



QR 코드

본인 확인, WiFi에 연결, URL로 이동, 게임 또는 결제와 같은 다양한 작업에 대한 **QR(Quick Response)** 코드를 본 적이 있을 것입니다. QR 코드는 쉽게 읽을 수 있는 축약형 라벨이기 때문에 인기가 높습니다.

많은 스마트폰에 QR 코드 스캐너가 내장되어 있고, 이를 해독할 수 있는 다른 애플리케이션도 많이 있습니다.

이 수업에서는 QR 코드가 작동하는 방식과 Zumi 세계에 QR 코드를 구현하는 방법을 배웁니다.

QR 코드 스캔

QR 코드는 인간에게는 무작위로 배열된 픽셀처럼 보이지만 컴퓨터에는 암호화된 정보가 들어 있습니다!

다음 QR 코드를 살펴보세요. 그것이 무엇이라고 생각하나요?



이 QR 코드를 읽으려면 몇 가지 라이브러리를 가져와야 합니다.

In []:

```
from zumi.zumi import Zumi
from zumi.util.camera import Camera
from zumi.util.vision import Vision
import time

camera = Camera()
vision = Vision()
```

이제 다음 코드를 실행하고 Zumi의 카메라에 동일한 QR 코드를 표시합니다.
그것은 무엇을 뜻하는 걸까요?
숨겨진 메시지를 보려면 셀의 출력을 보세요...



In []:

```
camera.start_camera()
try:
    for i in range(50):
        frame = camera.capture()
        vision.find_QR_code(frame)
        camera.show_image(frame)
        camera.clear_output()
finally:
    print("완료!")
    camera.close()
```

find_QR_code() 함수

새로운 기능을 소개할 시간입니다!

위에서 본 `find_QR_code()` 함수는 매개변수로 제공된 이미지를 처리하고 해석된 메시지가 있는 코드 주위에 사각형을 그립니다.

QR 코드는 어떻게 작동하나요?

앞서 본 QR 코드는 "Hello!"로 인코딩(암호화)되었지만 어떻게 작동합니까?

2진수

2진수는 컴퓨터에서 사용하는 언어입니다. 컴퓨터는 전자 장치이기 때문에 컴퓨터는 전자 펄스로 메시지를 보내야 합니다.

펄스가 HIGH이면 1로 해석됩니다. LOW이면 컴퓨터는 0으로 해석합니다.

컴퓨터는 이 펄스를 인간이 처리할 수 있는 것보다 훨씬 빠르게 보낼 수 있습니다!

이러한 개별 값을 **비트**라고 하며 두 개의 값만 가질 수 있는 숫자입니다. 이 두 값은 0과 1입니다.



QR 코드에 인코딩된 것과 동일한 메시지인 "Hello!"가 화면에 표시되면 QR코드를 잘 읽은 것입니다.

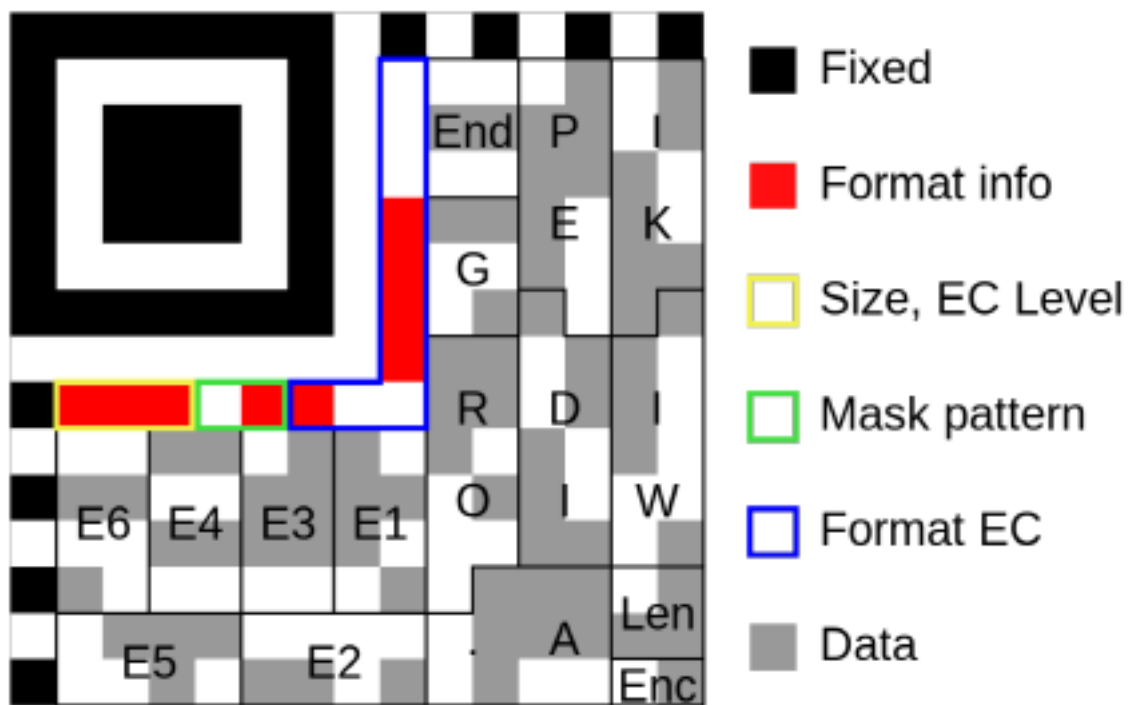
이 2진수 코드는 어떻게 QR 코드의 흑백 사각형으로 변환될까요? 2진수는 HIGH, LOW의 두가지 값만 사용한다는 것을 기억하세요.

이것은 검은색 사각형이나 흰색 사각형, "켜기" 또는 "끄기"로 나타낼 수 있습니다.

그 메시지를 뱀처럼 네모 모양으로 감싸면 QR코드가 나옵니다!

서식에 대한 몇 가지 세부사항과 규칙이 더 있지만 일반적인 개념은 동일합니다.

다음은 "wikipedia.org"에 대한 QR 코드를 분류한 것입니다.



https://commons.wikimedia.org/wiki/File:Micro_QR_Encoding.svg

get_QR_message() 함수

우리가 QR코드를 사용하면서 좋은 점은 디코딩(해석)할 코드를 별도로 작성할 필요가 없다는 것입니다.

이제 작동 방식을 알았으므로 새로운 기능을 사용하여 이 작업을 진행할 수 있습니다.

먼저 사진을 찍어 QR 코드가 감지되었는지 확인해야 합니다.

즉, if 문을 사용하여 QR 코드가 있는지 확인해야 합니다.

예를 들어 사진을 찍고 프레임 (frame) 변수에 저장했다고 가정하면 먼저 QR 코드를 감지합니다.

```
qr_code = vision.find_QR_code(frame)
```

그런 다음 `get_QR_message(qr_code)` 값을 출력하여 메시지를 표시할 수 있습니다. 인쇄물에서 QR 코드 중 하나를 가져와 Zumi의 카메라 앞에 놓습니다. 그런 다음 코드를 실행하여 메시지를 확인하세요!

In []:

```
camera = Camera()
camera.start_camera()
frame = camera.capture()
camera.close()
qr_code = vision.find_QR_code(frame)
message = vision.get_QR_message(qr_code)
print(message)
```

하지만 Zumi가 QR 코드를 보지 못하면 어떻게 되나요? QR 코드 없이 코드를 실행해보세요. 출력된 결과는 무엇인가요?

None vs "None"

QR 코드가 없는 이미지에서 스캔을 시도한 후 메시지를 출력려고 하면 메시지가 `None`으로 표시됩니다.

이 `None`은 값이 없음을 나타내는 파이썬의 특수 값입니다.

이것은 `None`이라는 단어가 포함된 문자열 값인 `"None"`과 다릅니다.

반복문

이전 레슨에서 반복문, 구체적으로는 `for` 반복문에 대해 배웠습니다. 이번에는 `while` 반복문에 대해 알아보겠습니다.

`for` 반복문과 `while` 반복문에는 기본적으로 같은 부분이 있습니다.

- 반복문의 조건식 : 반복문을 실행하거나 정지시키는 이유. 일반적으로 부등식의 형태입니다 (예를 들어 $i < 5$).
- 반복문의 실행문 : 반복문이 실행하는 내용. 반복문이 완료될 때까지 반복.

while 반복문

`for` 반복문과 `while` 반복문의 가장 중요한 차이점은 목적에 있습니다.

- `for` 반복문 : 반복문을 실행하는 횟수가 정해져 있을 때 사용합니다.
- `while` 반복문 : 반복문이 몇 번 실행될지 모를 때 사용합니다

`while` 은 조건이 참이라면 들여쓰기된 코드를 반복하게 됩니다.
`i` 값이 100보다 작다면 반복합니다.

In []:

```
i = 1
while i < 100: # s가 100보다 작을 동안 반복
    print(i)
    i = i * 2 # i를 두 배 한다
```

while 문은 반복할 횟수를 모르기 때문에, 무한으로 반복되어 실행될 수 있습니다.
이런 경우를 방지하기 위해, 조건식을 만들 때 주의해야 합니다.

아래 예제는 실행되면 끝없이 실행되므로, 정지 버튼을 눌러 멈춥시다.

In []:

```
i = 1
while i > 0:
    i = i + 1
    print(i)
    time.sleep(1)
```

카메라를 사용해서 QR코드를 찾아야하는 경우에 원하는 횟수만큼만 카메라를 확인하여 QR코드를 찾을 수 있습니다.
하지만, 그 횟수안에 찾을거라는 보장은 없습니다.
그래서 QR 코드를 찾을 때까지 반복문을 실행하고, 찾게 되면 반복문을 종료하도록 합니다.

In []:

```
zumi = Zumi()
camera = Camera()
camera.start_camera()
message = None
try:
    while message == None:
        image = camera.capture()
        qr_code = vision.find_QR_code(image)
        message = vision.get_QR_message(qr_code)
        camera.show_image(image)
        print(message)
        camera.clear_output()
finally:
    camera.close()
    print(message)
```

break 문

break 문은 반복문의 조건이 충족되지 않더라도 끝내야 할 경우 사용합니다.
for문과 while 모든 반복문에서 사용이 가능합니다.

In []:

```
for i in range(100):
    print(i)
    if i == 5: # i가 5가 되는 경우에
        break # 반복문이 종료됩니다.
```

In []:

```
i = 0 # i 는 0 이라고 지정한 후 ,

while True: # 무한 루프
    print(i) # i 마다 프린트를 한다.
    if i == 5: # 만약, i 가 5 와 같다면,
        break # break 를 실행시킨다.
    i = i + 1 # 한 루프마다 i 를 1 씩 증가시킨다.
```

마찬가지로 QR 코드를 찾을 때까지 반복문을 실행하고, 찾게 되면 반복문을 종료하도록 합니다. 이 경우에는 while 문의 조건이 True입니다. 따라서 무한 반복문이 되어 끝없이 실행됩니다. 그래서 실행문에 조건을 달아서 break문으로 빠져나올 수 있게 합니다.

In []:

```
zumi = Zumi()
camera = Camera()
camera.start_camera()
message = None
try:
    while True:
        image = camera.capture()
        qr_code = vision.find_QR_code(image)
        message = vision.get_QR_message(qr_code)
        camera.show_image(image)
        print(message)

        if message != None:
            break

    camera.clear_output()
finally:
    camera.close()
    print(message)
```

QR 메시지 같은지 비교하기

우리는 QR 코드의 메시지로 판단을 하고 결정을 내릴 수 있습니다.

if 문에 대한 강의를 기억하시나요?

이 경우 메시지는 문자열입니다.

"apple"이 "banana"보다 큰지 묻는 것은 실제로 의미가 없지만 두 메시지가 일치하거나 동일한지 확인할 수는 있습니다. 파이썬에서 두 값이 같은지 확인하는 것은 다음과 같습니다.

예를 들어

```
if message == "apple":
    # 실행되는 코드!
```

이 예에서 message는 변수입니다. 변수 내부의 값이 "apples"과 정확히 같으면 if 문이 실행됩니다.

QR로 의사 결정하기

두 개의 QR 코드를 선택하고 앞서 사용한 코드로 QR의 메시지를 확인합니다.
예를 들어, "학교"와 "박물관"을 선택하는 경우 각각의 위치를 설정합니다.
프로그램 시작 시 Zumi에게 QR 코드를 보여주세요. 그러면 주미는 해당위치로 이동할 것 입니다.



의사 코드가 어떻게 생겼는지 살펴보겠습니다.

카메라 시작
사진을 찍어 변수 'image'에 저장
카메라를 닫기
'image'에서 QR 코드 (있는 경우)를 찾고 결과를 'qr_code' 변수에 저장합니다.
QR 메시지를 받아 변수 'message'에 저장
메시지가 "school"라고 표시되면 학교로 운전
그렇지 않으면 메시지가 "museum"이라고 표시되면 박물관으로 운전합니다.

이제 아래에 코드를 작성하세요. 나중에 if 문에서 사용할 수 있도록 이미지, QR 코드 및 메시지를 변수에 저장해야 합니다.
if 문의 콜론과 들여쓰기를 잊지 마세요!

In []:

```
zumi = Zumi()
zumi = Zumi()
camera = Camera()
camera.start_camera()
message = None
try:
    # 여기에 코드를 쓰세요.

finally:
    camera.close()
    # 여기에 코드를 쓰세요.
```

해결 방안

[해결 방안을 보려면 클릭하세요!](#)

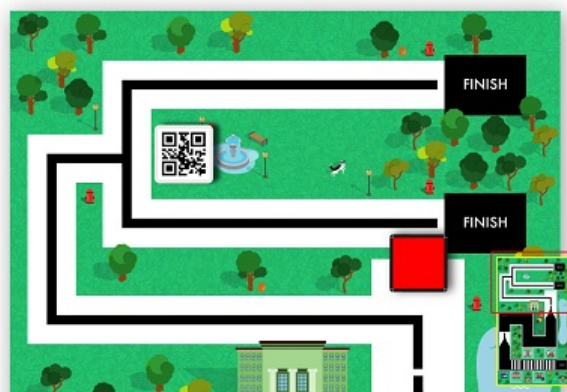
```

zumi = Zumi()
camera = Camera()
camera.start_camera()
message = None
try:
    while message == None:
        image = camera.capture()
        qr_code = vision.find_QR_code(image)
        message = vision.get_QR_message(qr_code)
        camera.show_image(image)
        print(message)
        camera.clear_output()
finally:
    camera.close()

print(message)
if message == "school":
    zumi.forward(40, 2)
    zumi.stop()
elif message == "museum":
    zumi.forward(40, 4)
    zumi.stop()

```

대회 : 어떻게 사용되나요?



대회 맵의 두 구역에는 QR 코드가 포함됩니다.

대회의 첫 번째 부분에서 주미에게 운전하고 주차할 위치를 알려주려면 QR을 읽어야 합니다.

대회의 세 번째 부분에서 주미는 QR 코드를 읽고 운전할 결승선을 결정해야 합니다.

In []:

```
qr_message = None

camera.start_camera()
qr_message = None
try:
    while qr_message == None:

        # 출력창을 지웁니다. (이미지가 계속 늘어나지 않도록)
        camera.clear_output()

        # 화면을 frame 변수에 저장합니다.
        frame = camera.capture()

        # QR코드를 화면에서 찾습니다.
        qr_code = vision.find_QR_code(frame)

        # QR코드에서 문자열만 가져옵니다.
        qr_message = vision.get_QR_message(qr_code)

        # 화면을 출력합니다.
        camera.show_image(frame)

    camera.close()

finally:
    camera.close()

print(qr_message)

if qr_message == "school":
    print("go school!")
elif qr_message == "museum":
    print("go museum!")
elif qr_message == "building":
    print("go building!")
elif qr_message == "factory":
    print("go factory!")
elif qr_message == "office":
    print("go office!")
```