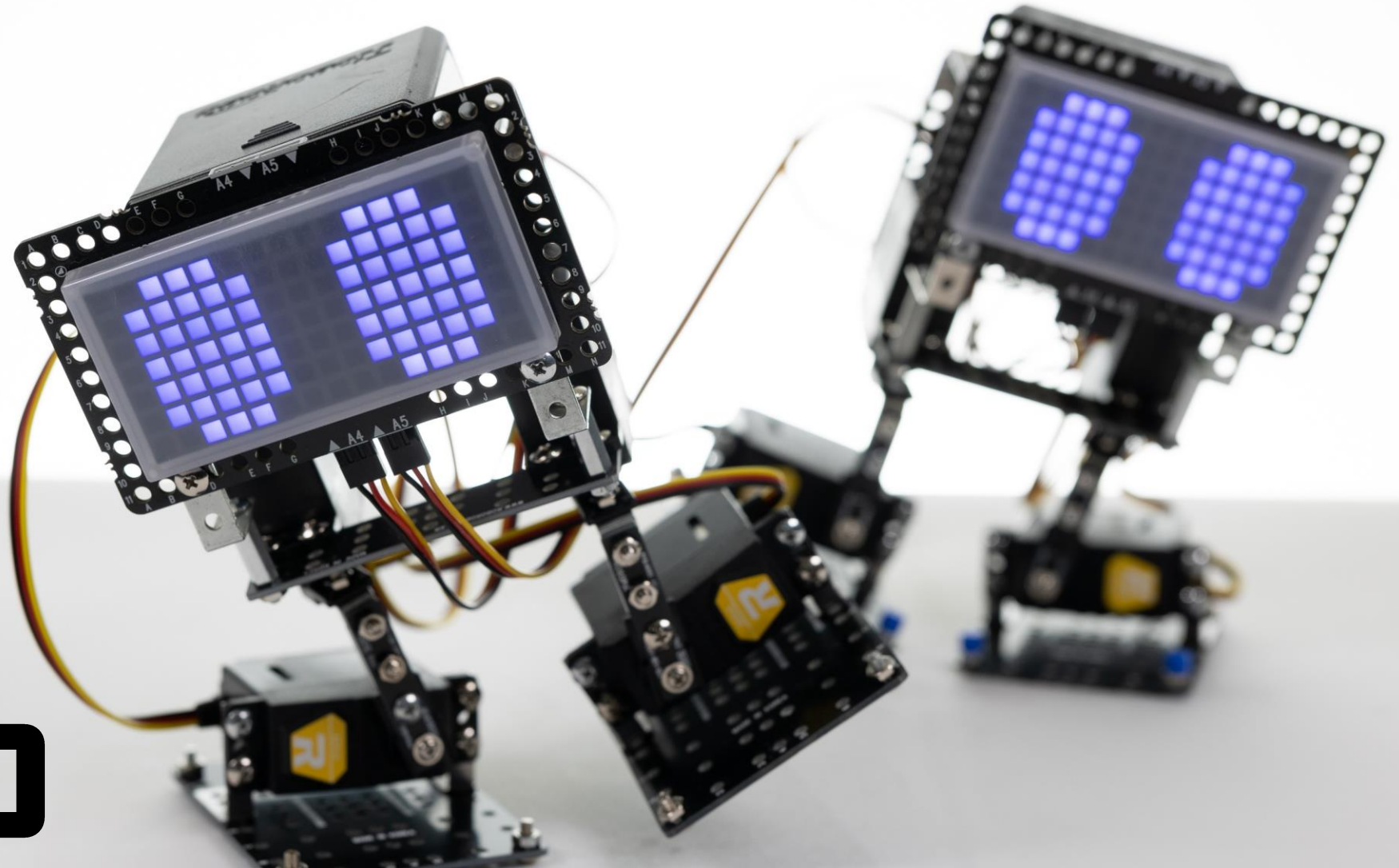


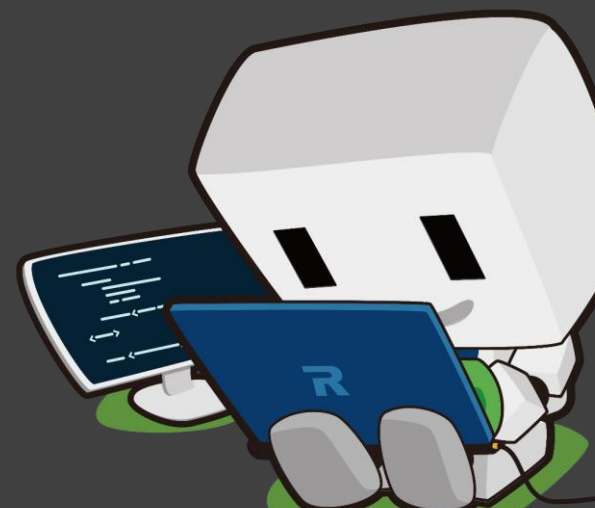
# AI 이족보행 로봇 [Walker]

Hi



DIYGO

# Step. 10 Python Coding





# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

해당 차시는..

Python으로 이족보행 로봇 [워커]의 **서보모터**를 제어하는 간단한 체험형 실습을 진행하도록 구성되어 있습니다. (Python문법 등에 대한 보강자료 없음)

Python과 Visual Studio Code, 로봇 제어 라이브러리를 활용한 실습을 진행할 예정이며, 기본 명령어를 따라서 타이핑 해 보며 [워커]를 제어하는 과정을 통해 Python을 비롯한 텍스트 코딩 시스템을 살펴보는 체험실습을 진행해 보시기 바랍니다.

**Python 및 Visual Studio Code의 업데이트 상황에 따라 표기되는 명칭 및 이미지가 다를 수 있습니다.**



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Python 설치



웹브라우저에 '파이썬'을 검색하나

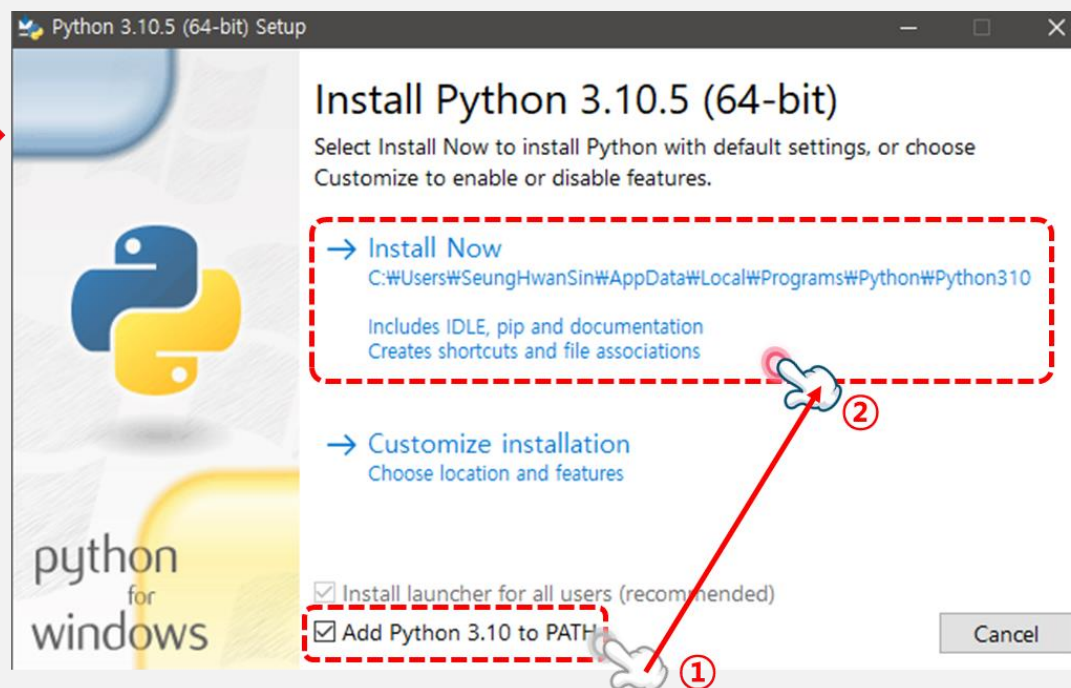
https://www.python.org 후 사이트로 이동해 최신버전의 파일을 다운로드



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Python 설치



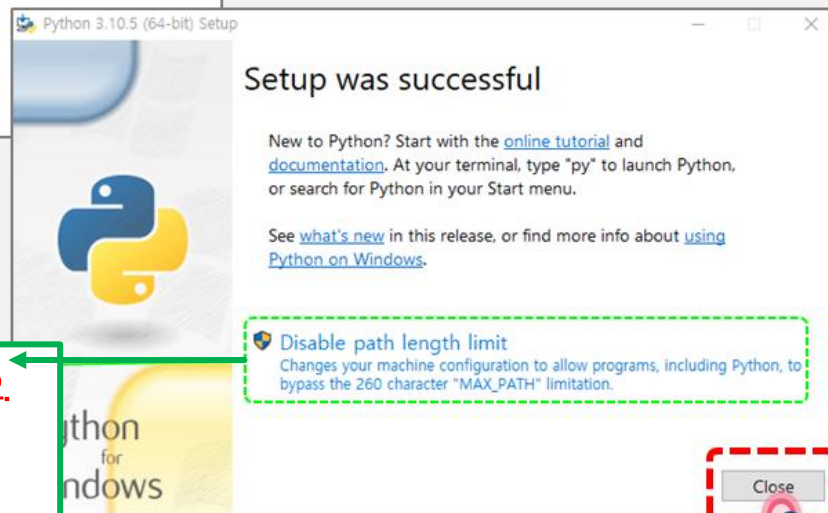
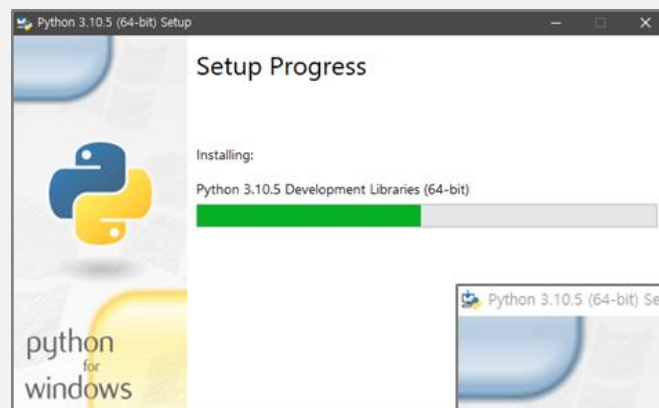
“Add python 3.10 to Path” 옵션은 파이썬이 어느 곳에서든 실행될 수 있는 기능이에요(체크)



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Python 설치



경로에 대한 최대길이를 해제하는 내용이에요.  
경로가 260자를 넘지 않으면 꼭 필요한  
부분은 아니에요.

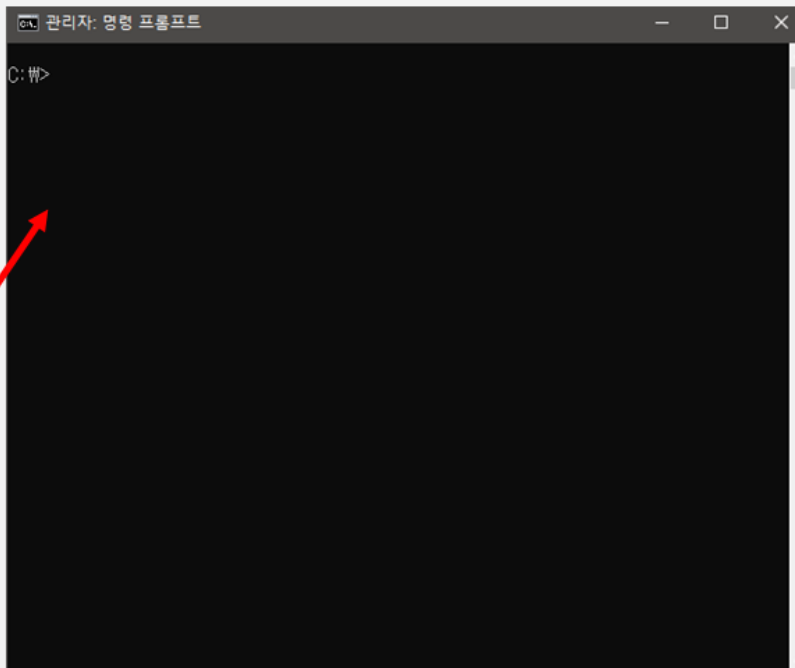




# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

pip 설치 (pip는 Python 패키지를 설치하고 관리하는 패키지 매니저입니다.)



‘윈도우키’를 누르고, cmd를 입력한 뒤 명령 프롬프트를 ‘관리자 권한으로 실행’ 해 주세요.



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

pip 설치 (pip는 Python 패키지를 설치하고 관리하는 패키지 매니저입니다.)

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

위 명령어를 복사 또는 타이핑한 뒤 엔터키를 누르면 인터넷 환경에서 pip를 내려 받아요.

(인터넷 연결 상태에서 진행해요.)

```
C:\> 관리자: 명령 프롬프트
C:\> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 2617k  100 2617k    0     0  8474k      0  --:--:-- --:--:-- --:--:-- 8498k
C:\> _
```

입력한 뒤 [엔터] 키를 입력하면  
인터넷환경에서  
pip를 다운받습니다.





# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

pip 설치 (pip는 Python 패키지를 설치하고 관리하는 패키지 매니저 입니다.)

```
python get-pip.py
```

위 명령어를 복사 또는 타이핑한 뒤 엔터키를 누르면 내려받은 pip를 설치해요.

```
C:\> 관리자: 명령 프롬프트

C:\> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload   Total   Spent    Left   Speed
100 2617k  100 2617k    0     0  8474k      0  --:--:-- --:--:-- --:--:-- 8498k

C:\> python get-pip.py
Collecting pip
  Using cached pip-22.1.2-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.1.2
    Uninstalling pip-22.1.2:
      Successfully uninstalled pip-22.1.2
  Successfully installed pip-22.1.2

C:\>
```

입력한 뒤 [엔터] 키를 입력하면

다운로드 받은 pip를 설치합니다.

설치 완료!



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Visual Studio Code 설치

The screenshot shows a Google search for 'Visual Studio Code'. The search bar contains the text 'Visual Studio Code'. Below the search bar, the search results show the official Visual Studio Code website. The first result is 'Visual Studio Code - Code Editing. Redefined' with the URL 'https://code.visualstudio.com'. Below the title, there is a 'Download' link. A red dashed box highlights the 'Download' link and the text 'Visual Studio Code is free and available on your favorite ...'. To the right of the search results, there is a preview of the Visual Studio Code download page. The preview shows the title 'Download Visual Studio Code' and the text 'Free and built on open source. Integrated Git, debugging and extensions.' Below this, there are three main sections for different operating systems: Windows, Linux, and Mac. Each section has a download button and a list of available download links. A red arrow points from the 'Download' link in the search results to the 'Windows' download button in the preview.

Google Visual Studio Code

전체 이미지 동영상 뉴스 도서

검색결과 약 475,000,000개 (0.44초)

<https://code.visualstudio.com>

**Visual Studio Code - Code Editing. Redefined**

Visual Studio Code is a code editor redefined and optimized for web and cloud applications. Visual Studio Code is free and ...

**Download**

Visual Studio Code is free and available on your favorite ...

**Download Visual Studio Code**

Free and built on open source. Integrated Git, debugging and extensions.

**Windows**  
Windows 8, 10, 11

User Installer 64 bit 32 bit ARM  
System Installer 64 bit 32 bit ARM  
.zip 64 bit 32 bit ARM

**.deb**  
Debian, Ubuntu

.deb 64 bit ARM ARM 64  
.rpm 64 bit ARM ARM 64  
.tar.gz 64 bit ARM ARM 64

**.rpm**  
Red Hat, Fedora, SUSE

**Mac**  
macOS 10.11+

.zip Universal Intel Chip Apple Silicon

Snap Store

웹브라우저에 'Visual Studio Code'를 검색한 후 사이트로 이동해 최신버전의 파일을 다운로드



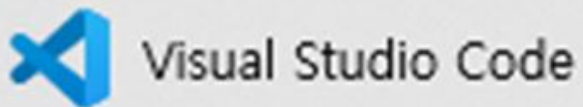
# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

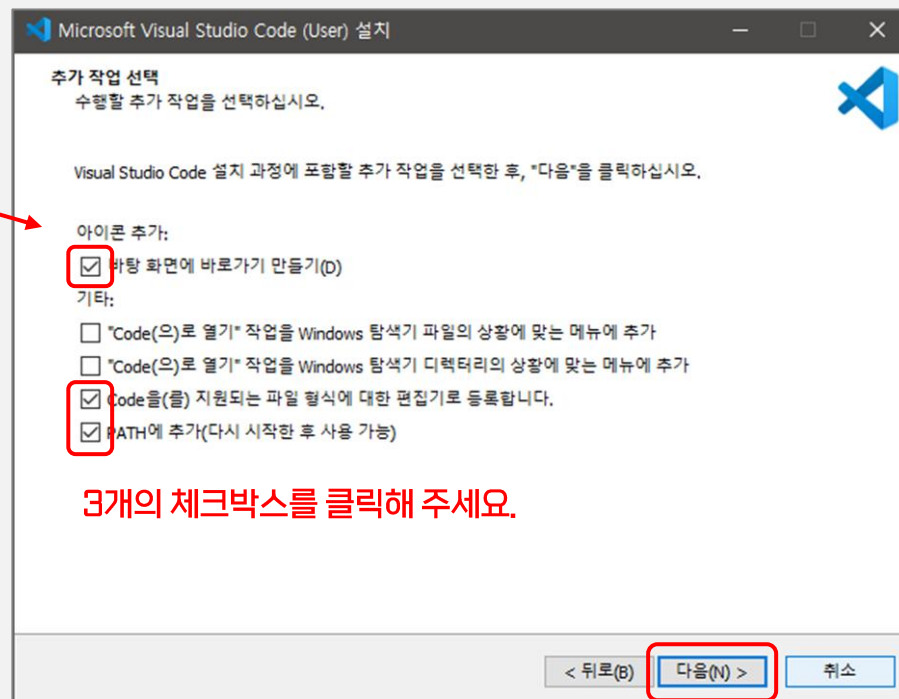
### Visual Studio Code 설치



최근에 추가한 앱



다운받은  
파일을 설치합니다.



3개의 체크박스를 클릭해 주세요.

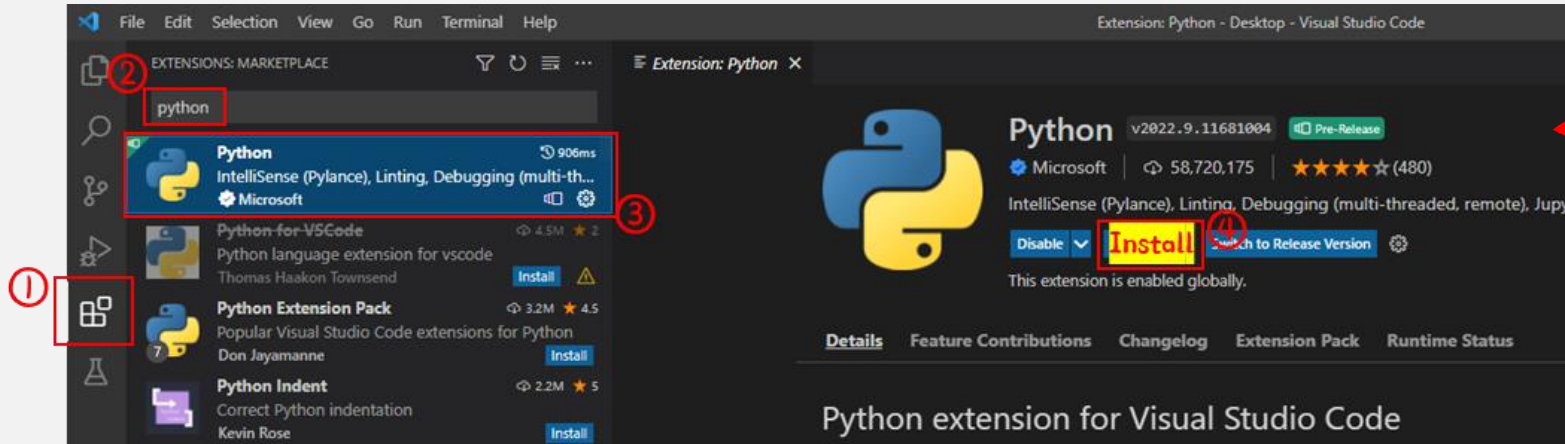
프로그램 설치



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Visual Studio Code 실행 - 파이썬 확장기능 설치



설치 후 실행

① extensions탭 클릭

② Python 검색

③ Python 확장프로그램 선택 (②~③의 과정을 통해 필요한 확장기능을 추가설치 가능)

④ Install 버튼을 클릭하여 확장프로그램 설치

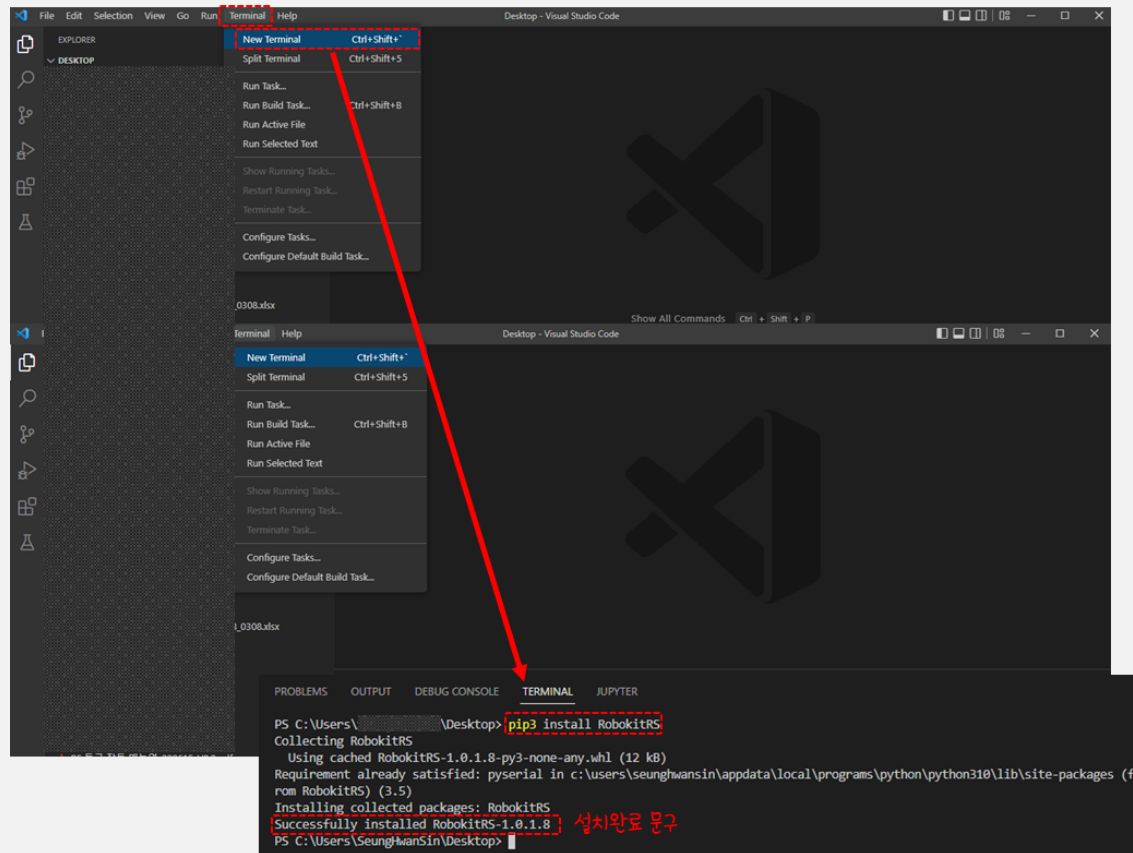
의 순서를 통해 Python 확장기능을 설치 합니다.



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Visual Studio Code 실행 - RobokitRS 라이브러리 설치 (인터넷 연결필요)



상단 메뉴바의

Terminal -> New Terminal 선택 후

아래 명령어를 통해 RobokitRS 라이브러리 설치

```
pip3 install RobokitRS
```

위 명령어 입력 후 엔터



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Visual Studio Code 실행 - 작업환경 설정

File -> Open Folder.. 를 선택하여  
작업 폴더를 지정합니다.

Yes 버튼 클릭하여 폴더지정 완료

작업 폴더는 본인이 사용하기 편한 경로를 설정해 줍니다.

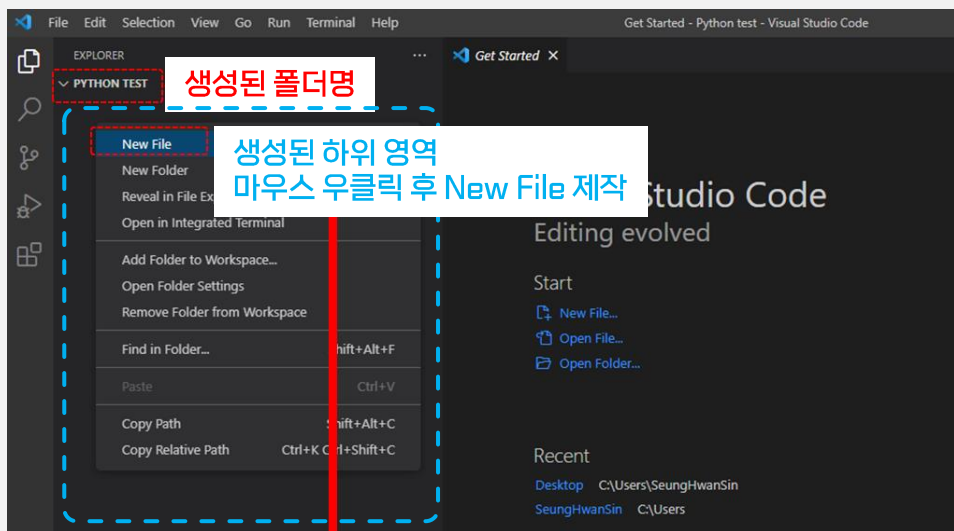




# Python으로 [워크] 제어하기

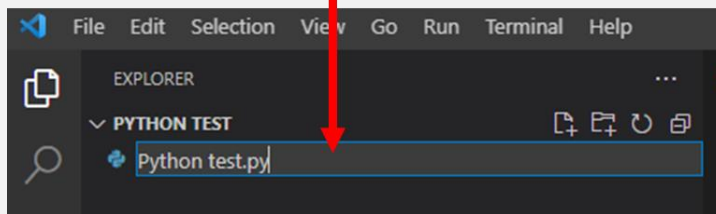
## Visual Studio Code를 활용한 Python 구문 실습

### Visual Studio Code 실행 - 작업환경 설정



생성된 폴더명

생성된 하위 영역  
마우스 우클릭 후 New File 제작



파일명.py  
형식으로 새 파일 만들기

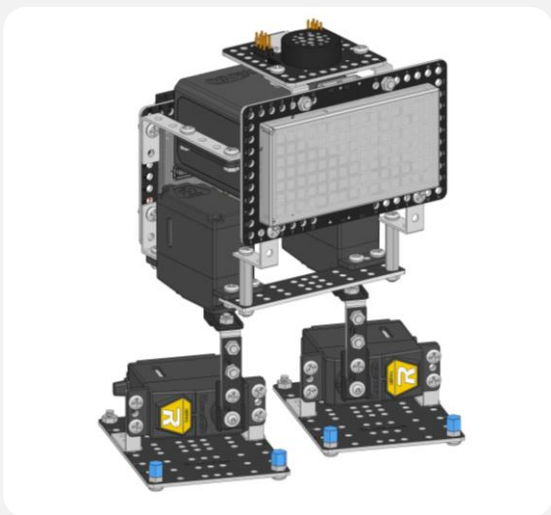
파일명 역시 자유롭게 기입 후  
.py를 붙여서 생성하면 완료



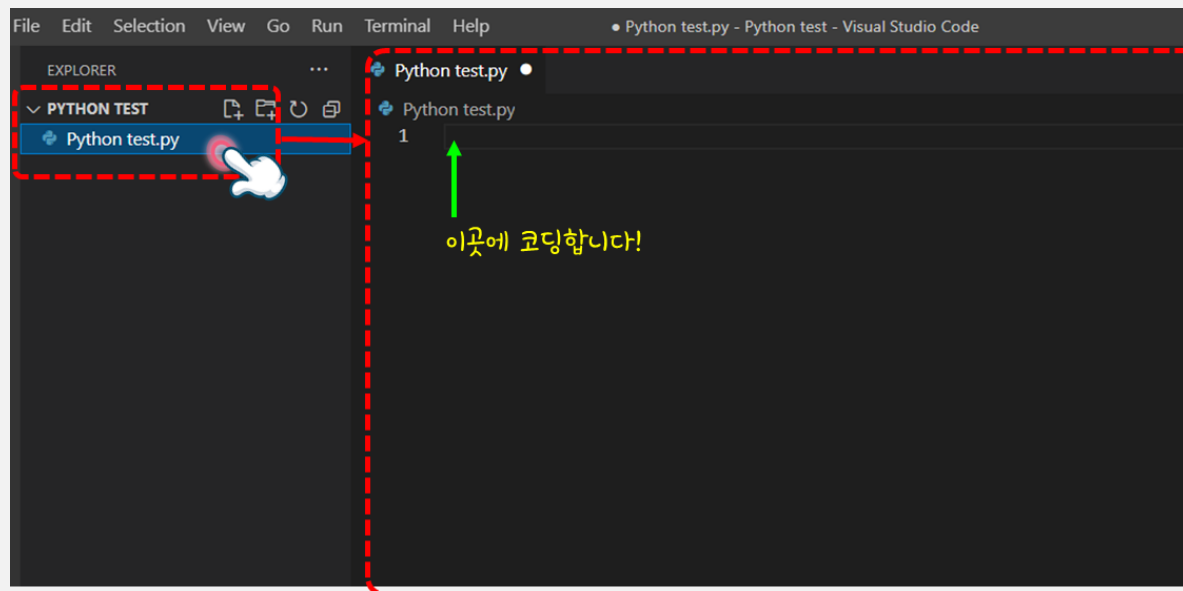
# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기



이족보행 로봇 [워커]를  
준비하세요.



로봇의 CPU보드와 컴보드에 블루투스 모듈을 꽂고, 전원을 켜세요.



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - 기본 세팅

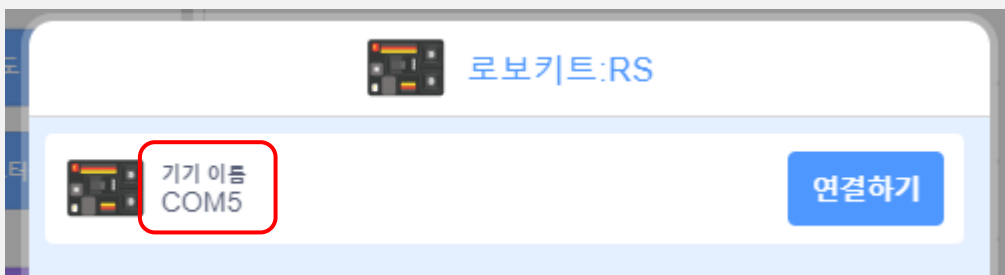
```
Python test.py •  
Python test.py > ...  
1 from RobokitRS import *  
2 rs = RobokitRS.RobokitRS()  
3 rs.port_open("com5")
```

RobokitRS 라이브러리의 모든 항목을 가져옴

RobokitRS 라이브러리로 'rs'라는 이름을 가진 새로운 객체를 생성

지정한 Com포트 번호(5번)를 열어 통신함.

(Com포트 번호의 경우 사용자 환경에 따라 다르며, Rogic에서 사용하는 번호와 동일해요)



#### 라이브러리란?

쉽게 말해 [타인이 만들어 놓은 프로그램 덩어리] 를 의미합니다.

RobokitRS 라이브러리는 회로모듈을 제어할 수 있는 명령어들이 포함된 덩어리 입니다.



# Python으로 [위커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

Rogic을 통해 보행로봇 [위커]를 제어하던 명령어 블록과 동일한 역할을 수행할 수 있는 Python 명령어를 알아보까요?

The diagram illustrates the correspondence between visual programming blocks and Python code for robot control:

- Blue block:** 2 번 핀 서보모터를 0 도로 회전하기 (Turn servo motor on pin 2 to 0 degrees) is linked to the Python snippet: 

```
for i in range(50):  
    rs.servo_write(2,i)  
    rs.delay(0.1)
```
- Orange block:** 1 초 기다리기 (Wait 1 second) is linked to the Python snippet: 

```
while True:  
    rs.servo_write(2,90)  
    rs.delay(3)  
    rs.servo_write(2,90)  
    rs.delay(3)
```
- Orange block:** 10 번 반복하기 (Repeat 10 times) is linked to the Python snippet: 

```
while True:  
    rs.servo_write(2,90)  
    rs.delay(3)  
    rs.servo_write(2,90)  
    rs.delay(3)
```
- Orange block:** 무한 반복하기 (Repeat forever) is linked to the Python snippet: 

```
while True:  
    rs.servo_write(2,90)  
    rs.delay(3)  
    rs.servo_write(2,90)  
    rs.delay(3)
```



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

#### 서보모터 제어 명령어

`servo_write(pin:int, position:int)`

:int의 의미는? 정수를 입력할 수 있다는 뜻이예요.

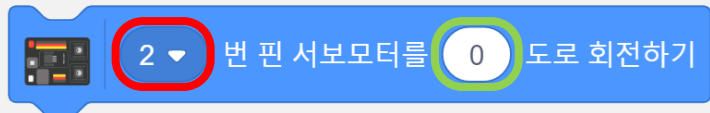
- 서보모터를 제어하는 명령어 함수입니다. `pin`(포트번호)에 연결된 서보모터를 `position`(각도) 위치로 이동시킵니다.

\* `pin`

-> min : 2, max : 13 (CPU보드의 디지털 포트번호)

\* `position`

-> min : -120, max : 120 (서보모터로 표현 가능한 각도 값)



`servo_write(2, 0)`

위와 같이 2번 포트의 서보모터를 0도로 지정할 수 있어요.



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

#### 기다리기 명령어

`delay(waitingtime:float)`

:float의 의미는? 실수(소수)를 입력할 수 있다는 뜻이에요.

- `waitingtime` 동안 CPU의 동작을 대기시킬 수 있습니다.

\* waitingtime -> 단위 : 초, 기본값 : 1 (기본값은 세팅하지 않을 때를 말합니다.)



위와 같이 Rologic의 기다리기 블록 기능을 Python 명령어로 표현할 수 있어요.



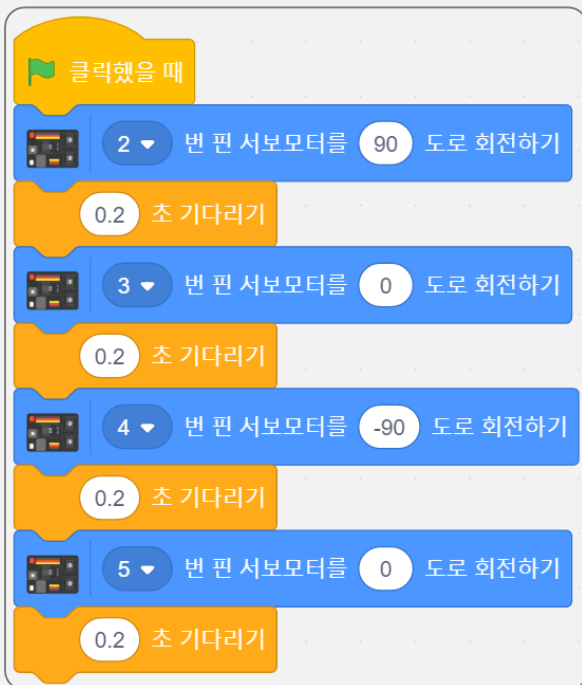


# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python으로 서보모터 제어하기

Rogic에서 코딩하던, [워커]의 기본자세 코딩을 파이썬으로 다음과 같이 나타낼 수 있어요.



```
1  from RobokitRS import *
2  rs = RobokitRS.RobokitRS()
3  rs.port_open("com5")
4  rs.servo_write(2,90) → 2번 서보모터 90도로 설정
5  rs.delay(0.2) → 0.2초 기다리기
6  rs.servo_write(3,0) → 3번 서보모터 0도로 설정
7  rs.delay(0.2) → 0.2초 기다리기
8  rs.servo_write(4,-90) → 4번 서보모터 -90도로 설정
9  rs.delay(0.2) → 0.2초 기다리기
10 rs.servo_write(5,0) → 5번 서보모터 0도로 설정
11 rs.delay(0.2) → 0.2초 기다리기
```



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python으로 서보모터 제어하기

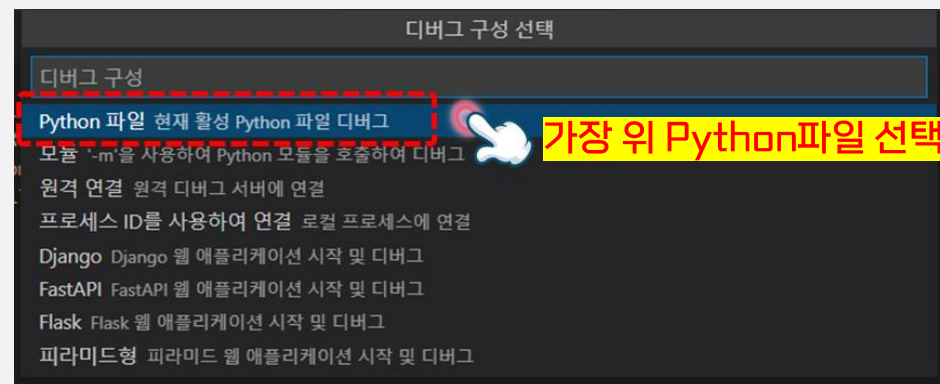
코딩 후 키보드의 F5 키를 누르면, 로봇으로 동작을 확인할 수 있어요 (로봇전원 on 확인)

```
1  from RobokitRS import *
2  rs = RobokitRS.RobokitRS()
3  rs.port_open("com5")
4  rs.servo_write(2,90)
5  rs.delay(0.2)
6  rs.servo_write(3,0)
7  rs.delay(0.2)
8  rs.servo_write(4,-90)
9  rs.delay(0.2)
10 rs.servo_write(5,0)
11 rs.delay(0.2)
```

코딩 후

F5 버튼

누르기



F5 버튼(실행 단축키)을 누르면

최초 1회에 한해 위처럼 선택 문구가 나타나요.

실행단축키

F5 -> 실행

Shift + F5 -> 실행중지



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python으로 서보모터 제어하기

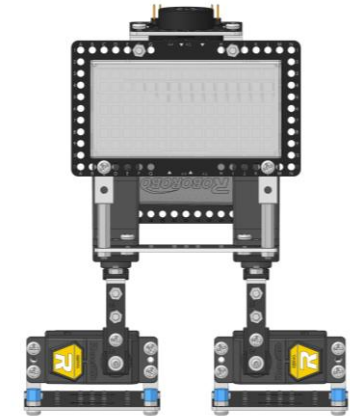
코딩 후 키보드의 F5 키를 누르면, 로봇으로 동작을 확인할 수 있어요 (로봇전원 on 확인)

```
Python test.py > ...
1  from RobokitRS import *
2  rs = RobokitRS.RobokitRS()
3  rs.port_open("com5")
4  rs.servo_write(2,90)
5  rs.delay(0.2)
6  rs.servo_write(3,0)
7  rs.delay(0.2)
8  rs.servo_write(4,-90)
9  rs.delay(0.2)
10 rs.servo_write(5,0)
11 rs.delay(0.2)
```

문제 출력 디버그 콘솔 **터미널** JUPYTER

```
C:\Users\...\AppData\Local\Programs\Python\Python310\lib\site-packages\RobokitRS\pins
RSBoard serial port open
Thread init done
Pin mode init done
Sysex init done
RSBoard init done
```

1. 진행상태를 알려주는 메시지 출력 후



2. 기본자세를 취해요.



# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

동작 실행 중 에러문구가 등장한다면?

```
Python test.py > ...
1  from RobokitRS import *
2  rs = RobokitRS.RobokitRS()
3  rs.port_open("com5")
```

**Exception has occurred: SerialException ×**  
could not open port 'COM5': PermissionError(13, '액세스가 거부되었습니다.', None, 5)  
File "C:\Users\\_\_\_\_\_\Desktop\Python test\Python test.py", line 3, in <module>  
rs.port\_open("com5")

장치의 연결상태를 다시 확인해주어야 해요.

- 블루투스 모듈이 제대로 연결되어 있는지, CPU에 전원이 켜져 있는지 확인하세요.
- USB포트에 연결된 컴보드를 뽑았다가 다시 한번 연결해 준 뒤 재실행 시켜 보세요.



# Python으로 [워커] 제어하기

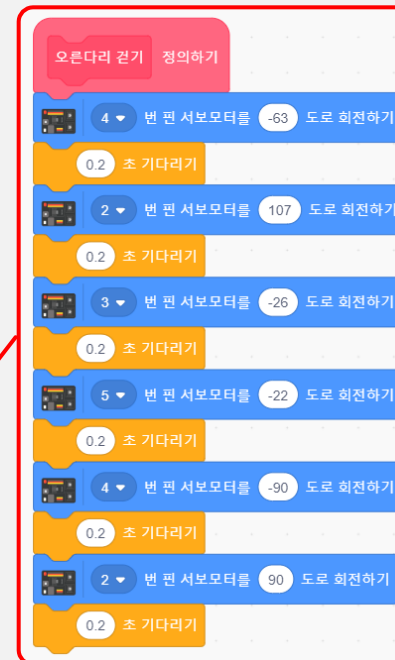
## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python으로 [워커]걷기 동작 만들기

Rogic에서 코딩했던 내용을 그대로 명령어로 타이핑 해 볼 거예요.

- 클릭했을 때
- 기본자세
- 오른다리 걷기
- 왼다리 크게걷기
- 오른다리 크게걷기
- 왼발 걸어 기본자세

```
1 from RobokitRS import *
2 rs = RobokitRS.RobokitRS()
3 rs.port_open("com5")
4 rs.servo_write(2,90)
5 rs.delay(0.2)
6 rs.servo_write(3,0)
7 rs.delay(0.2)
8 rs.servo_write(4,-90)
9 rs.delay(0.2)
10 rs.servo_write(5,0)
11 rs.delay(2)
12
13 rs.servo_write(4,-63)
14 rs.delay(0.2)
15 rs.servo_write(2,107)
16 rs.delay(0.2)
17 rs.servo_write(3,-26)
18 rs.delay(0.2)
19 rs.servo_write(5,-22)
20 rs.delay(0.2)
21 rs.servo_write(4,-90)
22 rs.delay(0.2)
23 rs.servo_write(2,90)
```



서보모터 제어 명령어와  
기다리기 명령어를  
반복적으로 사용하면  
손쉽게 동작을 만들 수 있어요.

[왼다리 크게걷기]  
[오른다리 크게걷기]  
[왼발 걸어 기본자세]

동작도 코딩한 뒤 실행해 보세요.



# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

#### Python의 무한 반복 명령어

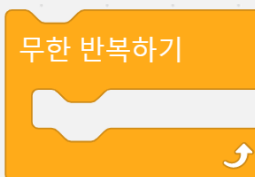
Python에서 사용하는 반복 구문은 `while(True):` 로 표현해요.

**while(True):**

반복할 명령어 (1)

반복할 명령어 (2)

=



위와 같이 반복하고자 하는 명령어들을 반복구문 아래 줄에 타이핑 하면 해당 내용들을 계속해서 반복하게 돼요.

이 때, 명령어 끝의 **:** 기호나, 하위 동작명령어 작성 시 **들여쓰기** 등이 오차 없이 작성되어야 제대로 동작할 수 있어요.



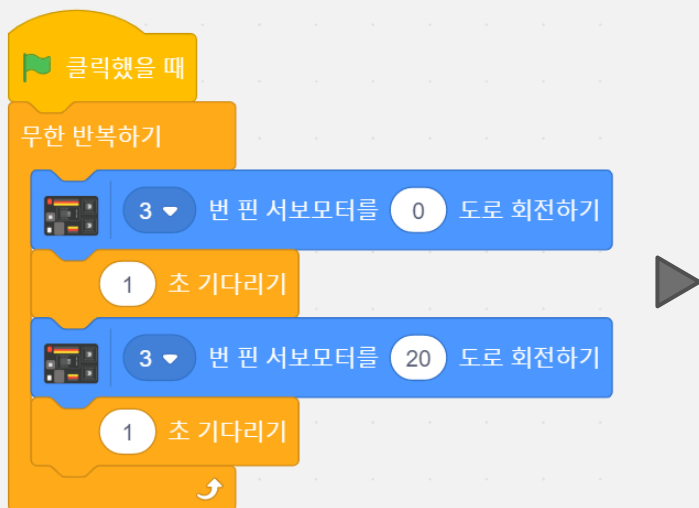


# Python으로 [워크] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

#### Python의 무한 반복 명령어



```
1 from RobokitRS import *
2 rs = RobokitRS.RobokitRS()
3 rs.port_open("com5")
4 while(True):
5     rs.servo_write(3,0)
6     rs.delay()
7     rs.servo_write(3,20)
8     rs.delay()
```

아래 명령어 무한반복

3번 서보모터 0도로 설정

1초 기다리기

3번 서보모터 20도로 설정

1초 기다리기

위와 같이 Rologic의 반복문을 Python으로 표현할 수 있어요.

(들여쓰기는 키보드 tab 키를 눌러서 표현할 수 있어요.)

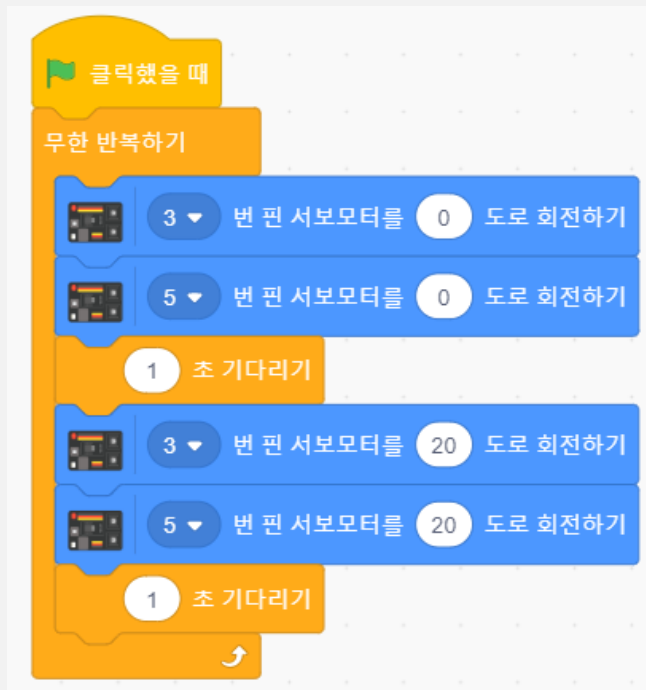


# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### 코딩하기 - Python 명령어 알기

#### Python의 무한 반복 명령어



```
1  from RobokitRS import *
2  rs = RobokitRS.RobokitRS()
3  rs.port_open("com5")
4
5
6
7
8
9
10
```

좌측의 Rologic 코드를 Python으로 작성하여 (10줄)  
[워커]의 양쪽 무릎모터를 동시에 동작시켜 보세요.

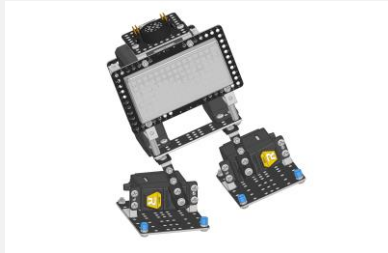


# Python으로 [워커] 제어하기

## Visual Studio Code를 활용한 Python 구문 실습

### Python으로 서보모터 제어하기

다양한 형태의 동작 만들기



[워커]로 표현했던 다양한 움직임들을  
Python코드로 직접 작성해서 동작시켜 보세요.