



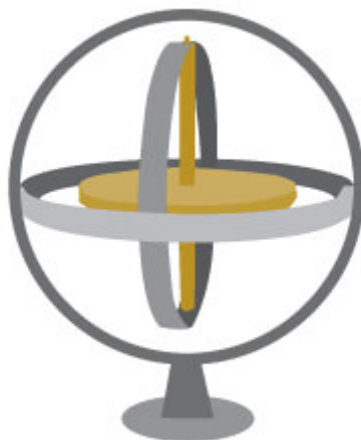
## 자이로스코프

주행 예제를 만들 때 Zumi는 지정한 각도를 사용하여 왼쪽 또는 오른쪽으로 회전했습니다. Zumi는 자신이 회전하고 있는 각도를 어떻게 알 수 있을까요? Zumi에는 **MPU** 또는 모션 처리 장치가 장착되어 있으며 직선 주행과 정확한 회전에 필요한 두 개의 매우 중요한 센서인 **자이로스코프**와 **가속도계**가 있습니다. 이 강의는 자이로스코프에 관한 것입니다.

## 자이로스코프란?

자이로스코프를 생각할 때 자이로스코프를 돌리거나 뒤집든, 방향에 관계없이 동일한 방향으로 균형을 유지하는 회전 바퀴를 생각할 수 있습니다. 어떻게? 이것은 운동량 보존으로 설명할 수 있습니다.

이전에 이 용어에 대해 들어본 적이 없을 수도 있지만 경험한 적이 있을 것입니다. 예를 들어, 바퀴가 달린 의자에서 회전한 적이 있다면 팔과 다리를 안으로 가져올 때 회전 속도가 더 빨라지는 것을 눈치 챘을 것입니다. 운동량은 보존되어야 하므로 질량(팔과 다리)을 더 가까이 가져갈 때 속도가 증가합니다.



전자 제품에서 자이로스코프는 이렇게 보이지 않습니다. 그들은 방향의 변화를 감지하기 위해 모션감지를 사용하는 작은 칩입니다. 이 움직임은 센서에 전달된 전류를 변경한 다음 디지털 값으로 변환합니다.

## 라이브러리 가져오기

In [ ]:

```
from zumi.zumi import Zumi
import time

zumi = Zumi()
```

## 자이로스코프 재설정

Zumi의 자이로스코프를 사용하기 위해. 다른 프로그램의 측정에 영향을 받지 않고 Zumi의 방향 변화를 정확하게 측정하려면 재설정해야 합니다. `reset_gyro()` 함수를 사용하면 롤, 피치 및 요에 대한 모든 각도가 다시 0으로 설정됩니다. `zumi=Zumi()` 를 실행하면 항상 이 작업이 수행되지만 나중에 프로그램에서 각도를 재설정해야 하는 경우 이 함수를 사용할 수 있습니다.

In [ ]:

```
zumi.reset_gyro()
```

## 리셋 드라이브

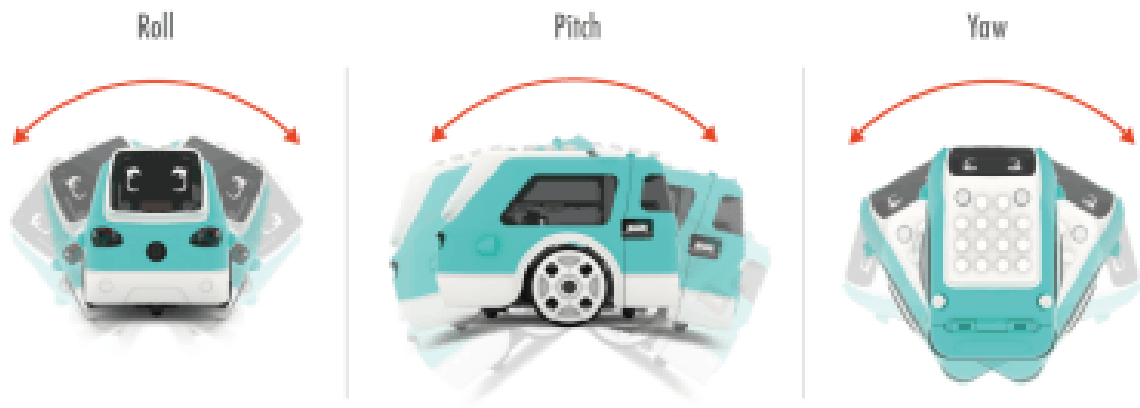
주행중에 사용하는 경우는 `reset_drive()` 함수를 사용합니다. `reset_gyro()`를 포함하여, 주행과 관련된 모든 파라미터 값이 초기화 됩니다.

In [ ]:

```
zumi.reset_drive()
```

## 측

회전 속도를 측정하는 데 사용할 수 있는 측이 두 개 이상 있습니다. 가장 관심을 가질 측은 요(yaw) 또는 좌우 회전을 측정하는 것입니다. Zumi가 앞뒤로 기울이고 있는지 또는 좌우로 기울이고 있는지도 측정할 수 있습니다. 이 세 측을 롤, 피치 및 요라고 합니다.



아래에 X, Y, Z의 세 축을 모두 읽는 세 개의 코드가 있습니다. 각 축을 실행하고 Zumi의 화면에서 각도가 어떻게 영향을 받는지 확인할 수 있습니다. Zumi를 이동하기 전에 평평한 바닥에 놓인 상태로 코드를 시작했는지 확인하세요.

## X 축 각도

In [ ]:

```
zumi.reset_gyro()
for i in range(0,30):
    x_angle = int(zumi.read_x_angle())
    print(x_angle)
    time.sleep(0.1)

print("완료")
```

## Y 축 각도

In [ ]:

```
zumi.reset_gyro()
for i in range(0,30):
    y_angle = int(zumi.read_y_angle())
    print(y_angle)
    time.sleep(0.1)

print("완료")
```

## Z 축 각도

```
In [ ]:
```

```
zumi.reset_gyro()
for i in range(0,30):
    z_angle = int(zumi.read_z_angle())
    print(z_angle)
    time.sleep(0.1)

print("완료")
```

데이터를 기반으로 X, Y, Z에 해당하는 방향을 알 수 있습니까?



운전할 때 Zumi 자이로스코프의 Z축 또는 요(yaw)에 대해 가장 신경을 쓰게 될 것입니다.

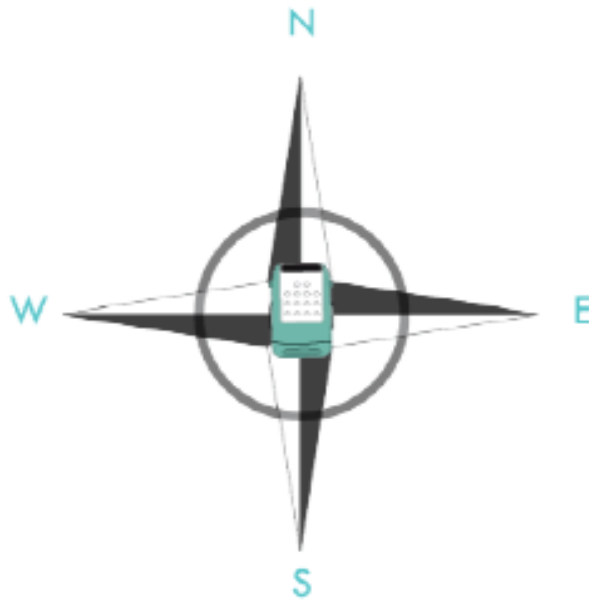
`zumi.read_z_angle()`을 호출하면 각도가 반환됩니다. 이후에는 프로그램에서 이것을 사용할 것입니다.

다음은 `read_z_angle()`을 사용하여 z축(요) 값을 가져오는 예입니다.

---

## 절대 각도

이전에서는 **상대** 각도를 사용하여 모양을 만들었습니다. 즉, 예를 들어 `zumi.turn_left(30)`를 호출하면 Zumi가 현재 보고 있는 방향에 따라 30도 회전합니다. 그러나 이것은 **heading** 또는 우리가 가리키는 방향을 추적하는 좋은 방법이 아닙니다. 길을 잃지 않기 위해 북쪽, 남쪽, 동쪽, 서쪽을 어떻게 사용하는지 생각해 보십시오.

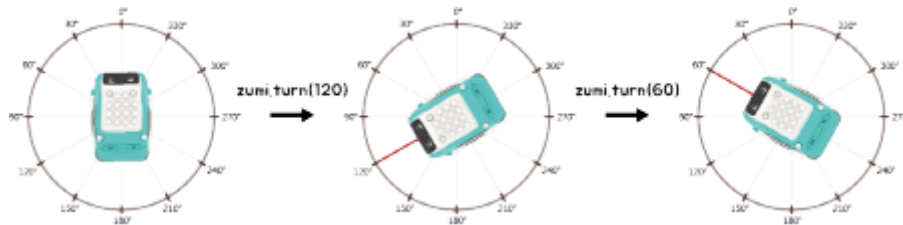


이전 섹션에서 Zumi 개체를 호출하거나 `reset_gyro()` 를 실행하면 Zumi가 모든 자이로스코프 값을 0으로 재설정한다는 것을 배웠습니다. 이 작업은 주미의 방향을 "북쪽"으로 재설정하는 것으로 생각하세요.

## zumi.turn()

`zumi.turn_left()` 및 `zumi.turn_right()` 에 대해 배웠지만 이제는 새로운 함수를 소개합니다.

절대각을 사용하려면 `zumi.turn()` 을 호출하세요. 작동 방식을 이해하려면 아래 그림을 참조하세요.



이 경우 0도는 북쪽, 90도는 서쪽, 180도는 남쪽, 270도는 동쪽입니다. 마찬가지로 -90도는 동쪽, -180도는 남쪽, -270도는 서쪽입니다. 양수 값을 사용하여 왼쪽으로 돌리고 음수 값을 사용하여 오른쪽으로 돌립니다.

## heading을 사용하는 사각형 그리기

이제 주변을 이동할 때 `turn()` 함수에 대한 새로운 지식을 가지고 나침반을 사용하는 것처럼 절대 방향으로 이동합니다.

다음과 같은 방식으로 Zumi에게 지시를 내리고 있다고 상상해 보십시오.

북쪽으로 1블록 이동  
 서쪽으로 1블록 이동  
 남쪽으로 1블록 이동  
 동쪽으로 1블록 이동

위의 이미지를 사용하면 도움이 됩니다.

In [ ]:

# 여기에 코드를 작성하세요.

## 해결 방안

[해결 방안을 보려면 클릭하세요!](#)

```
zumi.reset_gyro()  
zumi.forward()  
zumi.turn(90)  
zumi.forward()  
zumi.turn(180)  
zumi.forward()  
zumi.turn(270)  
zumi.forward()  
zumi.turn(0)
```