CS101 - `if` 조건문과 `while` 반복문

Lecture 3

School of Computing
KAIST

학습 목표:

- `if` 조건문의 형태 및 동작을 이해할 수 있다.
- `while`반복문의 형태 및 동작을 이해할 수 있다.

조건문

지금까지 우리가 만든 프로그램은 실행될 때마다 동일한 작업을 수행했습니다.

하지만, 로봇은 환경(상황)에 의존해서 움직여야 할 때가 자주 있습니다.

```
if it rains:
  listen_to_cs101_lecture()
else:
  eat_strawberries_in_the_sun()
```

조건

조건이 참이면, 이 작업을 수행한다

조건이 거짓이면, 이 작업을 수행한다

조건은 **참**(`True`)이나 **거짓**(`False`) 값을 가질 수 있습니다.

간단한 예시

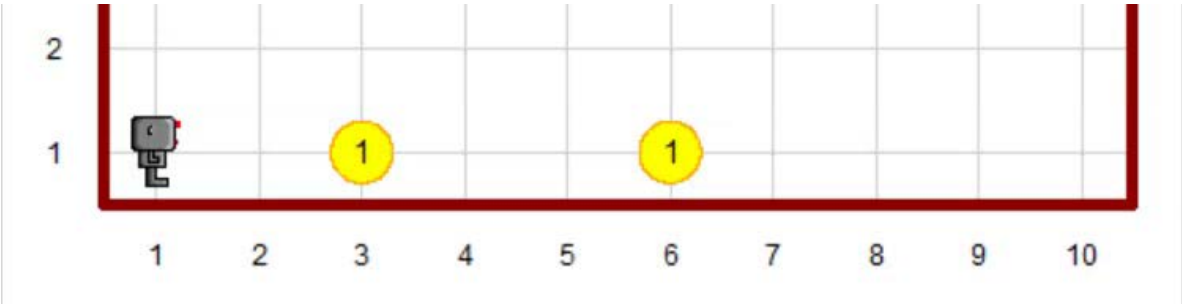
```
if True:
  print("CS101 is my favorite course")

if False:
  print("Every CS101 student will receive an A+")

if 3 < 5:
  print("3 is less than 5")
else:
  print("3 is larger than 5")
```

비퍼 감지

로봇을 9칸 전진시키면서 경로에 있는 모든 비퍼를 줍도록 하려고 합니다.



`hubo.pick_beeper()`는 비퍼가 없을 때 에러가 발생합니다.

다음 과정을 9회 반복합니다 :

- 한 칸 전진한다
- 비퍼가 있는지 확인한다
- 비퍼가 있으면, 비퍼를 줍는다

```
def move_and_pick():
    hubo.move()
    if hubo.on_beeper():
        hubo.pick_beeper()

for i in range(9):
    move_and_pick()
```

참의 반대는 거짓

방금 과정을 반대로 해 봅시다

: 현재 위치에 비퍼가 **없을** 때만 비퍼를 떨어뜨리고 싶습니다.

```
if not hubo.on_beeper():
    hubo.drop_beeper()
```

`not` 키워드는 조건을 반대로 바꿉니다.

: `not True` 는 `False` 이고, `not False` 는 `True` 입니다.

다음 코드의 결과는 어떻게 될까요?

```
print(not 3 < 5)
```

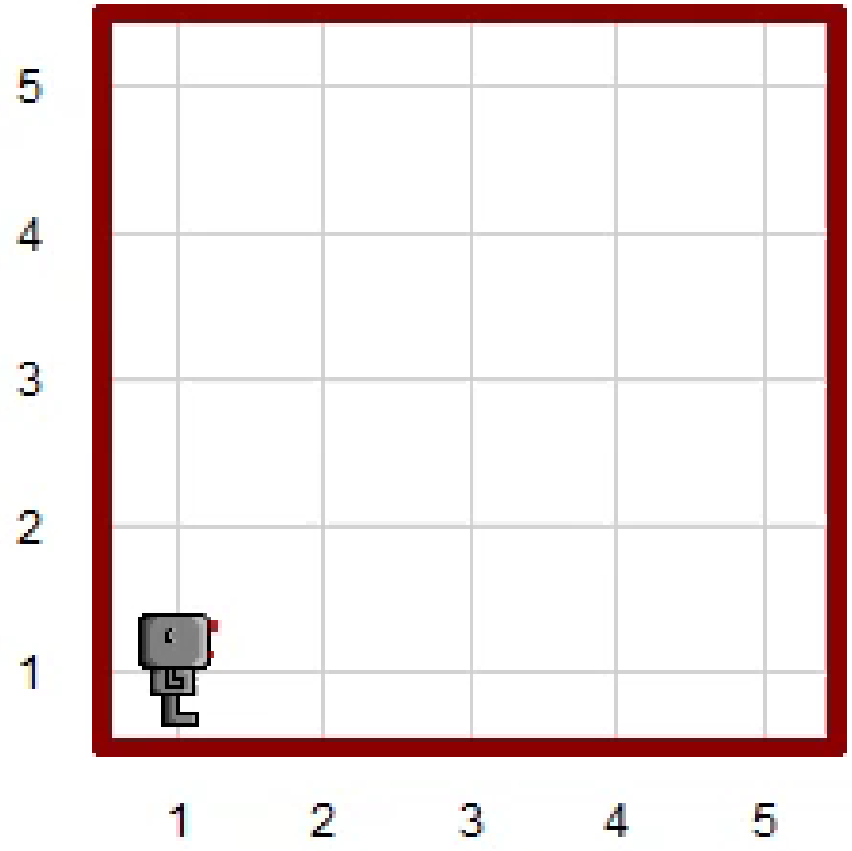
else란

로봇이 세계의 경계선을 따라서 움직이게 해 봅시다.

: 전방에 벽이 없으면 전진하고, 벽이 있으면 좌회전합니다.

```
def move_or_turn():
  if hubo.front_is_clear():
    hubo.move()
  else:
    hubo.turn_left()

for i in range(20):
  move_or_turn()
```

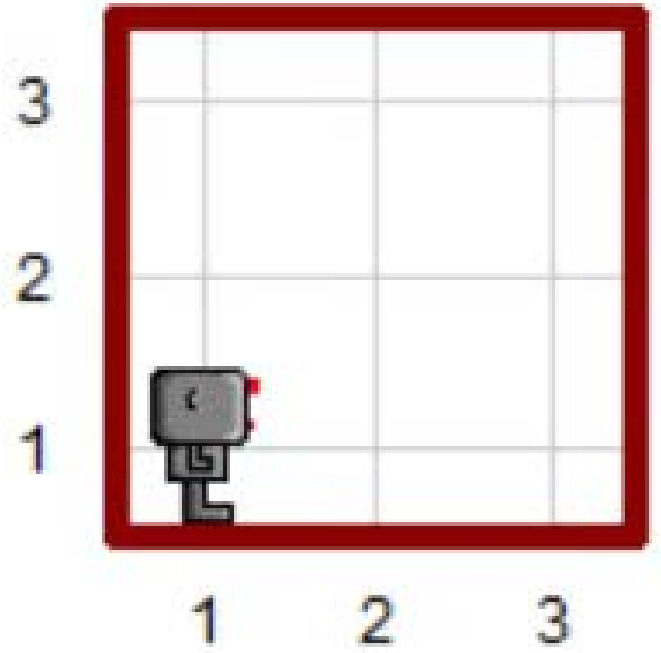


춤추고 노래하는 휴보

```
def dance():
    for i in range(4):
        hubo.turn_left()

def move_or_turn():
    if hubo.front_is_clear():
        dance()
        hubo.move()
    else:
        hubo.turn_left()
        hubo.drop_beeper()

for i in range(18):
    move_or_turn()
```

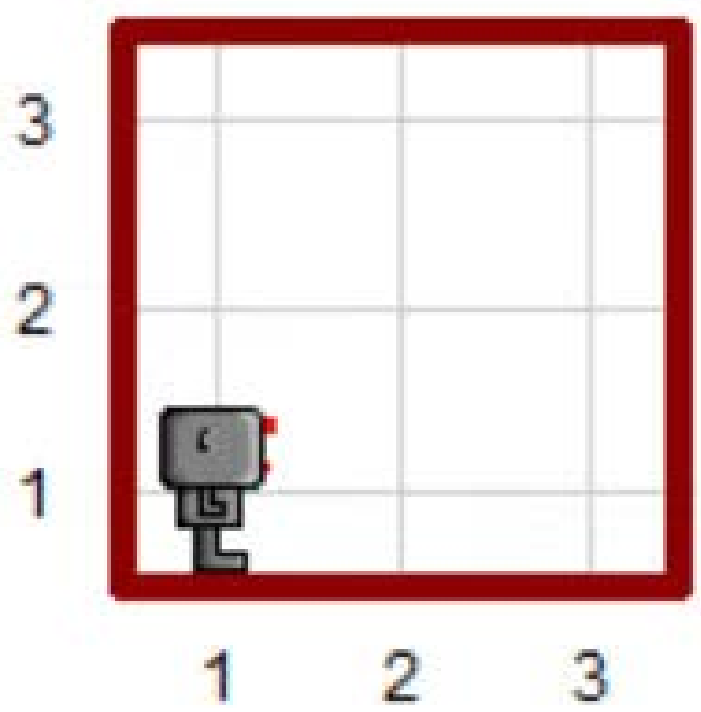


춤추고 노래하는 휴보

```
def dance():
    for i in range(4):
        hubo.turn_left()

def move_or_turn():
    if hubo.front_is_clear():
        dance()
        hubo.move()
    else:
        hubo.turn_left()
    hubo.drop_beeper()

for i in range(18):
    move_or_turn()
```



들여쓰기에 주의하세요!

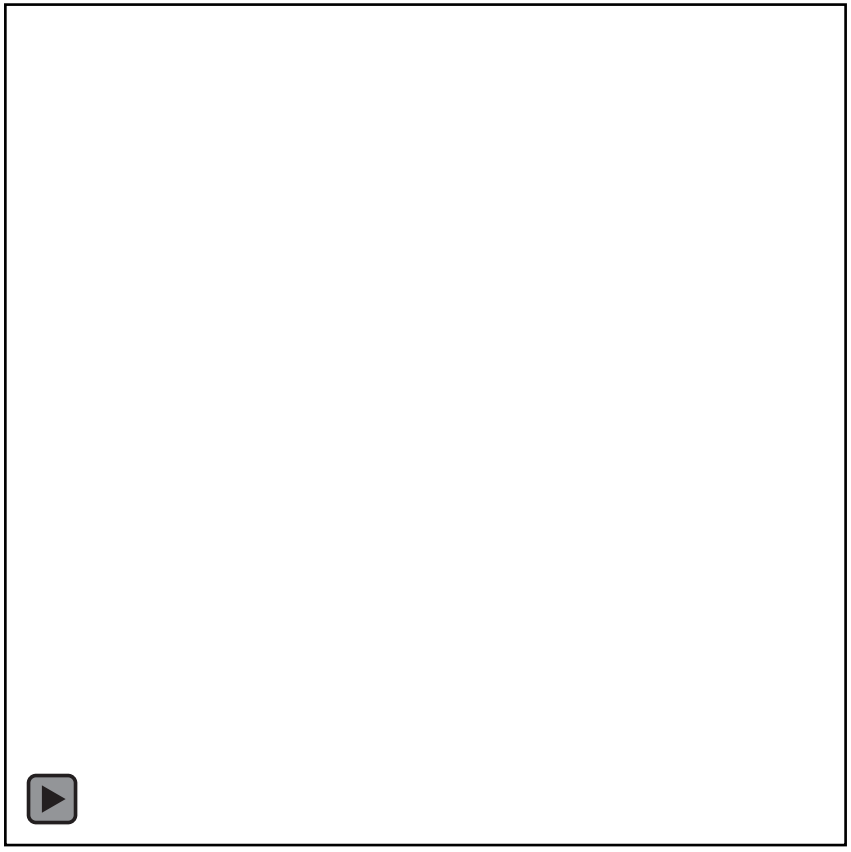
이제 로봇이 어떻게 행동할까요?

춤추고 노래하는 휴보

```
def dance():
  for i in range(4):
    hubo.turn_left()

def move_or_turn():
  if hubo.front_is_clear():
    dance()
    hubo.move()
  else:
    hubo.turn_left()
hubo.drop_beeper()

for i in range(18):
  move_or_turn()
```



… 이렇게 바꾸면요?

너무 많은 선택문

```
if hubo.on_beeper():
  hubo.pick_beeper()
else:
  if hubo.front_is_clear():
    hubo.move()
  else:
    if hubo.left_is_clear():
      hubo.turn_left()
    else:
      if hubo.right_is_clear():
        turn_right()
      else:
        turn_around()
```

문제점) 이 코드는 읽고 이해하기가 너무 힘들어요!

너무 많은 선택문

```
if hubo.on_beeper():
  hubo.pick_beeper()
elif hubo.front_is_clear():
  hubo.move()
elif hubo.left_is_clear():
  hubo.turn_left()
elif hubo.right_is_clear():
  turn_right()
else:
  turn_around()
```

elif 는 **else** 와 **if** 를 결합시킨 것으로,
복잡한 들여쓰기 없이 많은 연관된 조건문들을 표현할 수 있습니다.

while 반복문

for 반복문은 정해진 횟수만큼 명령을 반복합니다.
while 반복문은 주어진 조건이 참이라면 명령을 계속 반복합니다.

비퍼를 발견하기 전까지 계속 전진하려면

```
while not hubo.on_beeper():
    hubo.move()
```

정리 및 예습

본 강의 학습 목표:

- `if` 조건문의 형태 및 동작을 이해할 수 있다.
- `while`반복문의 형태 및 동작을 이해할 수 있다.

다음 강의 학습 목표:

- `if` 조건문과 `while`반복문을 사용하여 복잡한 미로를 탈출하는 프로그램을 작성할 수 있다.
- 프로그램을 작성할 때 따라야 하는 과정을 이해 할 수 있다.