

# Welcome !



2018 Spring CS101 Introduction to Programming



# Week 2



Programming with robots  
(Function, For loops, Conditional expressions, While loops)

**Elice**

# What is elice?

Online programming education platform

In CS101, we will use them for

- Programming tasks in lab
- Homework assignments

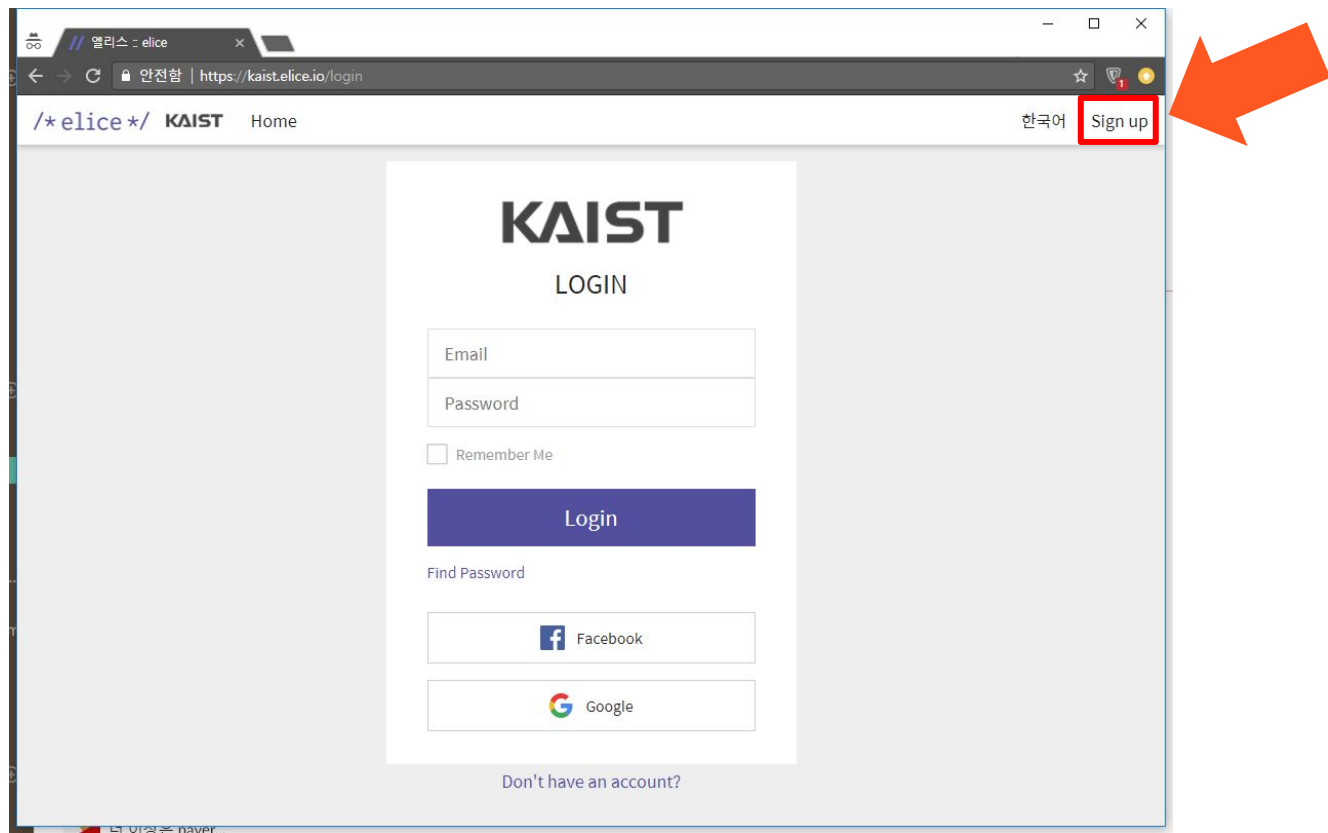


**No need to spend a lot of time setting up your development environment for CS101!**  
**Focus fully on your ideas & writing your code.**

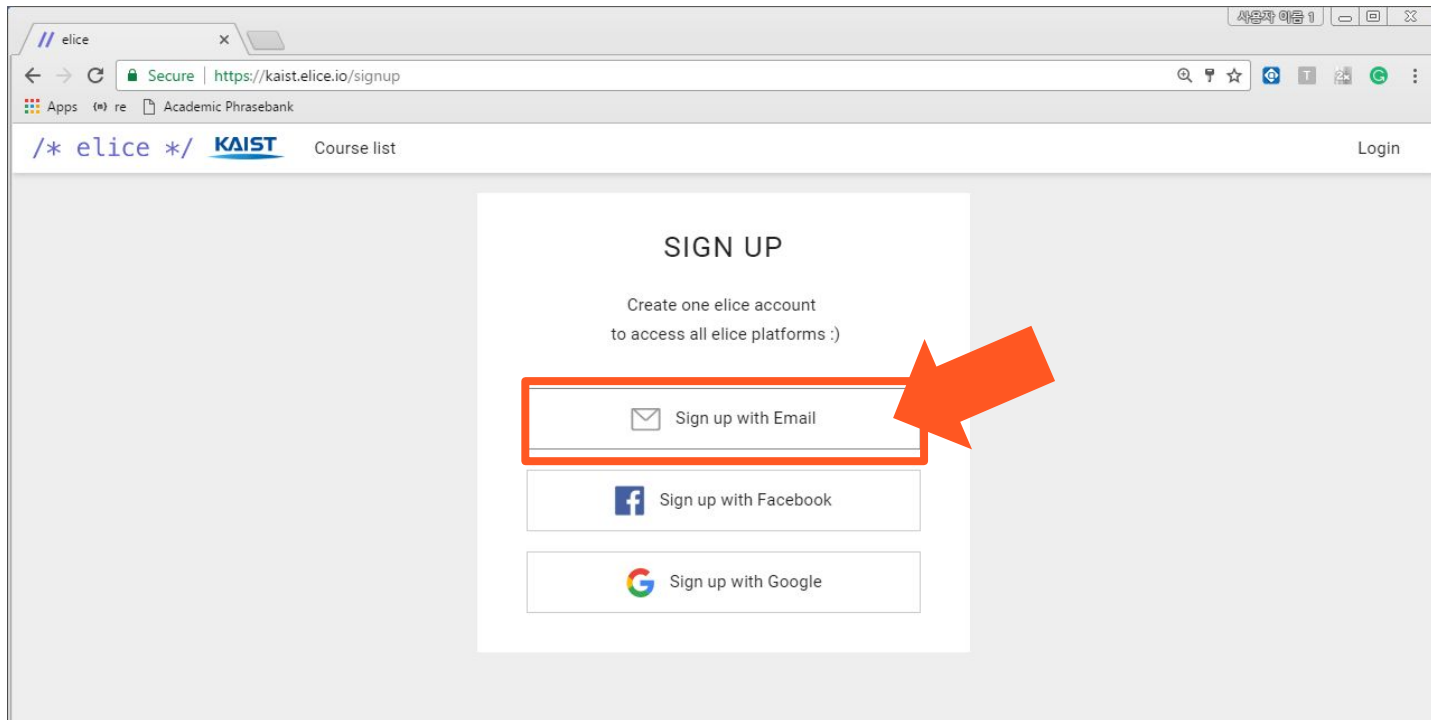
# What you need to know before signing up

1. Your real name
2. Your KAIST email address ( ...@kaist.ac.kr)
3. Your student ID

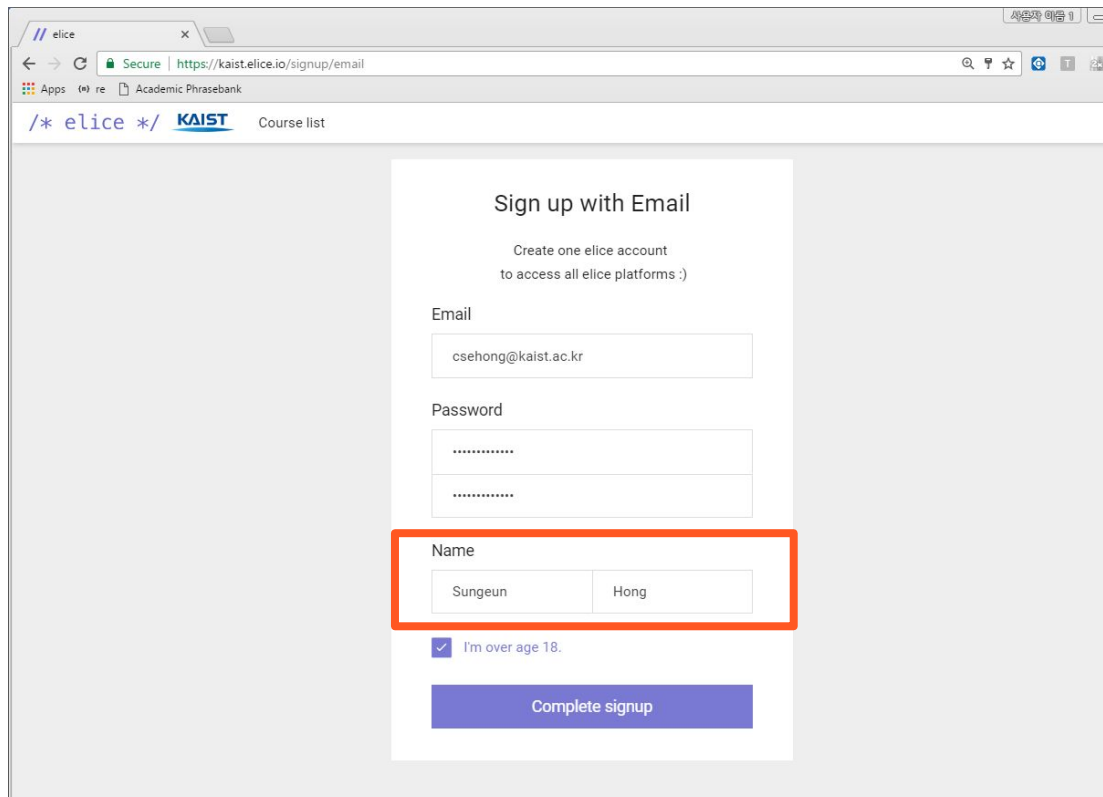
# Signing up (1) - kaist.elice.io



# Signing up (2)



# Signing up (3)



The screenshot shows a web browser window with the URL <https://kaist.elice.io/signup/email>. The page title is "/\* elice \*/ KAIST Course list". The main content is a "Sign up with Email" form. The form includes fields for "Email" (containing "csehong@kaist.ac.kr"), "Password" (with two masked input fields), and "Name" (with two input fields containing "Sungeun" and "Hong"). A checkbox labeled "I'm over age 18." is checked. A blue button labeled "Complete signup" is at the bottom. The "Name" section is highlighted with a red rectangle.

Sign up with Email

Create one elice account  
to access all elice platforms :)

Email

csehong@kaist.ac.kr

Password

.....

.....

Name

Sungeun Hong

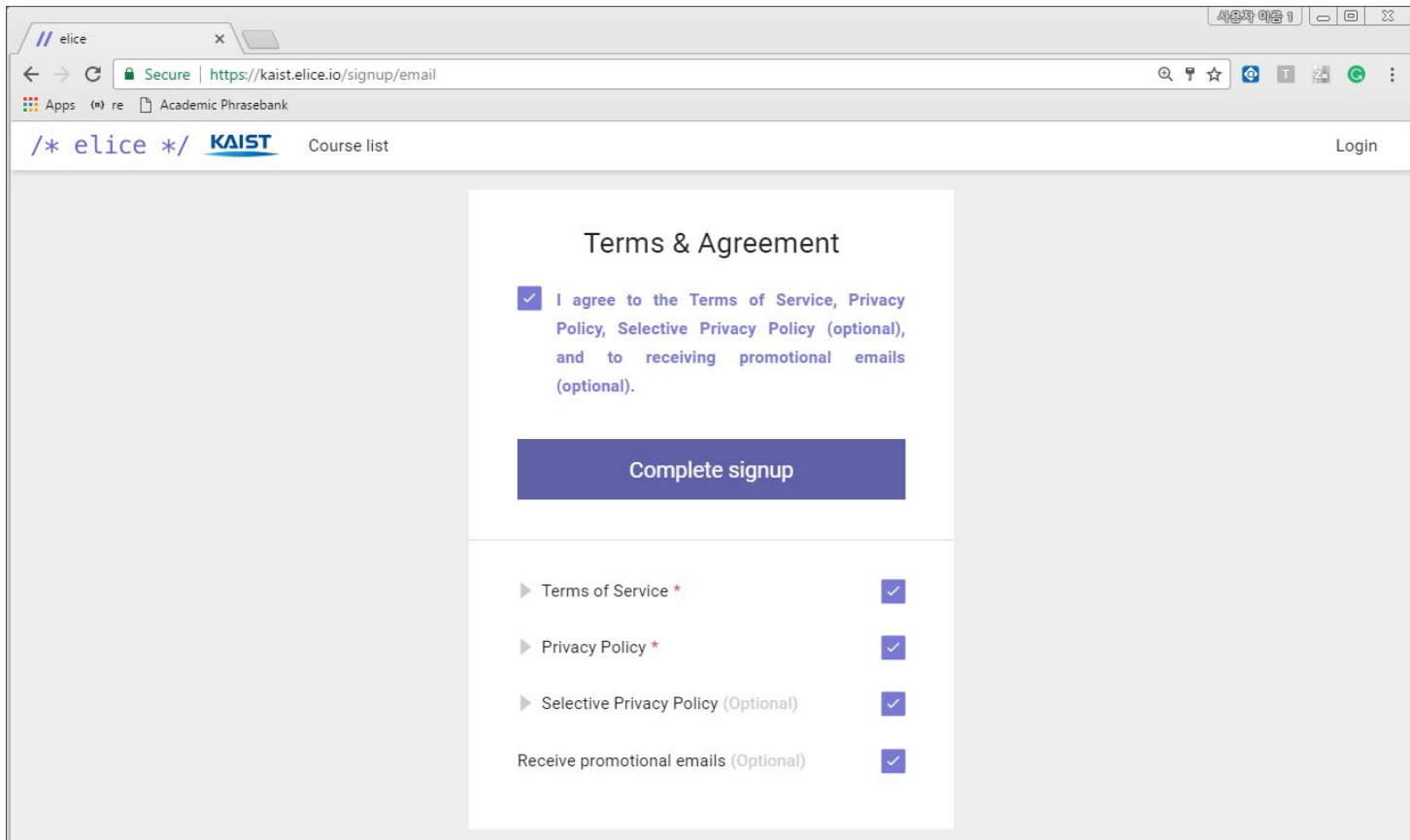
☒ I'm over age 18.

Complete signup

USE  
YOUR  
REAL  
NAME !!  
(Eng or Kor)



# Signing up (4)



The screenshot shows a web browser window with the URL <https://kaist.elice.io/signup/email>. The page header includes the elice logo, KAIST logo, and a "Course list" link. A "Login" link is in the top right corner. The main content area is titled "Terms & Agreement" and contains a checkbox that is checked, indicating agreement to the Terms of Service, Privacy Policy, Selective Privacy Policy (optional), and promotional emails. Below this is a "Complete signup" button. At the bottom, there is a list of terms with checkboxes, all of which are checked.

/\* elice \*/ **KAIST** Course list Login

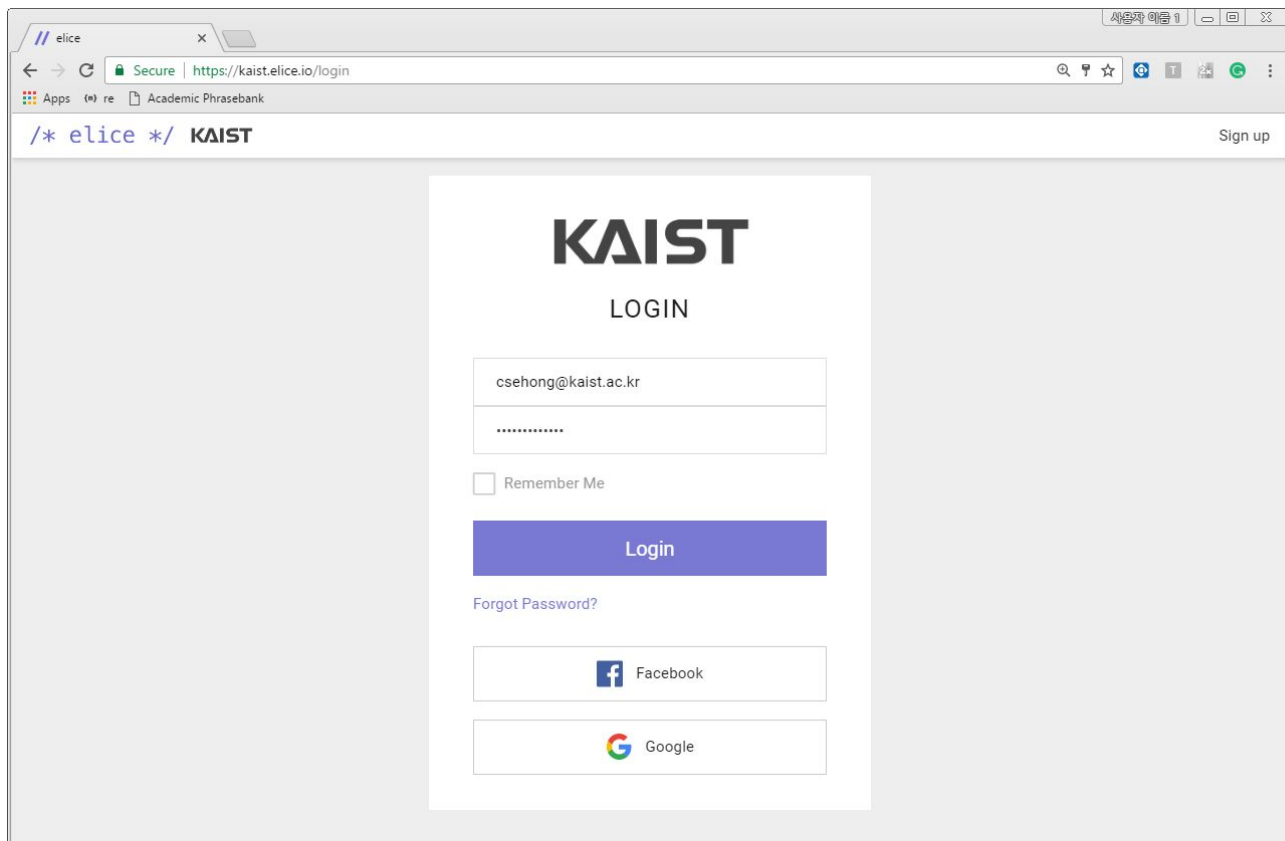
## Terms & Agreement

☒ 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)



The image shows a web browser window with the address bar displaying "Secure | https://kaist.elice.io/login". The browser's address bar also shows "elice" and "x" in the tabs. The page header includes the text "/\* elice \*/ KAIST" on the left and "Sign up" on the right. The main content area features a white box with the "KAIST" logo and the word "LOGIN" below it. There are two input fields for email and password, with the email field containing "csehong@kaist.ac.kr". Below the password field is a checkbox labeled "Remember Me". A blue "Login" button is positioned below the checkbox. Below the button is a link that says "Forgot Password?". At the bottom of the white box, there are two buttons for social login: "Facebook" and "Google".

elice x

Secure | https://kaist.elice.io/login

Apps (e) re Academic Phrasebank

/\* elice \*/ KAIST Sign up

# KAIST

## LOGIN


csehong@kaist.ac.kr


\*\*\*\*\*

☐ Remember Me

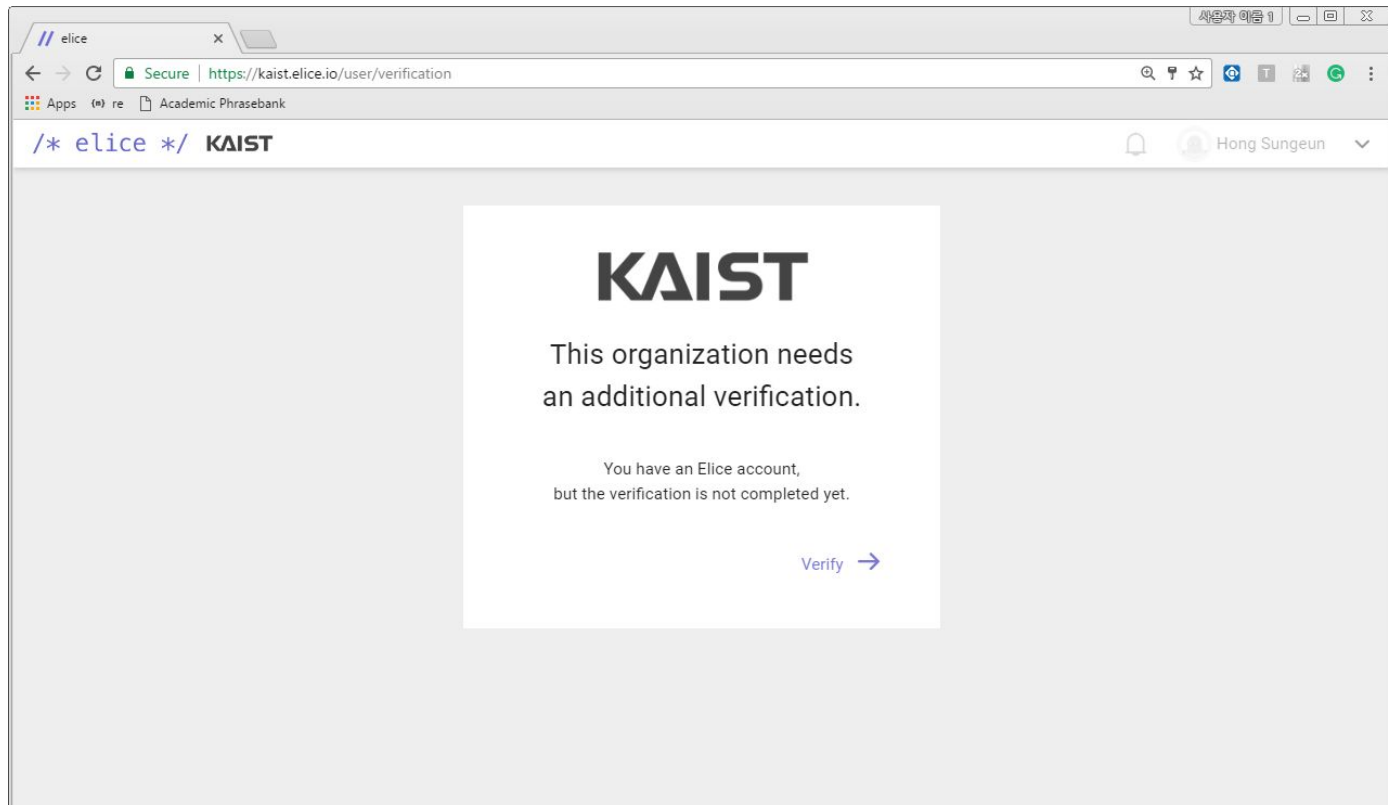
Login

[Forgot Password?](#)

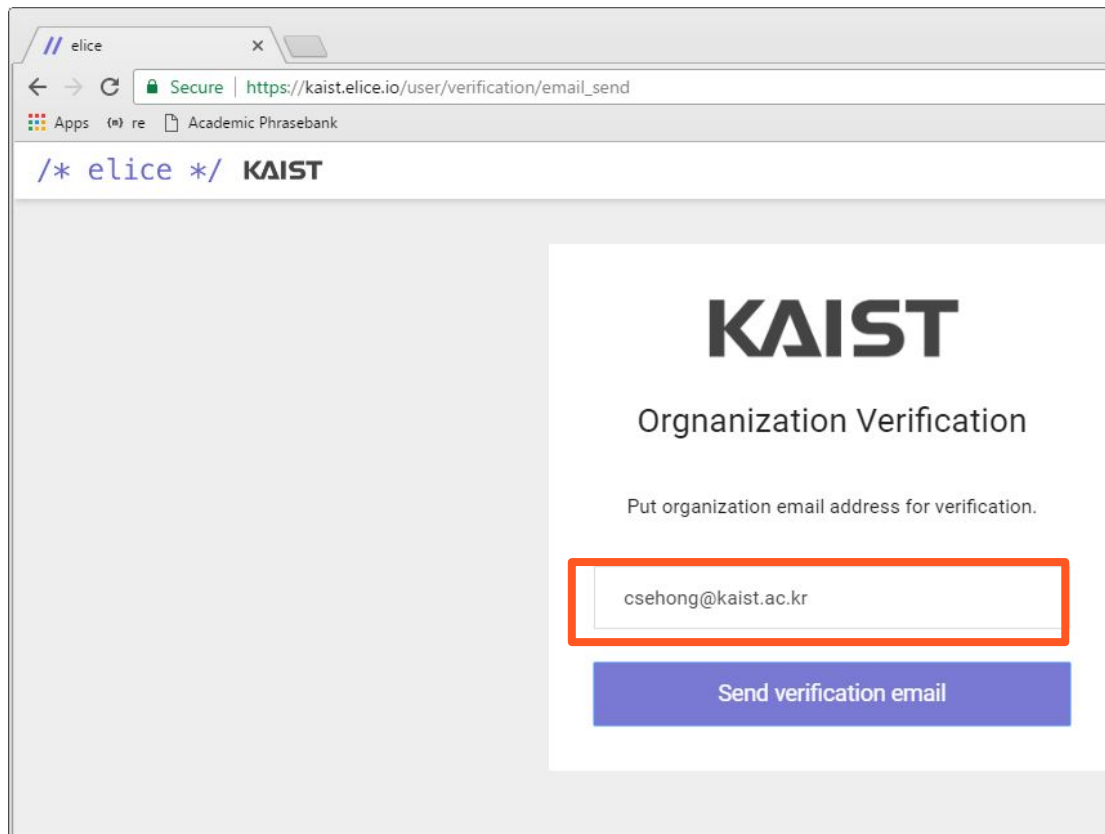
 Facebook

 Google

# Signing up (6)



# Signing up (7)

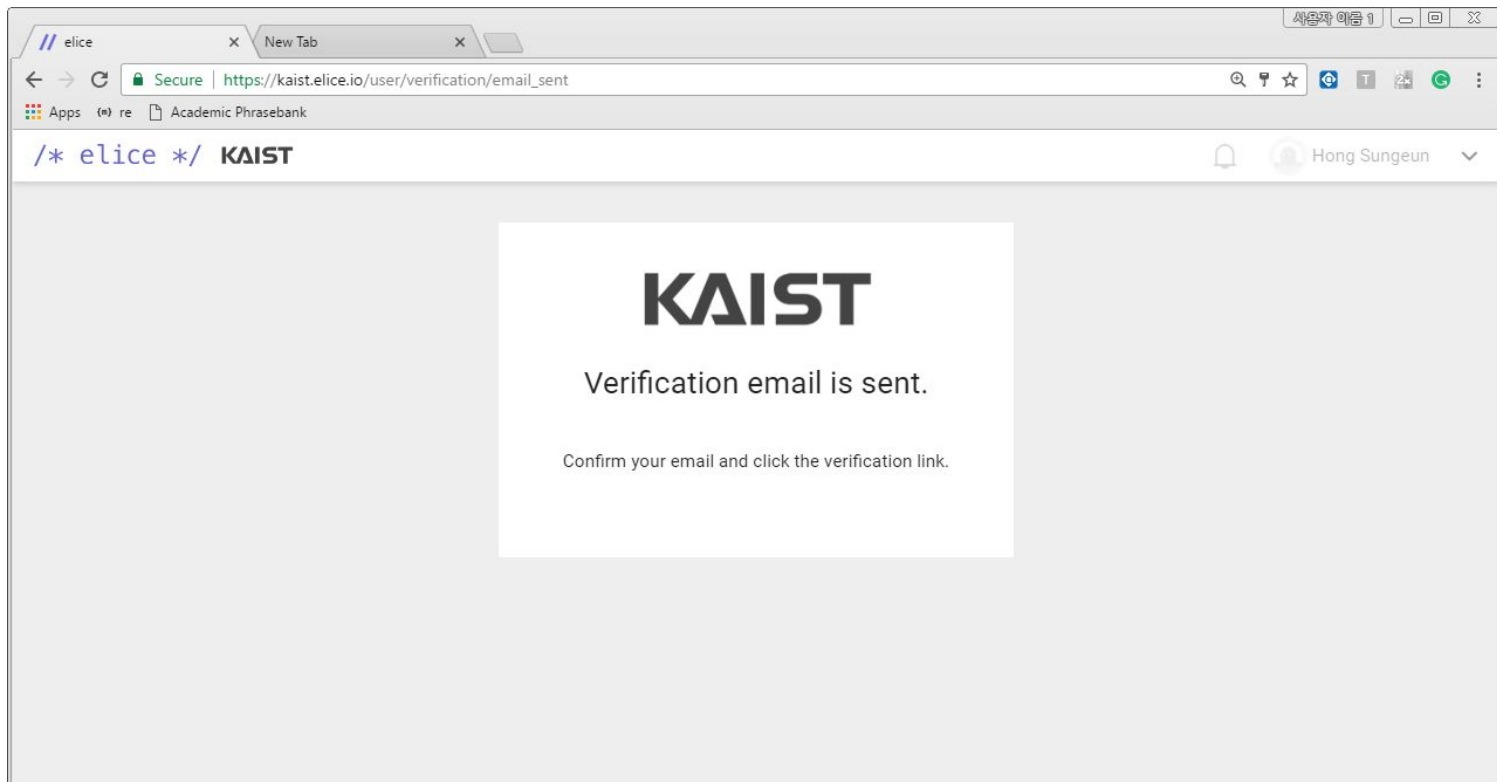


The screenshot shows a web browser window with the address bar displaying `https://kaist.elice.io/user/verification/email_send`. The page header includes the elice logo and the KAIST name. The main content area features the KAIST logo, the title "Organization Verification", and the instruction "Put organization email address for verification.". Below this, a text input field contains the email address `csehong@kaist.ac.kr`, which is highlighted with a red rectangular border. A blue button labeled "Send verification email" is positioned at the bottom of the form.

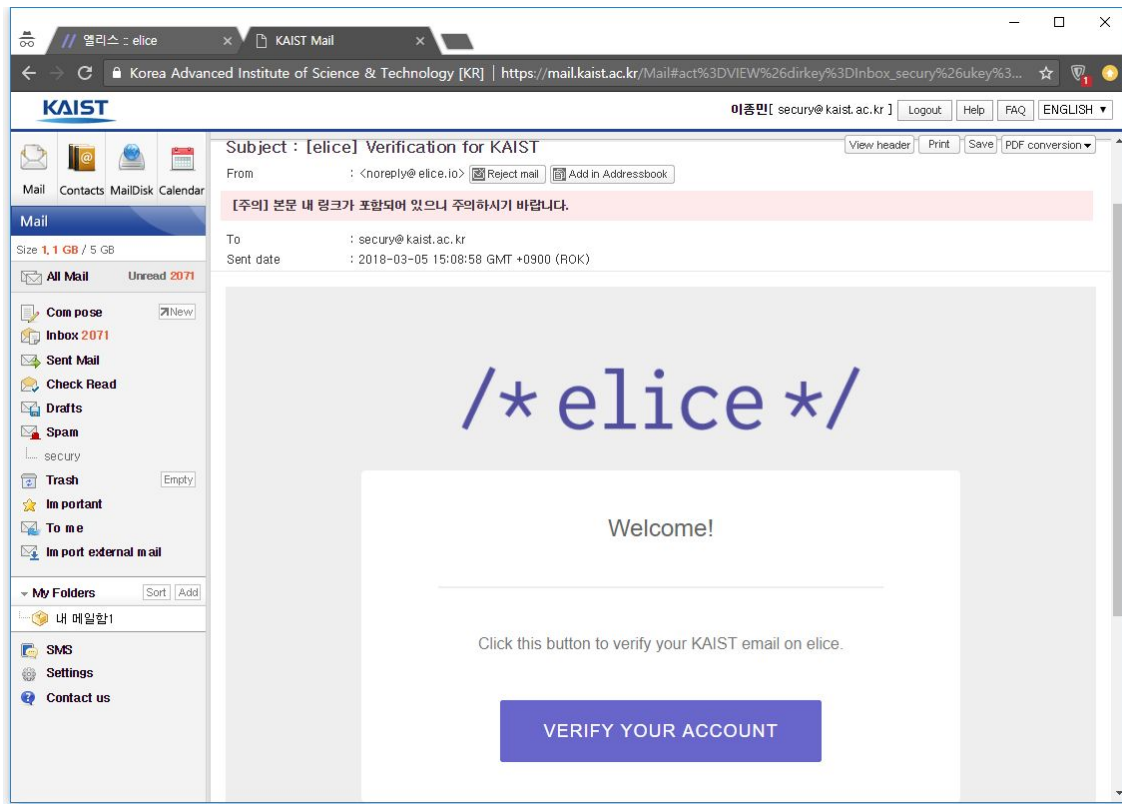
**MUST USE  
YOUR KAIST  
EMAIL !!**

**... @kaist.ac.kr**

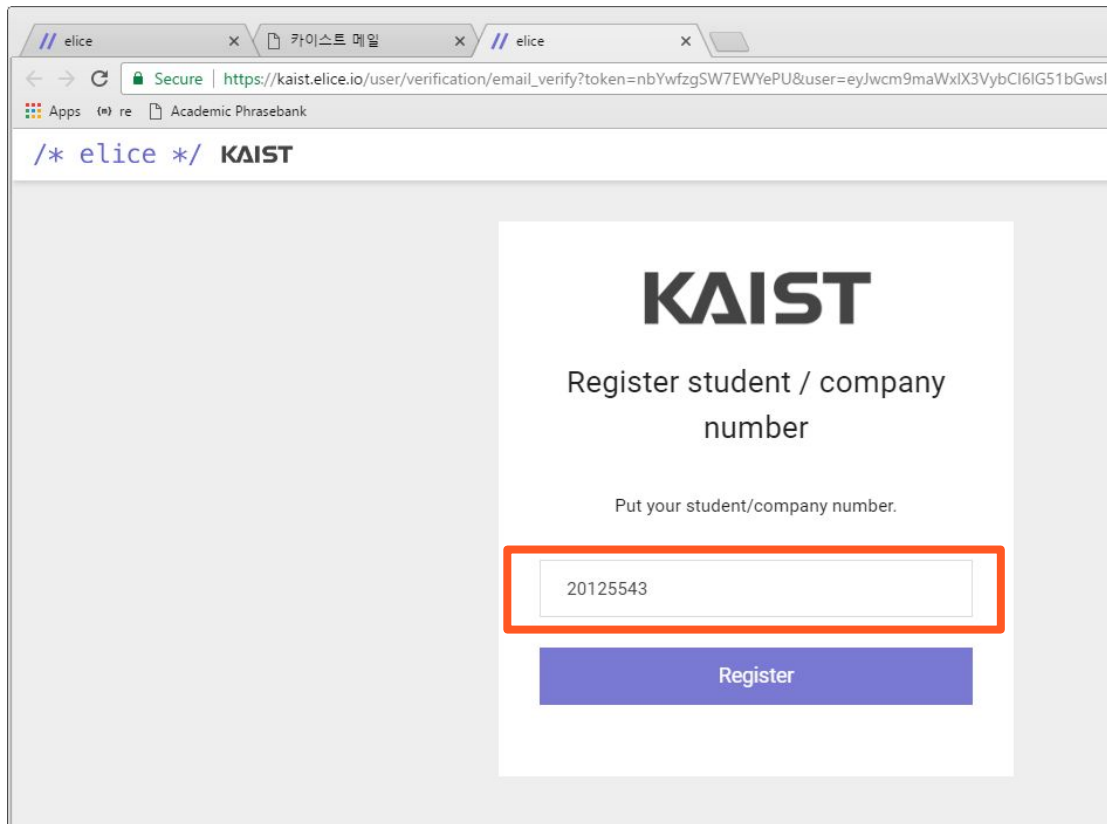
# Signing up (8)



# Signing up (9) - mail.kaist.ac.kr



# Signing up (10)



The screenshot shows a web browser window with three tabs. The active tab is titled 'elice' and shows a URL starting with 'https://kaist.elice.io/user/verification/email\_verify?token=nbYwfzgSW7EWYePU&user=eyJwcm9maWxlX3VybCI6IG51bGwsl'. The browser's address bar also shows 'Secure' and a lock icon. Below the address bar, there are links for 'Apps', 're', and 'Academic Phrasebank'. The main content area of the browser shows the KAIST logo and the text 'Register student / company number'. Below this, it says 'Put your student/company number.' and there is a text input field containing the number '20125543'. The input field is highlighted with a red border. Below the input field is a blue button labeled 'Register'.

elice 카이스트 메일 elice

Secure | [https://kaist.elice.io/user/verification/email\\_verify?token=nbYwfzgSW7EWYePU&user=eyJwcm9maWxlX3VybCI6IG51bGwsl](https://kaist.elice.io/user/verification/email_verify?token=nbYwfzgSW7EWYePU&user=eyJwcm9maWxlX3VybCI6IG51bGwsl)

Apps (re) Academic Phrasebank

/\* elice \*/ KAIST

**KAIST**

Register student / company number

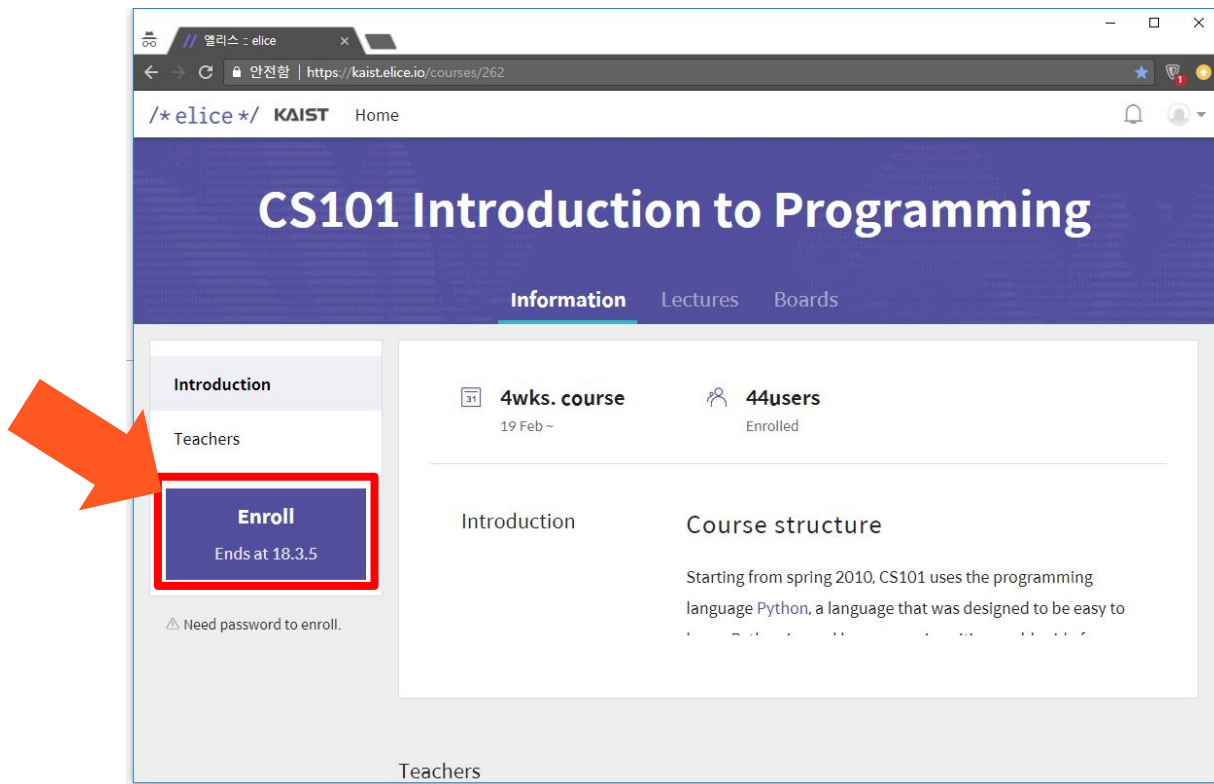
Put your student/company number.

20125543

Register

**MUST USE  
YOUR  
STUDENT ID!!**

# kaist.elice.io/courses/262



The screenshot shows a web browser window displaying the KAIST Elice website. The URL in the address bar is <https://kaist.elice.io/courses/262>. The page title is "CS101 Introduction to Programming". The navigation bar includes "Information", "Lectures", and "Boards". The left sidebar has a menu with "Introduction", "Teachers", and "Enroll". The "Enroll" button is highlighted with a red box and a red arrow. Below the "Enroll" button, it says "Ends at 18.3.5" and "Need password to enroll." The main content area shows course details: "4wks. course" (19 Feb ~), "44users" (Enrolled), and a section titled "Course structure" with the text "Starting from spring 2010, CS101 uses the programming language Python, a language that was designed to be easy to".

**CS101 Introduction to Programming**

Information Lectures Boards

Introduction Teachers **Enroll** Ends at 18.3.5

⚠ Need password to enroll.

4wks. course 19 Feb ~ 44users Enrolled

Introduction Course structure

Starting from spring 2010, CS101 uses the programming language Python, a language that was designed to be easy to

Teachers



# kaist.elice.io/courses/262

The screenshot shows a web browser window with the URL `https://kaist.elice.io/courses/262`. The page title is "CS101 Introduction to Programming". A modal dialog titled "ENROLL PASSWORD" is open, prompting the user to "Enroll password". The modal contains a password input field with a masked password "\*\*\*\*\*" and an "Enroll" button. In the background, the course page shows a sidebar with "Introduction", "Teachers", and an "Enroll" button that says "Ends at 18.3.5". A message at the bottom of the sidebar says "Need password to enroll."

## Password:

cs101\_2018\_kaist\_?!

# Signing up DONE!

The screenshot shows a web browser window with the URL `https://kaist.elice.io/courses/262/lectures`. The page title is `/*elice*/ KAIST Home`. The main heading is **CS101 Introduction to Programming**. The navigation bar includes `Information`, **`Lectures`**, `Boards`, and `Progress`. A search bar is located above the main content area.

The left sidebar lists the course structure:

- Lecture 1** (selected)
- Lab 1
- Supplement
- Lab 2
- Lecture 2
- Lecture 3
- Lab 3
- Lecture 4
- Lab 4

The main content area displays the details for **Lecture 1** (dated 26 Feb 2018) and **Lab 1** (dated 27 Feb 2018). The completion status for Lecture 1 is `0 / 1 Completed`, and for Lab 1, it is `0 / 7 Completed`.

The tasks listed under Lab 1 are:

- `</>` Sandbox for Robot
- `</>` Task 2: Hurdle
- `</>` Task 3: Newspaper Delivery
- `</>` Task 4: Harvest

**Honor Code**

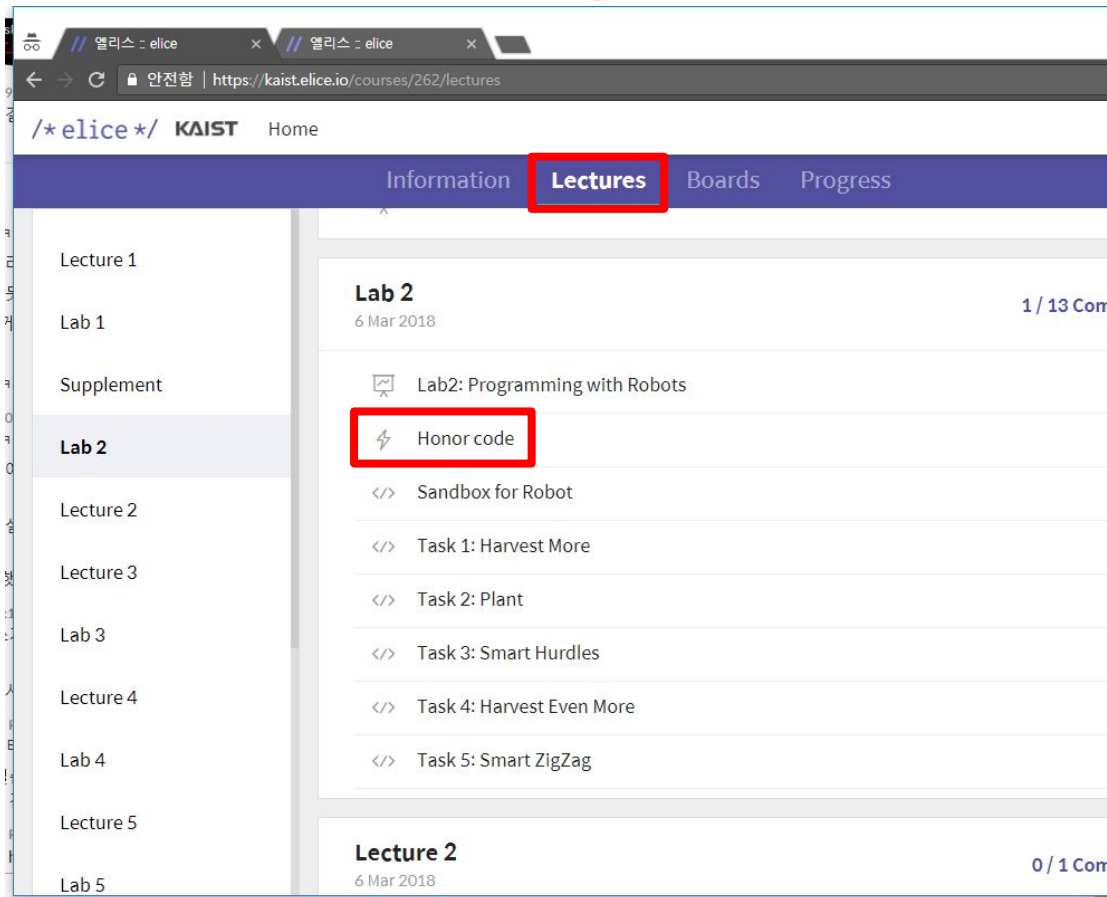
# Honor Code Agreement

The screenshot shows a web browser window with two tabs labeled '엘리스 - elice'. The address bar shows the URL 'https://kaist.elice.io/courses/262/lectures'. The page header includes the KAIST logo and a 'Home' link. A navigation bar at the top has four tabs: 'Information', 'Lectures' (highlighted with a red box), 'Boards', and 'Progress'. On the left side, a sidebar lists course items: 'Lecture 1', 'Lab 1', 'Supplement', 'Lab 2' (highlighted), 'Lecture 2', 'Lecture 3', 'Lab 3', 'Lecture 4', 'Lab 4', 'Lecture 5', and 'Lab 5'. The main content area displays 'Lab 2' with a date of '6 Mar 2018' and a completion status of '1 / 13 Completed'. Below this, a list of items is shown, each with an icon and a completion status (checkmark in a circle):

- Lab2: Programming with Robots (checked)
- Honor code** (checked, highlighted with a red box)
- Sandbox for Robot (checked)
- Task 1: Harvest More (checked)
- Task 2: Plant (checked)
- Task 3: Smart Hurdles (checked)
- Task 4: Harvest Even More (checked)
- Task 5: Smart ZigZag (checked)

At the bottom, a section for 'Lecture 2' is partially visible, dated '6 Mar 2018', with a completion status of '0 / 1 Completed'.

# Honor Code Agreement



You should  
agree with  
Honor Code.

Any cheating  
is strictly  
prohibited in  
CS101.

# Honor Code Agreement

are regarded as violations of academic integrity and honesty.

- Referring from other students/publisher's solutions, assignments, and reports.
- Allowing another student to refer from one's own work
- Submitting another student's work as his or her own
- Unpermitted collaboration or aid on take-home examinations and class assignments
- Plagiarism: the use of another person's original work without giving reasonable and appropriate credit to or acknowledging the author or source

The Honor Code works to the benefit of students, professors, and administrators in the department, which is based on the mutual trust that all those bound by it will uphold its principles. However, this makes

to abide by all of the above rules and policies, and pledge that I will neither give nor receive any unauthorized aid on examinations or other class assignments that are used by the instructor as the basis for grading."

You agree that you understand all the details written above and you will not violate academic integrity and honesty by writing your name here.

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

SECTION 1 OF 1

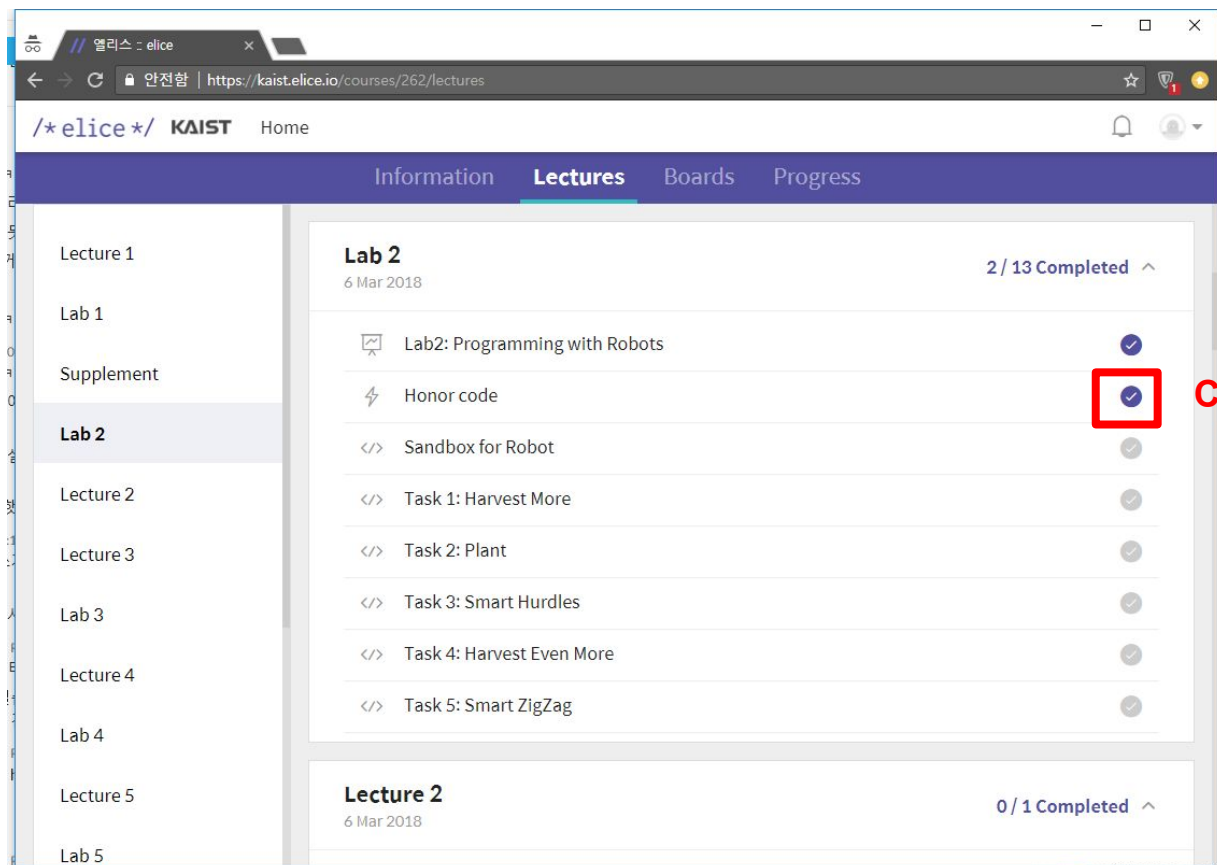
SUBMIT

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



The screenshot shows the KAIST Elice portal interface. The browser address bar displays the URL <https://kaist.elice.io/courses/262/lectures>. The page has a navigation bar with tabs for Information, Lectures, Boards, and Progress. The 'Lectures' tab is active. On the left, a sidebar lists various course items, with 'Lab 2' highlighted. The main content area displays 'Lab 2' with a date of '6 Mar 2018' and a completion status of '2 / 13 Completed'. Below this, a list of tasks is shown, each with a completion icon. The 'Honor code' task is highlighted with a red box and a blue checkmark.

Task	Status
Lab2: Programming with Robots	Completed
Honor code	Completed
Sandbox for Robot	Completed
Task 1: Harvest More	Completed
Task 2: Plant	Completed
Task 3: Smart Hurdles	Completed
Task 4: Harvest Even More	Completed
Task 5: Smart ZigZag	Completed

Completed!

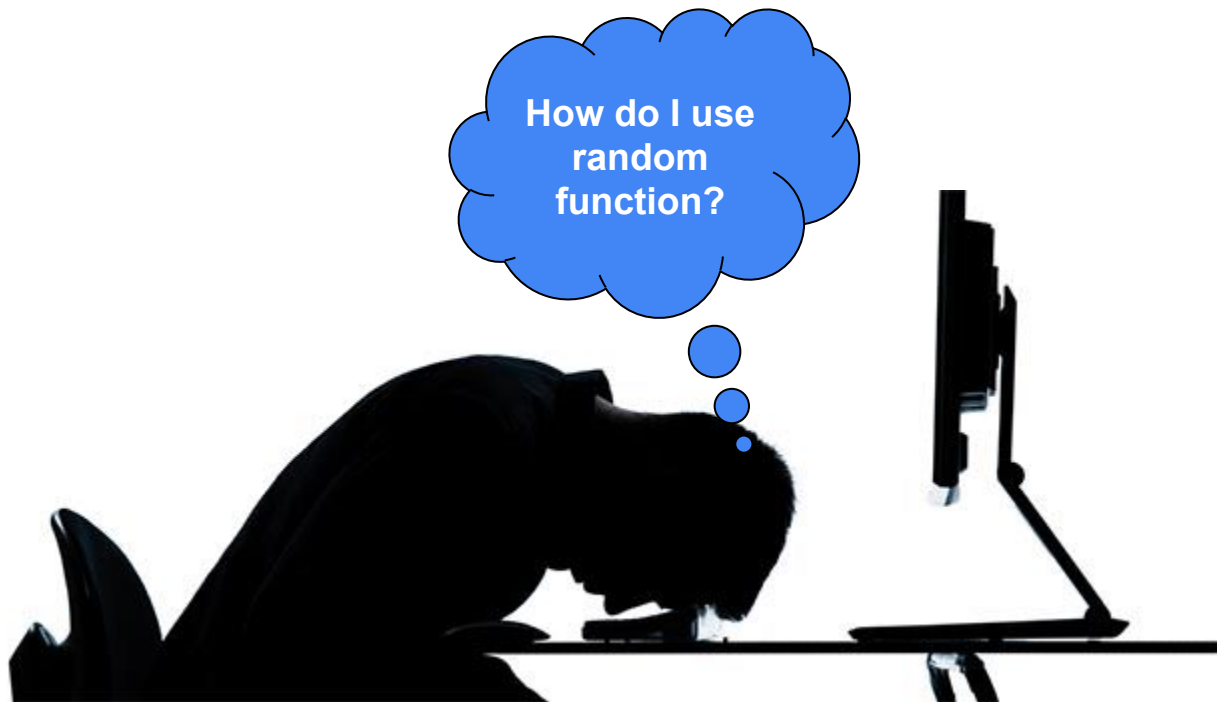
**questions?**



**Useful resources**

You're alone in your dorm room trying to study...

and stuck...





python 3 random example|



Google Search

I'm Feeling Lucky

Google.co.kr offered in: [한국어](#)



python 3 random example



All

Videos

Images

News

Maps

More

Settings

Tools

About 6,360,000 results (0.41 seconds)

## 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](http://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

<https://stackoverflow.com/questions/.../generate-random-integers-between-0-and-9> ▼

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.



python 3 random example



All

Videos

Images

News

Maps

More

Settings

Tools

About 6,360,000 results (0.41 seconds)

### 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](http://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

<https://stackoverflow.com/questions/.../generate-random-integers-between-0-and-9> ▼

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.



python 3 random example



All

Videos

Images

News

Maps

More

Settings

Tools

About 6,360,000 results (0.41 seconds)

### 9.6. random — Generate pseudo

<https://docs.python.org/3/library/random.html>

For integers, there is uniform selection from the basic function `random()`, which ... 1, January

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

### 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](http://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

<https://stackoverflow.com/questions/.../generate-random-integers>

Oct 22, 2010 - More info: <https://docs.python.org/3.0/library/random.html>  
10 pseudo random integers in range 0 to 10

<http://stackoverflow.com/...>

These two  
websites are  
usually good  
signs!

# 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

**Lab 1**  
27 Feb 2018

1 / 7 Completed ^

	Lab 1: Introduction to Lab	
	Sandbox for Robot	
	Task 1: ZigZag	
	Task 2: Hurdle	
	Task 3: Newspaper Delivery	
	Task 4: Harvest	
	Task 5: Harvest Again	

- How to define **function**
- How to use **for-loop**

**Lab 2**  
6 Mar 2018

2 / 13 Completed ^

	Lab2: Programming with Robots	
	Honor code	
	Sandbox for Robot	
	Task 1: Harvest More	
	Task 2: Plant	
	Task 3: Smart Hurdles	
	Task 4: Harvest Even More	
	Task 5: Smart ZigZag	

- How to use **conditional expressions**
- How to use **while-loop**

# Robot World (Read the robot notes!)

The screenshot shows a web browser window with the URL <https://kaist.elice.io/courses/262/lectures>. The page is titled "KAIST Home" and has a navigation bar with "Information", "Lectures", "Boards", and "Progress". The "Lectures" tab is selected. On the left, a sidebar lists "Lecture 1", "Lab 1", "Supplement", "Lab 2", "Lecture 2", "Lecture 3", "Lab 3", "Lecture 4", "Lab 4", "Lecture 5", and "Lab 5". The "Supplement" section is highlighted. The main content area shows a list of items under the "Supplement" heading, dated "27 Feb 2018". The items are: "Learning programming with Robots" (checked), "Photo processing with cs1media" (checked), "Graphics tutorial (for lecture 5 & lab 5)" (checked), "Python 3 and IDE" (checked), and "How to communicate in CS101" (checked). Below this, the "Lab 2" section is shown, dated "6 Mar 2018", with items: "Lab2: Programming with Robots" (checked), "Honor code" (checked), "Sandbox for Robot" (checked), and "Task 1: Harvest More" (checked). A red box highlights the "Learning programming with Robots" item in the Supplement section.

Section	Date	Item	Status
Supplement	27 Feb 2018	Learning programming with Robots	✓
		Photo processing with cs1media	✓
		Graphics tutorial (for lecture 5 & lab 5)	✓
		Python 3 and IDE	✓
		How to communicate in CS101	✓
Lab 2	6 Mar 2018	Lab2: Programming with Robots	✓
		Honor code	✓
		Sandbox for Robot	✓
		Task 1: Harvest More	✓

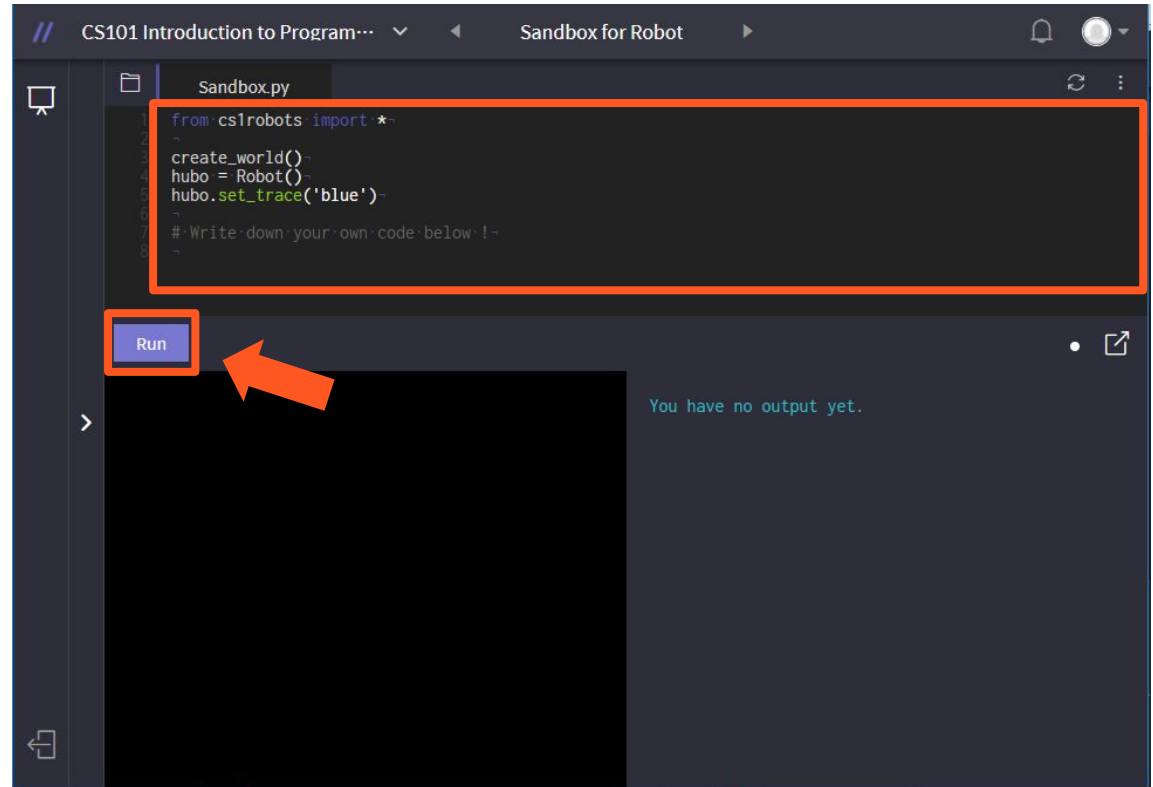
# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot()
```

```
hubo.set_trace('blue')
```



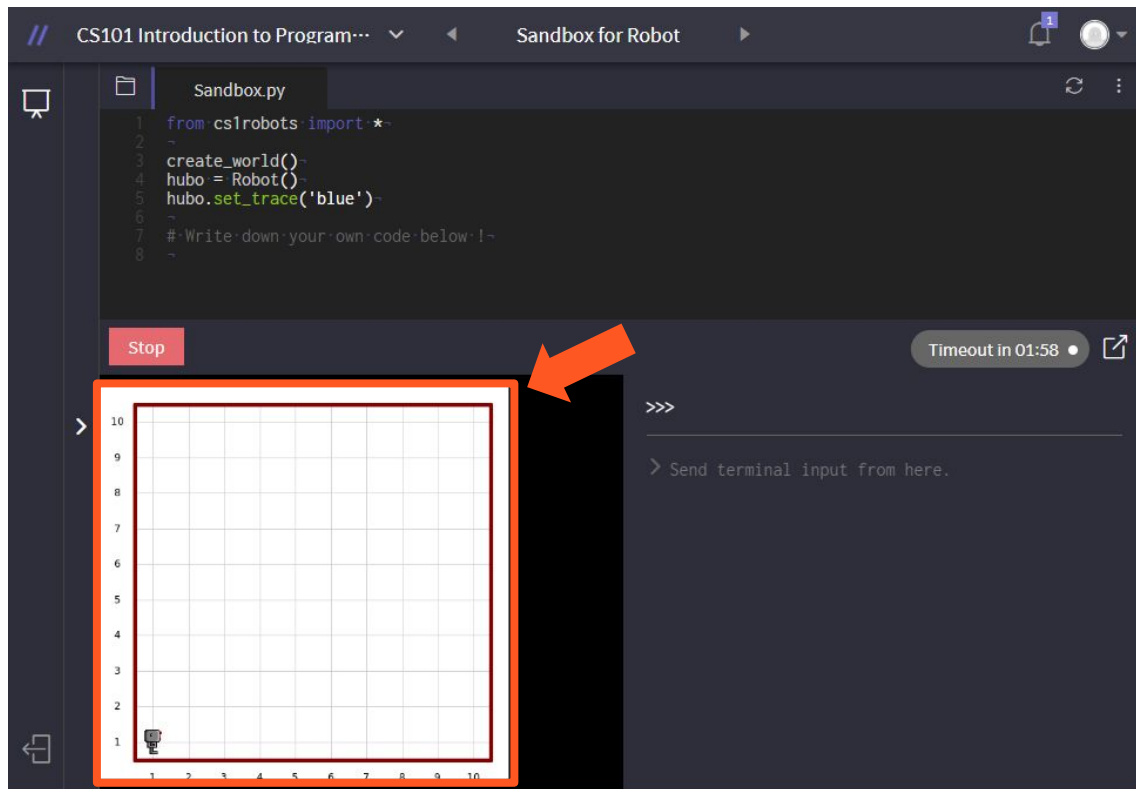
# Some Basics of *cs1robot* library

```
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
```

# Some Basics of *cs1robot* library

```
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 *cs1robot* library

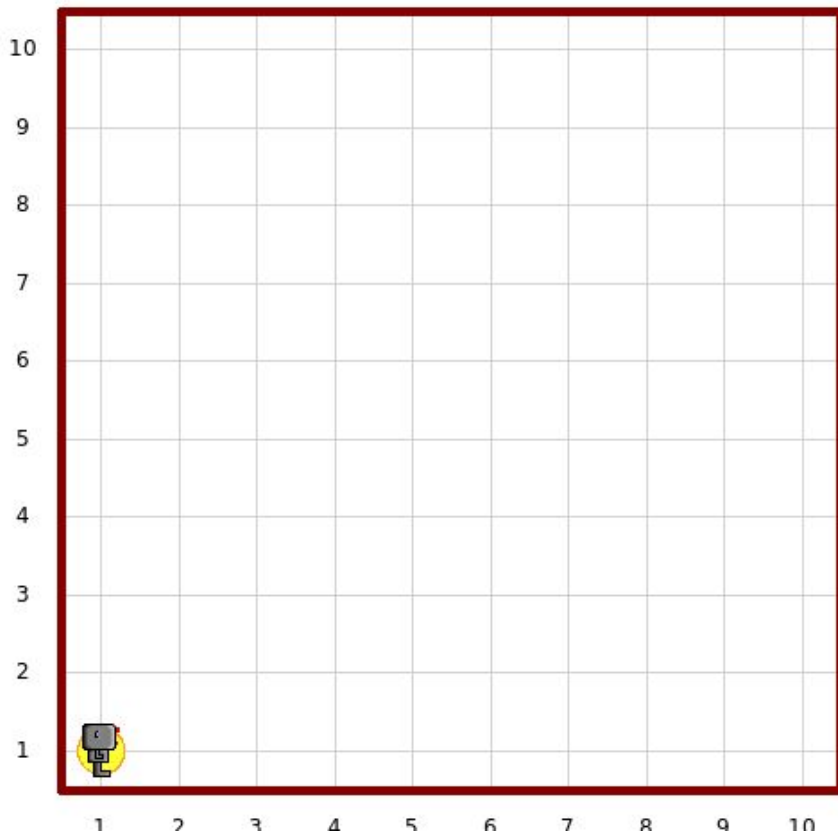
```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot beepers=10)
```

```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```



# Some Basics of *cs1robot* library

```
from cs1robots import *
```

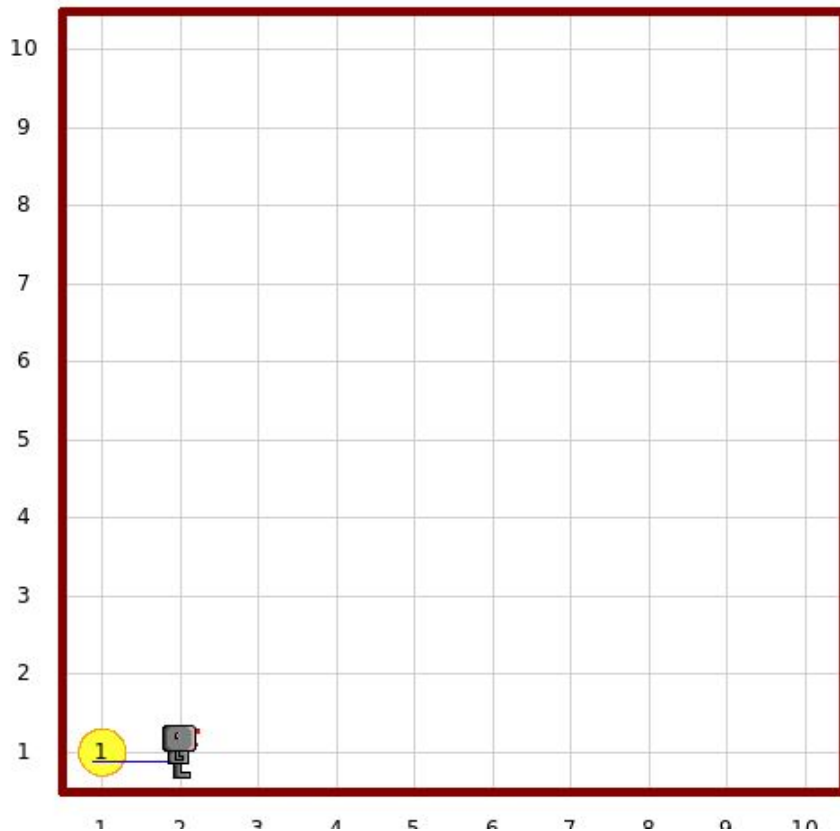
```
create_world()
```

```
hubo = Robot(b beepers=10)
```

```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```

```
hubo.move()
```





# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

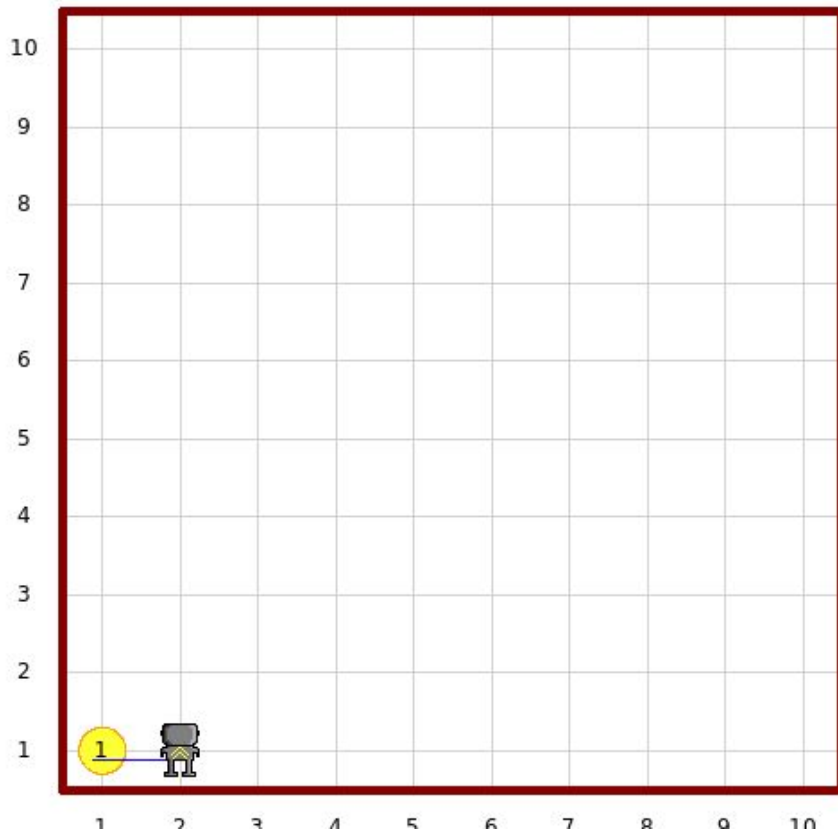
```
hubo = Robot(beepers=10)
```

```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```

```
hubo.move()
```

```
hubo.turn_left()
```



# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot(beepers=10)
```

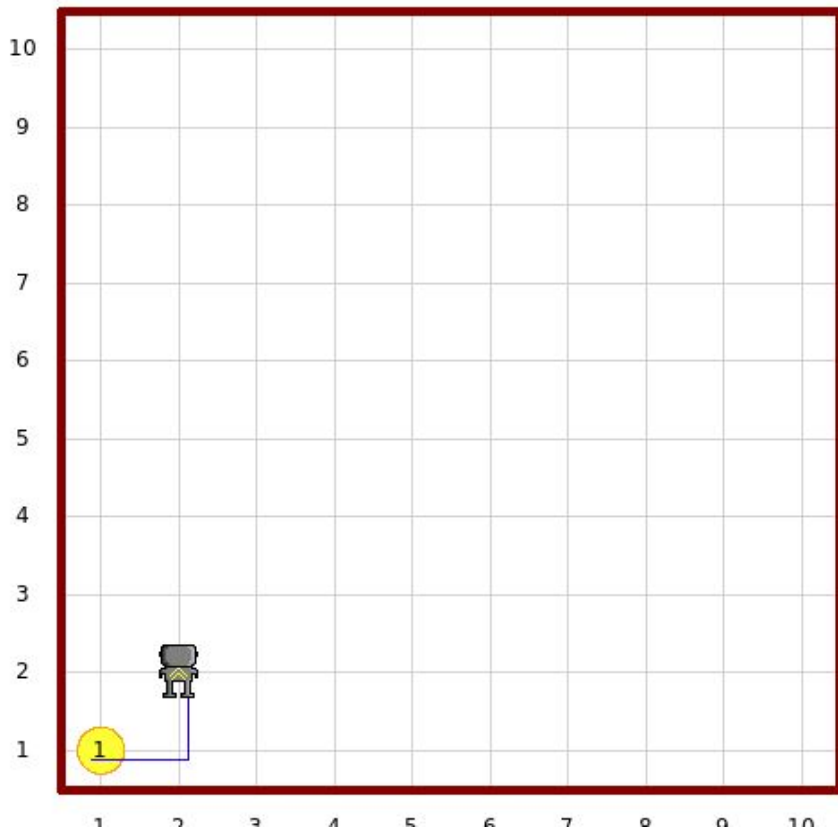
```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```

```
hubo.move()
```

```
hubo.turn_left()
```

```
hubo.move()
```



# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot(beepers=10)
```

```
hubo.set_trace('blue')
```

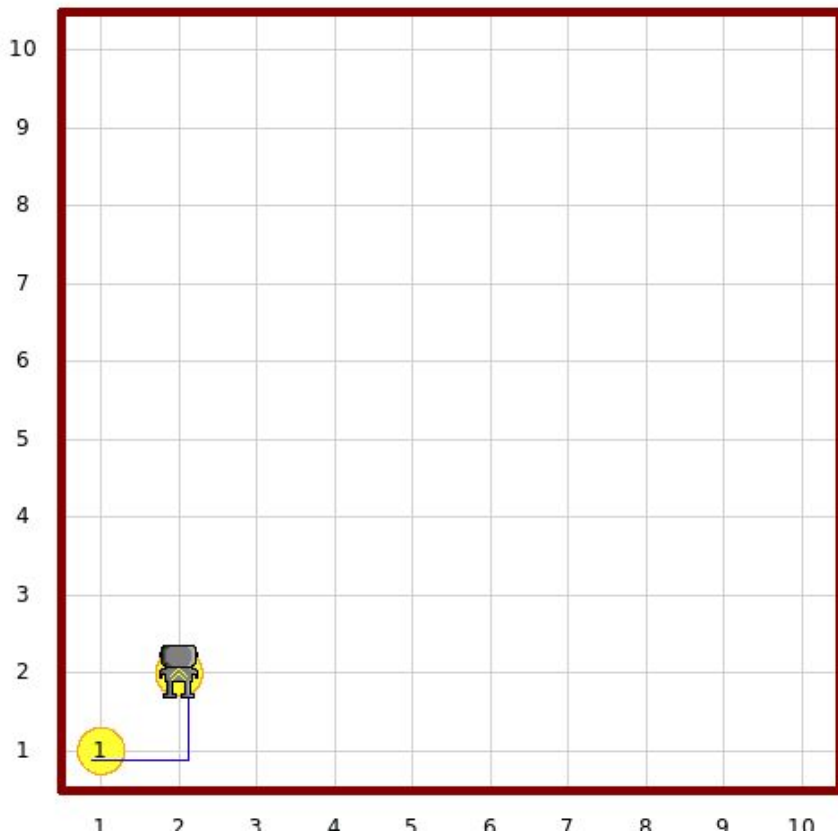
```
hubo.drop_beeper()
```

```
hubo.move()
```

```
hubo.turn_left()
```

```
hubo.move()
```

```
hubo.drop_beeper()
```



# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot beepers=10)
```

```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```

```
hubo.move()
```

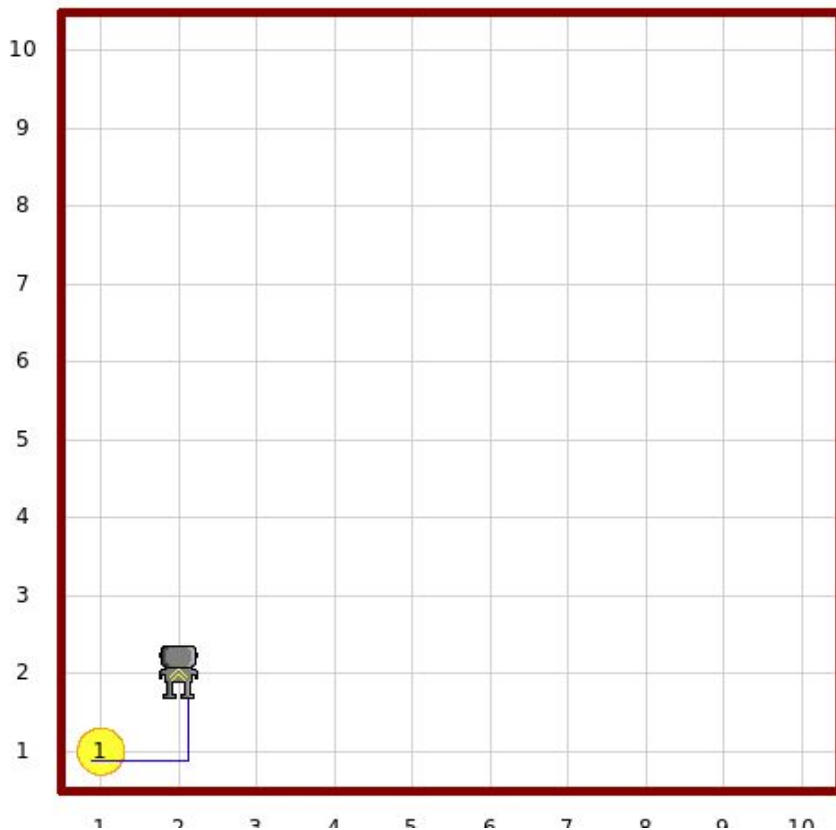
```
hubo.turn_left()
```

```
hubo.move()
```

```
hubo.drop_beeper()
```

```
hubo.pick_beeper()
```

✓ Before picking up a beeper, hubo should be on a beeper!



# Some Basics of *cs1robot* library

```
from cs1robots import *
```

```
create_world()
```

```
hubo = Robot(b beepers=10)
```

```
hubo.set_trace('blue')
```

```
hubo.drop_beeper()
```

```
hubo.move()
```

```
hubo.turn_left()
```

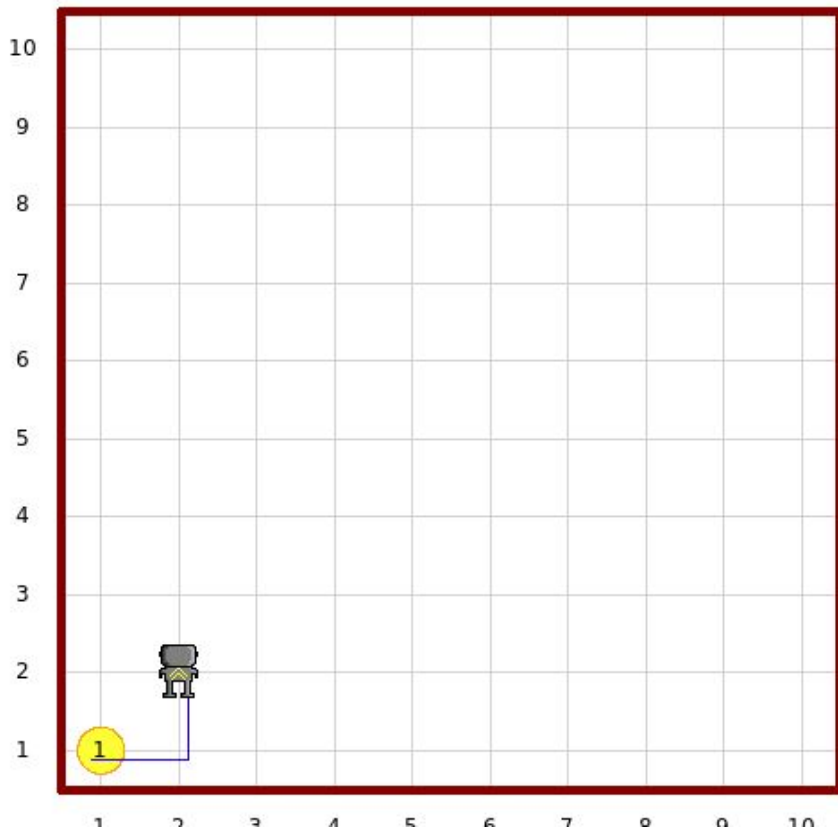
```
hubo.move()
```

```
hubo.drop_beeper()
```

```
hubo.pick_beeper()
```

```
hubo.pick_beeper()
```

**Error!**



# How to Load World?

```
from cs1robots import *
```

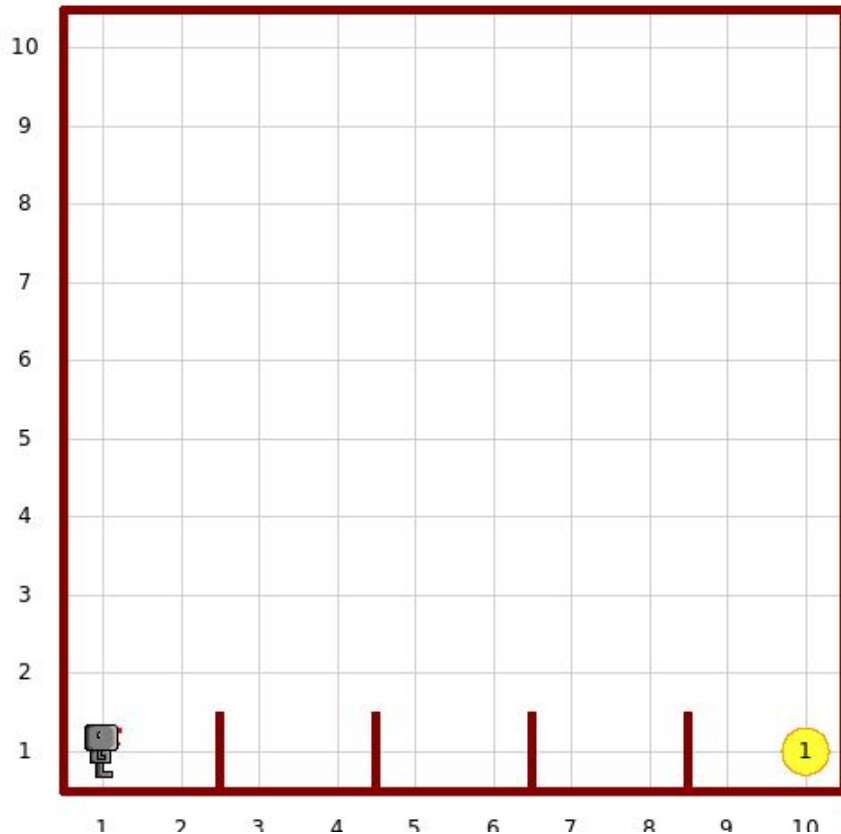
```
load_world('worlds/hurdles1.wld')
```

```
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()
```

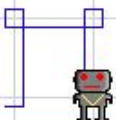


10  
9  
8  
7  
6  
5  
4  
3  
2  
1

**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



1 2 3 4 5 6 7 8 9 10

# How to Use For-Loop

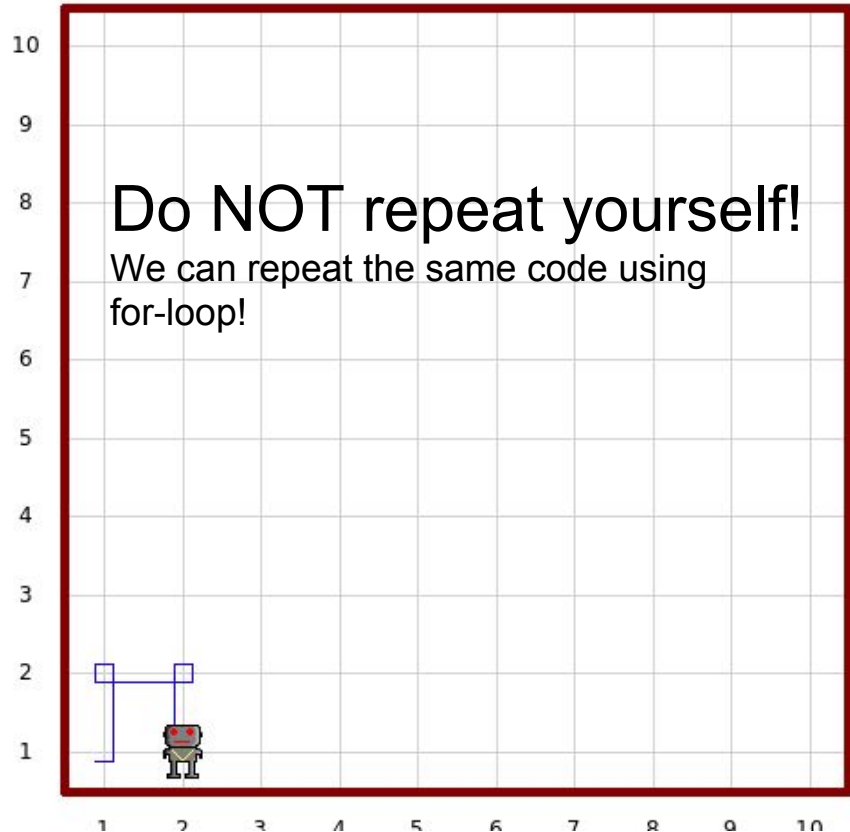
```
from cs1robots import *
```

```
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()
```



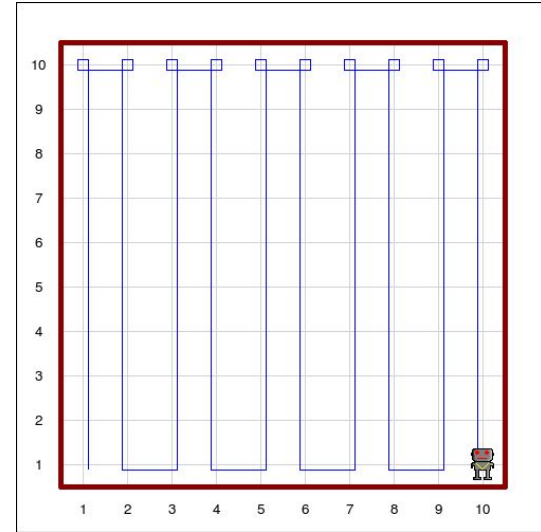


**Week 2**

**Today's Tasks 1**

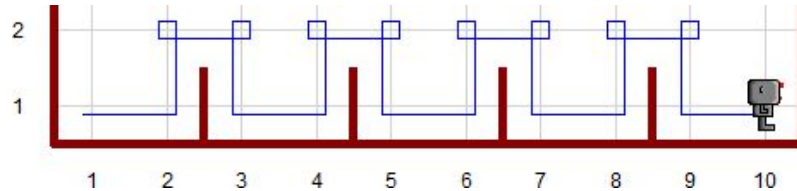
# Task 1: ZigZag

- Create a default world.
- Add a robot.
- Make the robot visit the entire world in a zigzag fashion.



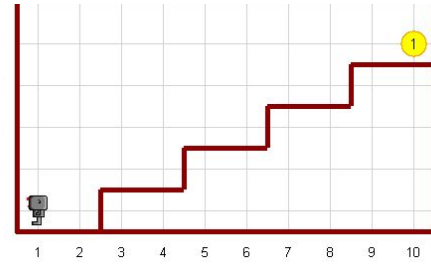
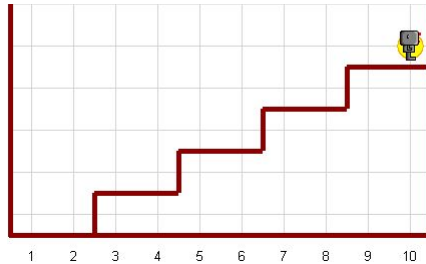
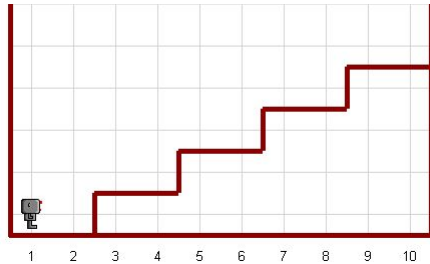
## Task 2: Hurdle

- Load the world named 'hurdles1.wld'.
- Add a robot.
- Make the robot jump all hurdles and pick up the beeper.



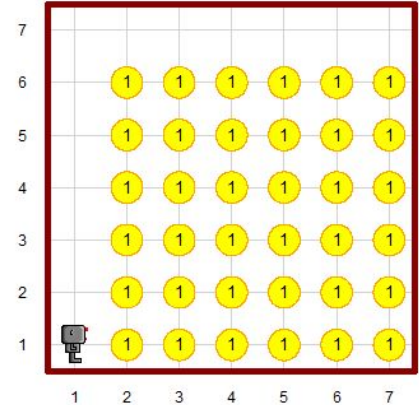
# Task 3: Newspaper delivery

- Load the world named 'newspaper.wld'.
- Add a robot with one beeper.
- Make the robot deliver newspapers and return to his starting point.



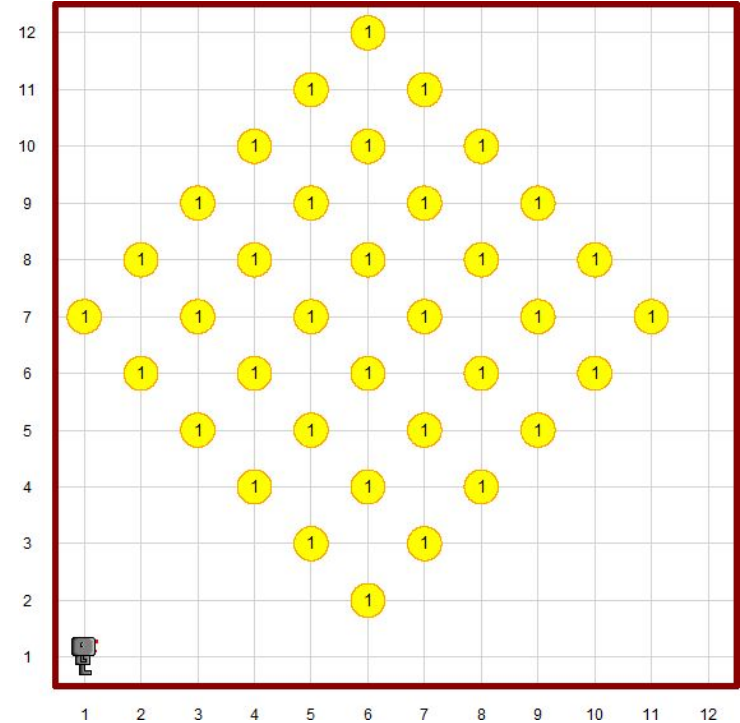
# Task 4: Harvest

- 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.



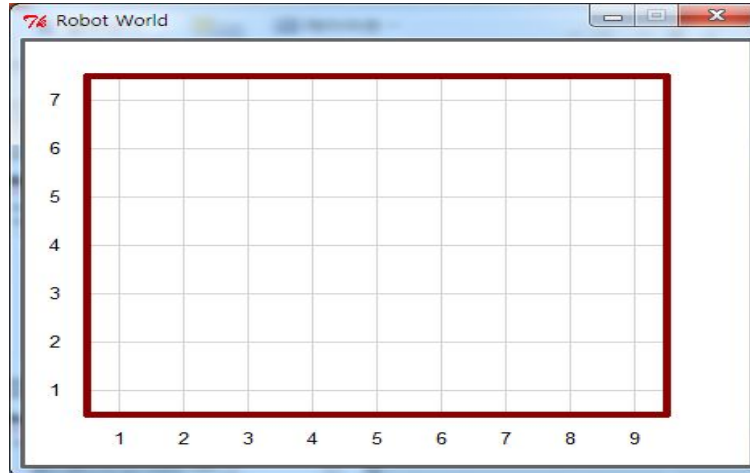
# **If-statement & While-statement**

# New functions

- Create a custom world

```
create_world(streets = 7, avenues = 9)
```

- ✓ 'streets' for the number of rows
- ✓ 'avenues' for the number of columns

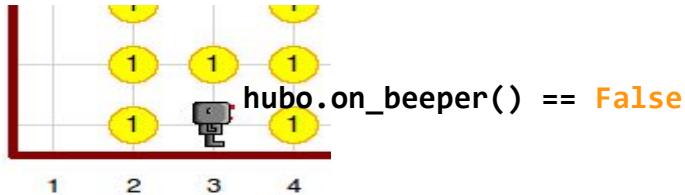
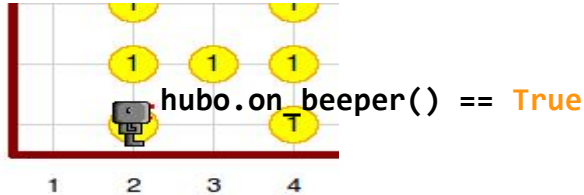




# New functions

- Can check if there is a beeper

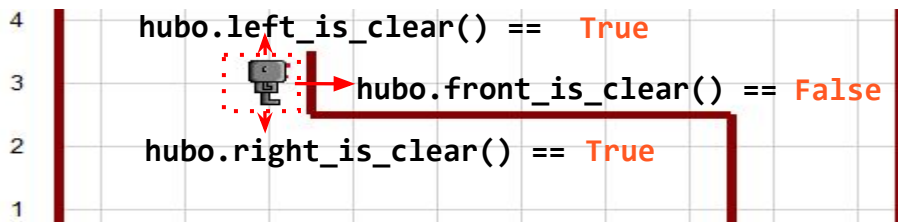
```
hubo.on_beeper()
```



# New functions

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

```
hubo.front_is_clear()  
hubo.left_is_clear()  
hubo.right_is_clear()
```



# If statements

- If statements **sequentially** checks the conditionals

**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

```
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

**while conditional\_expression:**  
works to do while conditional\_expression evaluates to True

- Example

```
n = 0
while n < 5:
    print(n)
    n = n + 1
```

Result)  
0  
1  
2  
3  
4

**Week 2**

**Today's Tasks**

# Tasks for Today!

- Read sections 10~13 in the robot notes

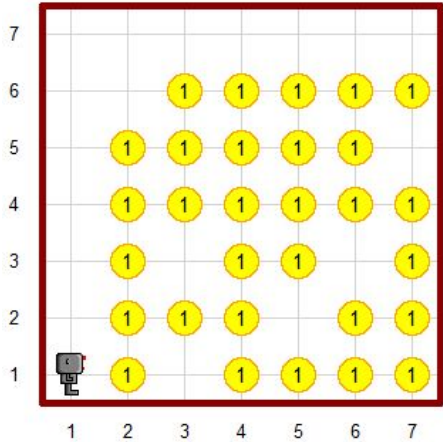
Five not so simple tasks

- 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

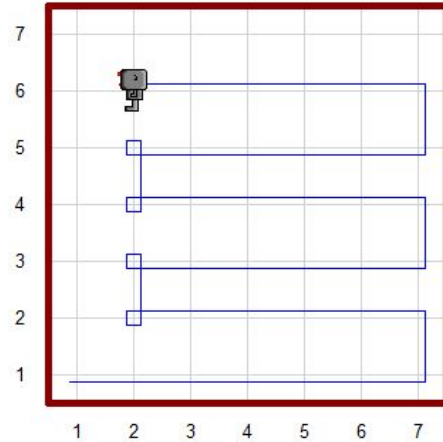
## Task 1 | Conditionals – Harvest More

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

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



## Before

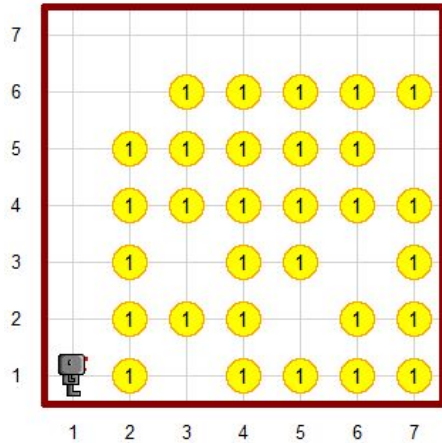


## After

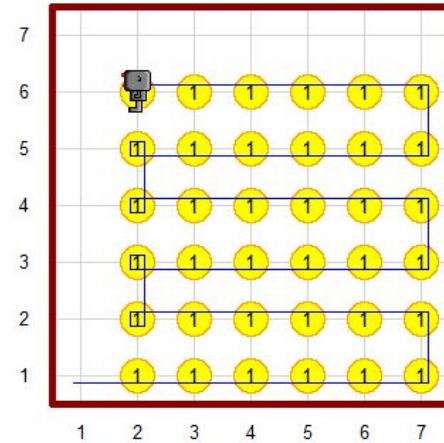


## Task 2 | Conditionals – Plant

- Write a program so that Hubo plants beepers in empty spots
- The finished screen should look like “harvest1.wld”  
`load_world(“worlds/harvest3.wld”)`



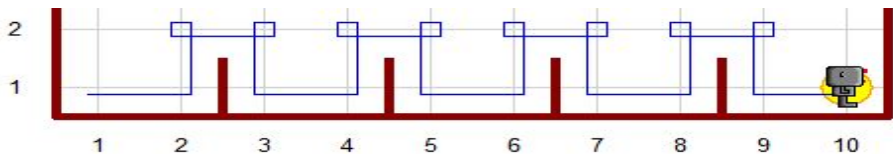
Before



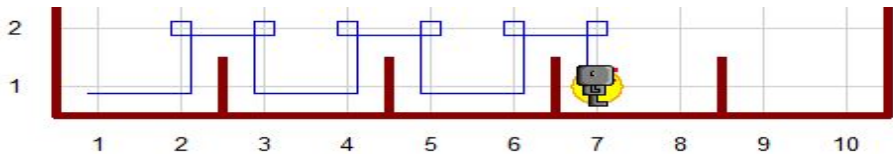
After

# Task 3 | Conditionals – Smart Hurdles

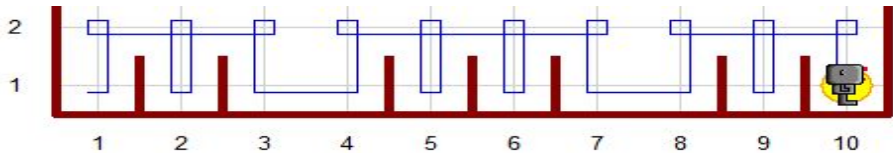
- Write **jump\_one\_hurdle()** in section 11
  - `move_jump_or_finish()` should be able to handle all three maps, “hurdles1.wld”, “hurdles2.wld” and “hurdles3.wld”. Check it yourself.
- Write a new program (similar to Hurdles3 in section 11) that uses a while loop. DO NOT USE a for-loop of fixed length
  - It should also work for all three hurdles



hurdles1.wld



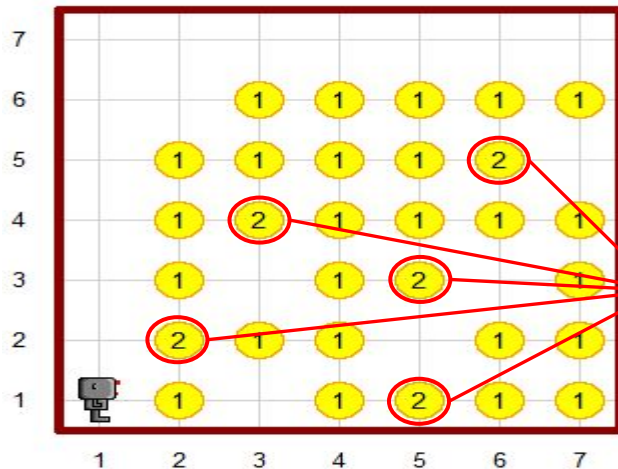
hurdles2.wld



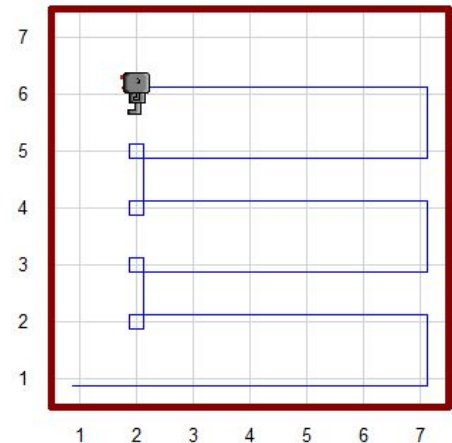
hurdles3.wld

# 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”)

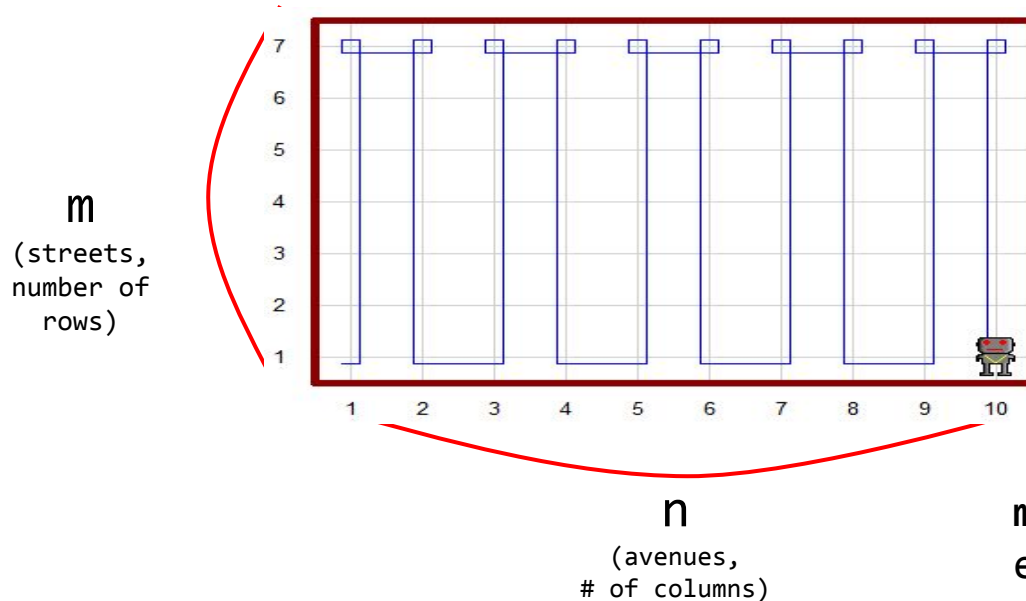


Have to harvest  
all carrots  
on one place



# 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



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

**questions?**