

파이썬을 활용한 프로그래밍 언어, 리보그




PYCON
APAC 2016

*교육용

소개



김무훈(BetaFish)

-  [@MuhunKim](#)
- 서귀포고등학교 재학 / 2학년
- 웹 프론트 엔드 개발과 파이썬을 즐깁니다.
- 관심 분야 :
 - 오픈소스 개발
 - 정보 교육

발표할 내용

1. 리보그 소개
2. 리보그 + 파이썬 프로그래밍 = 문제 해결력
3. 객체지향 -> 리보그

“왜 이 언어를
파이콘에서
발표하나요?”

왜?

파이썬 = 간단 명료함 = 교육적인 언어

비전공자들에게 적합.

리보그? 쉽게 배우는 파이썬!

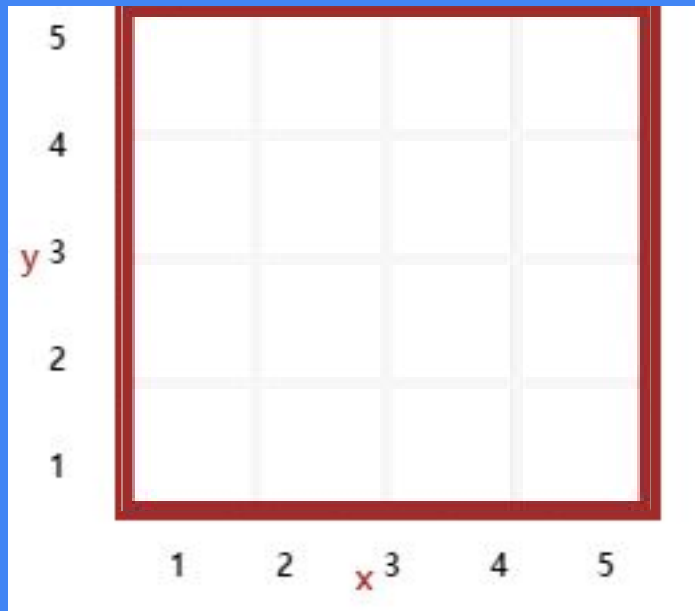
리보그?

카렐 언어의 영향을 많이 받은 언어

“격자_세상_속_로봇에게 완수해야 되는 작업을 파이썬 프로그래밍으로
해결하면서 컴퓨팅 사고력을 기르는 것을 목적으로 하는 언어”

격자 세상 속 로봇 —리보그

```
while True:
    if front_is_clear():
        move()
    else:
        turn_left()
```



파이썬을 활용한 언어, 리보그
파이썬 구문을 그대로 쓸 수 있음

리보그 언어의 목적

프로그래밍을 처음 배우는 학생들을 대상으로
파이썬으로 프로그래밍 개념을 쉽게 익히면서
문제해결력을 기르는 언어

리보그 언어의 목적

프로그래밍을 처음 배우는 학생들을 대상으로
파이썬으로 프로그래밍 개념을 쉽게 익히면서
문제해결력을 기르는 언어

더 나아가서 초심자가 객체지향 같은

고급 프로그래밍 개념을 쉽게 익힐 수 있는 언어

리보그의 역사

1981 ~

카렐 언어, Karel the Robot

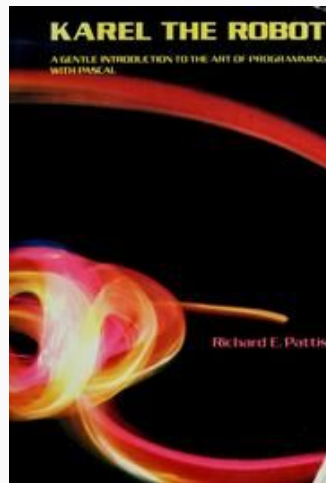
move

turnLeft

pickBeeper

putBeeper

turnOff



<https://www.cs.mtsu.edu/~untch/karel/book.html>

1981년 당시 스탠포드 대학에 재학 중이었던 리처드 패티스 교수가 카렐을 설계함

Karel the Robot : 왜 이름이 카렐인가요?

R.U.R.

위키백과, 우리 모두의 백과사전.

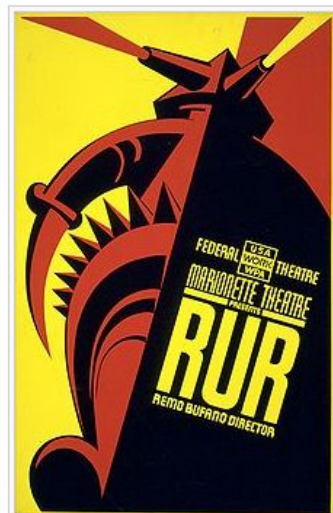
R.U.R.은 1920년에 체코의 작가 카렐 차페크가 쓴 SF 희곡이다. R.U.R.은 Rossumovi Univerzální Roboti (Rossum's Universal Robots)의 약자이다.^[1] 그러나, 체코어 원본에 부제를 영문인 Rossum's Universal Robots로 달아두었다.^[2] 1921년 1월 25일에 초연되었고, 영어와 SF세계에 robot라는 단어를 처음으로 등장시켰다.^[3]

출판된 뒤에는 매우 빠르게 유명해져서^{[4][5][6]} 1923년에는 30개 나라 언어로 번역되었다.^{[4][7]}

줄거리 [편집]

희곡은 roboti(robots)라 불리는 인조인간을 생산하는 공장에서 시작된다. 이 인조인간들은 현재 쓰이는 로봇의 정의에 정확하게 부합하는 로봇은 아니다. 인간으로 착각될 수 있고 스스로 생각할 수 있다는 점에서, 현대의 감각으로 말하자면 사이보그나 안드로이드, 아니면 클론에 가깝다. 그들은 처음에는 인간을 위해 일하는 것을 행복해하는 듯 했지만 변해버리고, 이 적대적인 로봇들의 반란은 인간의 멸종으로 이어진다. 차페크는 나중에 인간이 아닌 존재가 인간사회의 하수인그룹이 된다는 내용의 **도롱뇽과의 전쟁**(Válka s mloky)을 발표함으로써 같은 주제에 대해 다른 관점에서 접근을 시도한다.^[8]

R.U.R.은 어둡지만 희망이 없지는 않은 작품이며, 당시 유럽과 미국에서 모두 성공을 거두었다.^[9]



R.U.R. 상영 포스터

Karel the Robot : CODE

BEGINNING-OF-PROGRAM

BEGINNING-OF-EXECUTION

```
move;  
turnoff
```

END-OF-EXECUTION

END-OF-PROGRAM

* 매킨토시 첫출시일 : 1984년
카렐 : 1981년???

이후 여러 프로그래밍 언어로
카렐 구현체들이 생겨남
(Pascal, C++, Java, js, Python)

수 많은 구현체들이 생겨났으나..

접근성이 영 좋지 않거나
초심자에게 적합하지 않는 언어여서

..결과적으로 망했어요

RUR-PLE

2004

RUR-PLE (2004)

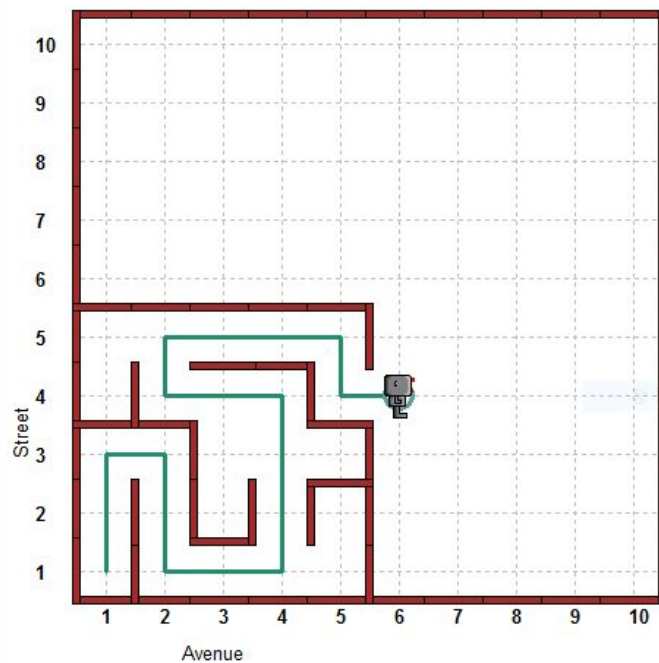
- Roberge's **Used Robot** (~~Rossum's Universal Robots~~)
 - 개발자 : 캐나다 물리학자 Andre Roberge
 - 파이썬! 파이썬!(python 2.5)
 - 2세대 - 리보그



<http://rur-ple.sourceforge.net>



```
1 def right():  
2     .turn_left()  
3     .turn_left()  
4     .turn_left()  
5  
6     turn_left()  
7     move()  
8     move()  
9     right()  
10    move()  
11    right()  
12    move()  
13    move()  
14    turn_left()  
15    repeat(move, 2)  
16    turn_left()  
17    repeat(move, 3)  
18    turn_left()  
19    move()  
20    move()  
21    right()  
22    move()  
23    right()  
24    repeat(move, 3)  
25    right()  
26    move()  
27    turn_left()  
28    move()  
29    turn_off()  
30  
31
```



1

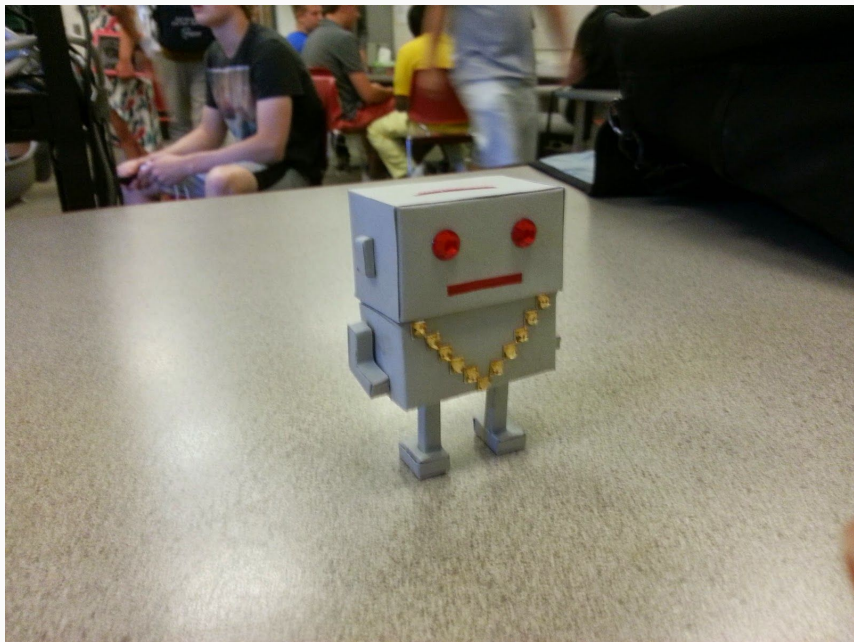
PEP8

```
move()  
turn_left()
```

```
def turn_right():  
    turn_left()  
    turn_left()  
    turn_left()
```

```
reeborg = UsedRobot()  
reeborg.move()
```

어디에서 쓰이나요?

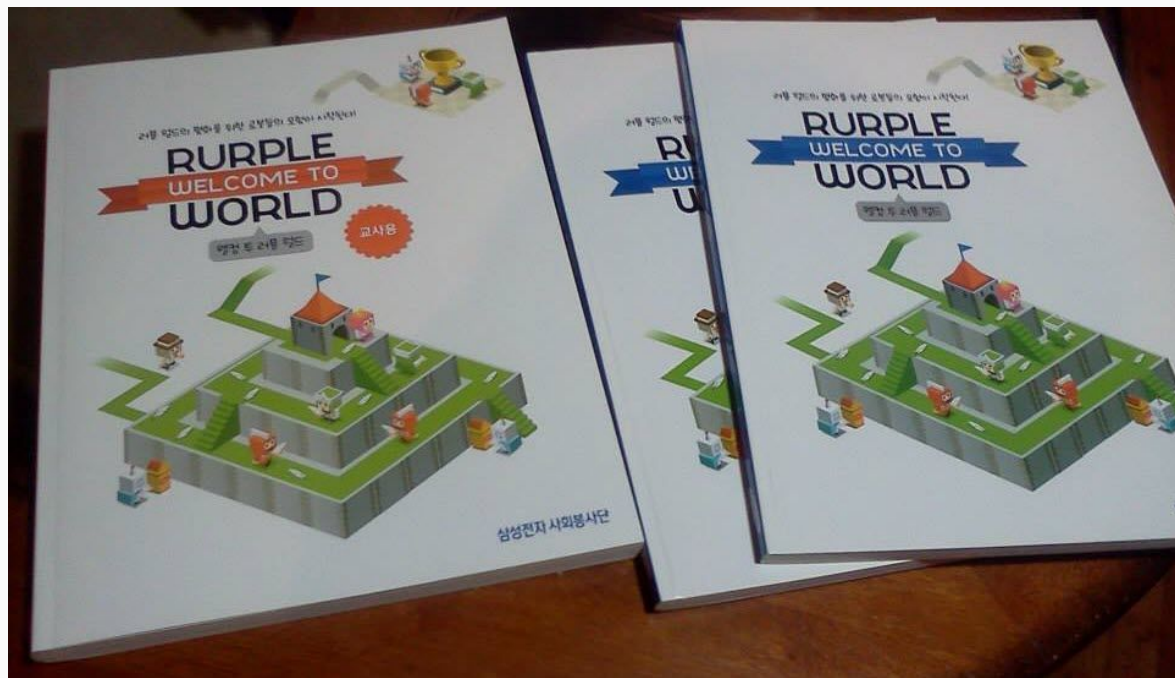


<http://aroberge.blogspot.kr/2014/06/reeborg-in-real-life.html>

미국 뉴저지의 한 고등학생이
할로윈때 쓸 리보그 의상을 제작함

많은 학교와 대학교에서 프로그래밍을 교육하기 위해 사용합니다.

어디에서 쓰이나요?



삼성 주니어 소프트웨어 아카데미의 교육과정

Q2

사용할 프로그래밍 언어를 선택해주세요.

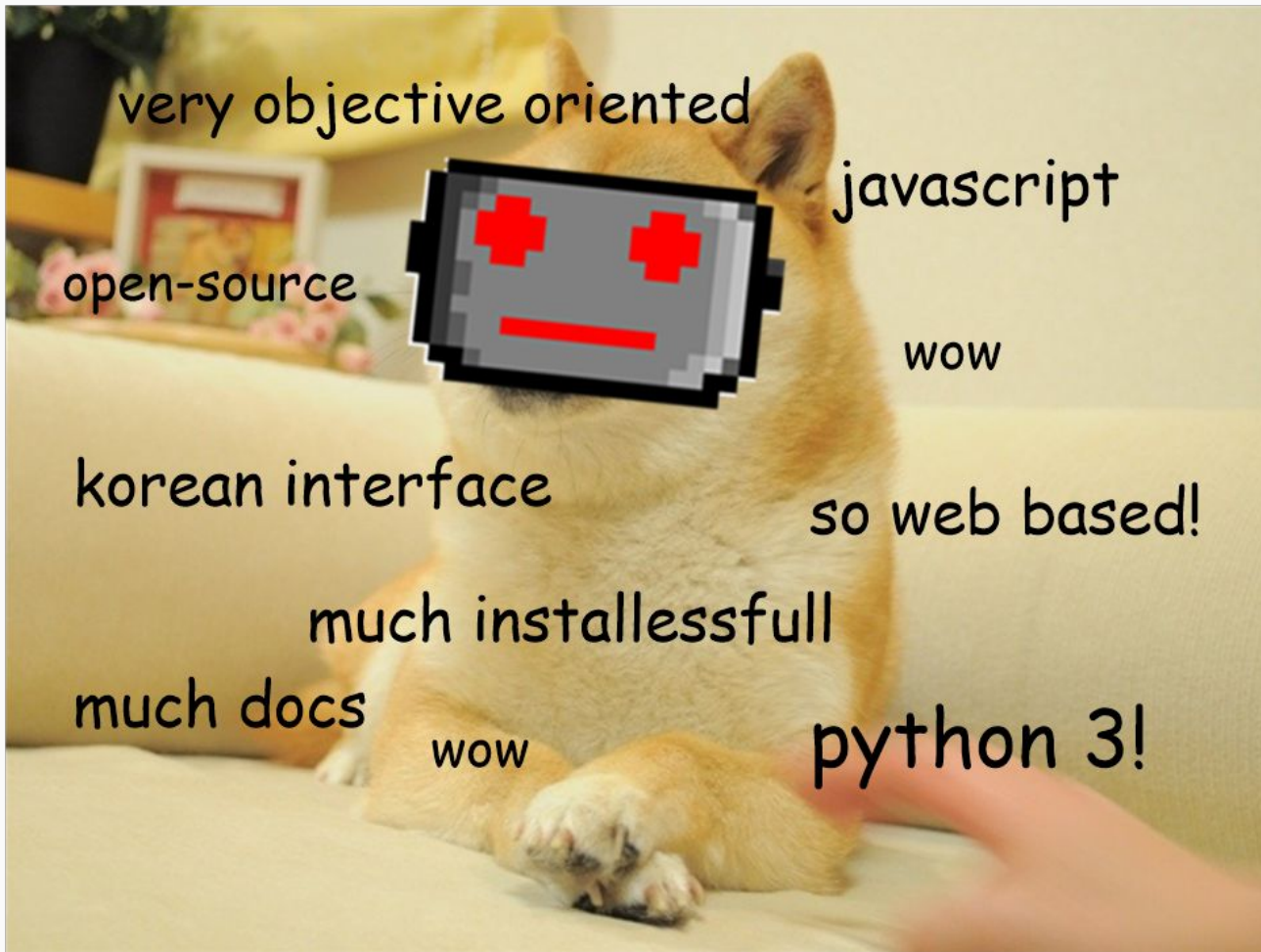
- ☐ 스크래치 ☐ 앱 인벤터 ☐ 엔트리 ☐ C / C++ ☐ 자바 ☐ 비주얼베이직 ☐ 러플
- ☒ 자바스크립트 ☐ ASP ☐ PHP ☐ 기타

<https://juniorswcup.com/contest/applyform/step3>

삼성 주니어 소프트웨어 대회
예선 접수 3번째 단계

그만 사용하세요!

대신에 리보그를 쓰세요..



[https://en.wikipedia.org/wiki/Doge_\(meme\)](https://en.wikipedia.org/wiki/Doge_(meme))

리보그 사용자 환경



미국 캘리포니아의 학생들이 리보그로
프로그래밍을 배우는 모습

특수 객체



목표 : 객체

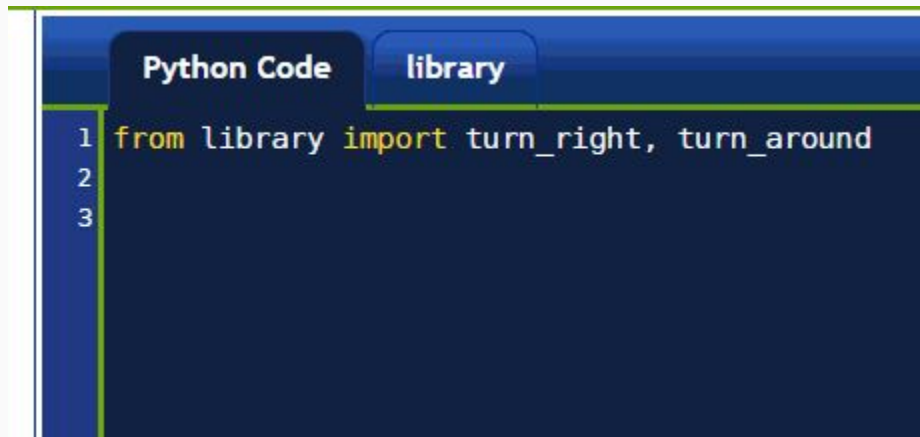


목표 : 로봇



리보그에게 정해진 최종 지점을 표시

리보그에서 라이브러리 사용하기



The image shows a screenshot of a code editor interface. At the top, there are two tabs: 'Python Code' and 'library'. The 'library' tab is currently selected. Below the tabs, the code editor displays the following Python code:

```
1 from library import turn_right, turn_around
2
3
```

<http://reeborg.ca/docs/ko/python/library.html>

import

```
import library  
library.turn_right()
```

앞에 library. 를 붙여주면 됩니다.

Etc..

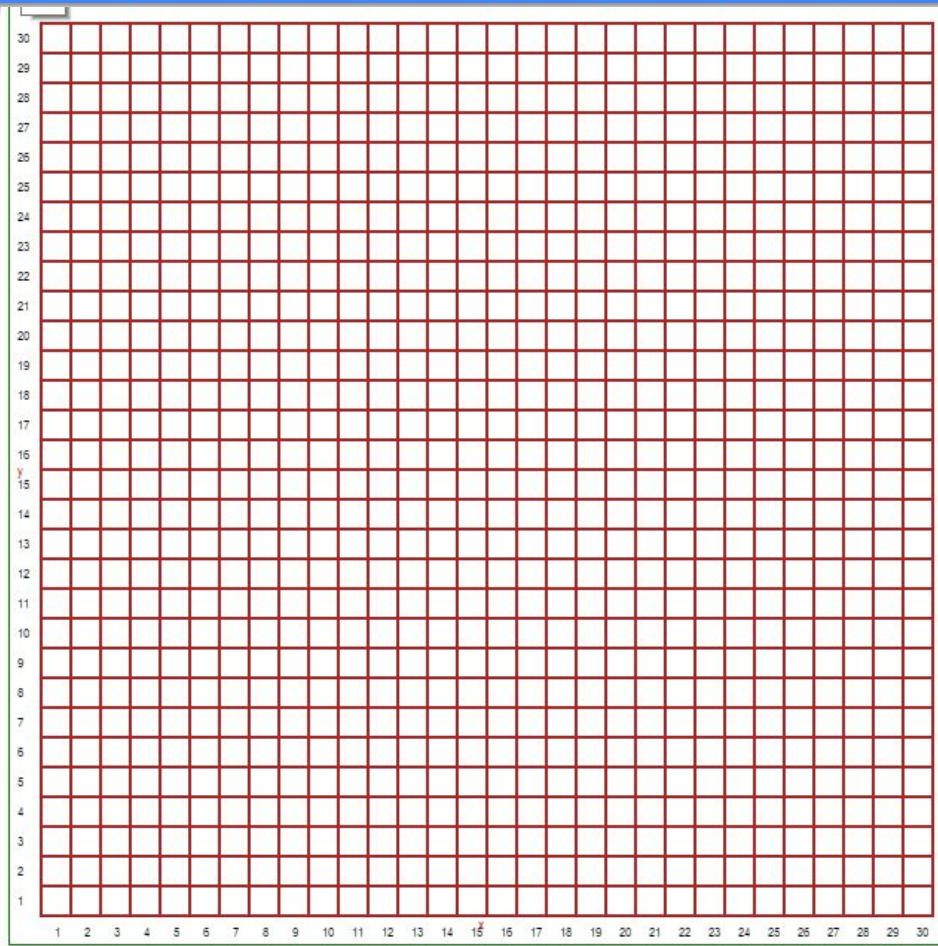
- `from library import turn_right as right`
- `from library import *`
- `from random import *` #Python 내장 모듈
- `from reeborg_fr import *`
- `avance() # franch : move()`

Etc..

- `from library import turn_right as right`
- `from library import *`
- `from random import * #Python 내장`
- `from reeborg_fr import *`
- `avance() # franch : move()`



random 모듈 응용 : 변태적인(?) 미로 생성



프레임이 기록될 때, 화면은 사실상
정지됨 (40초), 자바스크립트 코드가
훨씬 빠름

리보그와 비주얼 스크립트 언어

리보그 VS 비주얼 스크립트 언어



비교를 하자면..

리보그 VS 비주얼 스크립트 언어



리보그의 파이썬 코드 > 제한적인 블록 코딩

기본 명령어

<http://reeborg.ca/docs/ko/reference/commands.html>

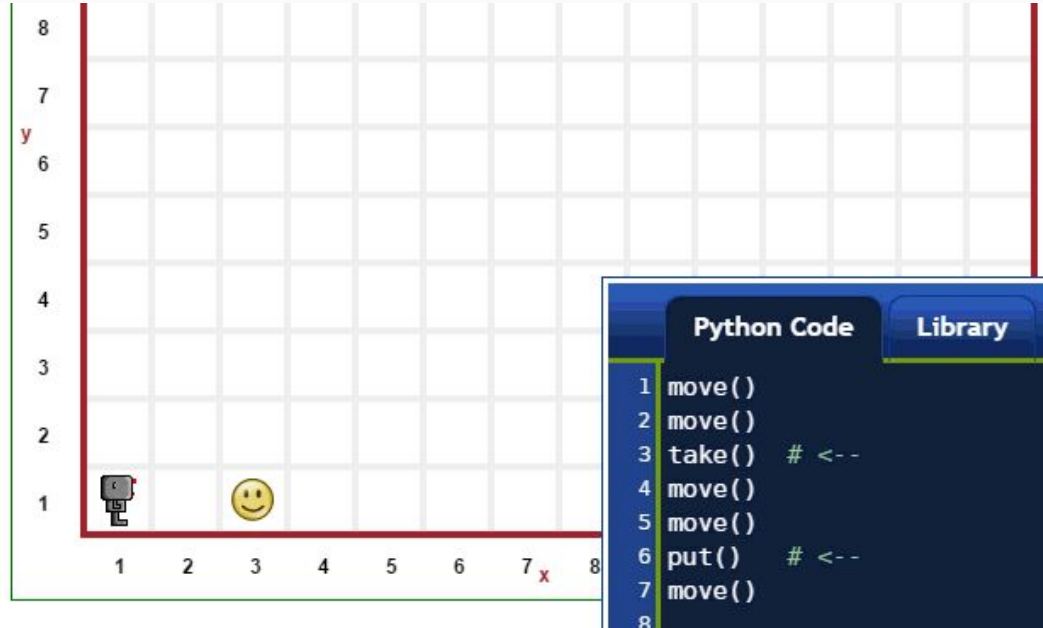
turn_right() ..?

```
def turn_right():  
    turn_left()  
    turn_left()
```

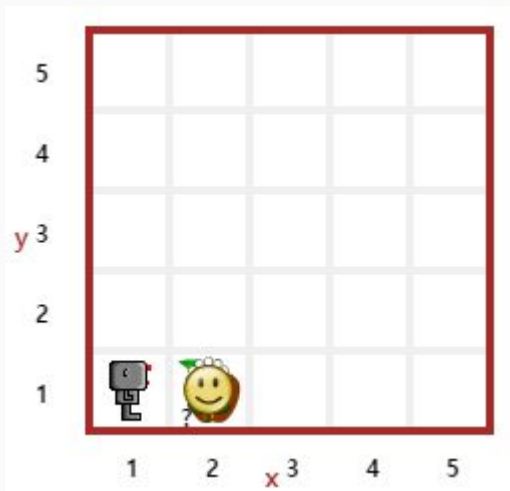
```
turn_left()  
turn_right()
```

* 카렐부터 이어진 관습

take() and put()



여러 객체를 집을려고 한다면 : X



```
move()  
while object_here():  
    take()
```

해결법 : 함수 인자 쓰기

```
while object_here("token"):  
    take("token") # "token": 😊  
move()  
put("token")
```

repeat n:

```
repeat n:  # 명령을 n 번 반복
```

```
    turn_left()
```

```
def turn_around():
```

```
    repeat 2:
```

```
        turn_left()  # 뒤돌아 선다.
```

```
def turn_right():
```

```
    repeat 3:
```

```
        turn_left()  #오른쪽으로(?) 돌기
```

for..in 권장

```
def turn_right():  
    for i in range(3):  
        turn_left()
```

```
turn_right() # 오른쪽으로 회전
```

기본 명령어 요약

리보그 제어

- `move()`
- `build_wall()`
- `pause()`
- `take()`
- `put()`
- `done()`
- `turn_left()`

리보그 상태

- `at_goal()`
- `object_here()`
- `right_is_clear()`
- `front_is_clear()`
- `carries_object()`
- `wall_in_front()`
- `wall_on_right()`

기본 명령어 요약

리보그 제어

- `move()`
- `build_wall()`
- `pause()`
- `take()`
- `put()`
- `done()`
- `turn_left()`

리보그 상태

- `at_goal()`
- `object_here`
- `right_is_clear`
- `front_is_clear`
- `carries_object`
- `wall_in_front`
- `wall_on_right`



기본 명령어 요약 : 특수 키보드



만일 타이핑 하는데 어려움을 겪고 계시다면, 리보그 월드의 상단에 “리보그의 키보드” 버튼이 있습니다. 명령어나 파이썬 키워드가 적힌 버튼을 클릭하면 편집기에 입력됩니다.

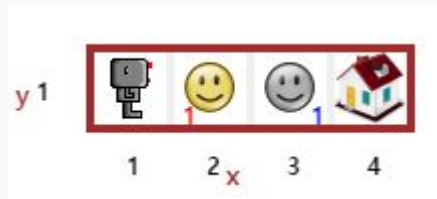
문제 해결 : 토큰 리보그에게 일 맡겨보기

직접 따라 해보면서 하셔도 됩니다.

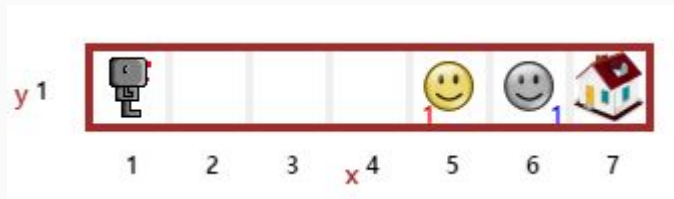
예제 : 토큰 1 ~ 6 풀기

리보그가 해야 하는 일

1. 토큰을 찾을때까지 이동한다.
2. 토큰을 집는다.
3. 다음 칸으로 이동한다.
4. 토큰을 자리에 놓는다.
5. 목적지에도착할때까지 이동한다.
6. 목적지 도착 : `done()`



토큰1



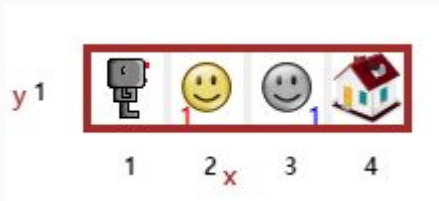
토큰2

NOTICE



토큰1에서 토큰6은 리보그에 내장된 월드 예제입니다.
(* 본 PT에서 쓰이는 월드는 발표하기 편하게 수정하였습니다.)

토콘1 : 무식하게 풀어보기



move() # 이동

take() #

집기

move()

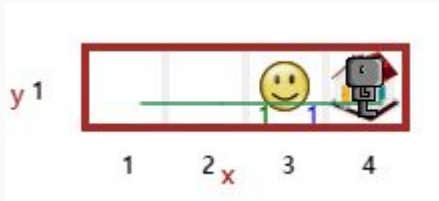
put() #

놓기

move()

done() # 목적지 도착

토큰1 : 무식하게 풀어보기 - 결과



move() # 이동

take() #

집기

move()

put() #

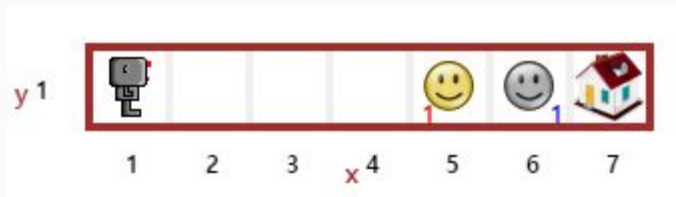
놓기

move()

done() # 목적지 도착

~~실제로 저의 학교 친구들이 러플(리보그-구버전) 수항평가를 저렇게 합니다.~~

토큰2 : 토큰2 세계에 써보기



move() # 이동

take() #

집기

move()

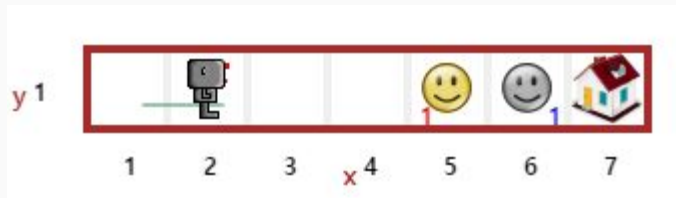
put() #

놓기

move()

done() # 목적지 도착

토큰2 : 토큰2 세계에 써보기



move() # 이동

take() #

집기

move()

put() #

리보그 : 뭔가가 잘못 났어요!

x

ReeborgError

여기서 객체 를 찾을수 없어요!

도착

토큰2 : 조금 고쳐보기(1)

이번에는 `def():` 를 써서 조금 고쳐보겠습니다.

```
def token2():  
    move()  
    if object_here():  
        # 무언가 수행한다.  
        # 다른 무언가 수행한다.  
        # 다시 다른 무언가 수행한다.  
        # 한가지 더 있다.  
    done()  
  
for i in range(42):  
    token2()  
  
# 주석 안의 내용이 될까요..?
```


토큰2 : 조금 고쳐보기(1)

이번에는 `def()`: 를 써서 조금 고쳐보겠습니다.

```
def token2():  
    move()  
    if object_here():  
        take() # 집기  
        move() # 이동  
        put() # 넣기  
        move()  
        done() # 목적지 도착  
  
for i in range(42): # 최대 x좌표에 있는 42 토큰까지 가능  
    token2()
```

답 : 전에 쓴 무식한 코드의 일부를 넣어봤습니다.

토큰2 : 조금 고쳐보기(1) - 결과

이번에는 `def()`: 를 써서 조금 고쳐보겠습니다.

```
def token2():
```



```
    move_until_done()
```

토큰2 : 무식하게 푼 코드를 아예 없애보기(2)

이번에는 고급스럽게(?) `while` 을 써봤습니다.

```
while not at_goal():  
    if object_here():  
        take() # 집기  
        move() # 이동  
        put() # 놓기  
    move()  
done()
```

토큰2 : 무식하게 푼 코드를 아예 없애보기(2) - 결과

이번에는 고급스럽게(?) `while` 을 써봤습니다.

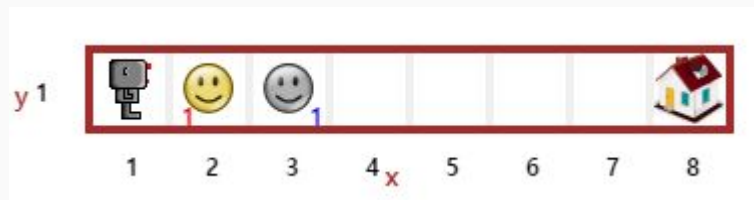


`done()`

토큰3, 4 : 토큰3과 토큰4에 써보기

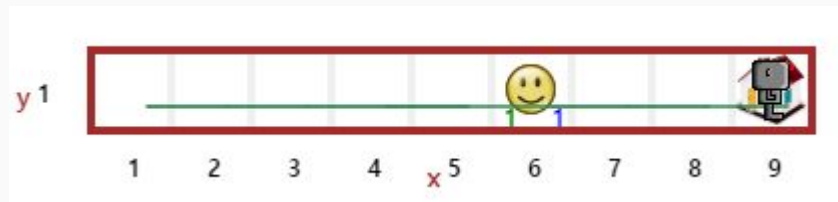


토큰3

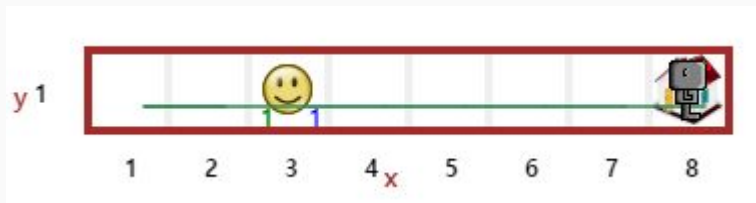


토큰4

토큰3, 4 : 토큰3과 토큰4에 써보기 - 결과

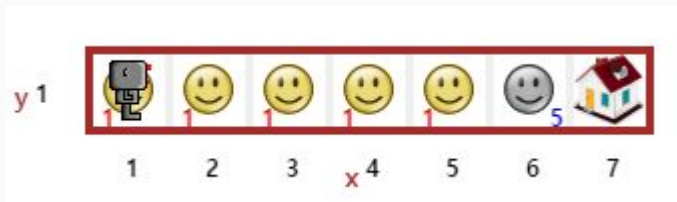


토큰3

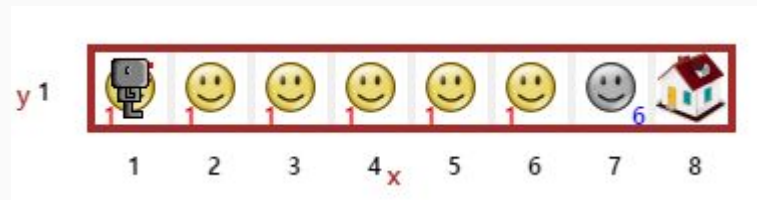


토큰4

토큰5, 6 : 토큰5와 토큰6에도 써보기

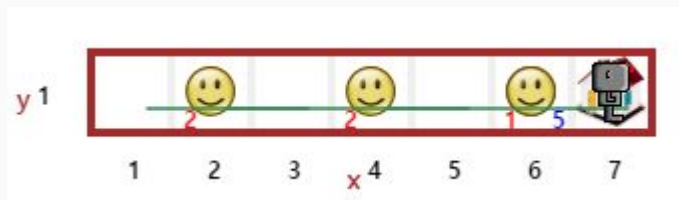


토큰5



토큰6

토큰5, 6 : 토큰5와 토큰6에도 써보기 - 결과



토큰5



토큰6

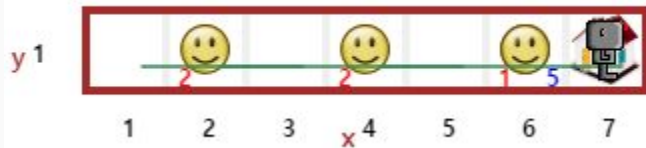
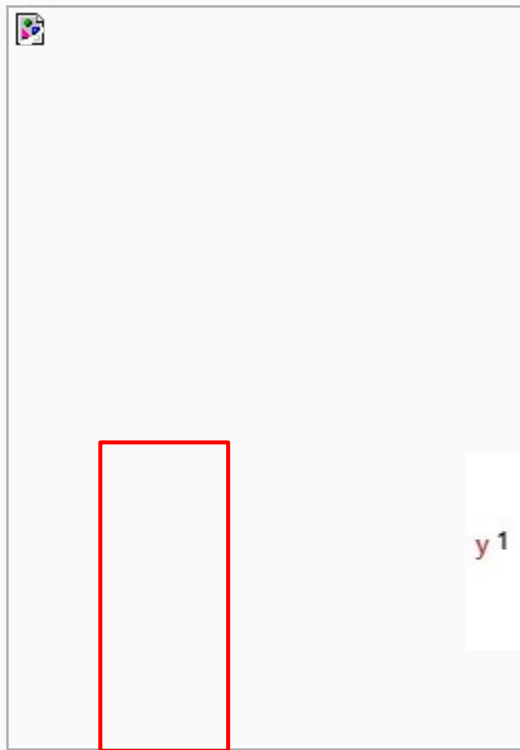
리보그 : 뭔가가 잘못 뒀어요!

리보그는 올바른 x 위치에 있습니다.✓

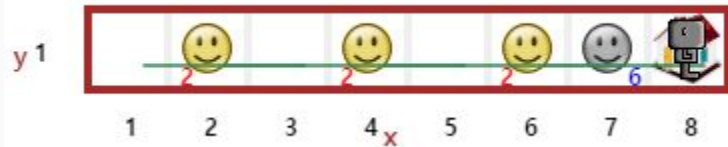
리보그는 올바른 y 위치에 있습니다.✓

하나 이상의 객체가 올바른 위치에 있지 않습니다.✗

토큰5, 6 : 맞게 고쳐보기 - 흐름도



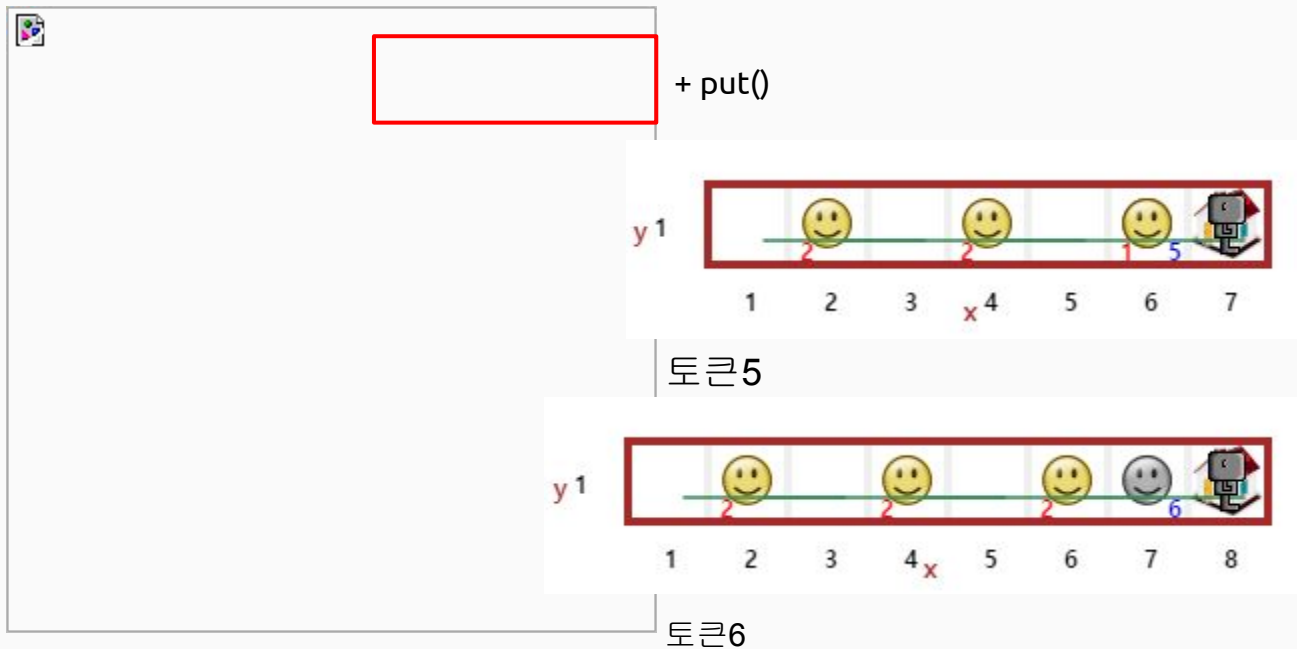
토큰5



토큰6

토큰을 집은 다음 바로 다음칸에 내려놓는걸 확인할 수 있습니다

토큰5, 6 : # 돌아서서 토큰 넣기



#돌아서서 토큰 넣기: `put()`을 `while not at_goal()` 루프 밖에 넣어봅시다.

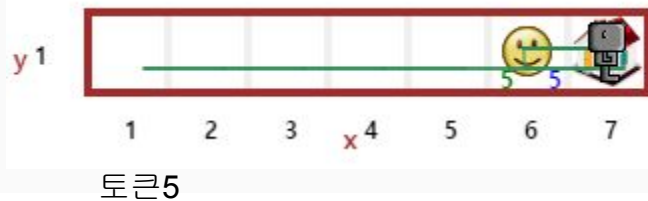
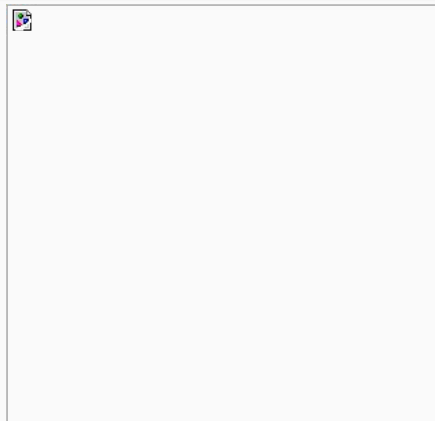
토큰5, 6 : # 돌아서서 토큰 넣기

```
def turn_around(): # 뒤돌아서기  
    turn_left()  
    turn_left()
```

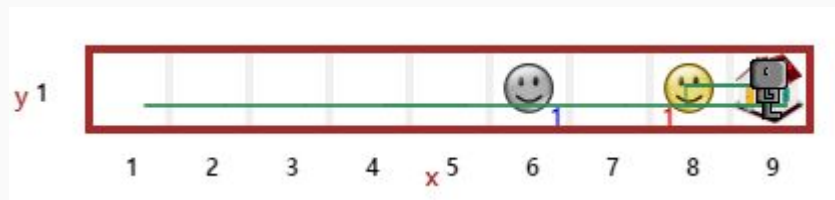
먼저 turn_around() 함수!

토큰5, 6 : # 돌아서서 토큰 넣기 - 결과

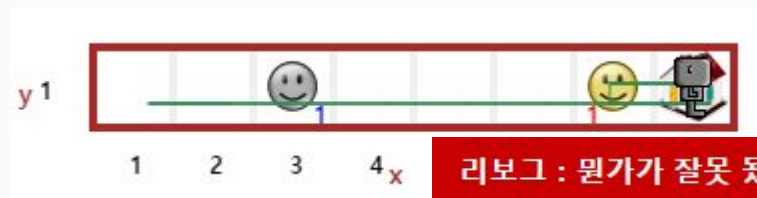
```
while not at_goal():  
    if object_here():  
        take()  
        move()  
    turn_around() # turn_left 2번  
    move()  
while carries_object():  
    put()  
    turn_around()  
    move()  
done()
```



그러면 토큰3와 토큰4에서는? - 결과



토큰3



토큰4

리보그 : 뭔가가 잘못 됐어요!

리보그는 올바른 x 위치에 있습니다.✓

리보그는 올바른 y 위치에 있습니다.✓

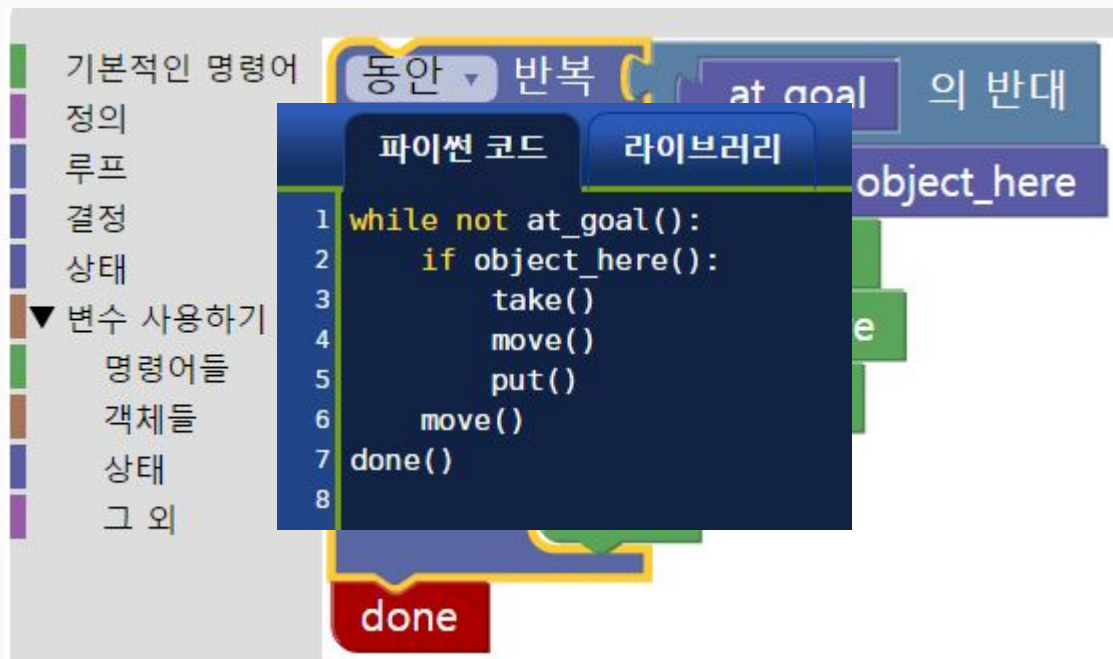
하나 이상의 객체가 올바른 위치에 있지 않습니다.✗

번외: 비주얼 스크립트 언어로 리보그 다뤄보기



비주얼 프로그래밍 에디터인 Blockly를 제공합니다.

번외: 비주얼 스크립트 언어로 리보그 다뤄보기



물론 파이썬 코드로 자동 변환 가능.

리보그로 객체지향 다뤄보기

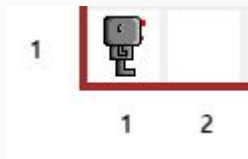
<http://reeborg.ca/docs/ko/python/oop.html>

NOTICE

이 발표에서는 객체지향 프로그래밍을 가르치는 것이 아니라,
객체지향 프로그래밍을 사용해서 어떤 작업을 수행할 수 있는지 소개합니다.

아주 간단한 첫번째 객체지향 프로그래밍

```
reeborg = UsedRobot()  
# UsedRobot : 리보그는 기름이 새고  
# 여러곳이 망가진 중고 로봇
```

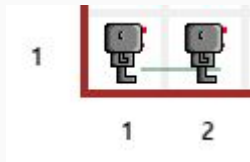


표준위치($x=1, y=1$) 로봇 생성.

*표준위치에 기존의 로봇과 새로 생성한 로봇이 같이 있습니다.

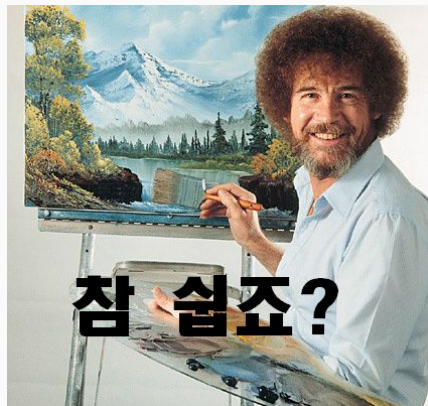
ex) 아주 간단한 객체지향 프로그래밍

```
reeborg = UsedRobot()  
reeborg.move() # 어때요, 참 쉽죠?
```



어때요, 정말 참 쉽죠?

*x좌표가 2인곳이 생성된 로봇입니다.



세번째 : 신형 로봇

```
class BetterRobot(UsedRobot):
```

```
    def turn_right(self):  
        self.turn_left()  
        self.turn_left()  
        self.turn_left()
```

```
reeborg =  
BetterRobot()  
reeborg.turn_right()
```

class 객체 안에 turn_right() 내장 가능.

리보그 내부 탐색하기

```
r =  
UsedRobot()  
dir_js(r)
```

dir_js()는 리보그에서 정의된 자바스크립트 함수

리보그 내부 탐색하기

```
r =  
UsedRobot()  
dir_js(r)  
  
__class__  
body
```

*__class__ 는 인스턴스 객체가 자신을 생성한 클래스 객체를 참조 가능한 내장 속성

리보그 내부 탐색하기

```
dir_js(r.body)
x
y
objects
_orientation
_is_leaky
_prev_x
_prev_y
_prev_orientation # 속성들
```

JavaScript

```
body
at_goal()
at_goal_orientation()
build_wall()
front_is_clear()
has_token()
dir_js(r) → is_facing_north()
move()
put()
token_here()
right_is_clear()
object_here()
take()
turn_left()
```


JavaScript

```
dir_js(r.body) → x  
                  y  
                  objects  
                  _orientation  
                  _is_leaky  
                  _prev_x  
                  _prev_y  
                  _prev_orientation
```

`_orientation:is_facing_south()`

```
class SouthRobot(UsedRobot):  
  
    def is_facing_south(self):  
        facing_south = self.body._orientation == RUR.SOUTH  
        return facing_south  
  
reeborg = SouthRobot()  
while not reeborg.is_facing_south():  
    reeborg.turn_left()
```

요약 : 특징

요약 : 특징

1. 프로그래밍을 처음 접하는 학생들이 정규 타이핑 방식으로 어렵지 않게 프로그래밍을 배울 수 있음
2. 리보그에서 배운 코딩 방식을 다양한 파이썬 도구에서도 그대로 사용할 수 있음
3. 입문자를 대상으로 객체 지향을 보다 쉽게 배울 수 있어서 고급 프로그래밍 개념을 배울 수 있음

<http://reeborg.ca/docs/ko>

or

<http://reeborg.ca/docs/en>

발표에서 다룬건 기본적인 내용입니다. 더 자세한 정보는 문서에..

<https://github.com/aroberge/reeborg>

and


<https://github.com/aroberge/reeborg-docs>

이 프로젝트에 관심있으신 분은 기여나 스텔..

마지막으로..

I proposed speakers about Reeborg in Pycon APAC


📌 ⌚ 🗑️ ✓ ⋮



나

Hi, Recently I proposed speakers about Reeborg in Pycon APAC(korea) - Abou...

6월 4일 ▾



보낸사람: **Andre Roberge**, 받는사람: 나 ↕

Hi,

Yes, absolutely. You do not need to ask my permission.

Good luck with your presentation.

André

...

6월 4일 ⋮

thanks to André Roberge.

참고자료

- PyCon 2015 Lightning Talks - Reeborg karel on steroids : <https://youtu.be/byClBydNI1A?t=10m9s>
- 리보그와 파이썬 배우기! : <http://reeborg.ca/docs/ko/> AND <http://reeborg.ca/docs/en/>
- xwMOOC 프로그래밍과 문제 해결 : <http://reeborg.xwmooc.org>
- ..오X워치 머그컵 : <http://www.ebay.co.uk/itm/Overwatch-Mug-gamer-blizzard-pc-uk-gift-/221934137171>

End

QnA

모든 것을 물어라. 무언가를 배워라. 아무 대답도 하지 말라.

"question everything learn something answer nothing"

— Euripides(고대 그리스 시인)