Welcome!

2018 Spring CS101 Introduction to Programming
Week 2

Programming with robots
(Function, For loops, Conditional expressions, While loops)
What is elice?

Online programming education platform

In CS101, we will use them for

- Programming tasks in lab
- Homework assignments

No need to spend a lot of time setting up your development environment for CS101!
Focus fully on your ideas & writing your code.
What you need to know before signing up

1. Your **real name**
2. Your **KAIST email address** ( ...@kaist.ac.kr)
3. Your **student ID**
Signing up (1) - kaist.elice.io
Signing up (2)

SIGN UP

Create one elice account to access all elice platforms :)

- Sign up with Email
- Sign up with Facebook
- Sign up with Google
Signing up (3)

USE YOUR REAL NAME !!
(Eng or Kor)
Terms & Agreement

- I agree to the Terms of Service, Privacy Policy, Selective Privacy Policy (optional), and to receiving promotional emails (optional).

Complete signup

- Terms of Service *
- Privacy Policy *
- Selective Privacy Policy (Optional)
- Receive promotional emails (Optional)
Signing up (5)
Signing up (6)

This organization needs an additional verification.

You have an Elice account, but the verification is not completed yet.

Verify ➔
Signing up (7)

MUST USE YOUR KAIST EMAIL!!

...@kaist.ac.kr
Confirmation email is sent.

Confirm your email and click the verification link.
Signing up (9) - mail.kaist.ac.kr
Signing up (10)

MUST USE YOUR STUDENT ID!!
CS101 Introduction to Programming

Enroll
Ends at 18:35

Introduction

Course structure

Starting from spring 2010, CS101 uses the programming language Python, a language that was designed to be easy to read and write.
cs101_2018_kaist_?!
Signing up DONE!
Honor Code
Honor Code Agreement
Honor Code Agreement

You should agree with Honor Code. Any cheating is strictly prohibited in CS101.
Honor Code Agreement

SECTION 1 OF 1
HONOR CODE AGREEMENT

QUESTION 1
You agree with the honor code by writing your name here.

Jongmin Lee

QUESTION 2
Enter today’s date in YYYY-MM-DD form.

2018-03-05

Submit

Read the whole text carefully.
Write down your name & date if you agree.
If you do NOT agree, you cannot take cs101 :(
Honor Code Agreement

Completed!
questions?
Useful resources
You’re alone in your dorm room trying to study...

and stuck...

How do I use random function?
python 3 random example
9.6. random — Generate pseudo-random numbers — Python 3.6.2 ...
https://docs.python.org/3/library/random.html
For integers, there is uniform selection from a range. ... Almost all module functions depend on the basic function random(), which ... 1, January pp.3–30 1998.

random — Generate pseudo-random numbers — Python v3.0.1 ...
https://docs.python.org/3.0/library/random.html
Almost all module functions depend on the basic function random(), which .... 3, 2, 5, 6, 4, 1] >>>
random.sample([[1, 2, 3, 4, 5], 3]) # Choose 3 elements [4, 1, 5].

9.6. random — Generate pseudo-random numbers — Python 2.7.13 ...
https://docs.python.org/2/library/random.html
Almost all module functions depend on the basic function random(), which ..... 3, 2, 5, 6, 4, 1] >>>
random.sample([[1, 2, 3, 4, 5], 3]) # Choose 3 elements [4, 1, 5].

How to use the Random Module in Python - Pythonforbeginners.com
www.pythonforbeginners.com/random/how-to-use-the-random-module-in-python ▼
Dec 24, 2012 - The random module provides access to functions that support many ... import random for i in range(3): print random.randint(0, 101, 5) ...

python - Generate random integers between 0 and 9 - Stack Overflow
Oct 22, 2010 - More info: https://docs.python.org/3/library/random.html#random.randint ... Generates 10 pseudo random integers in range 0 to 9 inclusive.
9.6. random — Generate pseudo-random numbers — Python 3.6.2 ...
https://docs.python.org/3/library/random.html
For integers, there is uniform selection from a range. ... Almost all module functions depend on the basic function random(), which ... 1, January pp.3–30 1998.

random — Generate pseudo-random numbers — Python v3.0.1 ...
https://docs.python.org/3.0/library/random.html
Almost all module functions depend on the basic function random(), which ... 3, 2, 5, 6, 4, 1] >>>
random.sample([1, 2, 3, 4, 5], 3) # Choose 3 elements [4, 1, 5].

9.6. random — Generate pseudo-random numbers — Python 2.7.13 ...
https://docs.python.org/2/library/random.html
Almost all module functions depend on the basic function random(), which ..... 3, 2, 5, 6, 4, 1] >>>
random.sample([1, 2, 3, 4, 5], 3) # Choose 3 elements [4, 1, 5].

How to use the Random Module in Python - Pythonforbeginners.com
www.pythonforbeginners.com/random/how-to-use-the-random-module-in-python
Dec 24, 2012 - The random module provides access to functions that support many ... import random for i in range(3): print random.randrange(0, 101, 5) ...

python - Generate random integers between 0 and 9 - Stack Overflow
Oct 22, 2010 - More info: https://docs.python.org/3/library/random.html#random.randint ... Generates 10 pseudo random integers in range 0 to 9 inclusive.
These two websites are usually good signs!

http://docs.python.org/3/...

http://stackoverflow.com/...
Useful resources

http://docs.python.org/3/
“the official python 3 documents”

http://stackoverflow.com/
“most beloved Q&A website in computer science”
Some Basics for Today's Tasks
Today's Tasks

- How to define function
- How to use for-loop
- How to use conditional expressions
- How to use while-loop
Robot World (Read the robot notes!)
Some Basics of `cs1robot` library

```python
from cs1robots import *
create_world()
hubo = Robot()
hubo.set_trace('blue')
```
Some Basics of *cs1robot* library

```python
from cs1robots import *
create_world()
hubo = Robot()
hubo.set_trace('blue')
```
Some Basics of *cs1robot* library

from cs1robots import * ← Import cs101 robot library!
create_world() ← Create a robot world (defined in cs1robots library)
hubo = Robot() ← Create a robot named 'hubo'
hubo.set_trace('blue') ← Turn on a trace for our robot (hubo) with 'blue' color
from cs1robots import *

create_world()
hubo = Robot(beepers=10)  ← Create a robot with 10 beepers
hubo.set_trace('blue')    (beeper=?? is optional)
Some Basics of \texttt{cs1robot} library

```python
from cs1robots import *
create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
```
Some Basics of cs1robot library

```python
from cs1robots import *

create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
hubo.move()
```
Some Basics of `cs1robot` library

```python
from cs1robots import *

create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
hubo.move()
hubo.turn_left()
```
Some Basics of *cs1robot* library

```python
from cs1robots import *

create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
hubo.move()
hubo.turn_left()
hubo.move()
```
Some Basics of *cs1robot* library

```python
from cs1robots import *

create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
hubo.move()
hubo.turn_left()
hubo.move()
hubo.drop_beeper()
```
Some Basics of *cs1robot* library

```
from cs1robots import *
create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')

hubo.drop_beeper()
hubo.move()
hubo.turn_left()
hubo.move()
hubo.drop_beeper()
hubo.pick_beeper()
```

✓ Before picking up a beeper, hubo should be on a beeper!
Some Basics of cs1robot library

```python
from cs1robots import *
create_world()
hubo = Robot(beepers=10)
hubo.set_trace('blue')
hubo.drop_beeper()
hubo.move()
hubo.turn_left()
hubo.move()
hubo.drop_beeper()
hubo.pick_beeper()
hubo.pick_beeper()
```

Error!
from cs1robots import *

load_world('worlds/hurdles1.wld')
hubo = Robot()
hubo.set_trace('blue')

# Write your own code below!

* Instead of using create_world, we can use load_world
How to Define Function

from cs1robots import *

create_world()
hubo = Robot()
hubo.set_trace('blue')

# Write your own code below!

def turn_right():
    hubo.turn_left()
    hubo.turn_left()
    hubo.turn_left()
    hubo.turn_left()
    hubo.move()
    turn_right()
    hubo.move()
    turn_right()
    hubo.move()

Do NOT repeat yourself!
If you have code that will be repeated several times, define that code as a function! Give your code a descriptive name.
→ Much simpler & human-readable code
from cs1robots import *

create_world()
hubo = Robot()
hubo.set_trace('blue')

# Write your own code below!

def turn_right():
    for i in range(3):
        hubo.turn_left()

hubo.turn_left()
hubo.move()
turn_right()
hubo.move()
turn_right()
hubo.move()

Do NOT repeat yourself!
We can repeat the same code using for-loop!
Week 2
Today's Tasks 1
Task 1: ZigZag

- Create a default world.
- Add a robot.
- Make the robot visit the entire world in a zigzag fashion.
Task 2: Hurdle

- Load the world named ‘hurdles1.wld’.
- Add a robot.
- Make the robot jump all hurdles and pick up the beeper.
Task 3: Newspaper delivery

- Load the world named ‘newspaper.wld’.
- Add a robot with one beeper.
- Make the robot deliver newspapers and return to his starting point.
Task 4: Harvest

- Load the world named ‘harvest1.wld’.
- Add a robot.
- Make the robot harvest all the carrots (beepers).
Task 5: Harvest again

- Load the world named ‘harvest2.wld’.
- Add a robot.
- Make the robot harvest all the carrots with the **shortest path** possible.
If-statement & While-statement
New functions

- Create a custom world
  
  ```
  create_world(streets = 7, avenues = 9)
  ```

  ✔ ‘streets’ for the number of rows
  ✔ ‘avenues’ for the number of columns
New functions

- Can check if there is a beeper

```python
hubo.on_beeper()
```

```
hubo.on_beeper() == True
```

```
hubo.on_beeper() == False
```
New functions

- Can check if there is a wall on each of the three sides

```python
hubo.front_is_clear() == True
hubo.left_is_clear() == False
hubo.right_is_clear() == True
```
If statements sequentially checks the conditionals

```python
if conditional_expression_1:
    works to do when `conditional_expression_1` evaluates to `True`

elif conditional_expression_2:
    works to do when `conditional_expression_1` evaluates to `False` &
    `conditional_expression_2` evaluates to `True`

elif conditional_expression_3:
    works to do when `conditional_expression_1` evaluates to `False` &
    `conditional_expression_2` evaluates to `False` &
    `conditional_expression_3` evaluates to `True`

... else:
    works to do when all the above conditions are `False`
```
If statements - Example

```python
score = 50
if score < 60:
    print('You got F grade')
elif score < 70:
    print('You got C grade')
elif score < 80:
    print('You got B grade')
else:
    print('You got A grade')
```

Guess what will be output
- when score = 55
- when score = 65
- when score = 70
- when score = 85
respectively?
While loops

- **while** statement loops until the **conditional** evaluates to true

```python
while conditional_expression:
    works to do while conditional_expression evaluates to True
```

- Example

```python
n = 0
while n < 5:
    print(n)
    n = n + 1
```
Week 2
Today’s Tasks
Tasks for Today!

- Read sections 10~13 in the robot notes

Five not so simple tasks
- Harvest More (page 6)
- Plant (page 7)
- Smart Hurdles (page 8 & 9)
- Harvest Even More (page 10)
- Smart ZigZag (page 10)

- When you have completed all the tasks, let a TA mark you off
Task 1 | Conditionals – Harvest More

- Modify your program from the ‘Harvest Again’ task (Week 01) so that it works for `harvest3.wld`

```python
load_world("worlds/harvest3.wld")
```

Before

![Before Image](image1)

After

![After Image](image2)
Task 2 | Conditionals – Plant

- Write a program so that Hubo plants beepers in empty spots
- The finished screen should look like “harvest1.wld”
  ```python
  load_world("worlds/harvest3.wld")
  ```
Task 3 | Conditionals – Smart Hurdles

- Write `jump_one_hurdle()` in section 11
  - `move_jump_or_finish()` should be able to handle all three maps, “hurdles1.wld”, “hurdles2.wld” and “hurdles3.wld”. Check it yourself.

- Write a new program (similar to Hurdles3 in section 11) that uses a while loop. DO NOT USE a for-loop of fixed length
  - It should also work for all three hurdles
Task 4 | While loop – Harvest Even More

- Modify Harvest More task
  - It should work even when there are more than one beeper on a spot (“harvest4.wld”)
  - It should also work for the previous worlds (“harvest1.wld” and “harvest3.wld”)

Have to harvest all carrots on one place
Task 5 | While loop – Smart ZigZag

- Rewrite ZigZag program so that the robot can visit every spot in an empty world of any size in zigzag fashion
  - It should work for even and odd numbers of streets and avenues

m (streets, number of rows)

n (avenues, # of columns)

m and n can be any integer except for m=1 or n=1
questions?