



## 운전 중 판단하기

우리는 걸을 때 항상 주변 환경을 감지합니다. 우리는 도중에 나타날 수도 있는 장애물을 감지할 수 있어야 합니다.

`forward()` 함수를 사용하는 동안에 주미는 다른 명령을 수행할 수가 없습니다.

이것은 `forward()` 함수를 사용하는 Zumi의 문제입니다.

`forward()` 를 사용하면 Zumi가 운전하는 동안 센서나 주변 환경을 확인할 수 없습니다.

눈을 감고 돌아다니고 다음 움직임을 결정하기 위해 움직임을 멈췄을 때 눈을 뜨는 것을 상상해 보세요.

이 수업에서는 Zumi가 멀티태스킹을 할 수 있도록 하는 새로운 함수를 사용하는 방법을 배웁니다. 주행중에 센서를 확인하며 판단을 내릴 수 있습니다.

## 라이브러리 포함

In [ ]:

```
from zumi.zumi import Zumi
import time
import IPython.display
zumi = Zumi()
```

## `forward()`

다음 코드를 실행하여 Zumi 앞에 손을 댍니다. 무슨 일이 일어나나요? Zumi가 멈추나요?

In [ ]:

```
zumi.forward(duration=5)
```

Zumi는 손으로 앞을 막아도 멈추지 않습니다.

Zumi가 정확한 거리를 이동할 수 있는 방법이 필요합니다!

## forward\_step()

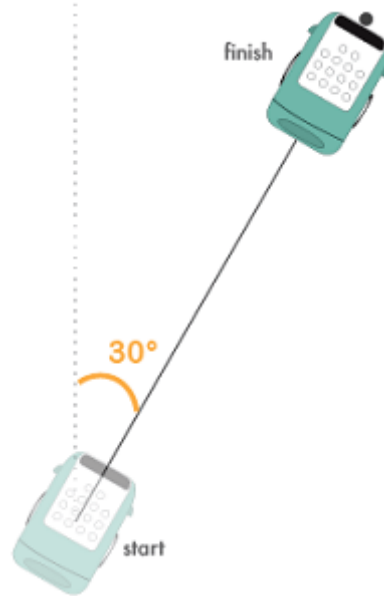
걸을 때 어떤 동작이 반복되고 있습니까? 오른발, 왼발, 오른발, 왼발... Zumi도 작은 걸음으로 한 스텝씩 앞으로 움직일 수 있습니다.

이러한 작은 단계가 반복문에 놓이면 Zumi가 앞으로 운전할 것입니다.

작은 걸음을 위한 구동 함수는 `zumi.forward_step()` 입니다.

`forward()` 와 달리 매개변수는 선택사항이 아닌 필수입니다. 이동할 속도와 **방향** 또는 방향이 필요 합니다.

아래 이미지에서 Zumi의 방향은 어디인가요?



`forward_step()` 을 사용하여 이 운전 패턴에 대한 코드를 보려면 아래 코드를 실행하세요.

In [ ]:

```
zumi.reset_gyro() # 주미의 자이로 센서 각도를 초기화합니다.  
  
for x in range(40): # 40번 반복합니다.  
    zumi.forward_step(40, -30) # -30 의 각도로 40의 속도로 이동합니다.  
  
zumi.stop() # 정지합니다.
```

프로그램 끝에 있는 `stop()` 은 정말 중요합니다!

`forward_step()` 을 사용하면 부드러운 주행이 가능하지만, 중간에 멈추지 않으므로 Zumi가 영원히(또는 배터리가 방전될 때까지) 운전하는 것을 막을 수 있는 방법은 없습니다.

`forward_step()` 을 사용하면 **반드시** `stop()` 과 함께 사용해야 합니다.

`for` 반복문이 완료된 후 코드는 다음 줄로 계속됩니다. `for` 반복문의 일부가 **아닌** 코드는 들여쓰지 않아야 합니다.

아래 셀에 `for` 반복문과 이 새로운 운전 명령을 연습 해보세요.

매개변수를 변경하고 반복하는 횟수에 따라 실행하는 데 걸리는 시간을 추정해 보세요.

In [ ]:

```
# 여기에 코드를 작성하세요.
```

`forward_step()`은 반복문에서만 작동하므로, 이 부분을 주의해야합니다.

아직은 센서나 조건문을 사용하지 않았기 때문에 이전의 주행과 큰 차이가 없을 지도 모릅니다.

---

## 파이썬 Tips: Try and Finally

반복문 및 `forward_step()`으로 작업할 때 Zumi의 주행(모터)이 끝없이 실행될 수 있는 문제가 발생할 수 있습니다.

이 잠재적인 문제를 해결하기 위해 `try` 및 `finally` 기능을 사용할 수 있습니다.

문제가 발생하거나 `try` 부분에서 코드를 중지하면 프로그램이 자동으로 `finally` 문으로 이동합니다.

In [ ]:

```
zumi.reset_gyro()
try:
    for x in range(30):
        zumi.forward_step(40, 0)
finally:
    zumi.stop()
```