소리

Zumi는 자신의 감정에 따라 소리를 재생할 수 있습니다! 소리는 주파수와 진폭으로 측정할 수 있습니다.

- 주파수는 초당 펄스 또는 진동 수이며 헤르츠 단위로 측정됩니다. 주파수가 높을수록 소리의 높낮이가 높아집니다.
- 진폭은 소리가 얼마나 크거나 강한지를 나타내며 데시벨로 측정됩니다. 진폭이 클수록 소리가 커집니다.

각각의 감정은 어떻게 들립니까? 행복은 저주파음일까 고주파음일까요? 화가난 경우는 진폭이 낮나요? 아니면 높나요? 이것은 Zumi에게 어떻게 적용됩니까?

`play_note()`를 사용하여 다양한 음표를 재생할 수 있습니다. 첫 번째 매개변수는 연주하려는 음입니다(C2에서 B6까지). 두 번째 매개변수는 선택 사항이며 음표가 재생되기를 원하는 시간(밀리초)을 나타냅니다. 기본값은 500ms로 설정되어 있지만 다음과 같은 두 번째 매개변수를 추가하여 변경할 수 있습니다.

음계는 다음과 같이 나타냅니다.
C D E F G A B C
도 레 미 파 솔 라 시 도

```
play_note(Note.GS3, 400).
```

이것은 400밀리초 동안 3옥타브 반올림 솔(G 샵)을 연주합니다.

라이브러리 가져오기

In [ ]:

```
from zumi.zumi import Zumi
from zumi.util.screen import Screen
from zumi.protocol import Note
import time

zumi = Zumi()
screen = Screen()
```

아래 코드를 시도하여 음계를 듣고 나만의 음악을 작곡해 보세요!

In [ ]:

```
zumi.play_note(Note.C4, 500)
zumi.play_note(Note.D4, 500)
zumi.play_note(Note.E4, 500)
zumi.play_note(Note.F4, 500)
zumi.play_note(Note.G4, 500)
zumi.play_note(Note.A4, 500)
zumi.play_note(Note.B4, 500)
zumi.play_note(Note.C5, 500)
```

여기에서 나만의 멜로디를 코딩해보세요! 소리를 통해서 Zumi가 작업을 완료했거나 물체를 감지했음을 알리는 것까지 다양한 용도로 사용할 수 있습니다.

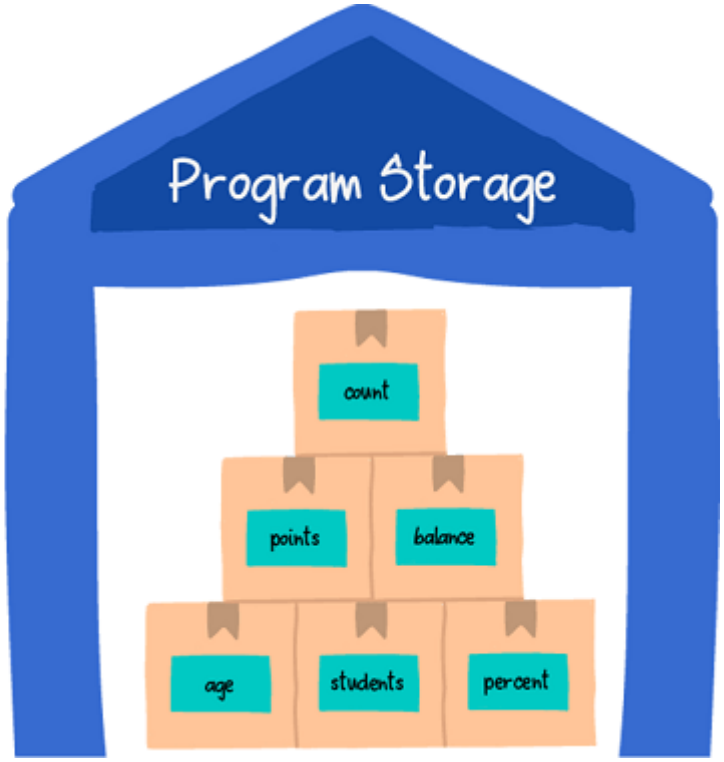
In [ ]:

```
# 여기에 여러분만의 멜로디를 만들어보세요.
```

변수란?

인간이 우리 뇌에서 기억을 찾듯이, 컴퓨터는 변수를 사용하여 프로그램 메모리의 값을 찾습니다.
변수는 값이 할당된 자리의 표시명입니다.
변수를 프로그램 메모리에 저장 창고로 생각하세요. 여러개의 변수를 만들 수 있으며, 다양한 값을 저장할 수 있습니다.

count = 2
students = 30
points = 100
balance = 250.75
age = 14
percent = 52.3



변수가 유용한 이유는?
변수에 들어가기 전에, 여러분은 그것이 유용할 수 있는 상황에 대해 이해해야 합니다.

방금 사용한 멜로디 프로그램을 떠올려 봅시다.

- 멜로디의 재생 시간을 1초로 변경해서 길게 음을 재생해봅시다
- 다시 멜로디의 재생 시간을 1.5초로 변경해서 더 길게 음을 재생해봅시다
- 다시 줄여서 0.8초로 재생시간을 변경합시다.
- 휴!😔 매번 코드의 각 줄을 바꾸기에는 너무 많은 일입니다. 어떻게 하면 이 코드를 더 효율적으로 만들 수 있을까요?

다음은 재생시간에 변수를 사용한 예제입니다.

In [ ]:

```
play_duration = 1000 # 1초

zumi.play_note(Note.C4, play_duration)
zumi.play_note(Note.D4, play_duration)
zumi.play_note(Note.E4, play_duration)
zumi.play_note(Note.F4, play_duration)
zumi.play_note(Note.G4, play_duration)
zumi.play_note(Note.A4, play_duration)
zumi.play_note(Note.B4, play_duration)
zumi.play_note(Note.C5, play_duration)
```

위의 변수를 사용한 예제에서 play_duration 변수 값을 변경해보세요.
어떻게 달라지나요?

지금부터, 여러분은 앞으로 수업에서 대부분 변수를 사용할 것입니다.
어떤 변수들은 숫자를 저장하지만, 다른 변수들은 단어나 그림을 저장할 수 있습니다!

화면

감정 표현

주미는 **OLED** 화면을 사용하여 감정을 표현할 수 있습니다.
주미의 눈의 종류에는 여러가지가 있습니다

* close_eyes()
* sleepy_eyes()
* sleepy_left()
* sleepy_right()
* blink()
* look_around_open()
* sleeping()
* look_around()
* glimmer()
* sad()
* happy()
* hello()
* angry()

원하는 표정을 사용해보세요

In [ ]:

```
screen.sad()
```

문자표시

주미의 눈을 그리는 것 외에도 주미가 화면에 메시지를 쓰도록 할 수도 있습니다!
`draw_text()` 함수를 사용하여 다음과 같은 메시지를 작성할수 있습니다.

In [ ]:

```
screen.draw_text("hello!")
```

자동으로 화면 중앙에 텍스트를 표시하려면 대신 이 함수를 호출하세요.

In [ ]:

```
screen.draw_text_center("hello!")
```

숫자로 텍스트를 작성하려면 모든 항목이 문자열 데이터 유형인지 확인해야 합니다.

In [ ]:

```
number = 10
screen.draw_text("ten " + str(number)) # the str() functions turns the number into a string
```

Zumi가 시간을 표시하도록 할 수도 있습니다!

In [ ]:

```
for i in range(0,50):
    screen.draw_text(time.ctime())
    time.sleep(0.1)
```

`draw_text()` 함수와 `draw_text_center()` 함수는 사용하게 되면 화면을 모두 지우게 됩니다.
여러메시지를 계속 출력하고 싶다면 어떻게 하면 좋을까요?

`pirnt()` 함수를 사용하면 됩니다.
`pirnt()` 함수는 화면을 지워주는 기능이 없으므로, 화면을 지우고 싶을때는 `clear_drawing()` `()` 함수를 같이 사용해주세요.

print(*messages,x=0, y=0, fill_in=True,font_size=12)
x: 문자열의 x 좌표.기본값은 0 입니다.
y: 문자열의 y 좌표.기본값은 0 입니다.

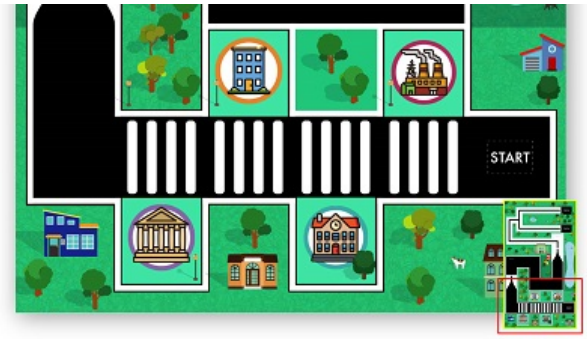
다음 줄을 표시할 때는 y 값을 변경해줍니다.
혹은 "\n" 줄바꿈 기호를 사용하여 줄을 바꿔줍니다.

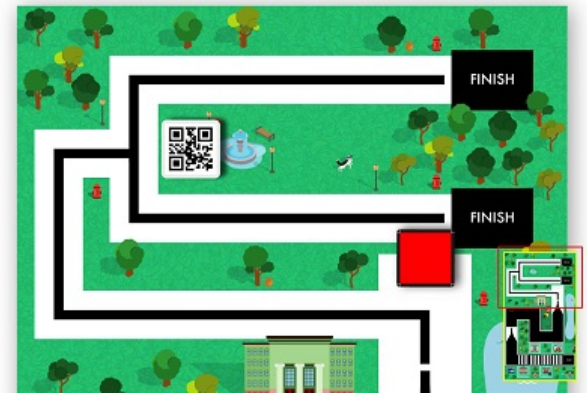
In [ ]:

```
screen.clear_drawing()
screen.print("12345678")
screen.print("hello", y=15)
screen.print("WnWn","Zumi")

time.sleep(1)
screen.clear_drawing()
screen.print("ZUMI",font_size=20)
```

대회 : 어떻게 사용되나요?





주미는 QR에서 읽은 목적지로 이동합니다.
모든 코스를 진행한 후에 이동한 목적지를 화면에 출력하도록 되어있습니다.

확인할때마다 바로바로 출력을 하는 방법과 하나의 변수에 모아두었다가 마지막에 출력하는 방법이 있습니다.

In [ ]:

```
screen.clear_drawing()

qr_message = "school"

screen.print(qr_message,font_size=30)

qr_message = "left"

screen.print(qr_message,y=33,font_size=30)
```

In [ ]:

```
screen.clear_drawing()

qr_message = "school"

destination = qr_message

qr_message = "left"
destination = destination + "₩n" +qr_message

screen.print(destination,font_size=30)
```