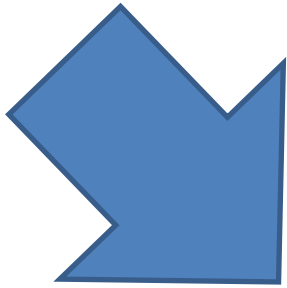


Course mechanics

(This is a brief version.

Please read the handout for full details).

Course Website



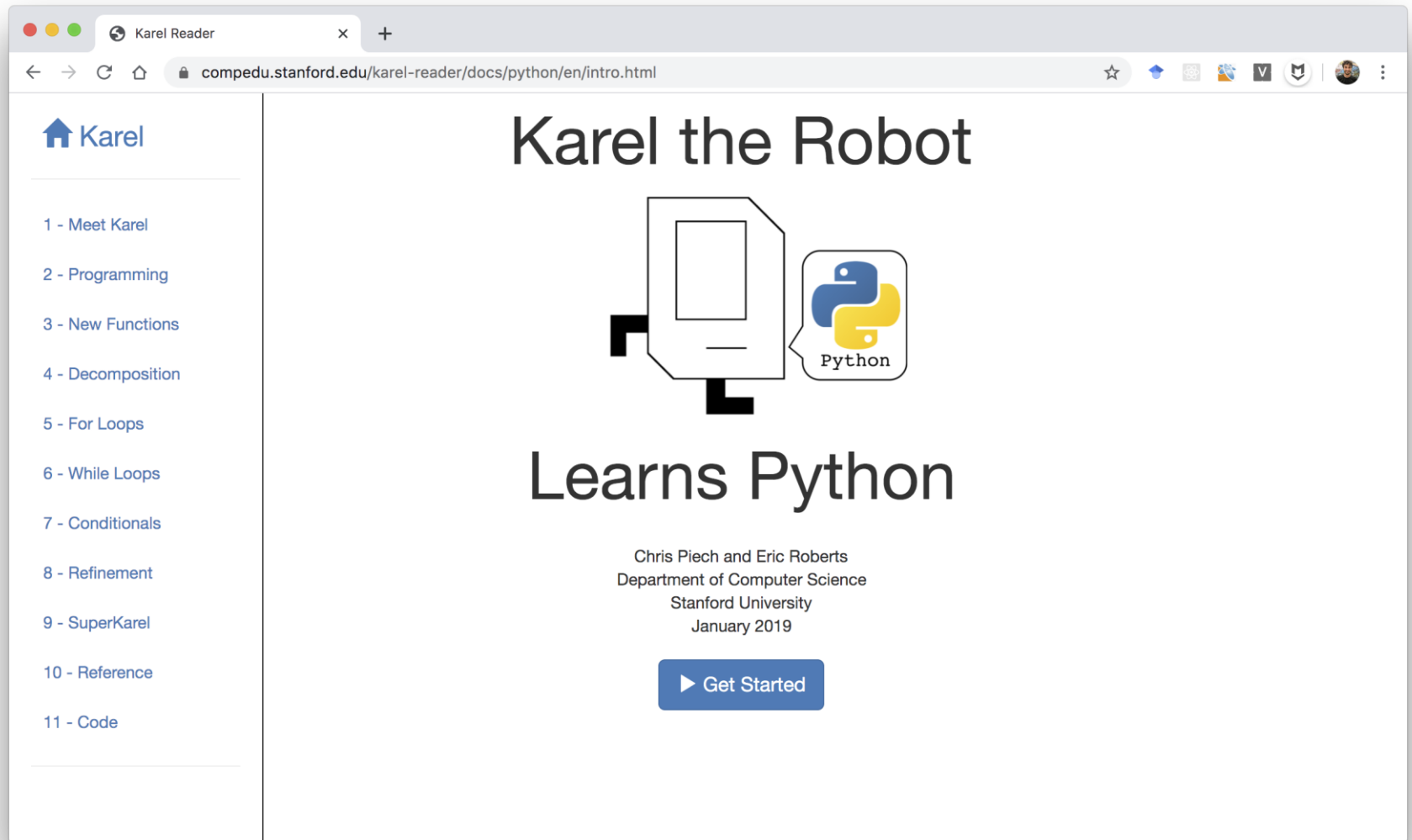
<https://cs101a.stickmind.com>



Prerequisite Test

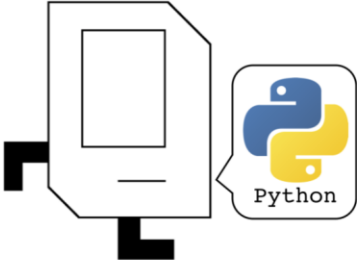


Online Text Books



The image is a screenshot of a web browser displaying the 'Karel the Robot' website. The browser's address bar shows the URL `compedu.stanford.edu/karel-reader/docs/python/en/intro.html`. The page features a large title 'Karel the Robot' at the top, followed by a graphic of a robot head with a Python logo on its side. Below the graphic is the subtitle 'Learns Python'. The authors, Chris Piech and Eric Roberts, and their affiliation with Stanford University are listed. A 'Get Started' button is prominently displayed. On the left, a sidebar contains a list of 11 chapters, starting with 'Meet Karel' and ending with 'Code'.

Karel the Robot



Learns Python

Chris Piech and Eric Roberts
Department of Computer Science
Stanford University
January 2019

[▶ Get Started](#)

- 1 - Meet Karel
- 2 - Programming
- 3 - New Functions
- 4 - Decomposition
- 5 - For Loops
- 6 - While Loops
- 7 - Conditionals
- 8 - Refinement
- 9 - SuperKarel
- 10 - Reference
- 11 - Code

Online Karel Reader

Karel Reader

compedu.stanford.edu/karel-reader/docs/python/en/chapter2.html

Home

Star

Share

Search

GitHub

Twitter

YouTube

Home

Karel

1 - Meet Karel

2 - Programming

3 - New Functions

4 - Decomposition

5 - For Loops

6 - While Loops

7 - Conditionals

8 - Refinement

9 - SuperKarel

10 - Reference

11 - Code

Chapter 2: Programming Karel

The simplest style of Karel program uses text to specify a sequence of built-in commands that should be executed when the program is **run**. Consider the simple Karel program below. The text on the left is the program. The state of Karel's world is shown on the right:

```
# File: FirstKarel.py
# -----
# The FirstKarel program defines a "main"
# function with three commands. These commands cause
# Karel to move forward one block, pick up a beeper
# and then move ahead to the next corner.
from karel.stanfordkarel import *

def main():
    move()
    pick_beeper()
    move()
```

Run Program

A 4x6 grid world. The robot is at (1,1). A beeper is at (2,1). A wall is at (4,1) and (4,2). The grid is labeled with coordinates (row, column) from (1,1) to (4,6).

Press the "Run" button to execute the program. Programs are typically written in a special application called an **Integrated Development Enviroment** (IDE) and most Karel programs are written in an IDE called PyCharm. Like an IDE, this reader has the ability to execute programs in order to help you see how things work as you learn.

The program is composed of several parts. The first part consists of the following lines:

```
# File: FirstKarel.py
# -----
```

What is CS101A?

Computer Science

“Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do.”

— Michael Fellows and Ian Parberry

“You must unlearn what you have learned”

— Yoda

Learning Goals

- *Learn how to harness computing power to solve problems.*
- To that end:
 - Explore fundamental techniques in computer programming.
 - Develop good software engineering style.
 - Gain familiarity with the Python programming language.

There are a lot of cool
programs you may one day
write

Computer Graphics



Pat Hanrahan, one of the founders of Pixar is a professor here.
He recently won the Turing Award – the Nobel Prize of Computer Science.

Consumer Applications



Computing in Medicine



(c) 2012 Intuitive Surgical, Inc.

Self-Driving Car

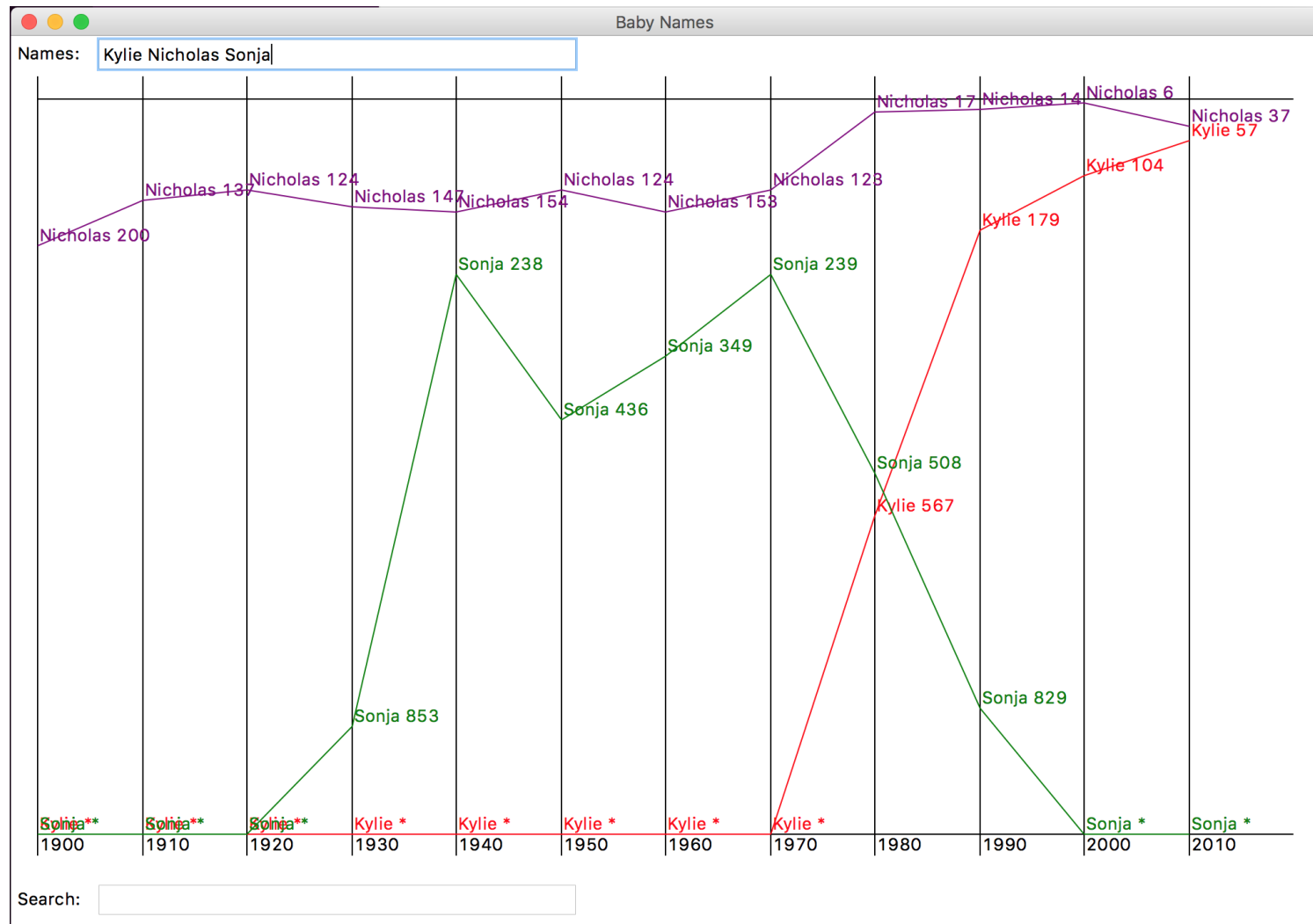


If only we could program
self-driving cars...

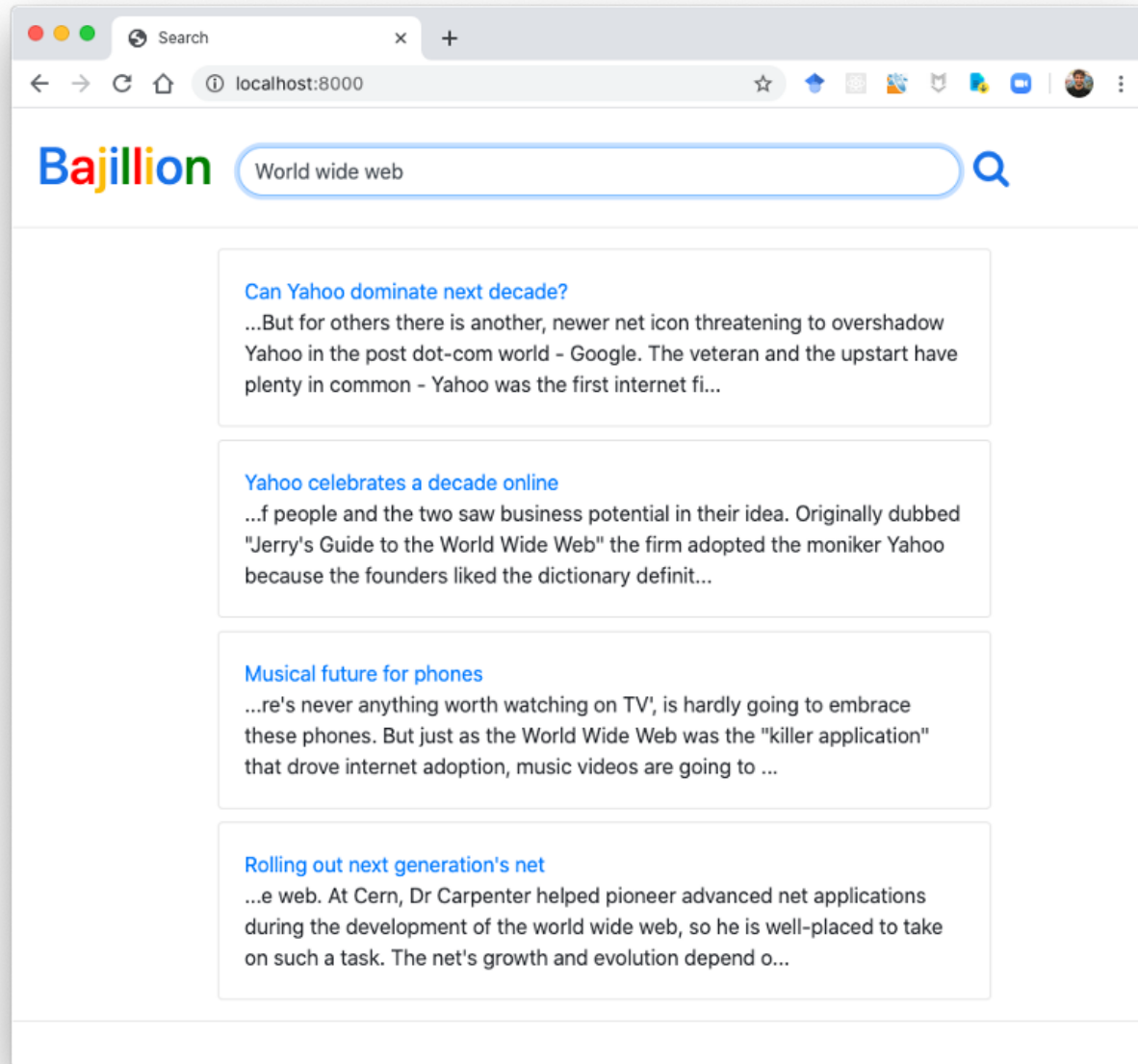
Image Transformation



Data Science



Internet Applications



Strive for Everyone to Succeed



Lets Get Started



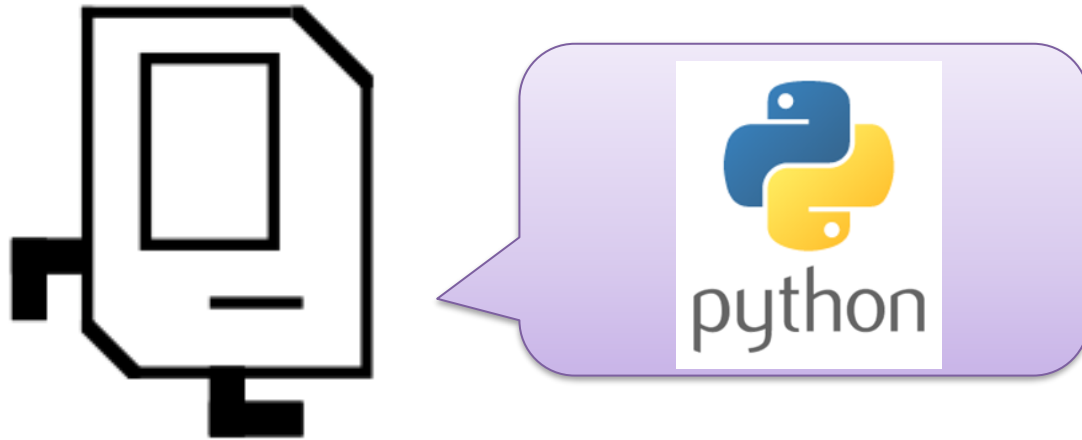
Meet Karel the Robot



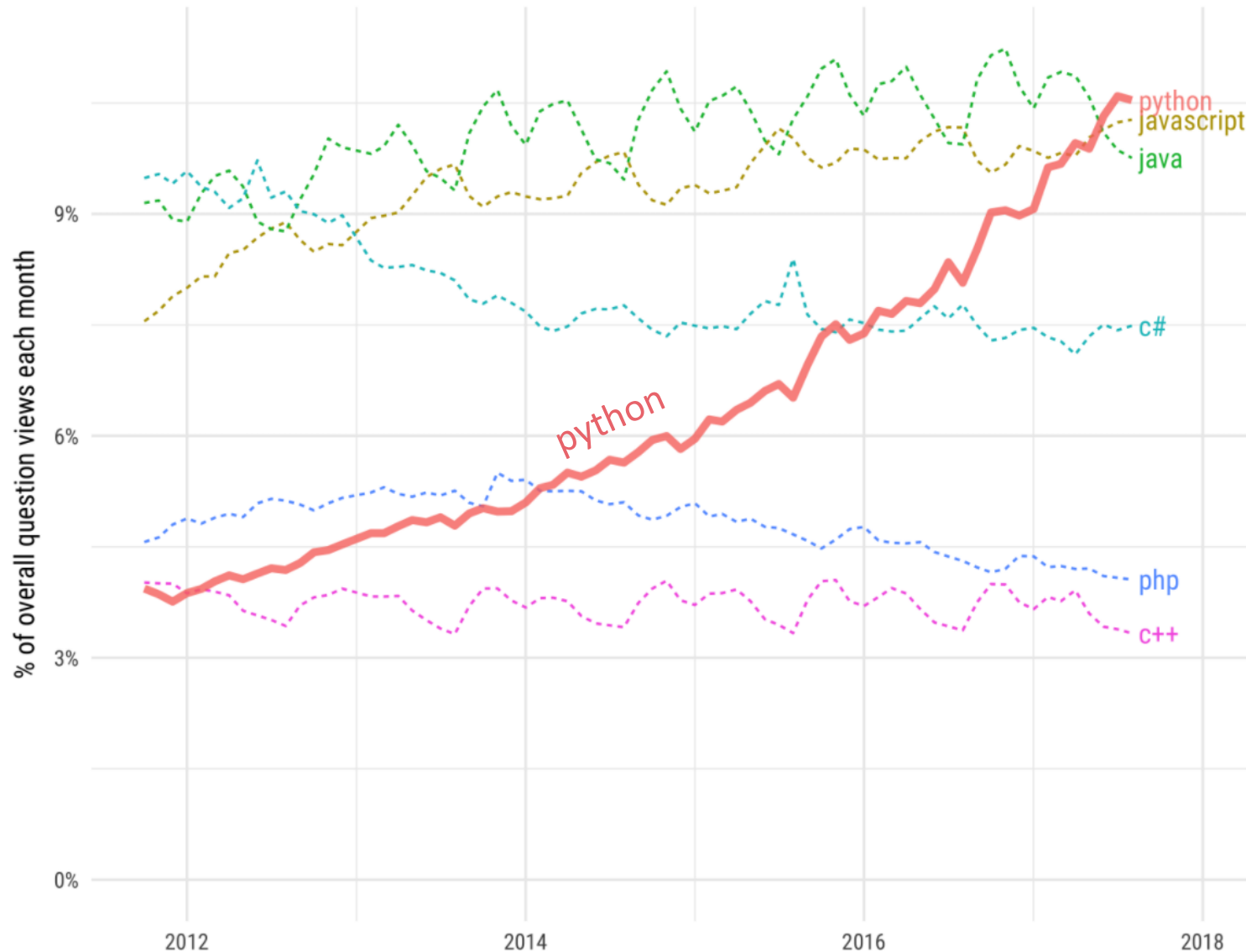
Good morning



Karel Speaks Python



Why Python?

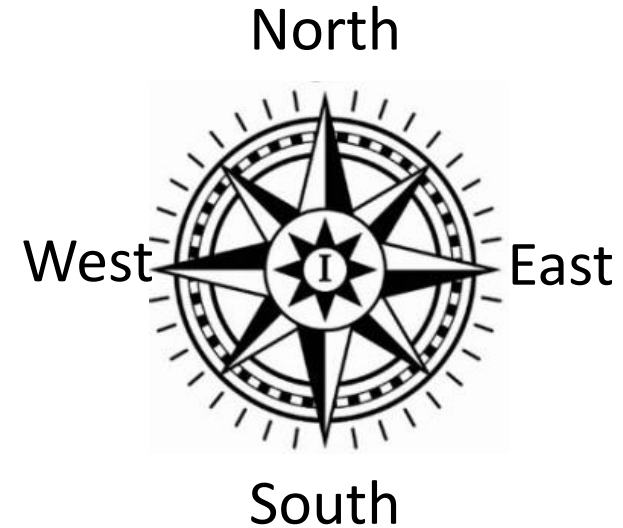


<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

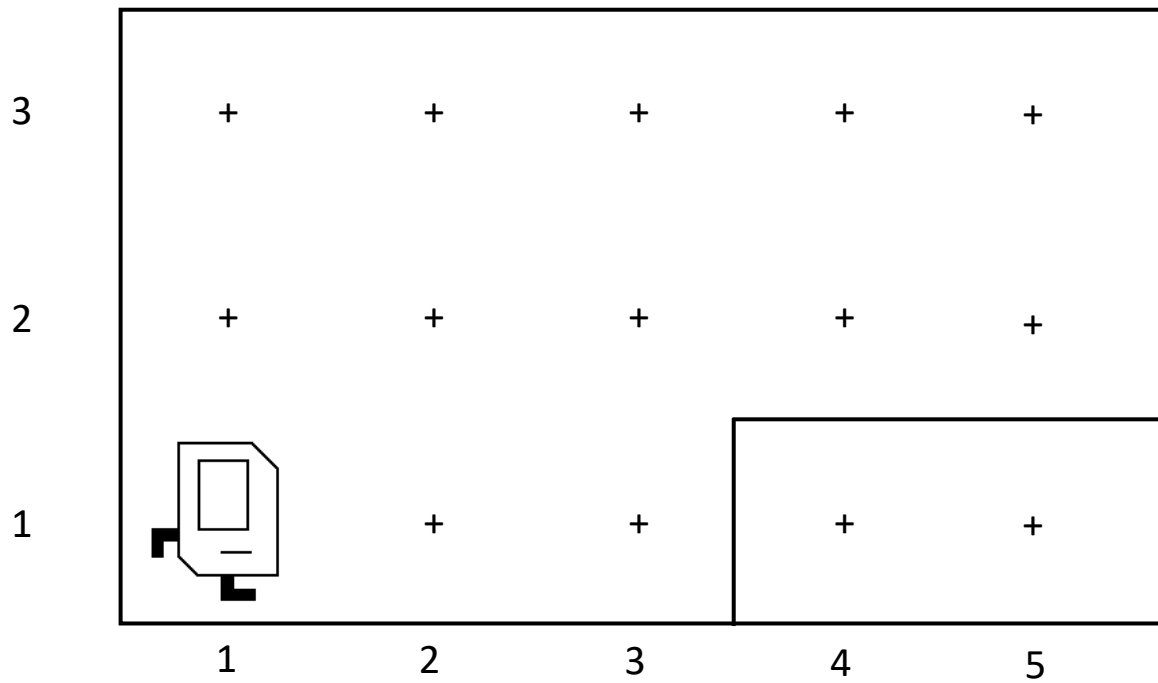
Guido van Rossum



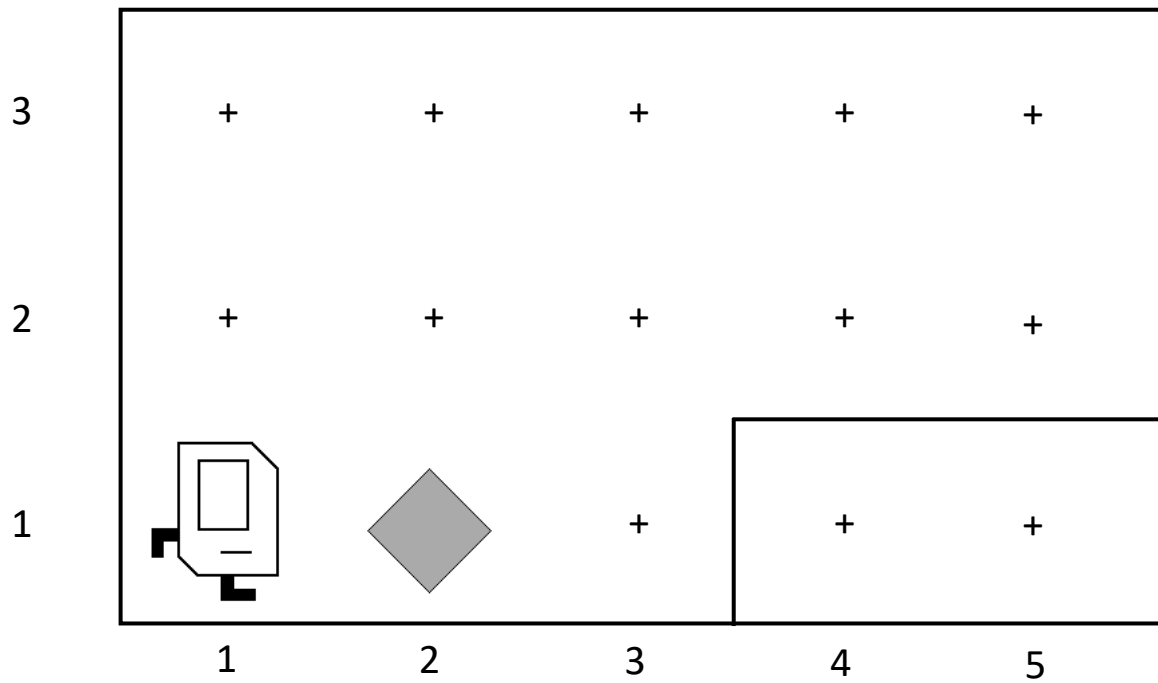
Karel's World



Walls



Beeper's



Knows Four Commands



`move()`

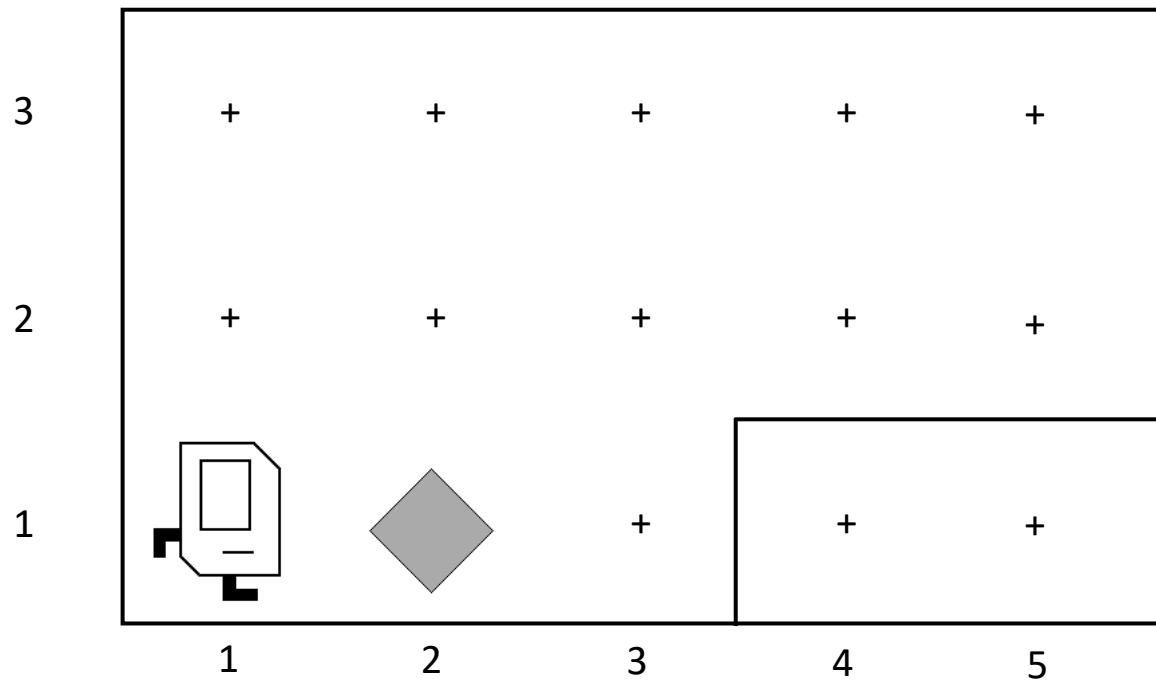
`turn_left()`

`put_beeper()`

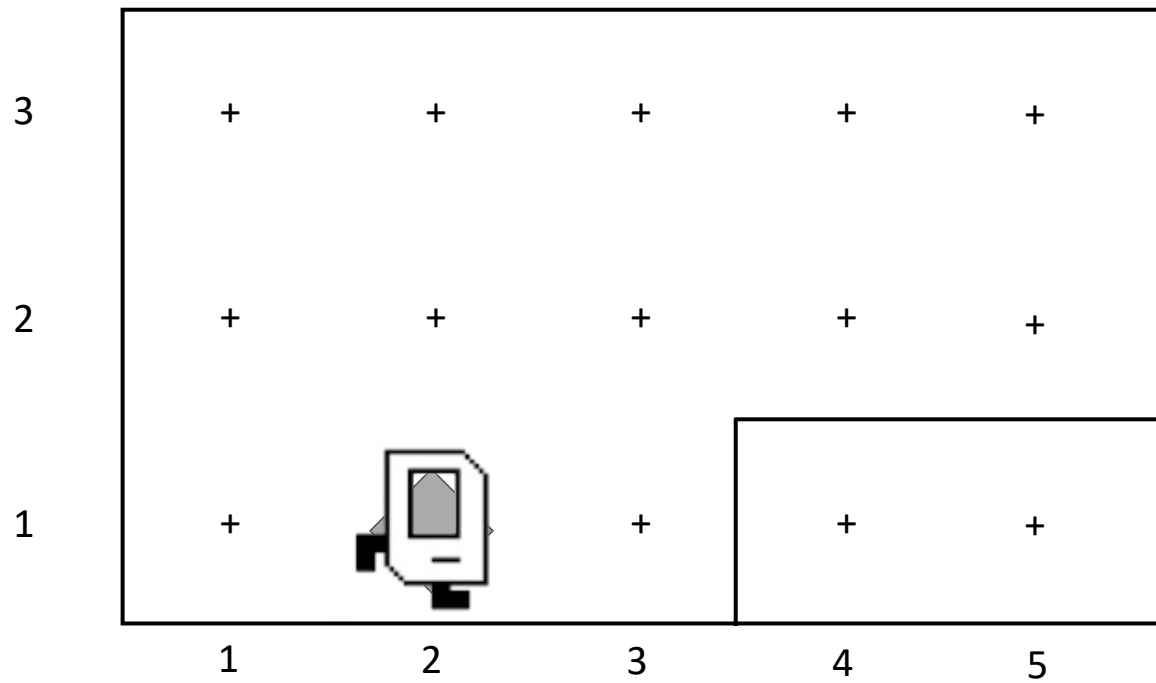
`pick_beeper()`

move ()

move ()

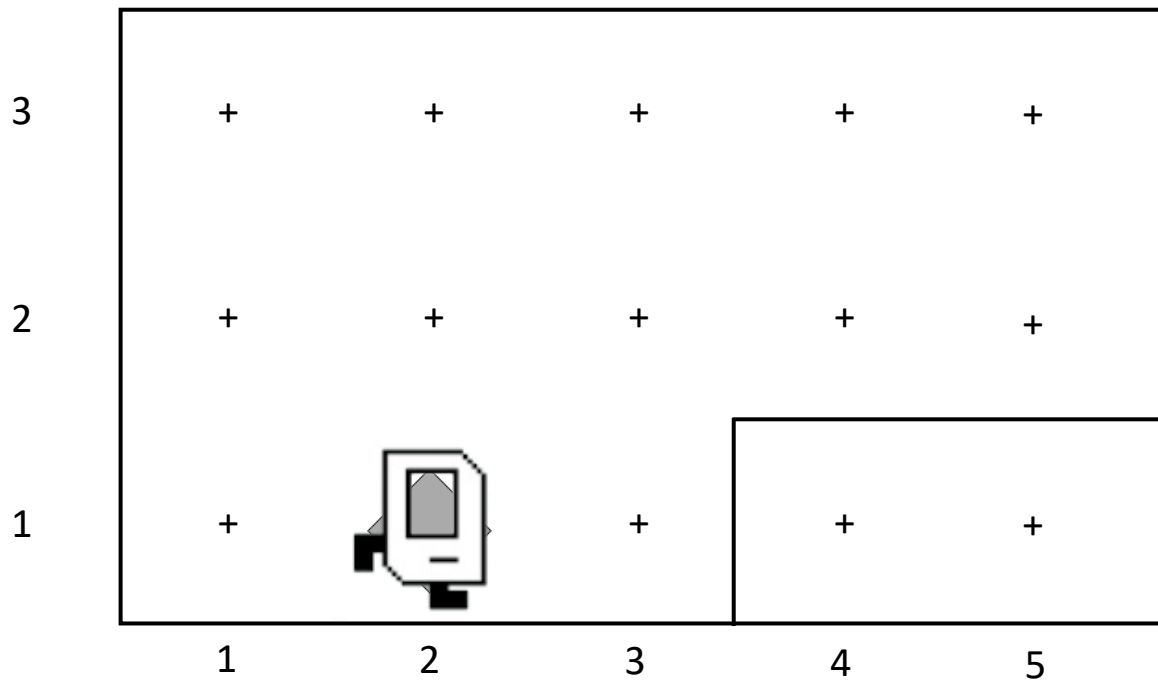


move ()

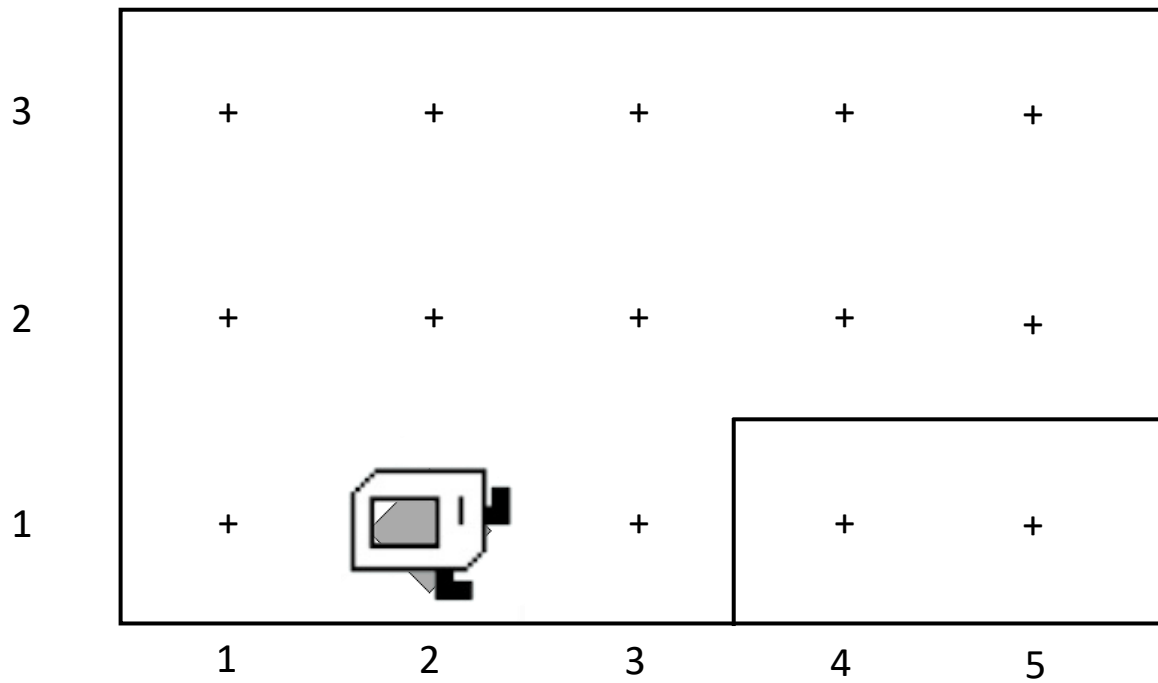


```
turn_left()
```

turn_left()

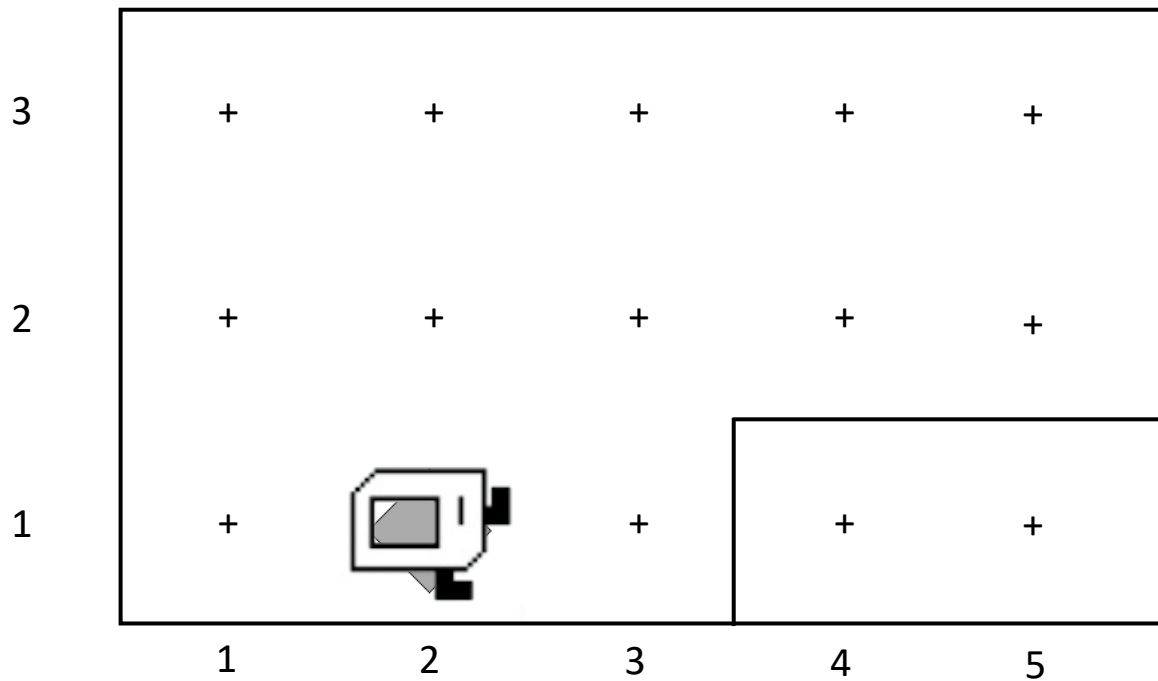


turn_left()

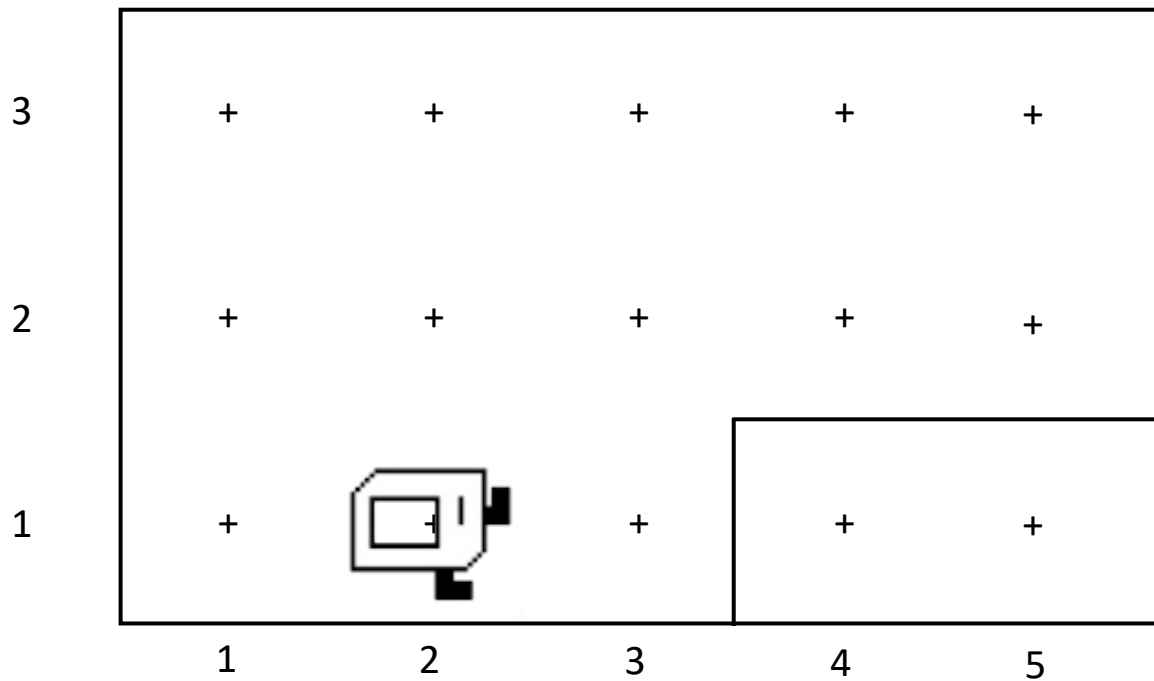


```
pick_beeper()
```

turn_left()

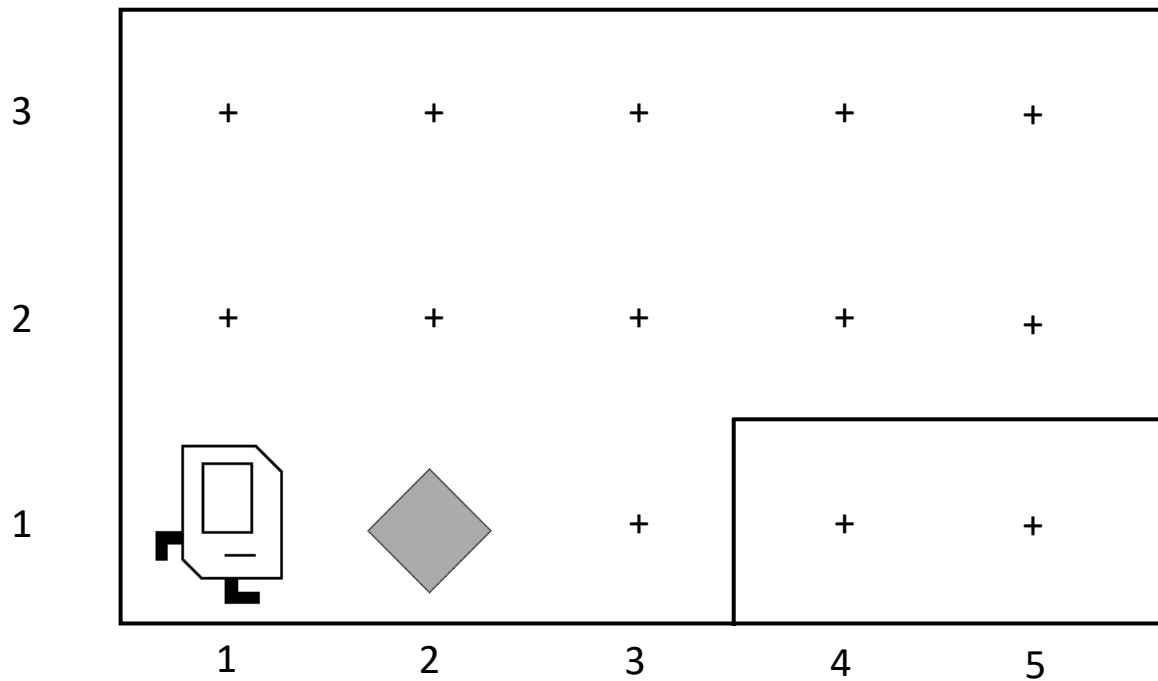


turn_left()

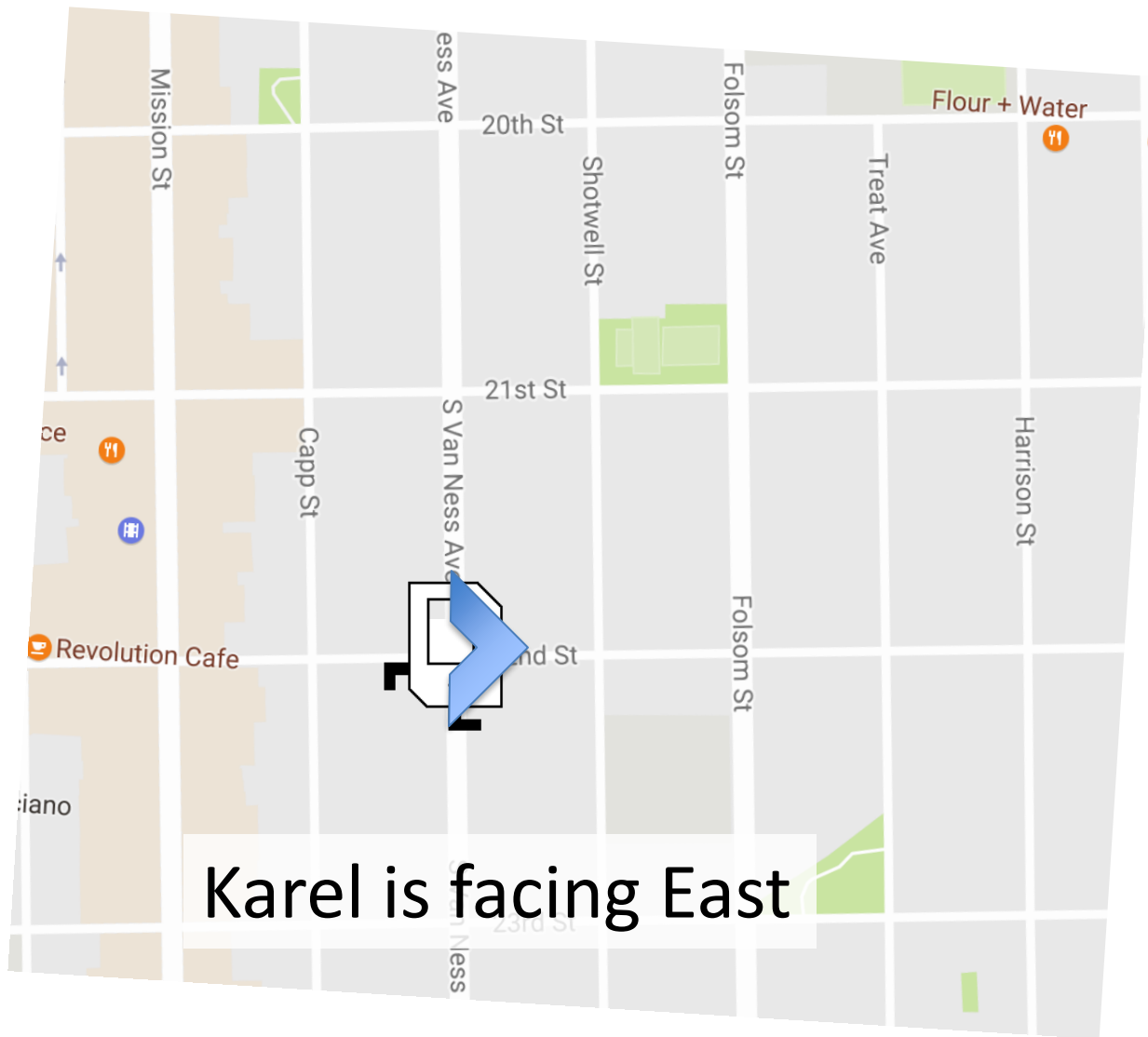


Make Sense?

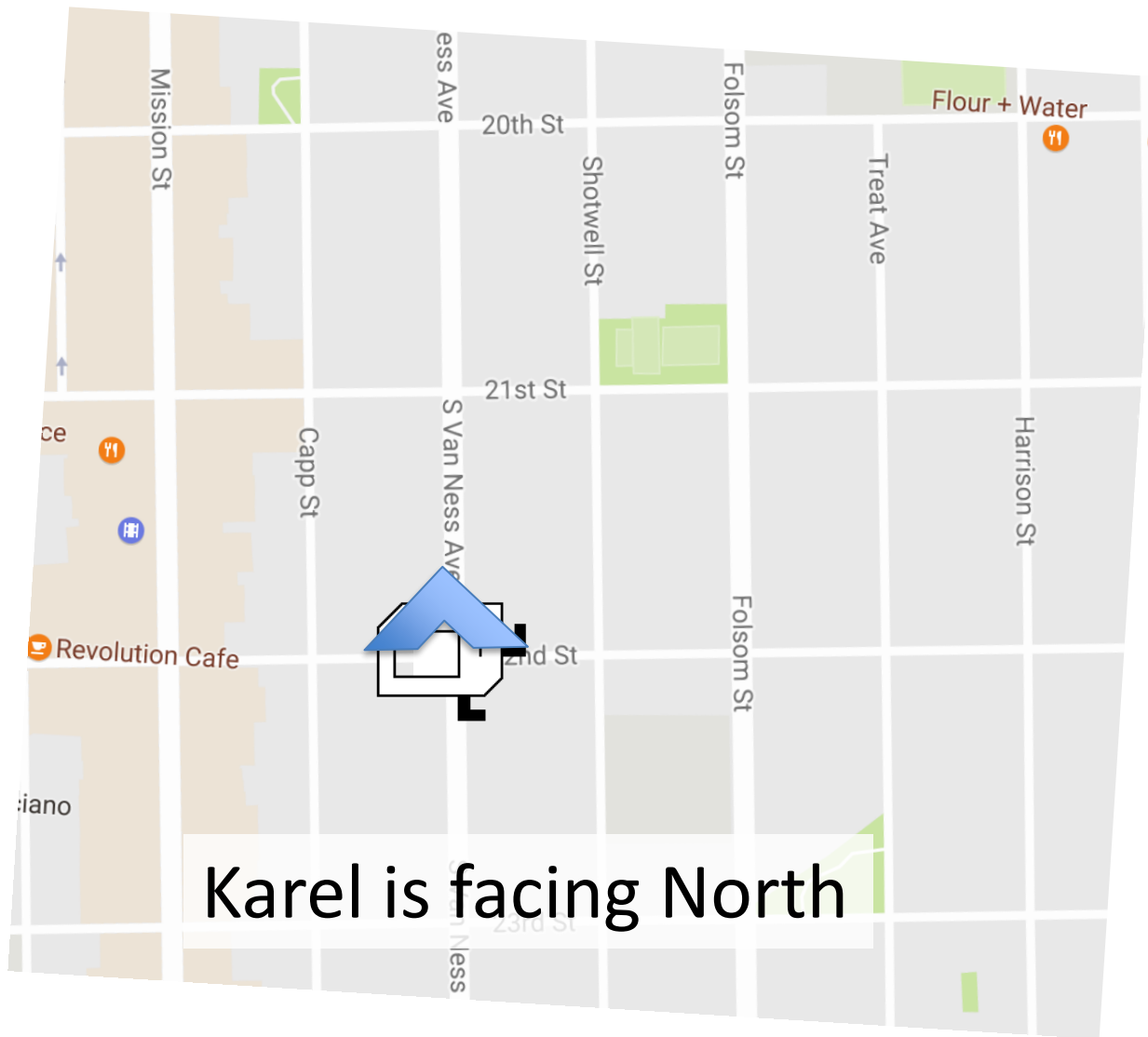
Bird's Eye View



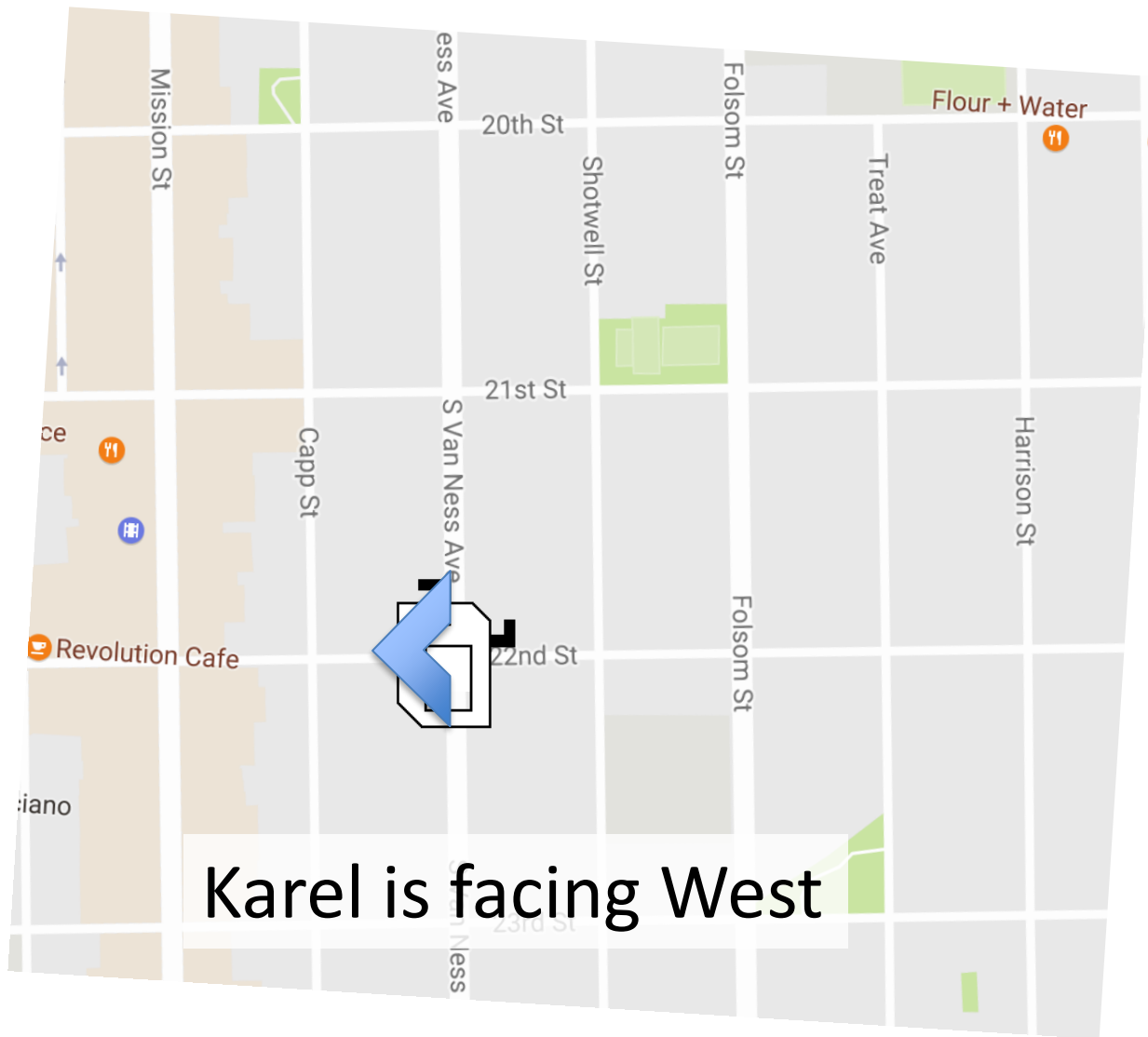
Bird's Eye View



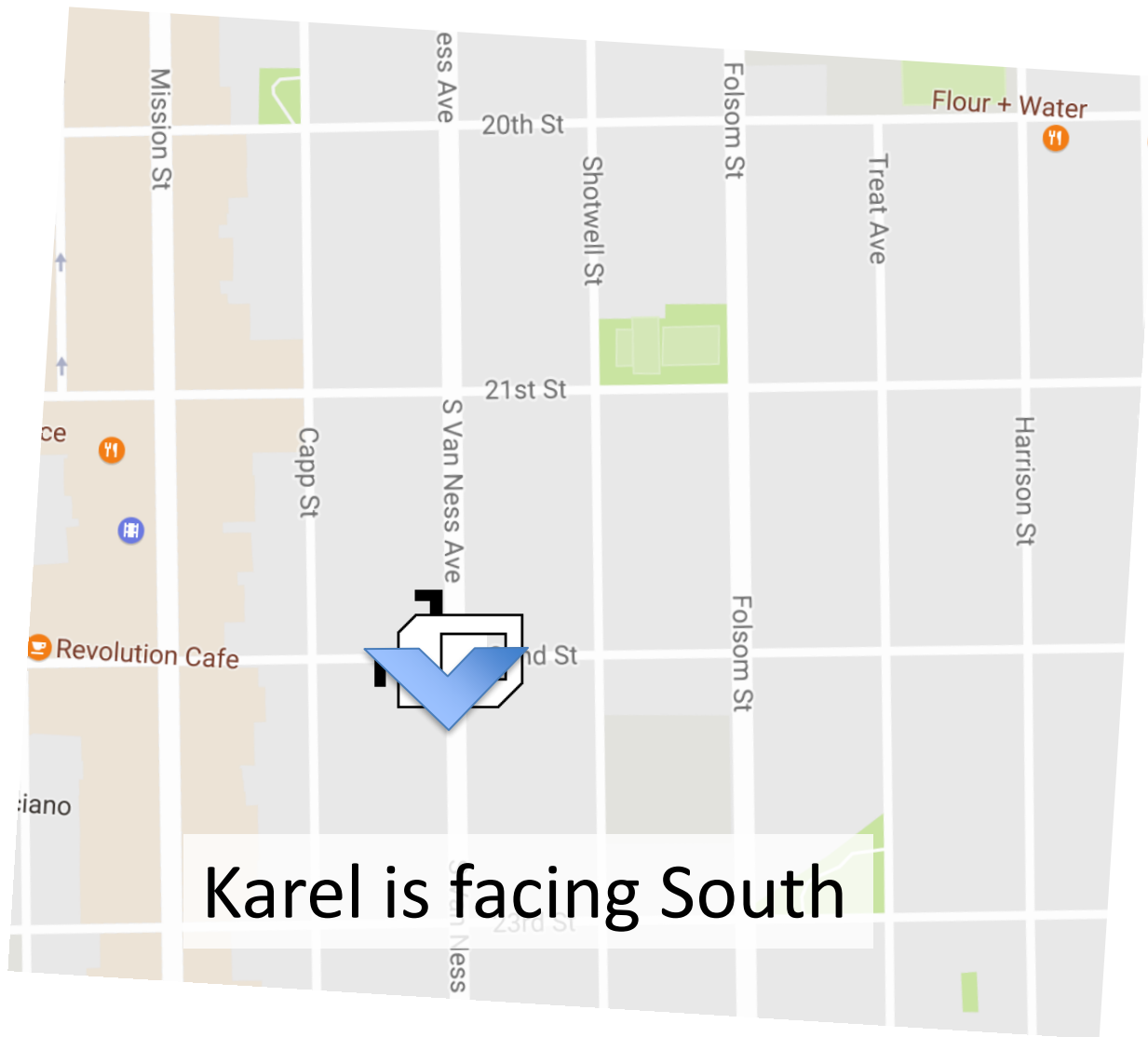
Turn Left



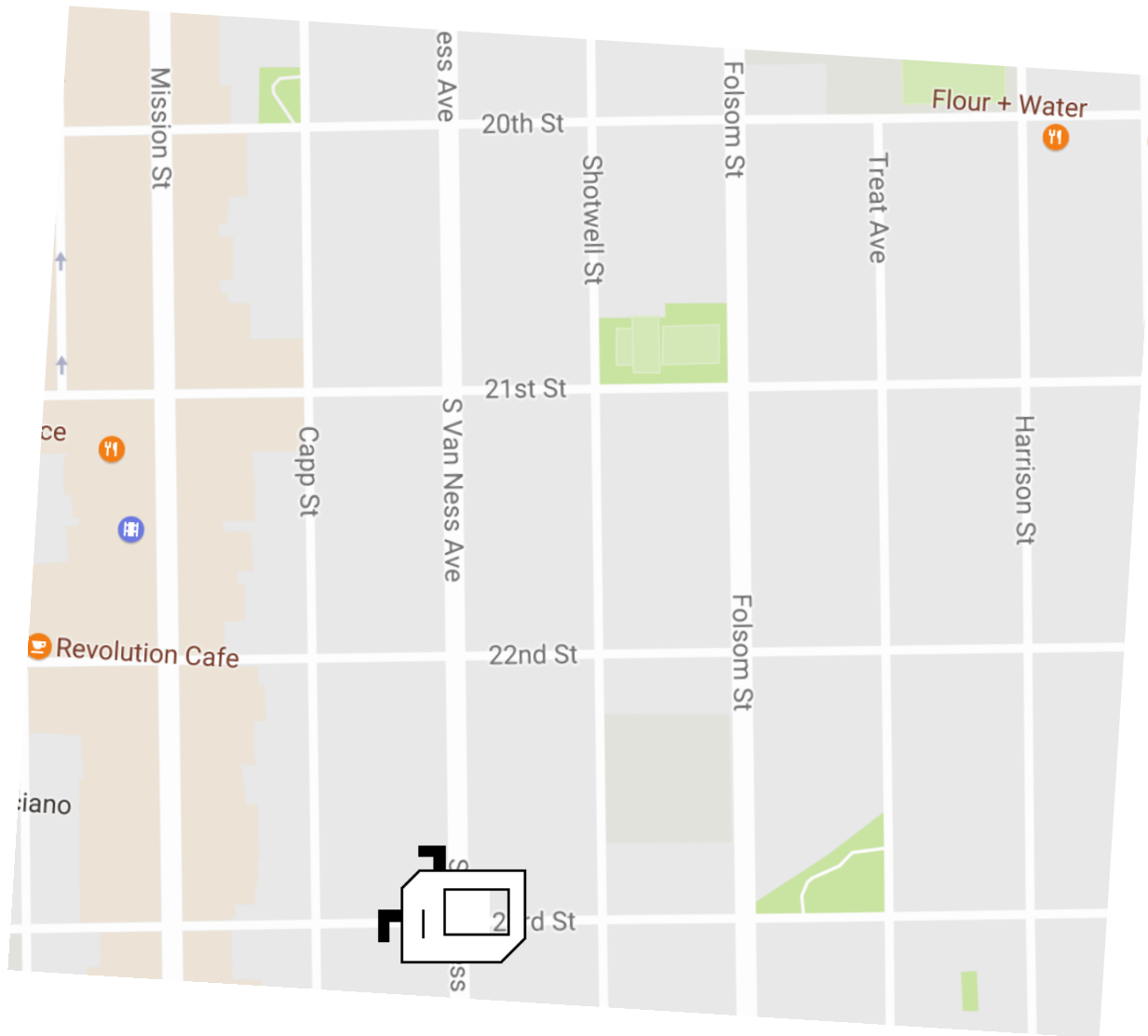
Turn Left



Turn Left



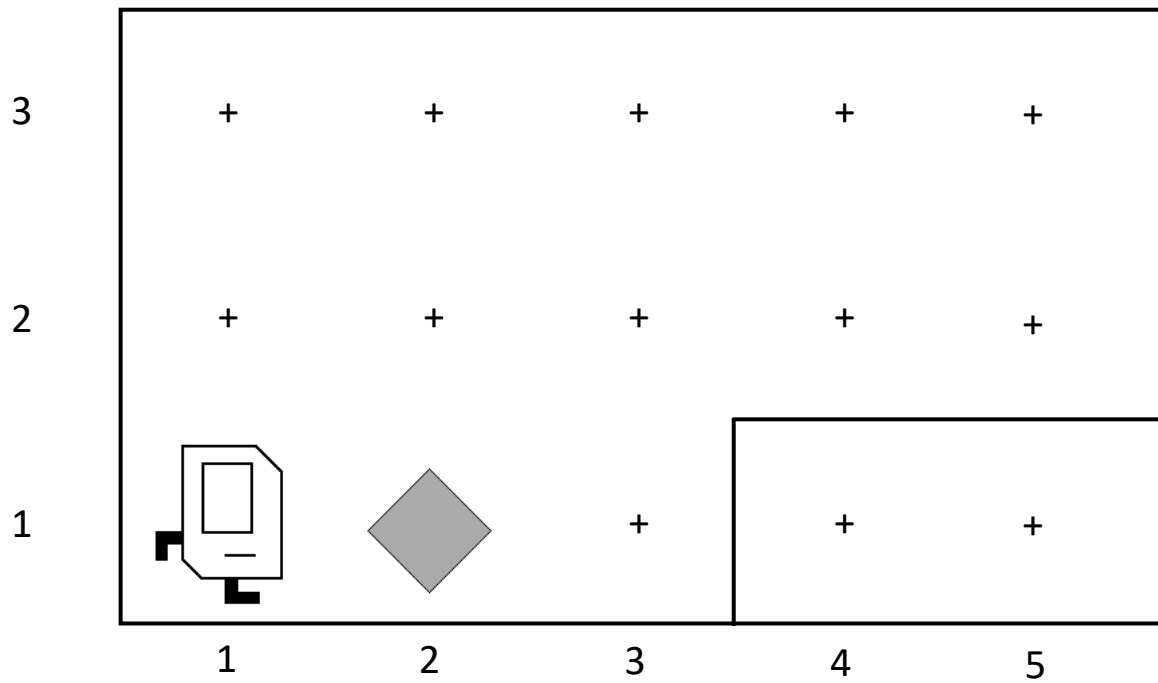
Move



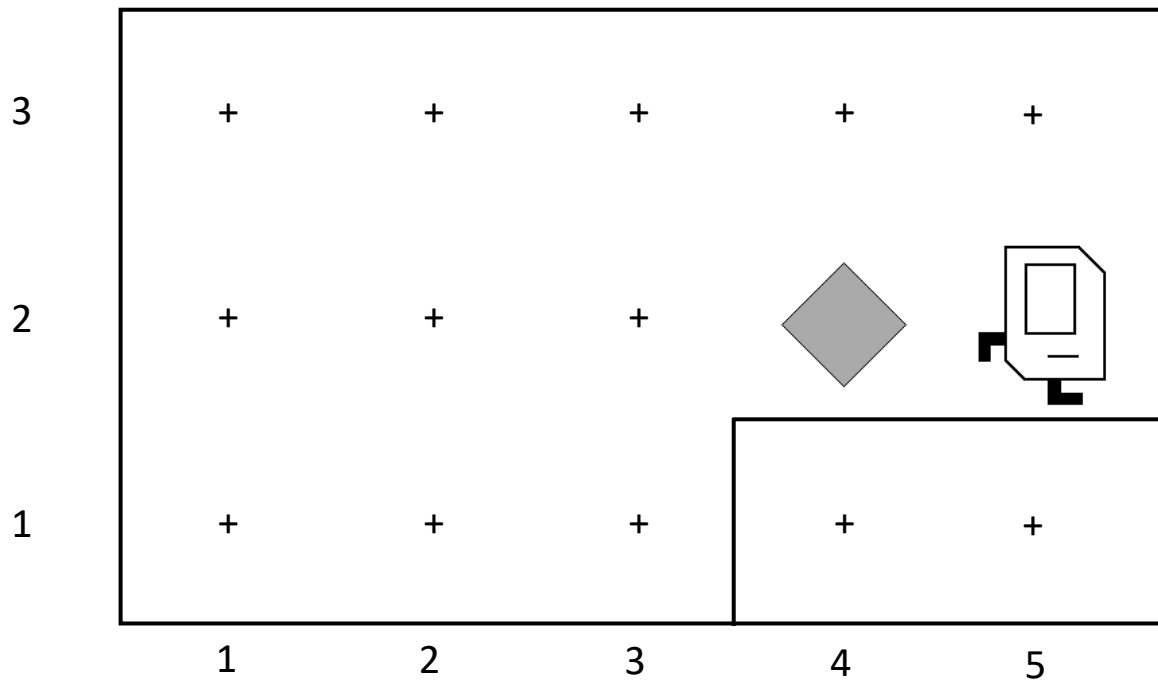
Learn By Doing



First Challenge



First Challenge



Function Definition

```
def name() :  
    function statements
```

This adds a new
command to Karel's
vocabulary

Anatomy of a Program

Import Packages

Program

Anatomy of a Program

Import Packages

Anatomy of a Program

Import Packages

main function

helper functions

start program

Anatomy of a Program

Import Packages

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

helper functions

start program

Anatomy of a Program

Import Packages

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()  
  
def turn_right():  
    turn_left()  
    turn_left()  
    turn_left()
```

start program

Anatomy of a Program

Import Packages

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()  
  
def turn_right():  
    turn_left()  
    turn_left()  
    turn_left()  
  
if __name__ == "__main__":  
    run_karel_program()
```

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

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

```
if __name__ == "__main__":  
    run_karel_program()
```

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

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

```
if __name__ == "__main__":  
    run_karel_program()
```

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

This piece of the program's
source code is called a
function.

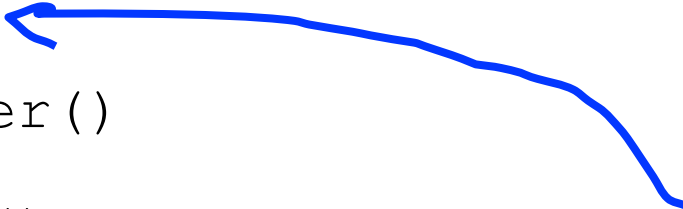
```
def turn_right():  
    turn_left()  
    turn_left()  
    turn_left()
```

```
if __name__ == "__main__":  
    run_karel_program()
```


Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```



This line of code gives the ***name*** of the function
(here, the name is: **main**)

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

```
if __name__ == "__main__":  
    run_karel_program()
```

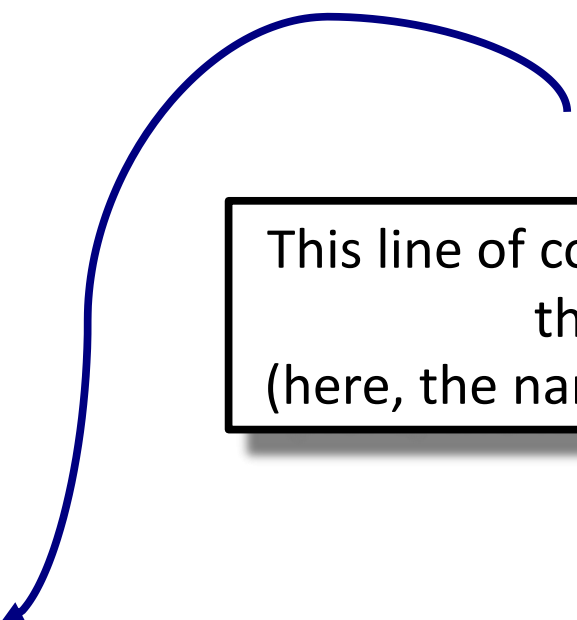
Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

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

```
if __name__ == "__main__":  
    run_karel_program()
```



This line of code gives the *name* of the function
(here, the name is: **turn_right**)

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():
```

```
    move()
    pick_beeper()
    move()
    turn_left()
    move()
    turn_right()
    move()
    put_beeper()
    move()
```

This is called a *code block*
(Note the indenting)

```
def turn_right():
```

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

```
if __name__ == "__main__":
    run_karel_program()
```

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

This is called a *code block*
(Note the indenting)

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

```
if __name__ == "__main__":  
    run_karel_program()
```

Anatomy of a Program

```
from karel.stanfordkarel import *
```

```
def main():  
    move()  
    pick_beeper()  
    move()  
    turn_left()  
    move()  
    turn_right()  
    move()  
    put_beeper()  
    move()
```

This is called a *code block*
(Note the indenting)

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

```
if __name__ == "__main__":  
    run_karel_program()
```

Why Study CS?

Joy of Building



Interdisciplinary



Closest Thing To Magic



Everyone is Welcome



The End