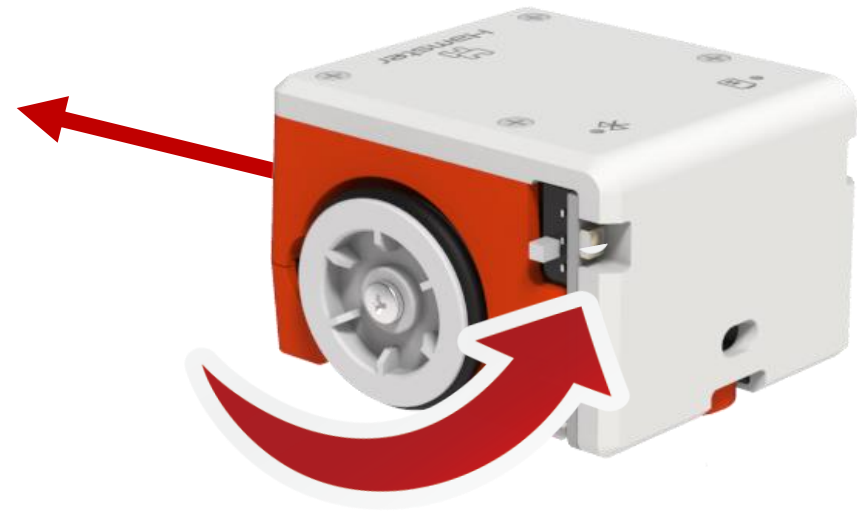
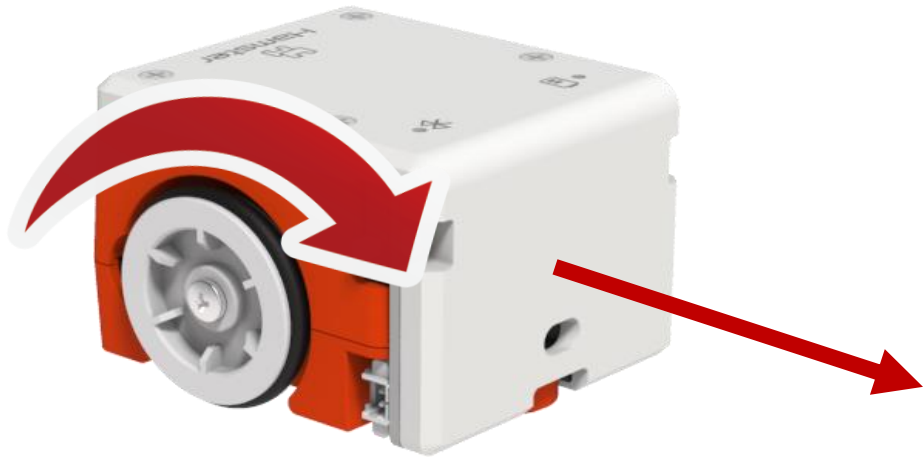
Running: .py

```
Hamster[0] Connected: COM6 FF:59:58:56:81:5F  
>>>
```

정상 연결 시 출력 메시지

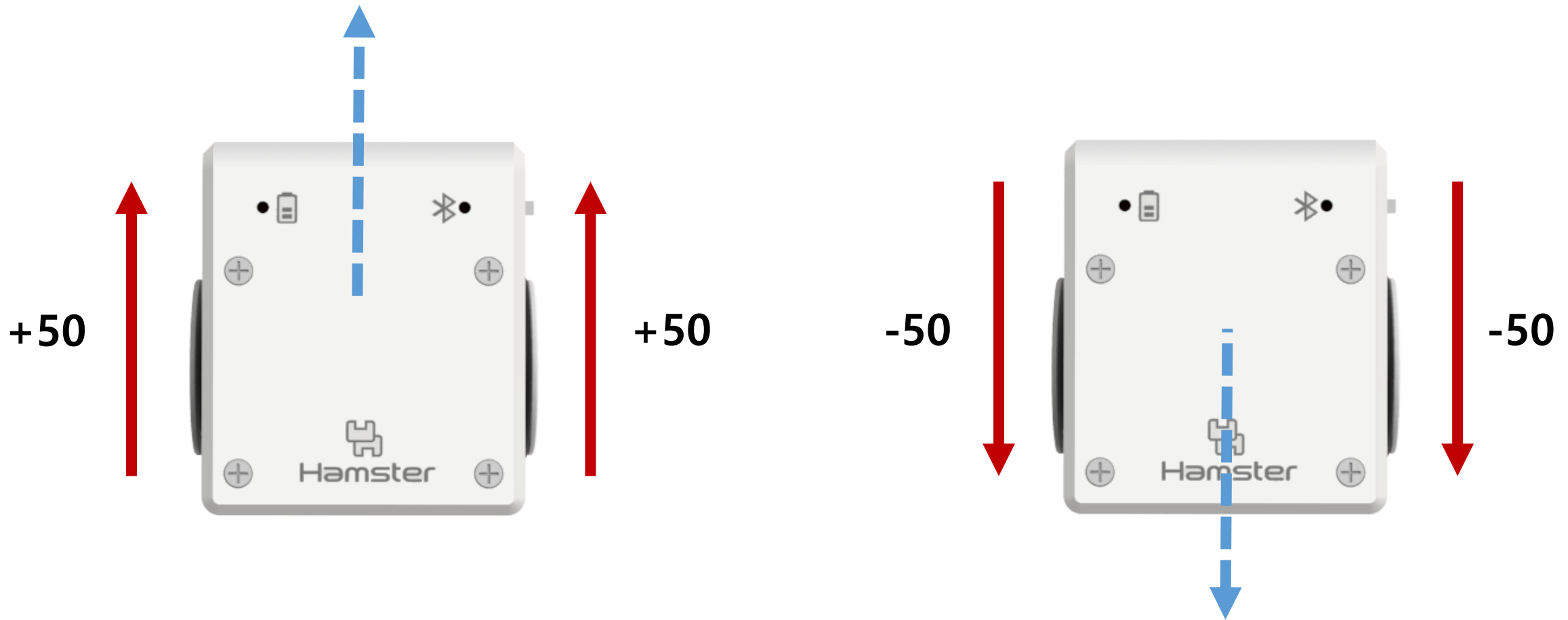
마이크로 로봇 모터 동작

모터 센서



마이크로 로봇 모터 동작

모터 센서



마이크로 로봇 모터 동작

모터 센서



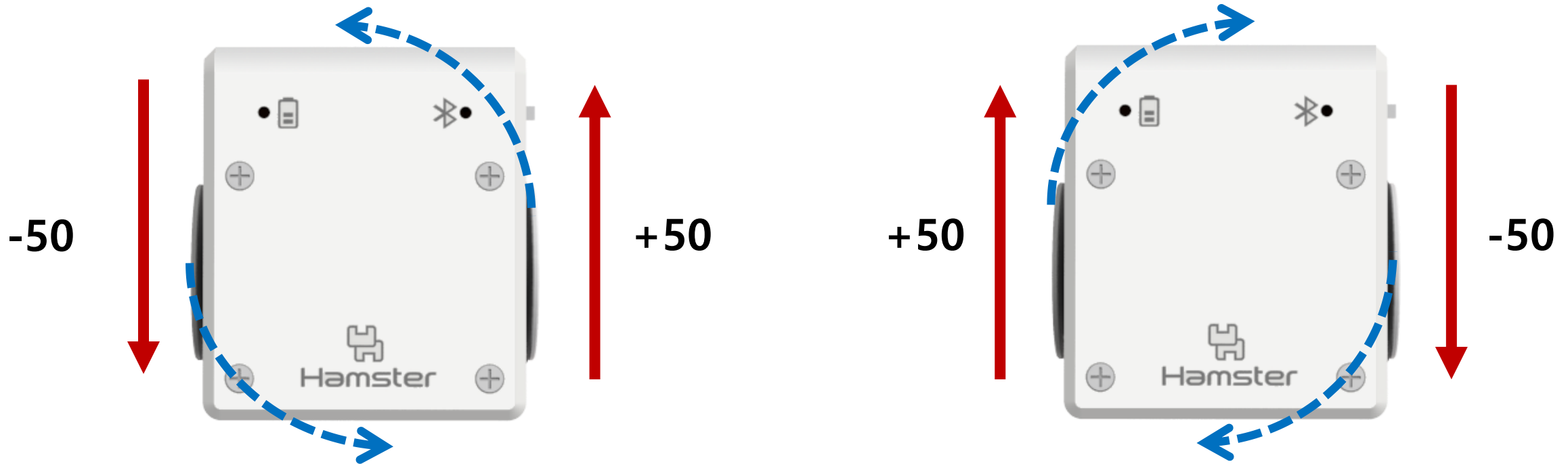
+50

-50



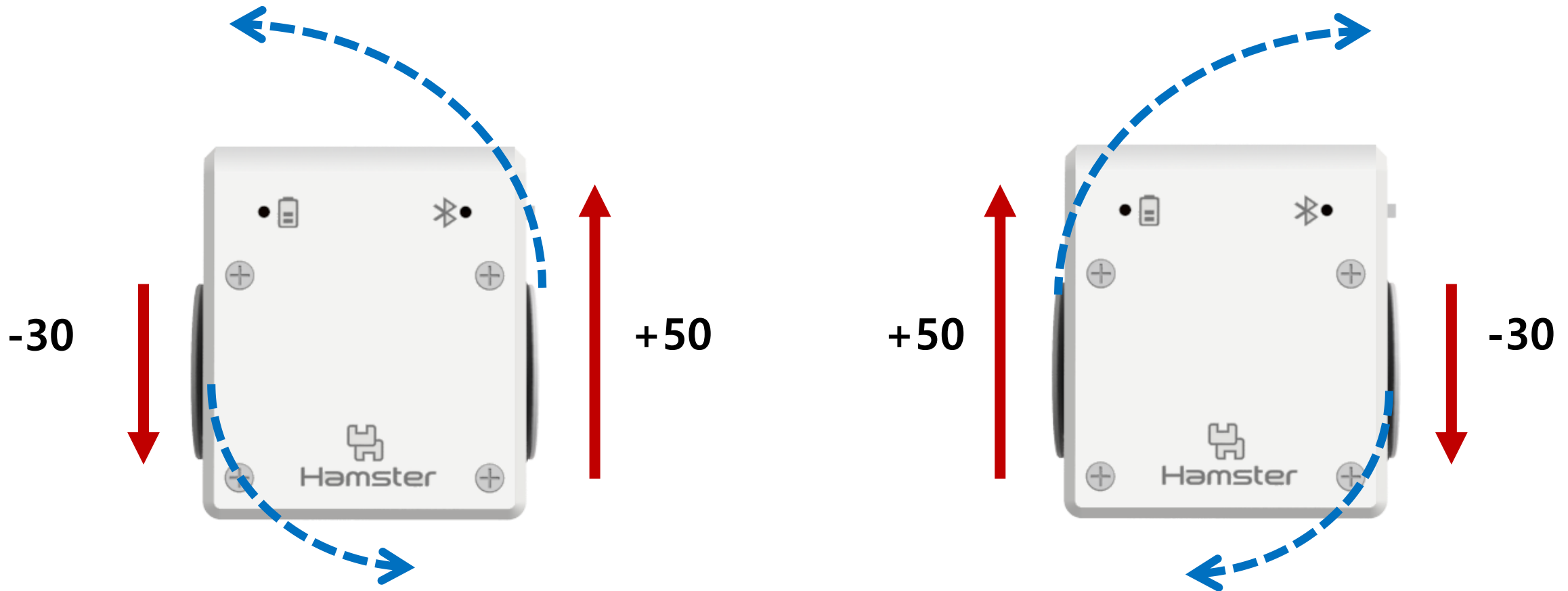
마이크로 로봇 모터 동작

모터 센서



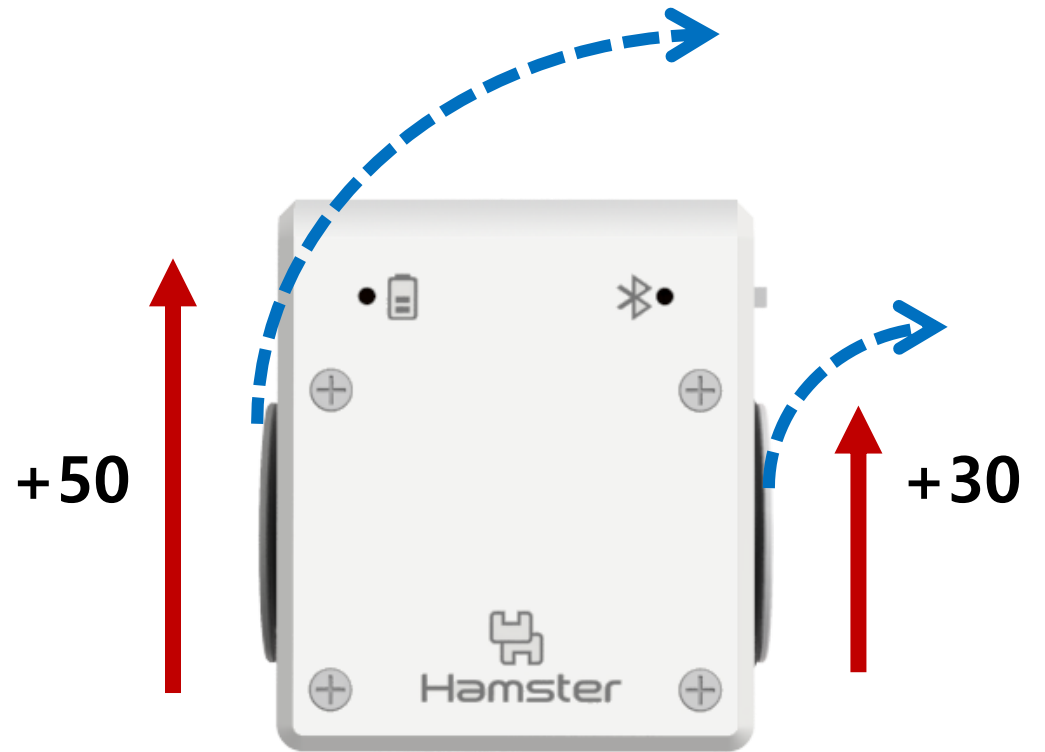
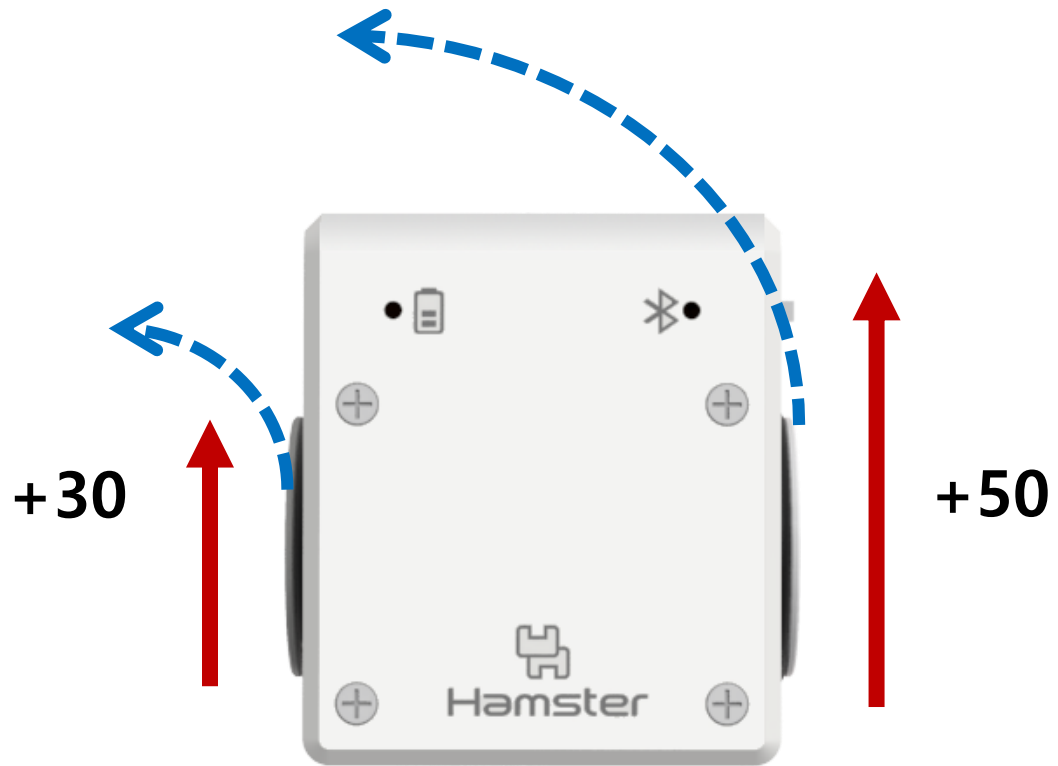
마이크로 로봇 모터 동작

모터 센서



마이크로 로봇 모터 동작

모터 센서



마이크로 로봇 모터 동작

모터 센서 - 과제 1

wheels()를 사용하여 햄스터 로봇을 움직여 보자

```
9  from roboid import *
10
11  hamster = Hamster()
12
13
14  # 1. 직진하기
15  hamster.wheels(50, 50)
16  wait(2000)
17
18  # 2. 뒤로가기
19  hamster.wheels(-50, -50)
20  wait(2000)
```

```
23  # 3. 오른쪽 회전하기
24  hamster.wheels(50, -50)
25  wait(2000)
26
27
28  # 4. 왼쪽 회전하기
29  hamster.wheels(-50, 50)
30  wait(2000)
31
32
33  # 실행 종료 후 멈추기
34  hamster.stop()
```

오른쪽, 왼쪽

- 입력 값은 양의 정수는 전진, 음의 정수는 후진, 0은 정지 상태
- 입력 값의 절댓값은 회전의 빠르기

마이크로 로봇 모터 동작

모터 센서 - 과제 2

각 방향으로 움직임이 정이 된 함수를 이용하여 햄스터 로봇을 움직여 보자

```
8  from roboid import *
9
10 hamster = Hamster()
11
12
13 # 1. 직진하기
14 hamster.move_forward(2, 50)
15
16
17 # 2. 뒤로가기
18 hamster.move_backward(2, 60)
19
20
21 # 3. 오른쪽 회전하기
22 hamster.turn_right(2, 60)
23
24
25 # 4. 왼쪽 회전하기
26 hamster.turn_left(2, 60)
27
28
29
30 # 실행 종료 후 멈추기
31 hamster.stop()
32
```

- 첫번째 매개변수는 지속시간, 두번째 매개변수는 이동 속도이다.
- 세밀한 속도 컨트롤은 어렵다.

마이크로 로봇 센서 동작

LED 센서

LED 색상	숫자	설명
Hamster.LED_OFF	0	전원 OFF
Hamster.LED_BLUE	1	파란색
Hamster.LED_GREEN	2	초록색
Hamster.LED_CYAN	3	하늘색
Hamster.LED_RED	4	빨간색
Hamster.LED_MAGENTA	5	보라색
Hamster.LED_YELLOW	6	노란색
Hamster.LED_WHITE	7	하얀색

마이크로 로봇 센서 동작

LED 센서 – 과제 1

햄스터 로봇의 좌우 LED를 켜보자

```
7  from roboid import *
8
9  hamster = Hamster()
10
11
12  # 1. 왼쪽 LED 켜기
13  hamster.left_led("blue")
14  wait(500)
15
16  hamster.left_led("off")
17  wait(500)
18
```

```
20  # 2. 오른쪽 LED 켜기
21  hamster.right_led("yellow")
22  wait(500)
23
24  hamster.right_led("off")
25  wait(500)
26
27
28  # 3. 양쪽 LED 모두 켜기
29  hamster.leds("red", "green")
30  wait(500)
31
32
33  hamster.leds("off")
```

마이크로 로봇 센서 동작

LED 센서 – 과제 2

로봇의 좌우 LED의 색과 모터를 움직여 이동시켜보자.

```
7 from roboid import *
8
9 hamster = Hamster()
10
11 hamster.wheels(50)
12 hamster.left_led(Hamster.LED_BLUE)
13 wait(500)
14 hamster.right_led(Hamster.LED_GREEN)
15 wait(500)
16
17 hamster.wheels(-50, -50)
18 hamster.left_led(Hamster.LED_CYAN)
19 wait(500)
20 hamster.right_led(Hamster.LED_RED)
21 wait(500)
22
```

```
23 hamster.wheels(-50, 50)
24 hamster.left_led(Hamster.LED_MAGENTA)
25 wait(500)
26 hamster.right_led(Hamster.LED_YELLOW)
27 wait(500)
28
29 hamster.wheels(50, -50)
30 hamster.leds(Hamster.LED_WHITE, Hamster.LED_WHITE)
31 wait(500)
32
33 # 실행 종료 후 멈추기
34 hamster.stop()
```

마이크로 로봇 센서 동작

피에조 센서

- 햄스터 로봇의 부저 소리 범위는 0~167771.5 Hz까지 출력

나이	최고 가청 주파수	나이	최고 가청 주파수
보청기 필요	8000 Hz	20대 중반	15800 Hz
50대 이후	10000 Hz	20대 초반	16700 Hz
40대 정도	12000 Hz	10대 후반	17700 Hz
30대 정도	14100 Hz	10대 초반	18800 Hz
20대 후반	14900 Hz	10대 이하	19900 Hz

마이크로 로봇 센서 동작

피에조 센서 - 과제 1

버저의 값을 바꿔 자신의 가청 주파수는 얼마인지 측정해보자.

```
9  from roboid import *
10
11  hamster = Hamster()
12
13  # 숫자는 Hz의 값 #hamster.buzzer(16500)
14  hamster.buzzer(134.42)
15
16
17  # 실행 종료 후 멈추기
18  hamster.stop()
19
```

마이크로 로봇 센서 동작

피에조 센서 - 과제 2

0 옥타브부터 8 옥타브 도까지 소리 내보자

```
10 from roboid import *
11
12 hamster = Hamster()
13
14 hamster.note(Hamster.NOTE_C_1)
15 wait(1000)
16
17 hamster.note(Hamster.NOTE_C_2)
18 wait(1000)
19
20 hamster.note(Hamster.NOTE_C_3)
21 wait(1000)
22
23 hamster.note(Hamster.NOTE_C_4)
24 wait(1000)
25
```

```
26 hamster.note(Hamster.NOTE_C_5)
27 wait(1000)
28
29 hamster.note(Hamster.NOTE_C_6)
30 wait(1000)
31
32 hamster.note(Hamster.NOTE_C_7)
33 wait(1000)
34
35 # 실행 종료 후 멈추기
36 hamster.stop()
```

마이크로 로봇 센서 동작

적외선 센서 - 과제 1

```
2 from roboid import *
3
4 hamster = Hamster()
5
6 buzzer_hz = 0
7 while True:
8     #proximity = hamster.left_proximity()
9     proximity = hamster.right_proximity()
10
11     if proximity < 10:
12         proximity = 0
13     buzzer_hz = (buzzer_hz * 5 + proximity * 70) / 10.0
14     hamster.buzzer(buzzer_hz)
15
16     wait(20)
```

실습01. 가속 모터

게임에서 부스터를 이용하여 빠르게 달린다. 이를 코드로 구현해 보면?

```
7  from roboid import *
8
9  hamster = Hamster()
10
11
12  for i in range(40, 100):
13      hamster.wheels(i, i)
14      wait(30)
15
16
17  hamster.stop()
18
```

실습01. 감속 모터

게임에서 브레이크를 밟아 급속도로 속도를 줄여야 한다. 이를 코드로 구현해 보면?

```
8  from roboid import *
9
10 hamster = Hamster()
11
12
13 for i in range(100, 0, -1):
14     hamster.wheels(i, i)
15     wait(30)
16
17
18
19 hamster.stop()
```

실습02. 피아노 연주

피에조 센서 이용하여 젓가락 송 노래를 연주를 해보자!

무엇이 무엇이 똑같은까~~

젓가락 두 짝이 똑같지요~~



실습02. 피아노 건반

피에조 센서 이용하여 피아노 건반을 만들어보자



```
1 from roboid import *
2
3 hamster = Hamster()
4
5 while True:
6     key = input()
7     if key:
8         if key == 'a':
9             hamster.note(Hamster.NOTE_C_4) # 도
10        elif key == 's':
11            hamster.note(Hamster.NOTE_D_4) # 레
12        elif key == 'd':
13            hamster.note(Hamster.NOTE_E_4) # 미
14        elif key == 'f':
15            hamster.note(Hamster.NOTE_F_4) # 파
16        elif key == 'g':
17            hamster.note(Hamster.NOTE_G_4) # 솔
18        elif key == 'h':
19            hamster.note(Hamster.NOTE_A_4) # 라
20        elif key == 'j':
21            hamster.note(Hamster.NOTE_B_4) # 시
22        elif key == 'k':
23            hamster.note(Hamster.NOTE_C_5) # 도
24        elif key == 'l':
25            hamster.note(Hamster.NOTE_D_5) # 레
26        elif key == ';':
27            hamster.note(Hamster.NOTE_E_5) # 미
28        elif key == '"':
29            hamster.note(Hamster.NOTE_F_5) # 파
30        elif key == ' ': # 스페이스 키
31            hamster.note(0)
32
33 wait(20)
```

실습03. 감속 모터

키보드 이벤트를 받아 가속 모터를 발생 시켜보자.

```
8  from roboid import *
9  import keyboard
10
11 hamster = Hamster()
12 count = 0
13 while True:
14     if keyboard.is_pressed("a"):
15
16         if count < 200:
17             count += 5
18
19             hamster.wheels(count, count)
20
21             wait(30)
22     else:
23         count = 0
24         hamster.wheels(0, 0)
```


프로젝트 01. 엉터리 씨름 대회

햄스터 로봇을 이용하여 엉터리 씨름 대회를 진행하고자 한다.

엉터리 씨름 대회는 **키보드 방향키를 이용하여 가속하는 햄스터로봇으로 상대팀을 밀치는 씨름이다.**

4인 1조가 되어 총 3팀으로 나눠져 게임을 한다.

게임은 토너먼트 형식으로 진행되며 뽑기를 통하여 부전승을 결정할 것이다.



프로젝트 01. 엉터리 씨름 대회

```
8 from roboid import *
9 import keyboard
```

```
10
11 hamster = Hamster()
12 count = 0
13 is_pressed_key = ""
```

```
14
15 while True:
```

```
16     if keyboard.is_pressed("w"):
```

```
17
```

```
18
```

```
19
```

```
20         count += 10
```

```
21         hamster.wheels(count, count)
```

```
22         is_pressed_key = "w"
```

```
23         wait(30)
```

```
24
```

```
25     elif keyboard.is_pressed("a"):
```

```
26         if is_pressed_key != "a":
```

```
27             count = 0
```

```
28         if count < 100:
```

```
29             count += 3
```

```
30         hamster.wheels(-count, count)
```

```
31         is_pressed_key = "a"
```

```
32         wait(30)
```

```
34     elif keyboard.is_pressed("d"):
```

```
35         if is_pressed_key != "d":
```

```
36             count = 0
```

```
37         if count < 100:
```

```
38             count += 3
```

```
39         hamster.wheels(count, -count)
```

```
40         is_pressed_key = "d"
```

```
41         wait(30)
```

```
42
```

팀 이름을 먼저 정해주세요!

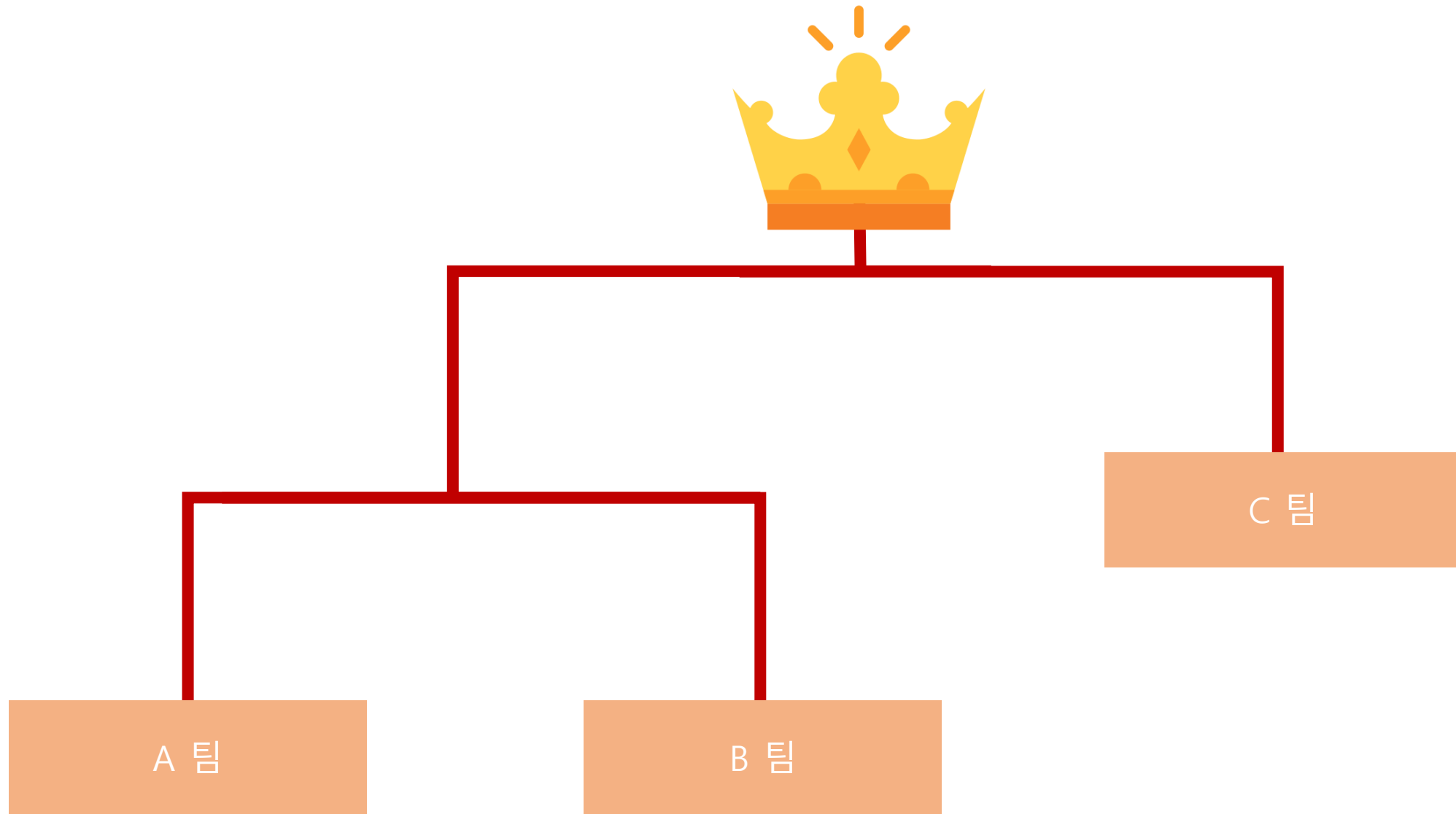
```
46
```

프로젝트 01. 엉터리 씨름 대회

```
8 from roboid import *
9 import keyboard
10
11 hamster = Hamster()
12 count = 0
13 is_pressed_key = ""
14
15 while True:
16     if keyboard.is_pressed("w"):
17         if is_pressed_key != "w":
18             count = 0
19             if count < 100:
20                 count += 10
21                 hamster.wheels(count, count)
22                 is_pressed_key = "w"
23                 wait(30)
24
25     elif keyboard.is_pressed("a"):
26         if is_pressed_key != "a":
27             count = 0
28             if count < 100:
29                 count += 3
30                 hamster.wheels(-count, count)
31                 is_pressed_key = "a"
32                 wait(30)
```

```
34 elif keyboard.is_pressed("d"):
35     if is_pressed_key != "d":
36         count = 0
37         if count < 100:
38             count += 3
39             hamster.wheels(count, -count)
40             is_pressed_key = "d"
41             wait(30)
42
43 else:
44     count = 0
45     hamster.wheels(0, 0)
46
```

프로젝트 01. 엉터리 씨름 대회



프로젝트 02. 햄스터 로봇 축구 대회

햄스터 로봇을 이용하여 로봇 축구 대회를 진행하고자 한다.

4인 1조가 되어 총 3팀으로 나뉘져 게임을 한다.

먼저 자신의 햄스터 로봇이 강력해질 수 있게 **마분지로 다양한 모양으로 변신**이 가능하다.

게임은 총 3번 앞서 경기한 엉터리 씨름 대회 우승팀이 부전승으로 올라간다.

경기 제한 시간은 **전반전 4분, 쉬는시간 2분, 후반전 4분으로** 진행된다.

동점일 경우 승부차기(각 2번의 기회)!

단 연결이 끊기거나 배터리가 없어 로봇이 움직이지 않는 경우 퇴장!

최종 우승한 팀에게는 특별한 **선물**이 있다.



프로젝트 02. 햄스터 로봇 축구 대회

