Code in Place 2023

Stanford CS106A

Section - Week 1

Welcome to Section!

If the Zoom session disconnects:

Try rejoining again with the “Section Video Call” button you used to join.

If that doesn’t work, check the Section Forum for a status update from me.

- If we need to recreate a new private Zoom room, I’ll post a link ASAP.

Today’s Agenda

Introductions
Icebreakers

Norms
Guidelines

Concept Review
Karel, Control Flow

Practice Problems
“Hospital Karel”

Introductions

Hi, I'm David, your Section Leader!

- Massachusetts Institute of Technology (MIT) - Computer Science
- Worked as a Software Engineer and Product Manager
- Attended Code in Place 2021 as a student
- Love photography, video games, movies
- Cool superpower I'd like to have:
  - Felix Felicis ("Liquid Luck" potion) from Harry Potter

How about you?

- Where are you from?
- Your coding experience
- Cool superpower you'd like to have

Course Weekly Timeline

Lectures	Section	Assignment
2 Video Lectures	**Live Zoom Session**	**Coding Challenges**
Both released every Monday. ~1 hour per lecture.	Practice lecture concepts. Prepare you for assignment.	Solidify your understanding. Optional: “Extension” projects

About Section

Section Goals

- Practice what you've been learning in lecture and get you ready to do the assignments.
- Have an interactive, collaborative session with like-minded peers.
- Finishing the live exercises in Section isn't the main goal. Learning is.
    - Some weeks, we might get more exercises than can be expected to finish in our allotted time
    - Sample solutions will be provided an hour after Section.

Expectations

- **Watch each week's lectures before coming to section!**
    - Two lectures are posted every Monday
    - Section is designed assuming you've watched lectures
- `section != lecture`
    - In section, we will do only a brief overview (5 min) of lecture concepts.
- Majority of the time will be solving sample problems together

Norms

Expectations & Guidelines

- Be kind, courteous, and thoughtful to your peers
  - We have students from different backgrounds (differences in language proficiency, culture, coding experience, etc.).
- Please ask questions!
- Make mistakes! Learning through mistakes is the fastest way to learn!
- Write your own code in your browser's IDE while following along.
- If we run overtime, no pressure to stay!

CIP is a Beginner-Friendly Course

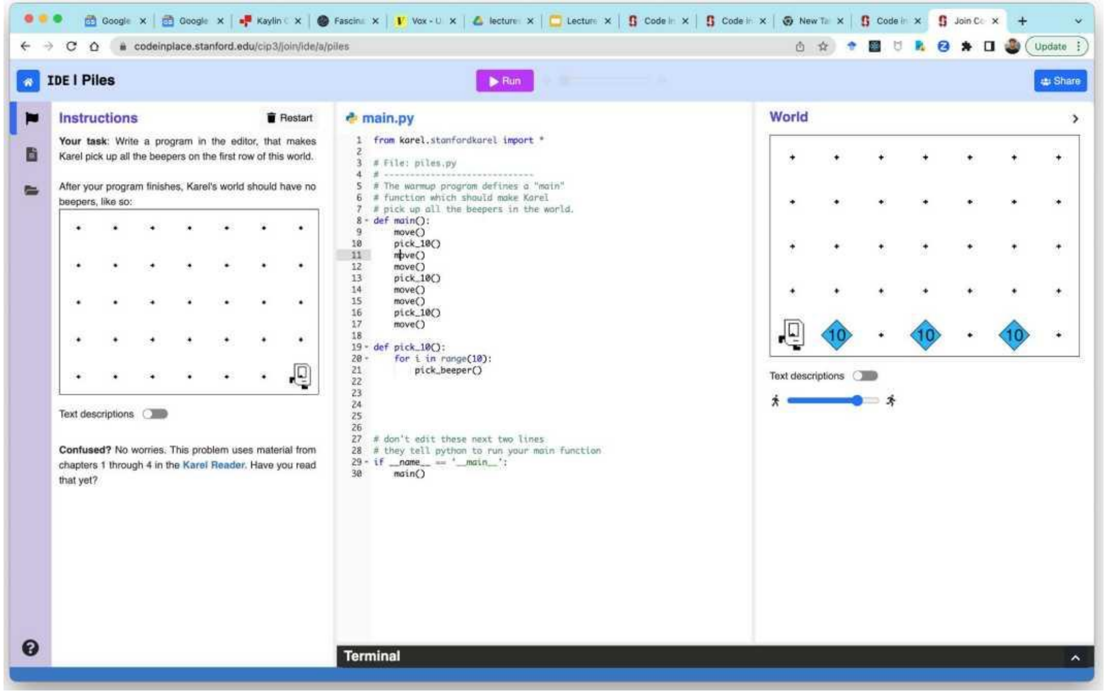
- If you have extensive programming experience, please help me to teach the other students (give them hints, etc.).
- It would be great if we get an even distribution of students contributing to the discussion, asking questions, and sharing their thoughts.

Norms

Zoom Etiquette

- We want to make this remote learning experience conducive to learning.
- We would appreciate if everyone could turn on their cameras.
    - However, we understand some people may be uncomfortable with cameras on, which is also fine!
- Please mute your microphone when you're not speaking.
    - To prevent picking up background noise.
- You can type in chat! And use Zoom's "reactions".
- Don't be afraid to tell anybody (me or students) to type what they just said in chat.
    - Accommodate connectivity issues, global accents, etc.

Brief Tour of Course's Web Platform



Your task: Write a program in the editor, that makes Karel pick up all the beepers on the first row of this world.

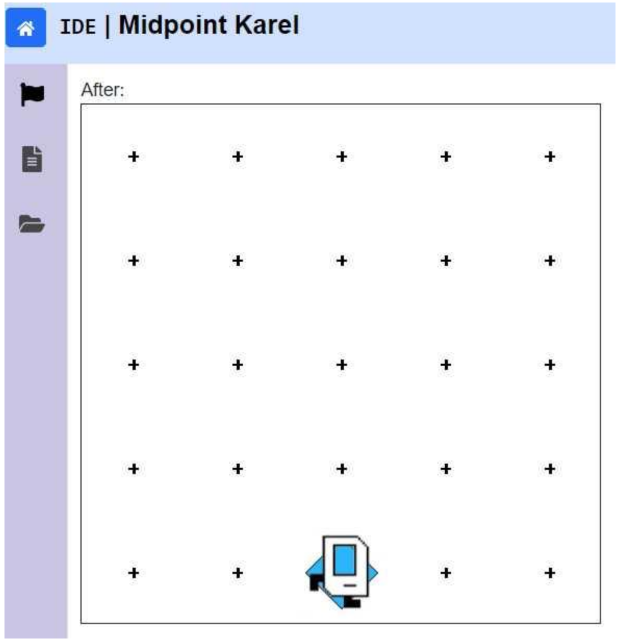
After your program finishes, Karel's world should have no beepers, like so:

Text descriptions

Confused? No worries. This problem uses material from chapters 1 through 4 in the Karel Reader. Have you read that yet?

```python
from karel.stanfordkarel import *

# File: piles.py
# -----------------------------
# The warmup program defines a "main"
# function which should make Karel
# pick up all the beepers in the world.
def main():
    move()
    pick_10()
    move()
    move()
    pick_10()
    move()
    move()
    pick_10()
    move()

def pick_10():
    for i in range(10):
        pick_beeper()




# don't edit these next two lines
# they tell python to run your main function
if __name__ == '__main__':
    main()
```

Some Basic Python Functionality is TURNED OFF….for now!



For the duration of Karel unit (Weeks 1 & 2):

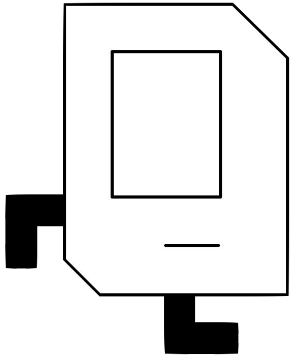
- Basic Python functionality is turned off for simplicity
  - Focus on building blocks of Karel commands
- Examples of disabled functionality:
  - print statements
  - conditional keywords (`NOT`, etc.)
  - variables, lists, dictionaries
  - input arguments to functions
  - function return statements

Challenge yourself to solve the optional "Karel Extension" coding assignment named **"Midpoint Karel"**

- There are multiple ways to solve the problem!
- "Midpoint Karel" is always a memorable fan favorite, every year

Concepts Review (Lightning Round!)

Karel is like a LEGO set with only 4 types of blocks....but you can do a lot!



Karel only knows four commands at the beginning:

```
move()

turn_left()

put_beeper()

pick_beeper()
```

Helpful website to visualize Karel (by Sophia Westwood):

www.karelhelper.com

Control Flow

`for` **Loops**

`while` **Loops**

`if/else`

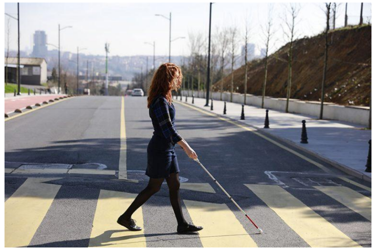
for **Loop**

```
for i in range(count):
    statements                # note indenting
```

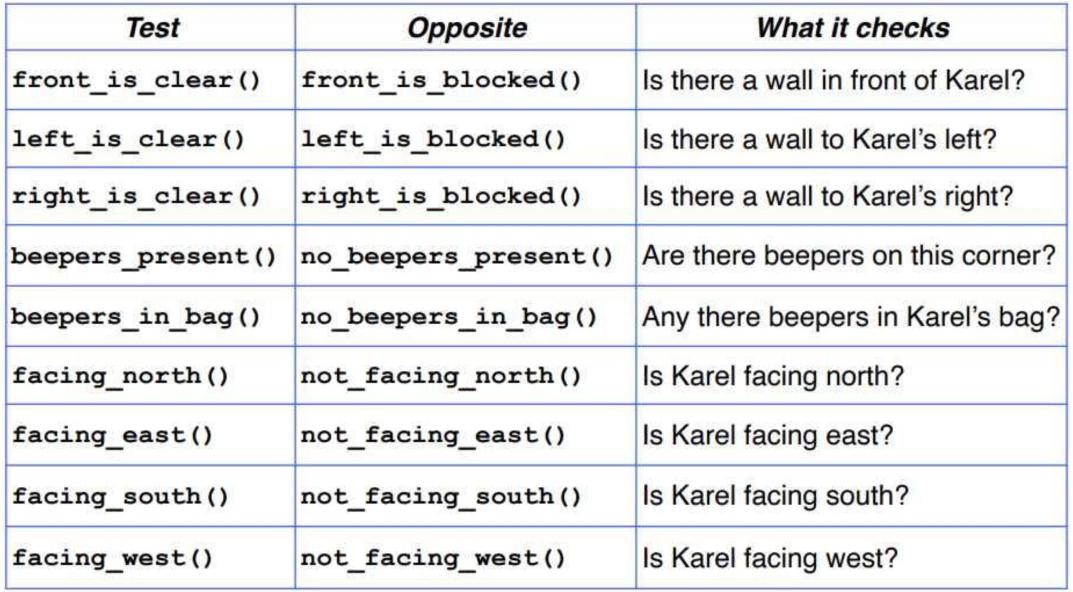
```
# Example:
def turn_right():
    for i in range(3):
        turn_left()           # note indenting
```

`while` Loop

```
while condition:
    statements                # note indenting


# Example:
def move_to_wall():
    while front_is_clear():
        move()                # note indenting
```



Conditions Karel Can Check For



Test	*Opposite*	*What it checks*
`front_is_clear()`	`front_is_blocked()`	Is there a wall in front of Karel?
`left_is_clear()`	`left_is_blocked()`	Is there a wall to Karel's left?
`right_is_clear()`	`right_is_blocked()`	Is there a wall to Karel's right?
`beepers_present()`	`no_beepers_present()`	Are there beepers on this corner?
`beepers_in_bag()`	`no_beepers_in_bag()`	Any there beepers in Karel's bag?
`facing_north()`	`not_facing_north()`	Is Karel facing north?
`facing_east()`	`not_facing_east()`	Is Karel facing east?
`facing_south()`	`not_facing_south()`	Is Karel facing south?
`facing_west()`	`not_facing_west()`	Is Karel facing west?

Check out "Karel Reader", Chapter 10 for full reference

Which Type of Loop Should You Use?

`for` Loop

You <u>know</u> exactly how many times to repeat (definite loop)

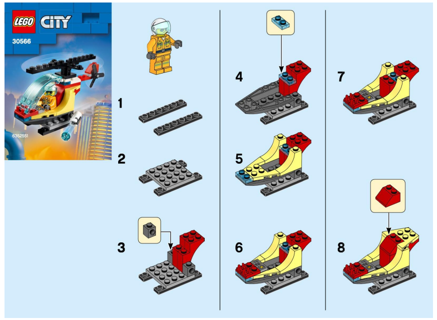
`while` Loop

You <u>don't know</u> exactly how many times to repeat (indefinite loop)

`if-else` **Statement**

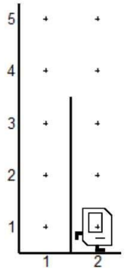
```
if condition:
    statements               # note indenting
else:
    statements


# Example:
def invert_beepers():
    if beepers_present():
        pick_beeper()        # note indenting
    else:
        put_beeper()         # note indenting
```

Decomposition



Break down a problem into more manageable sub-problems

- "Divide and conquer" approach
- Which sorts of tasks should be decomposed?
- Rule of Thumb: Each function should execute a single purpose.
- Code becomes easier to read!   For both you and everyone else.



```
turn_left()
while right_is_blocked():
      move()
turn_right()
move()
turn_right()
move_to_wall()
turn_left()
```

→

```
ascend_hurdle()
move()
descend_hurdle()
```

Live Exercise: “Hospital Karel”