

# CS106A: Programming Methodology

Midjourney AI “magical portal to Stanford university”

# Mehran Sahami

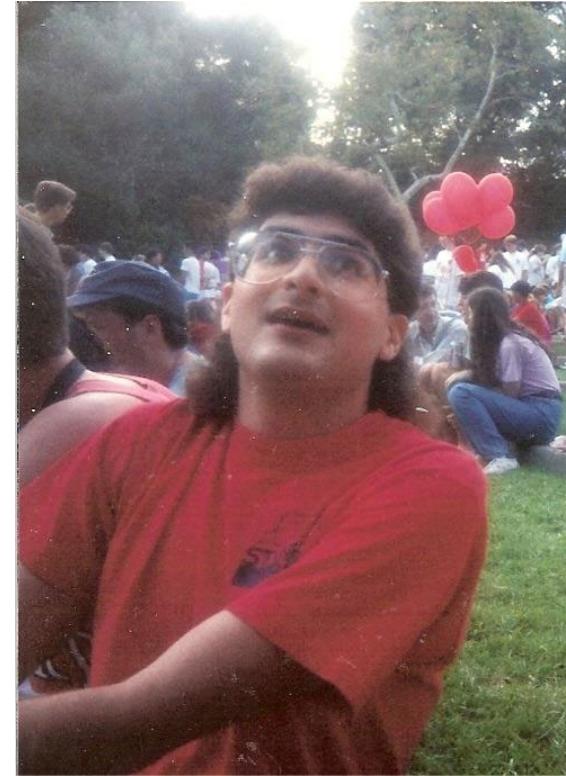


- Childhood: Iran
- High School: San Diego
- Stanford University Ph.D. in Machine Learning  
(Before Machine Learning was cool)
- Spent a decade in tech industry before coming back as professor
  - Love of teaching is why I came back
  - Chair of Computer Science department



# Mehran Sahami

- Took CS106A my freshman year at Stanford
  - It changed my life
- But it did not make me cut my mullet
  - It should have...
- Now, my oldest son is a freshman CS major in college (not here)
  - He does not have a mullet because he has much better fashion sense than I ever had.



# Chris Gregg



- Childhood: Central New York (Finger Lakes)
- B.S., Johns Hopkins, Electrical Engineering
- MEd., Harvard, physics teaching
- Ph.D., University of Virginia, Computer Engineering
- U.S. Navy, 7 years active duty, 15+ years reserves, retired as a Commander
- Taught high school for 7 years, Brookline, MA, Santa Cruz, CA
- At Stanford for nine years
- I love to tinker! Ask me about my typewriter projects ([#1](#), [#2](#))!
- Associate Chair for Education, Computer Science Department



# Chris Gregg



- I have a five-year old daughter, Celeste, and two whippets, Jupiter and Juno
- Celeste loves Greek Mythology
- She hasn't started coding...yet

Sahami and Gregg, CS106A, Stanford University



# Head TA: Ngoc Nguyen

- Born in Vietnam, grew up in Kentucky!
- Studied Symbolic Systems
- Currently getting a Master's in Computer Science!



I ate where it all began at the original KFC



Sahami and Gregg, CS106A, Stanford University



# Section Leaders



Clinton



Colin



Daniel



Daniel



Dev



Dwight



Eric



Esteban



Evan



Francesca



Francisco



Grant



Hannah



Jack



Jason



Jayendra



Jennifer



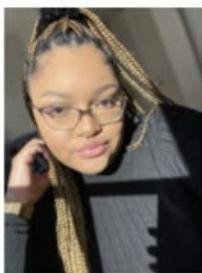
Jerry



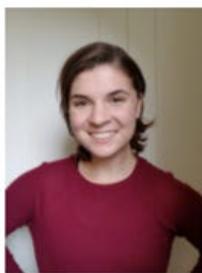
Jin-Hee



Julie



Kayla



Laura



Lauren



Minh

\* Actually some past section leaders

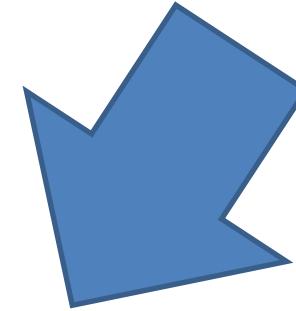
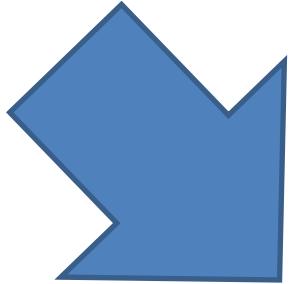
Sahami and Gregg, CS106A, Stanford University



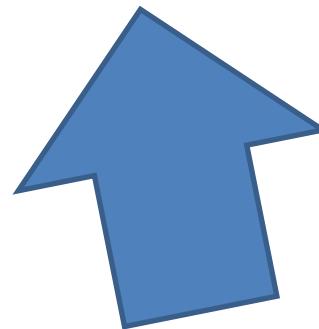
Sahami and Gregg, CS106A, Stanford University



# Course Website



<http://cs106a.stanford.edu>



# Prerequisite Test

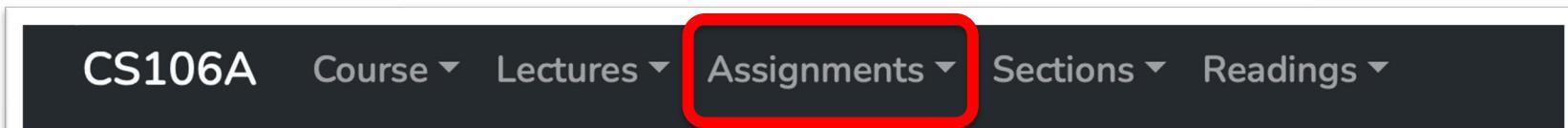


Sahami and Gregg, CS106A, Stanford University



# Getting To Know You

- Assignment #0 on website (“Tell us about yourself”)



CS106A Course ▾ Lectures ▾ **Assignments ▾** Sections ▾ Readings ▾



**CS106A: Programming Methodologies**  
Spring 2025  
Monday, Wednesday, Friday 2:30pm - 3:20pm in-person in [Hewlett 200](#)

“It is a really difficult time for my family right now, but I will still be trying my best in this class.”

- Please be safe, compassionate, and kind. So will we.



# Lectures and Sections

- Lectures MWF 2:30pm-3:20pm
  - Will be recorded (available on Canvas)
- Weekly 50-min section led by awesome section leaders (the backbone of the class!)
  - Section signups will be on class webpage (not Axess)
  - Signups begin on Thursday at 5pm and close Sunday at 5pm.



Sahami and Gregg, CS106A, Stanford University



# LaIR Hours in Durand Building



CS in Durand

1. In this door with card
2. Elevator to 3rd floor
3. To your right exiting elevator

LaIR helper hours: evenings Sunday through Thursday  
(starting Sunday)

Sahami and Gregg, CS106A, Stanford University



# Grading Scale

**Functionality** and **style** grades for the assignments use the following scale:

- ++** A submission so good it “makes you weep”
- +** Exceeds requirements (and has great style)
- ✓ +** Satisfies all requirements, with good functionality and style
- ✓** Meets the requirements, but perhaps with small problems
- ✓ -** Has some somewhat serious problems
- Is worse than that, but shows real effort and understanding
- Better than nothing



# Interactive Grading



One on one feedback  
from your section leader

- Chance for you to get more feedback than just a grade
- Opportunity to really develop “style” as a programmer
  - We’ll talk more about that soon
- We can put focus on *learning* rather than grading



# What we will ask you to do

- 6 programming assignments 40%
- Diagnostic exam 20%
- Final exam 35%
- Section participation 5%

Get 4 free “late days” (on assignments).  
Each late day is a 24-hour period



# Optional Contest



Sahami and Gregg, CS106A, Stanford University



# Online Text Books

A screenshot of a web browser window titled "Karel Reader". The URL in the address bar is "compedu.stanford.edu/karel-reader/docs/python/en/intro.html". The main content area displays the title "Karel the Robot" in large font, followed by an illustration of Karel the Robot facing right, with the Python logo integrated into its body. Below this, the text "Learns Python" is displayed. At the bottom left, there is a sidebar with a list of chapters: 1 - Meet Karel, 2 - Programming, 3 - New Functions, 4 - Decomposition, 5 - For Loops, 6 - While Loops, 7 - Conditionals, 8 - Refinement, 9 - SuperKarel, 10 - Reference, and 11 - Code. A blue button at the bottom right says "► Get Started".



# Online Karel Reader

Karel Reader

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

## 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

Press the "Run" button to execute the program. Programs are typically written in a special application called an **Integrated Development Environment** (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.

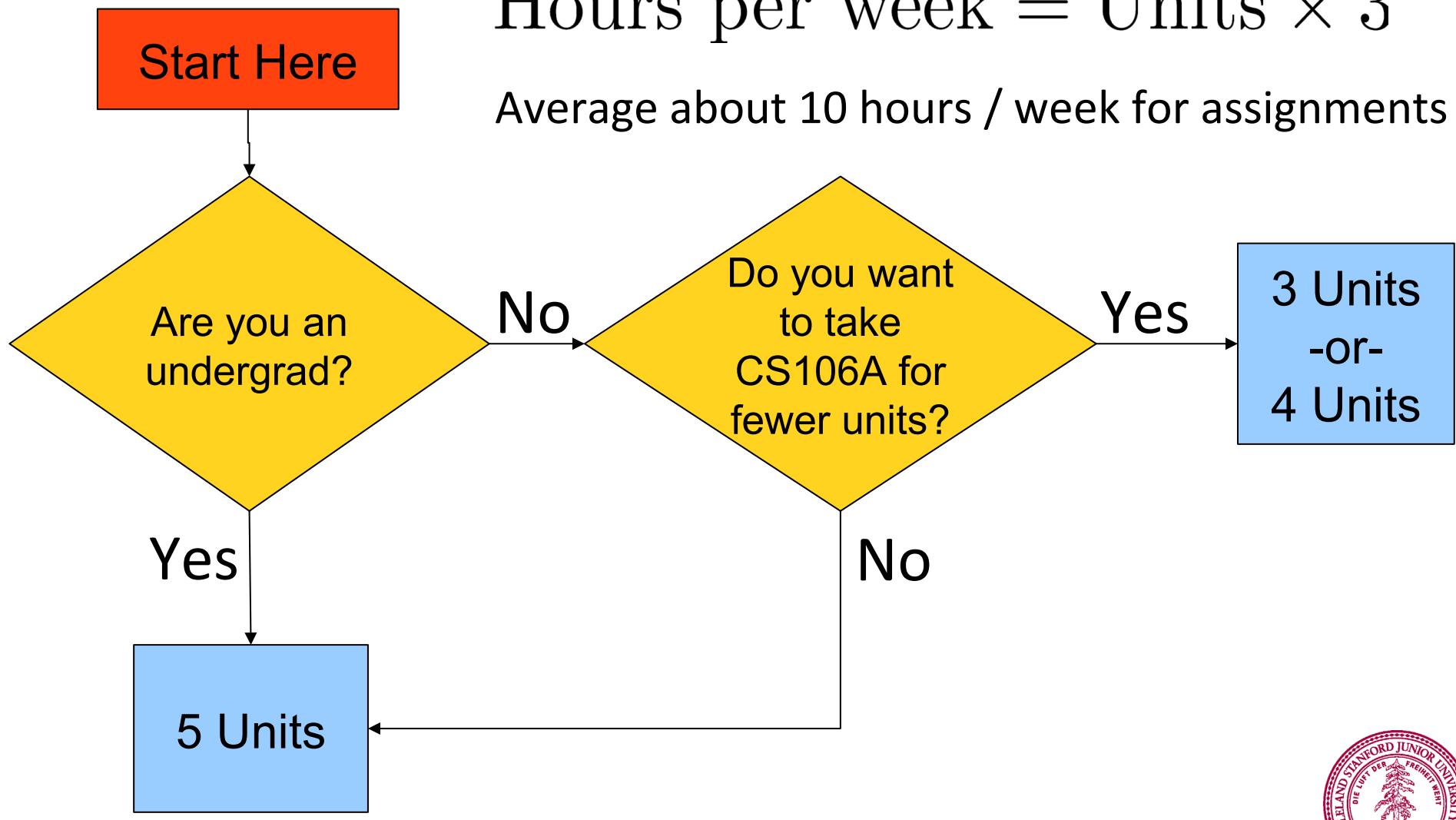
The program is composed of several parts. The first part consists of the following lines:

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



# CS106A Units

Hours per week = Units  $\times$  3  
Average about 10 hours / week for assignments





# Where Should You Start?

- No/light previous programming (many students start here) → CS106A
- Limited previous programming (e.g., written “short” programs) → CS106A
- AP exam: CS Principles, score 4 or 5 → CS106A
- AP exam: CS A, score 4 or 5 → CS106B
- No AP, significant previous programming experience → CS106B
- *Extensive* prior experience and/or multiple prior CS classes → CS106B
- Just want to satisfy “Ways” and know that will be all you’ll take → CS105 or 106A



Sahami and Gregg, CS106A, Stanford University



# 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.



Sahami and Gregg, CS106A, Stanford University



# Computer Graphics



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

# Consumer Applications



You have a supercomputer in your pocket! (that tracks everything you do, everywhere you go...)

Sahami and Gregg, CS106A, Stanford University



# Self-Driving Car

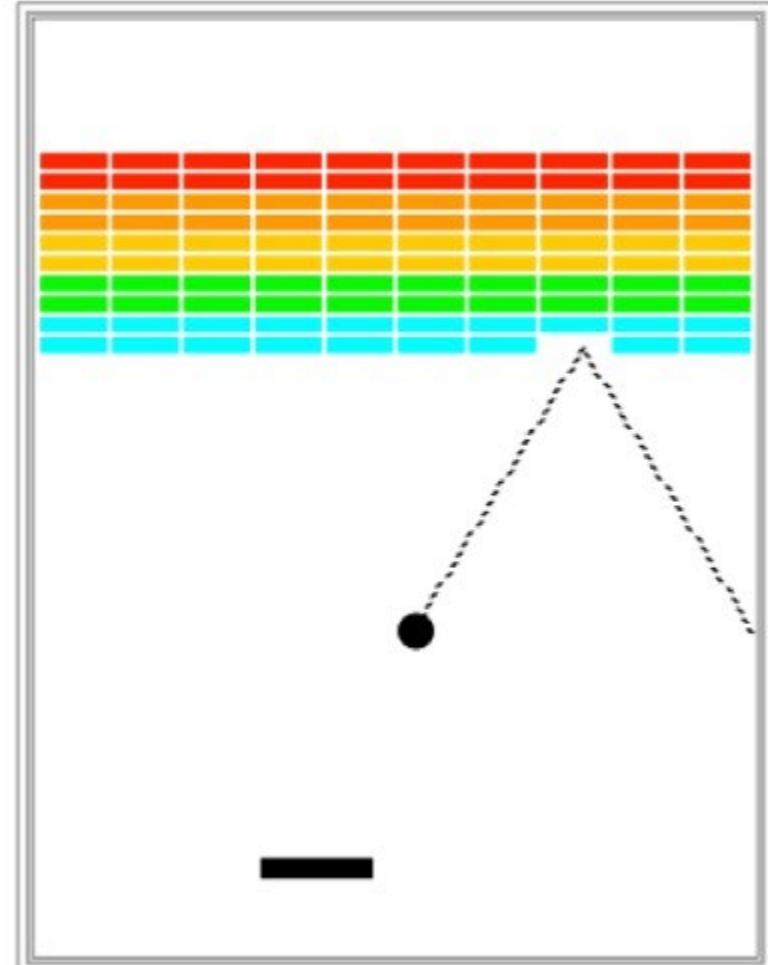
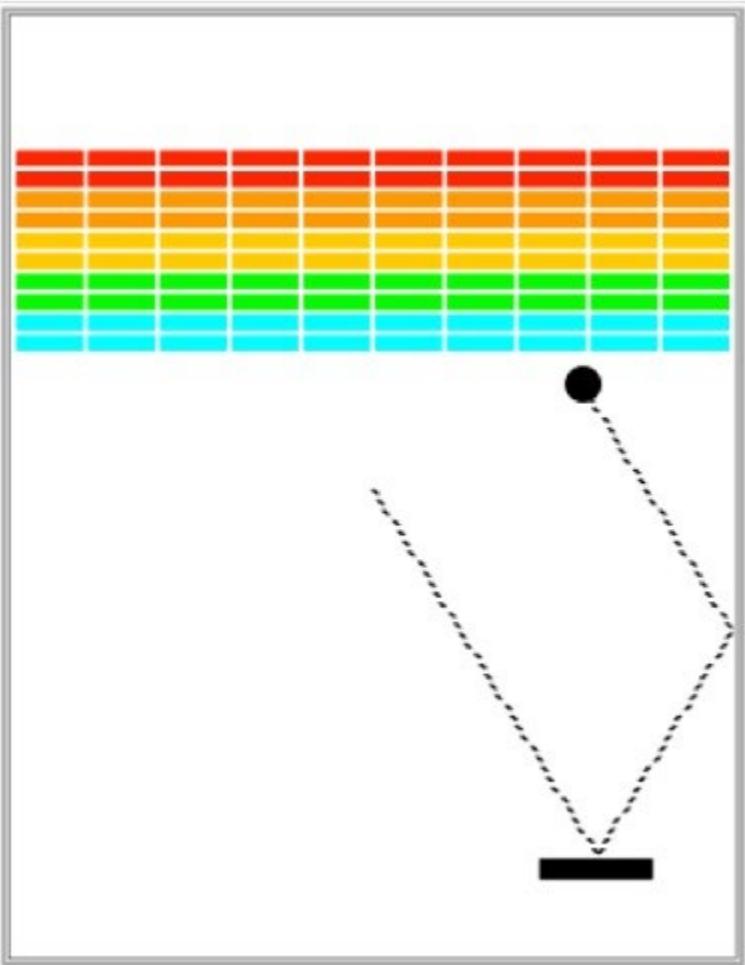


Sahami and Gregg, CS106A, Stanford University

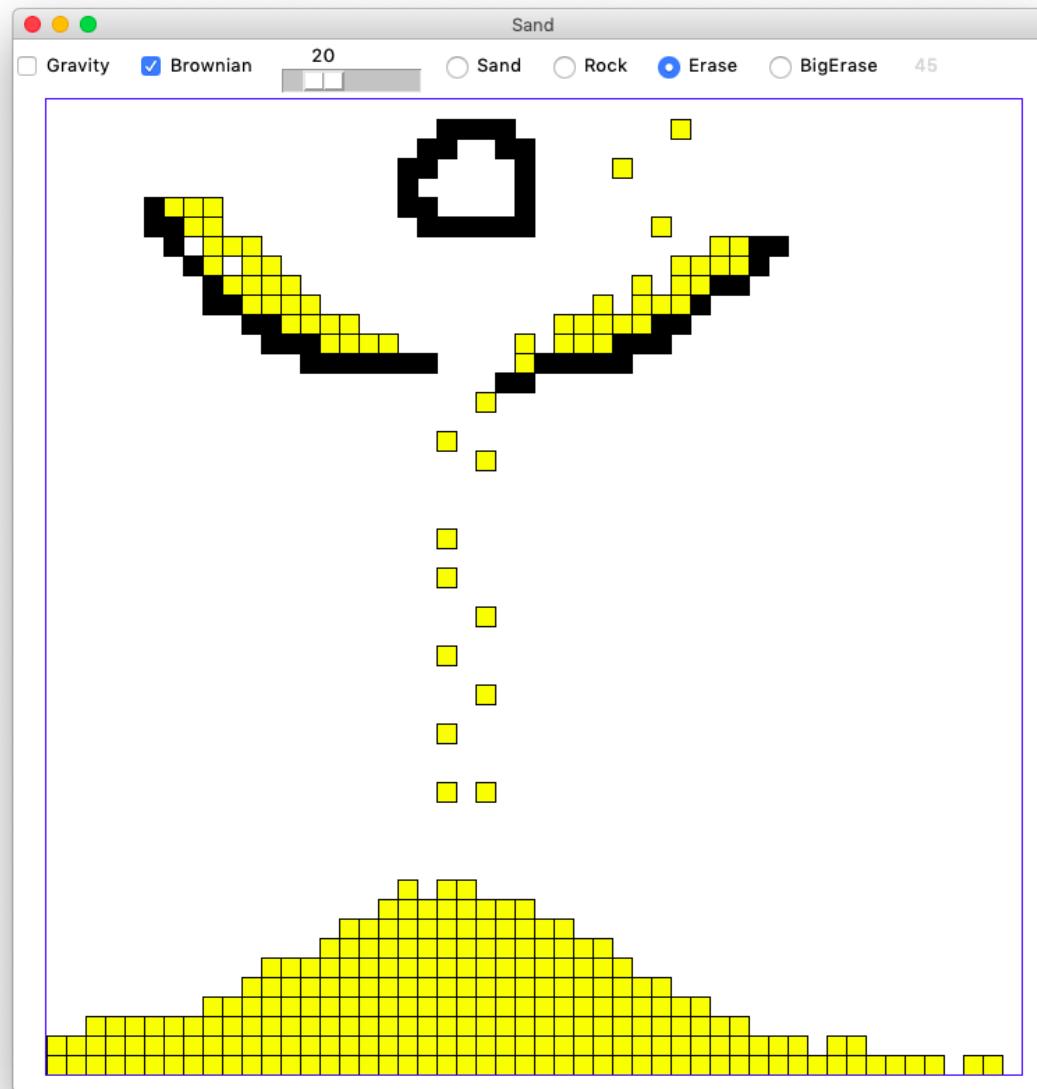




# Graphics and Games



# Simulations



Sahami and Gregg, CS106A, Stanford University



# Internet Applications

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:8000
- Search Bar:** World wide web
- Content Area:** Four search results are displayed in boxes:
  - Can Yahoo dominate next decade?**  
...But for others there is another, newer net icon threatening to overshadow Yahoo in the post dot-com world - Google. The veteran and the upstart have plenty in common - Yahoo was the first internet fi...
  - Yahoo celebrates a decade online**  
...f people and the two saw business potential in their idea. Originally dubbed "Jerry's Guide to the World Wide Web" the firm adopted the moniker Yahoo because the founders liked the dictionary definit...
  - Musical future for phones**  
...re's never anything worth watching on TV', is hardly going to embrace these phones. But just as the World Wide Web was the "killer application" that drove internet adoption, music videos are going to ...
  - Rolling out next generation's net**  
...e web. At Cern, Dr Carpenter helped pioneer advanced net applications during the development of the world wide web, so he is well-placed to take on such a task. The net's growth and evolution depend o...



# Strive for Everyone to Succeed

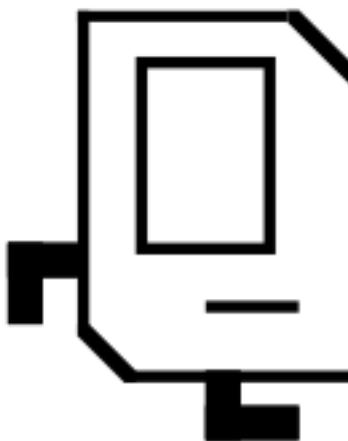


# Lets Get Started



# Meet Karel the Robot

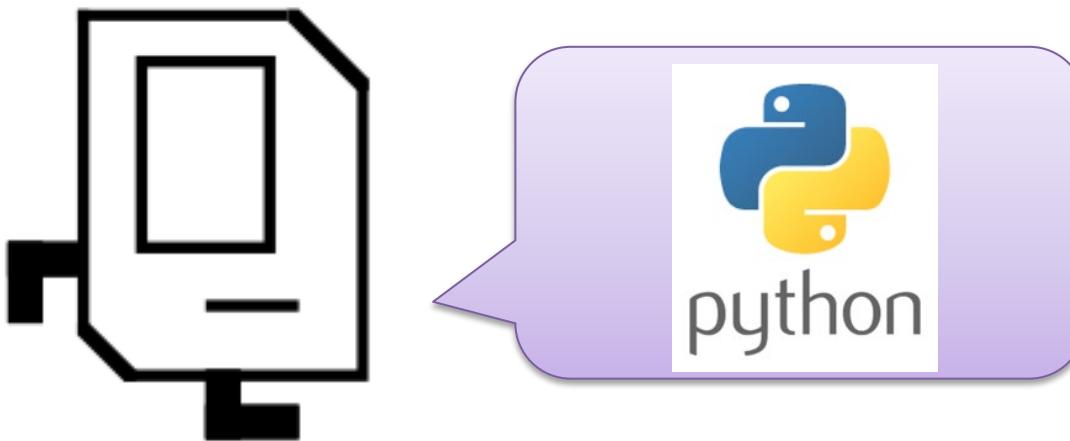
Rich Parris



Good morning



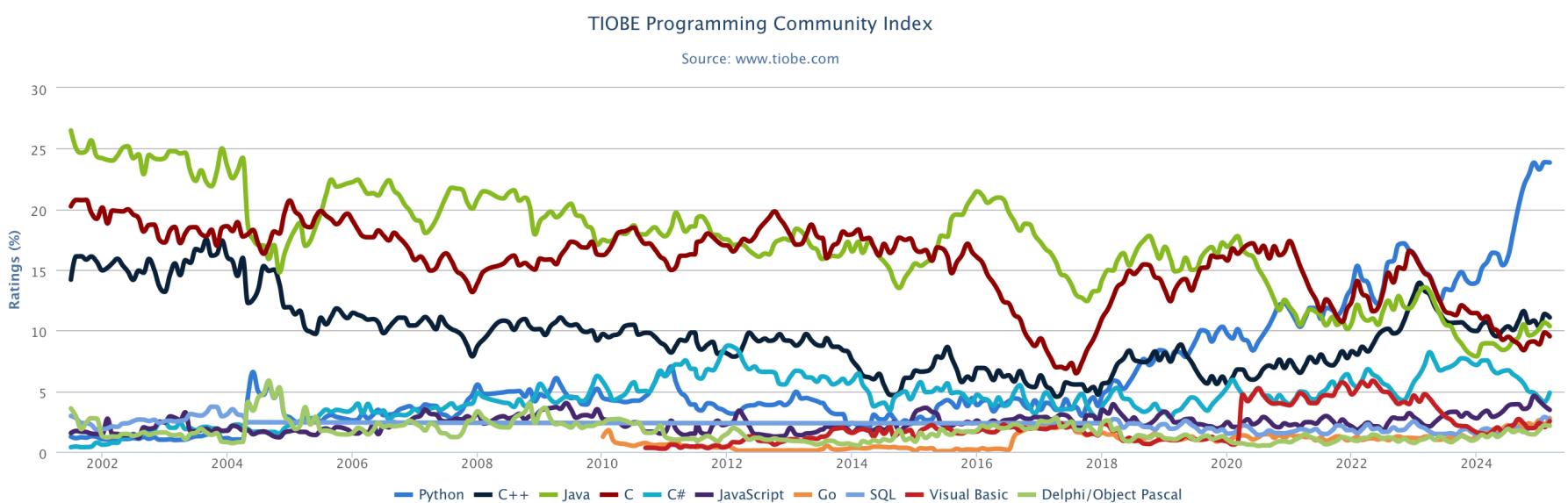
# Karel Speaks Python



Sahami and Gregg, CS106A, Stanford University



# Why Python?



<https://www.tiobe.com/tiobe-index/>



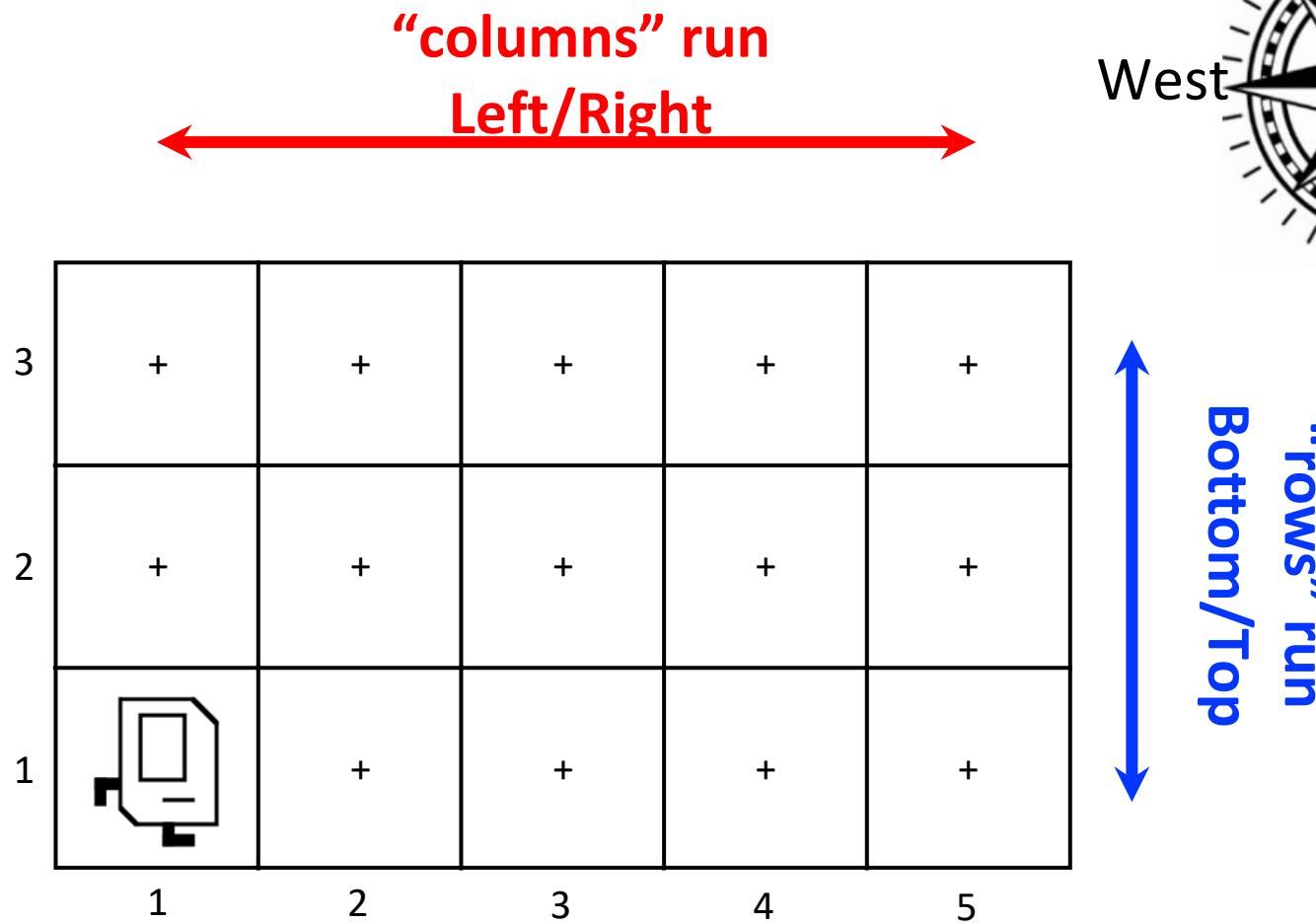
# Guido van Rossum



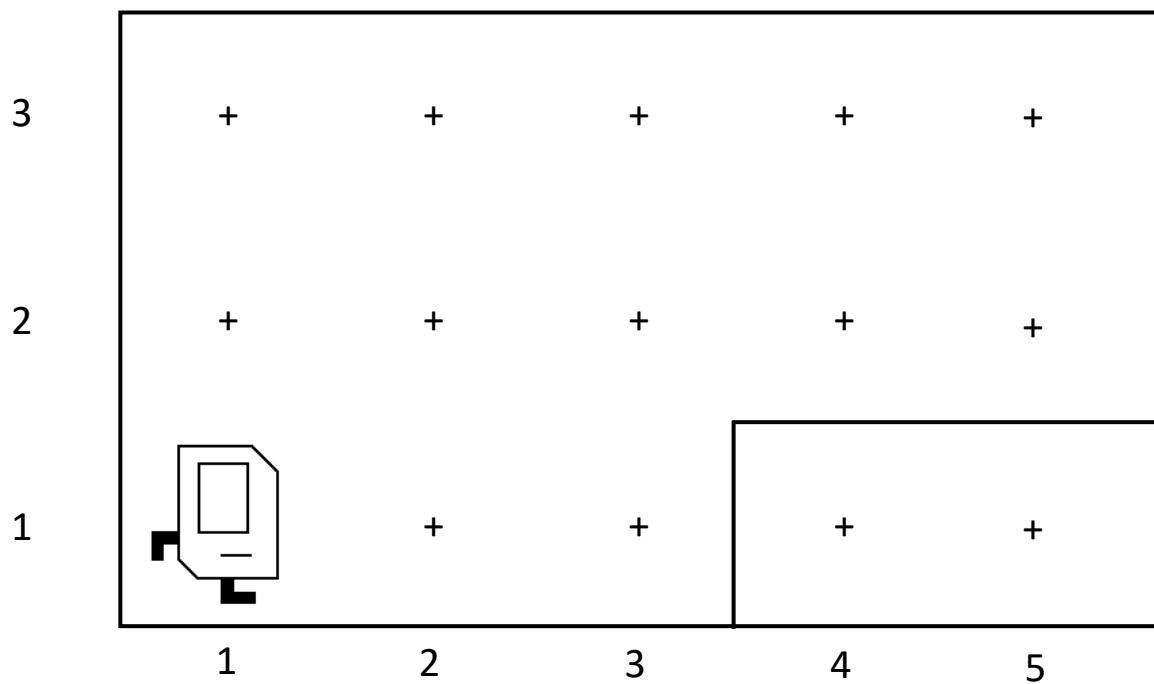
Sahami and Gregg, CS106A, Stanford University



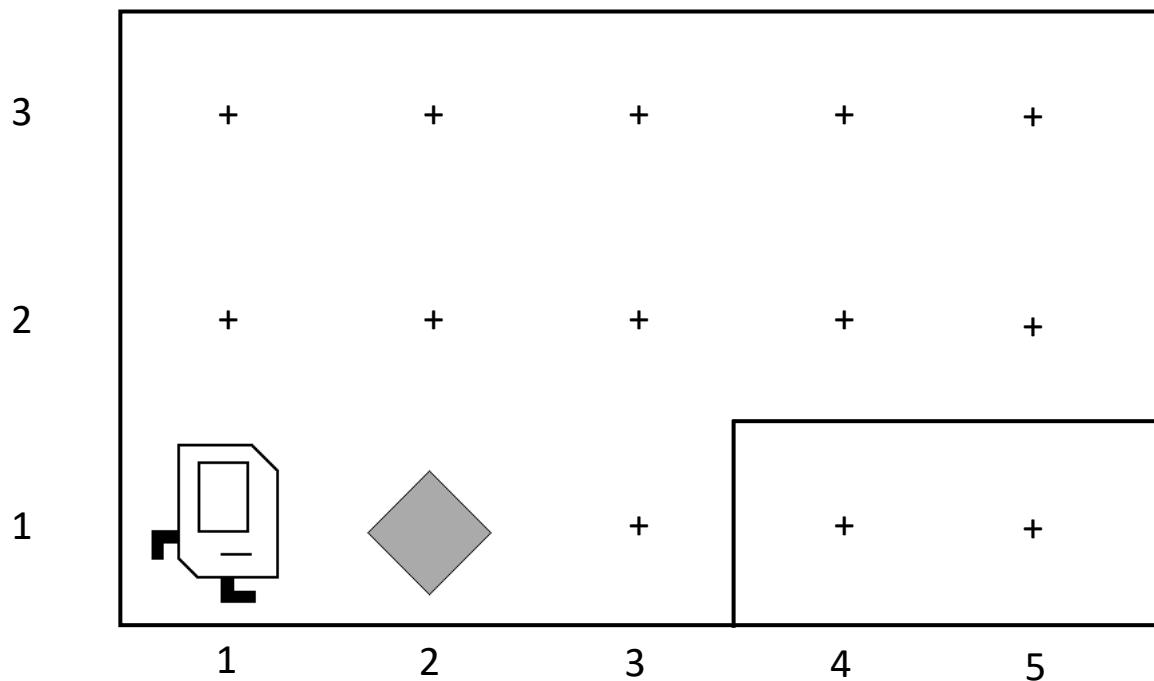
# Karel's World



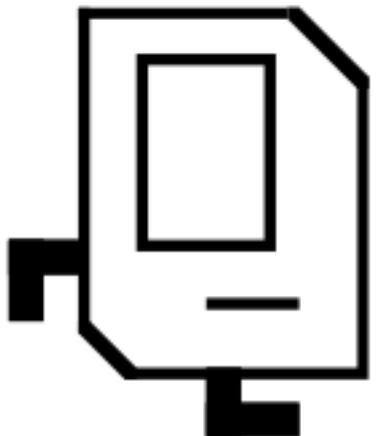
# Walls



# Beepers



# Knows Four Commands



move()

turn\_left()

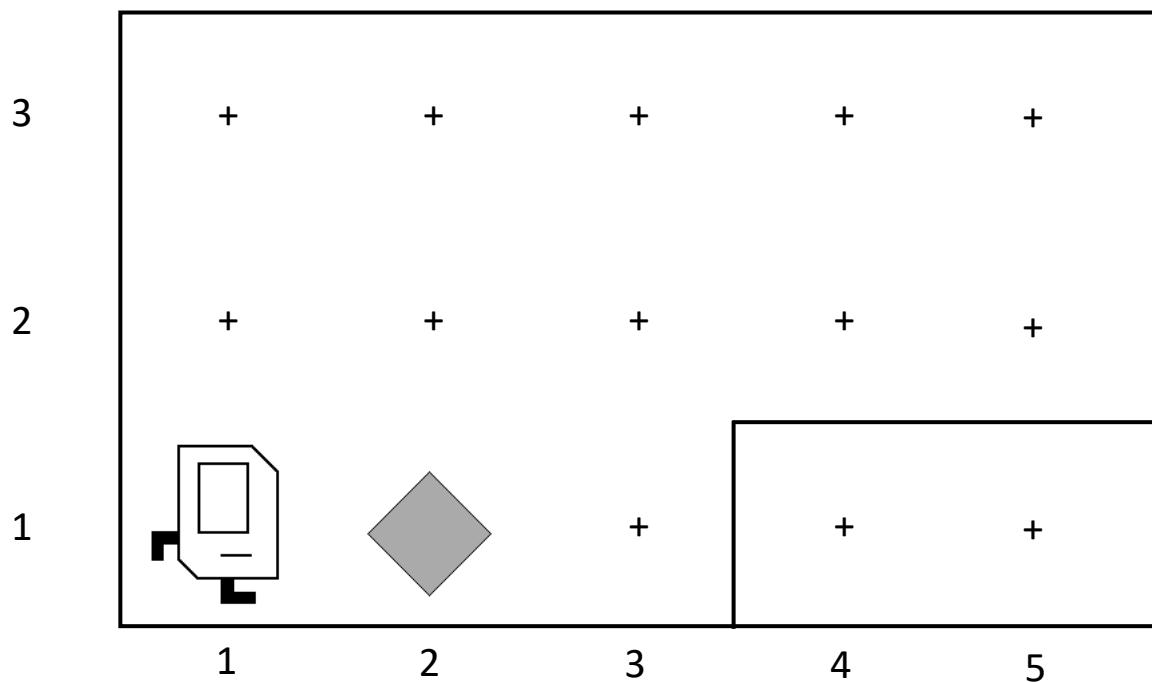
put\_beeper()

pick\_beeper()

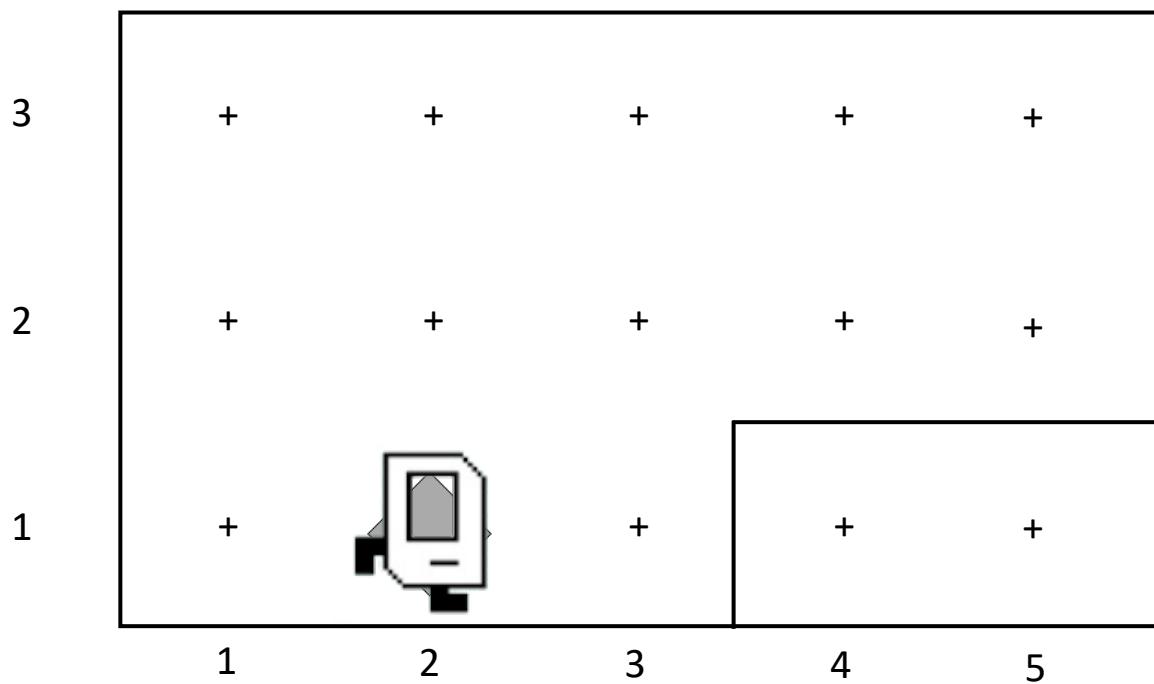


move()

# move()

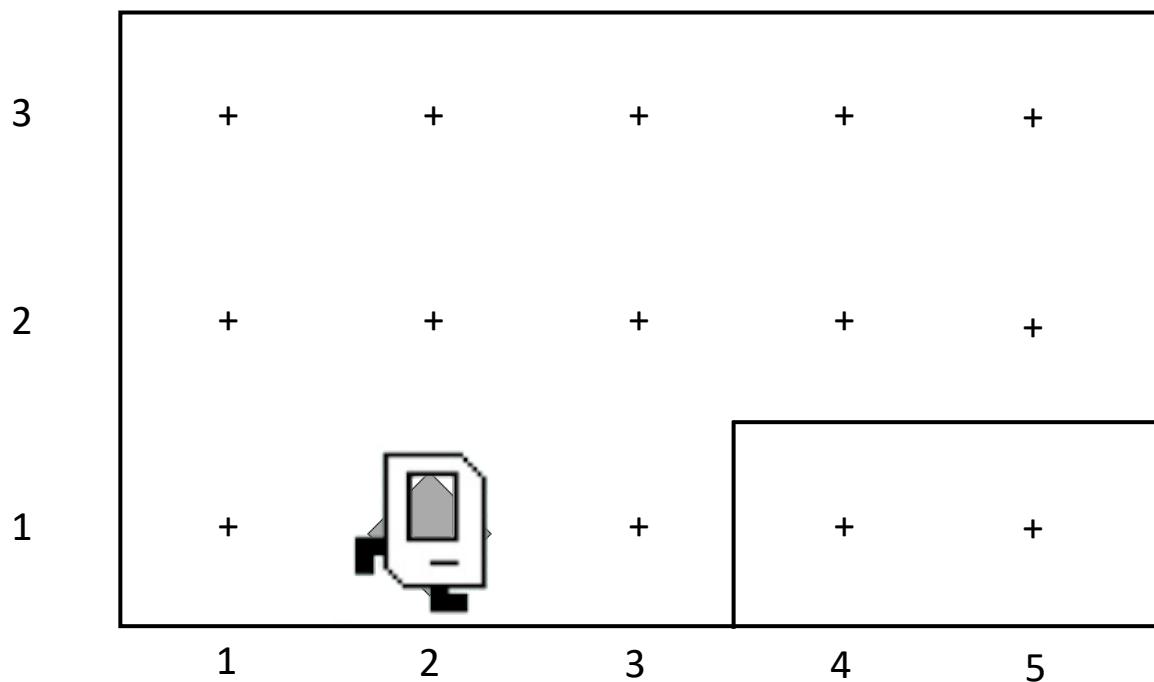


# move()

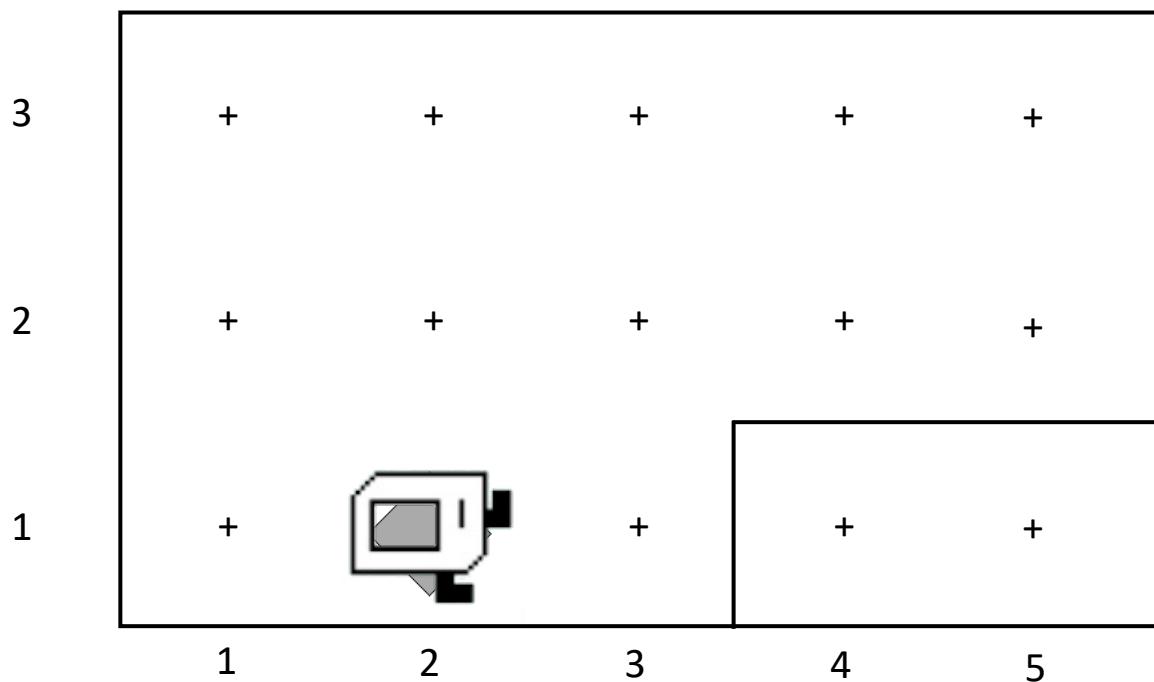


turn\_left()

# turn\_left()

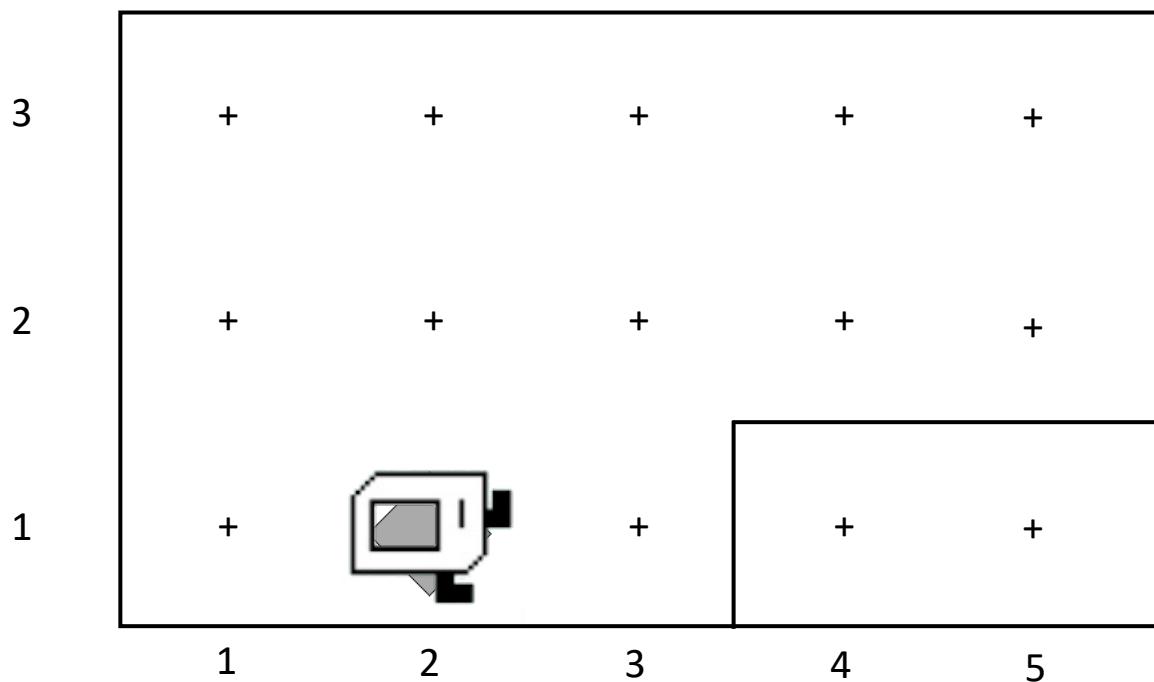


# turn\_left()

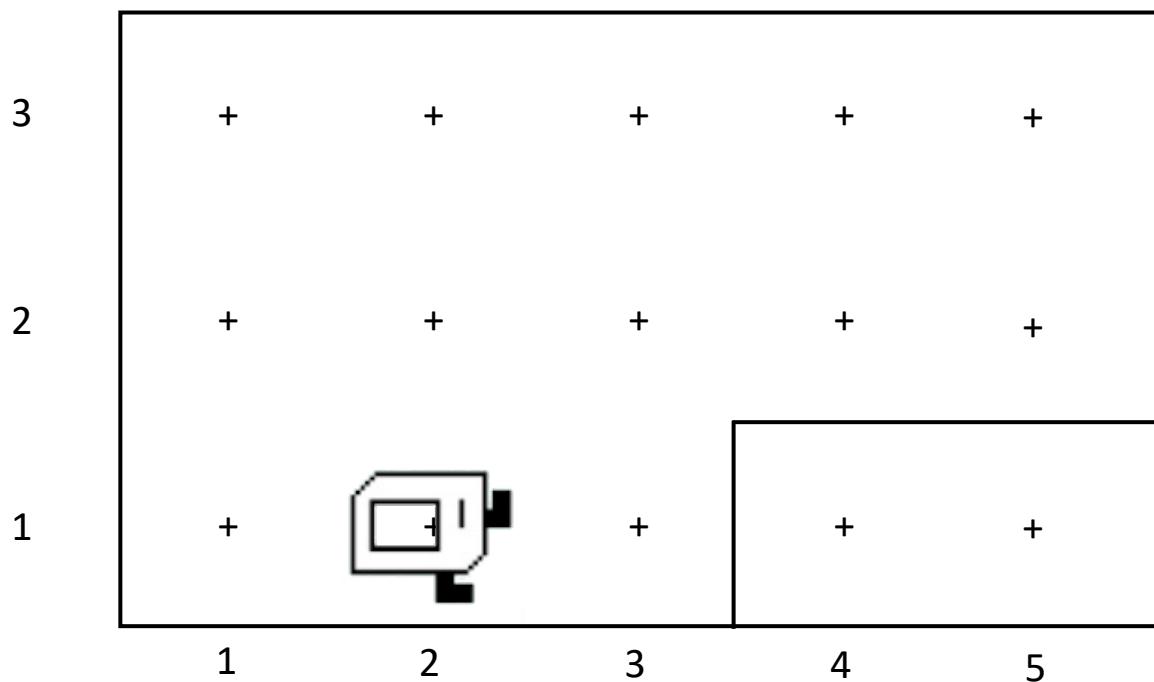


pick\_beeper()

# turn\_left()

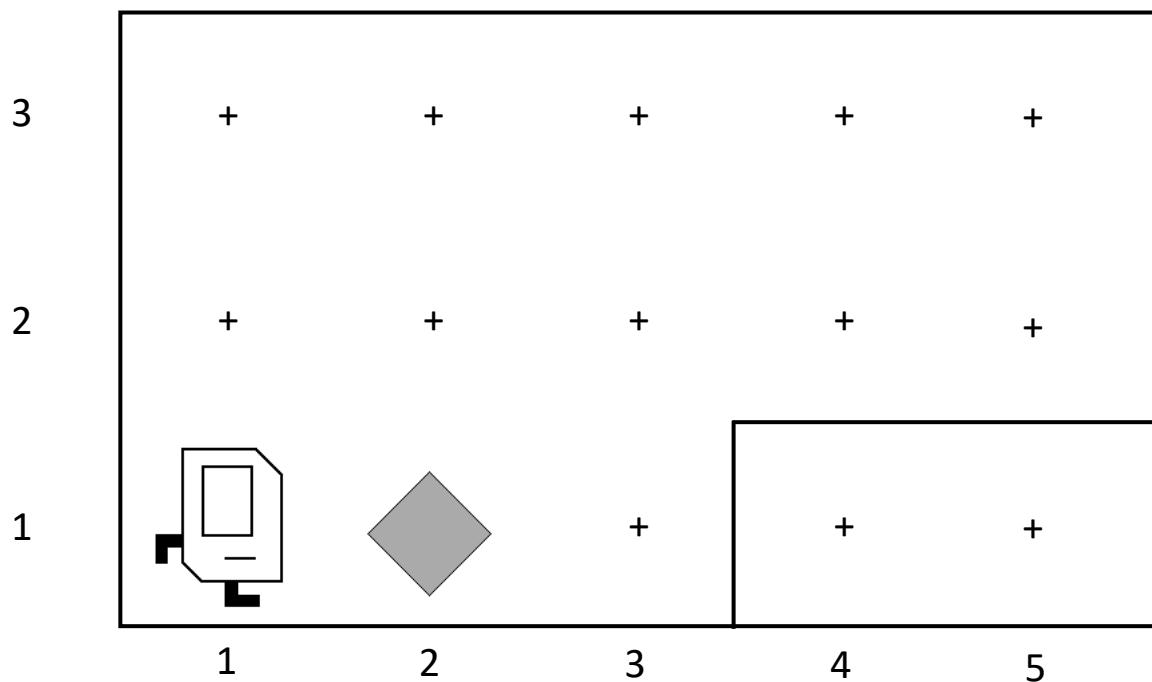


# turn\_left()

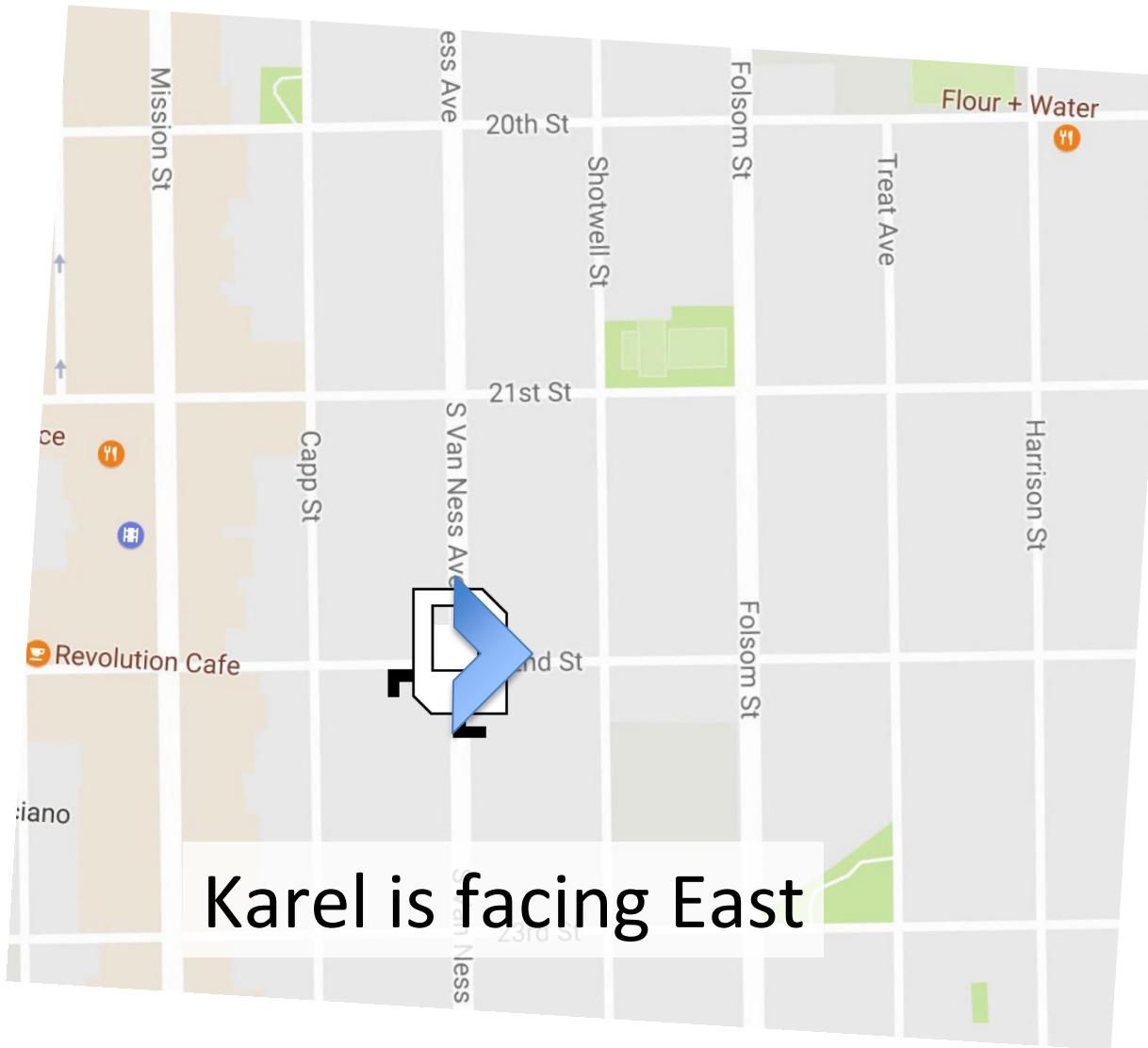


Make Sense?

# Bird's Eye View



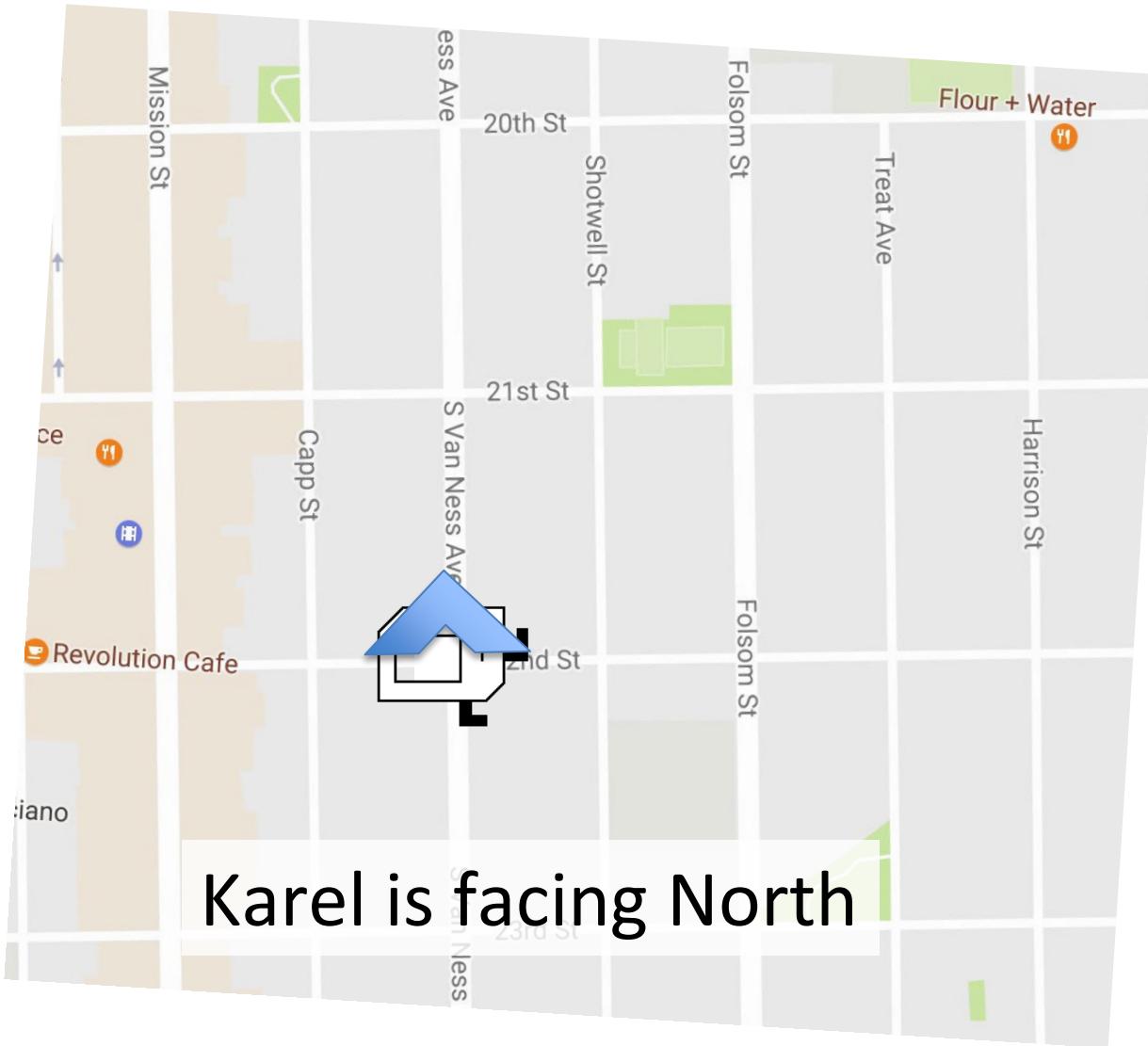
# Bird's Eye View



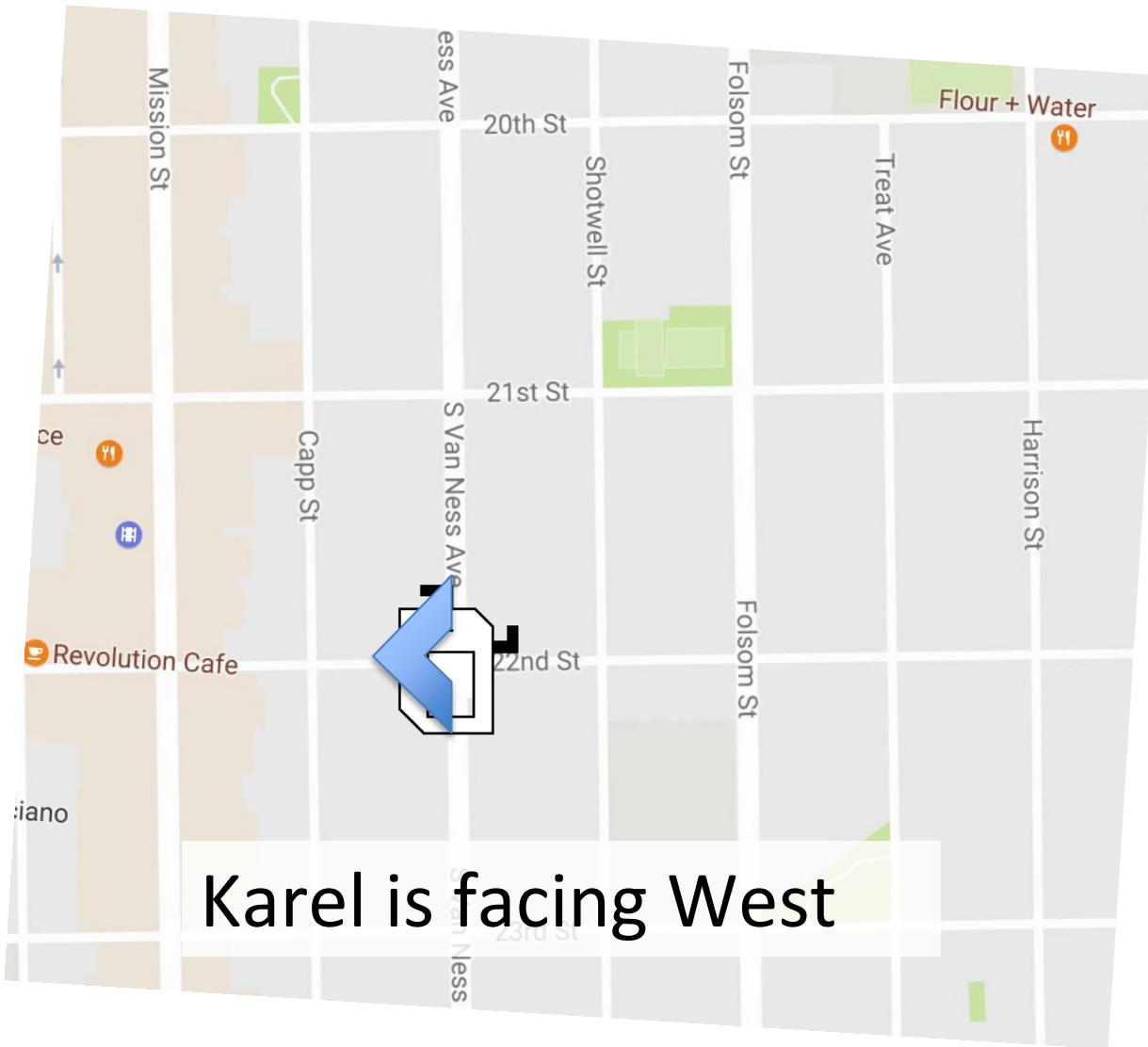
Sahami and Gregg, CS106A, Stanford University



# Turn Left



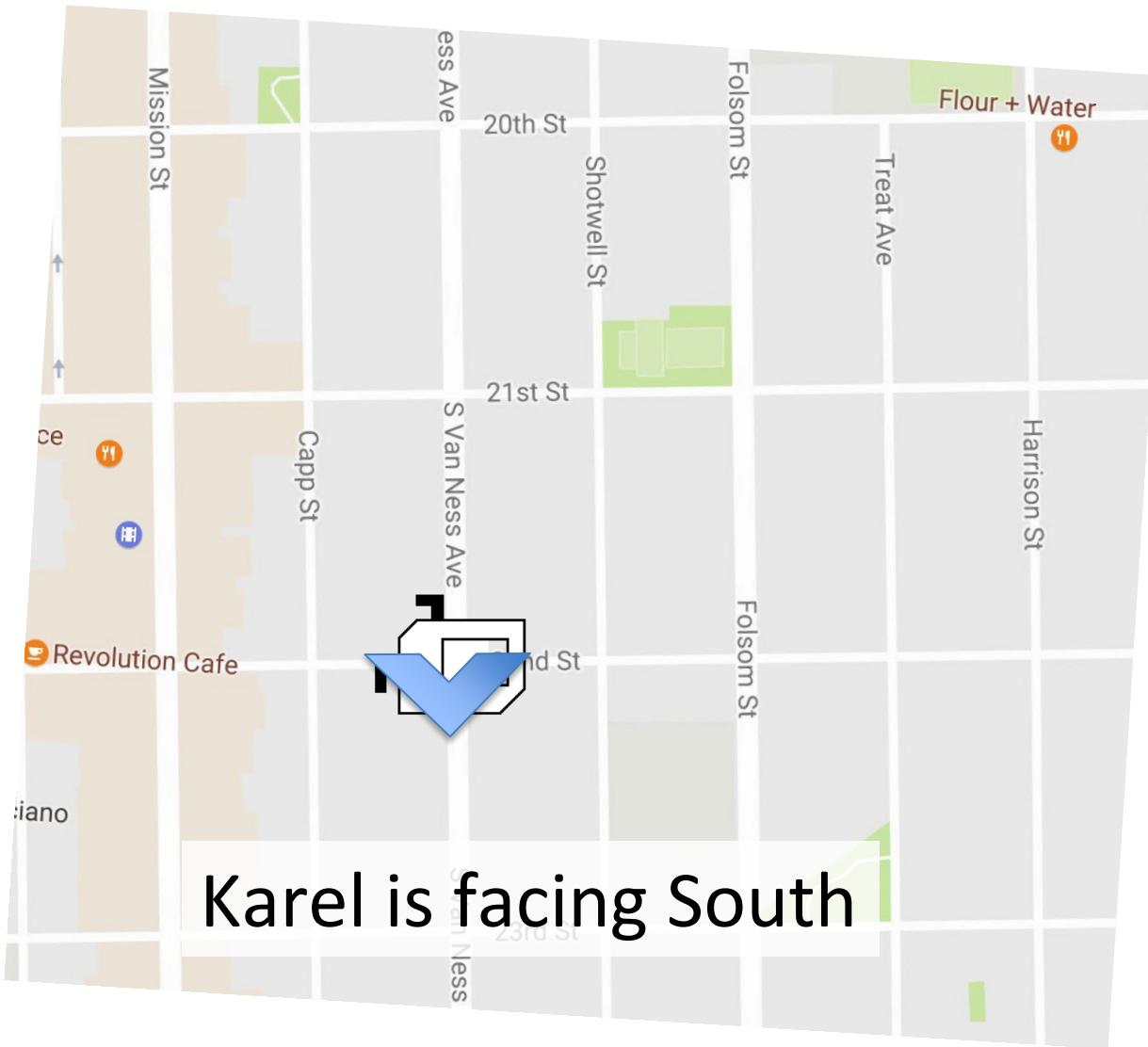
# Turn Left



Karel is facing West



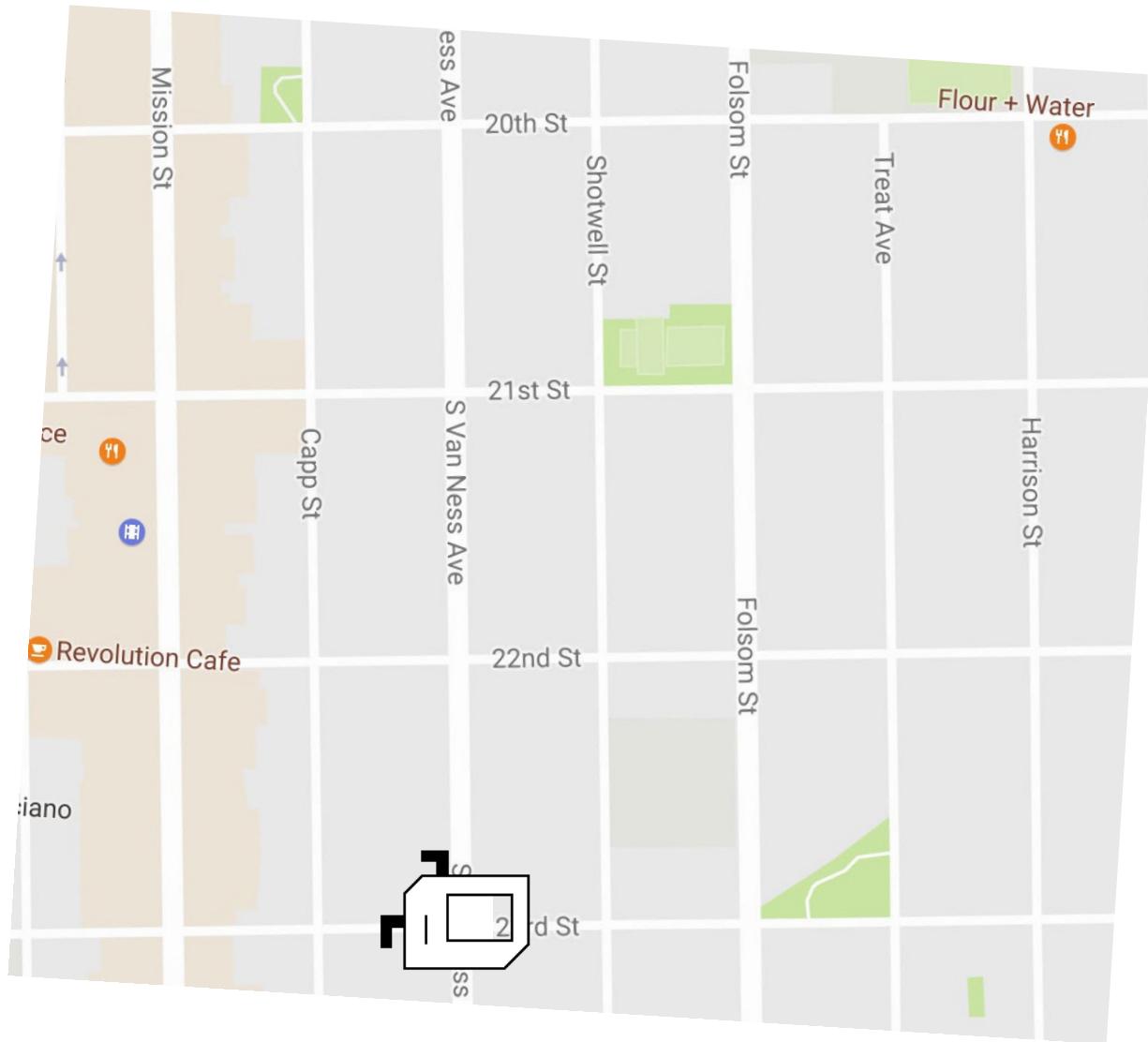
# Turn Left



Karel is facing South



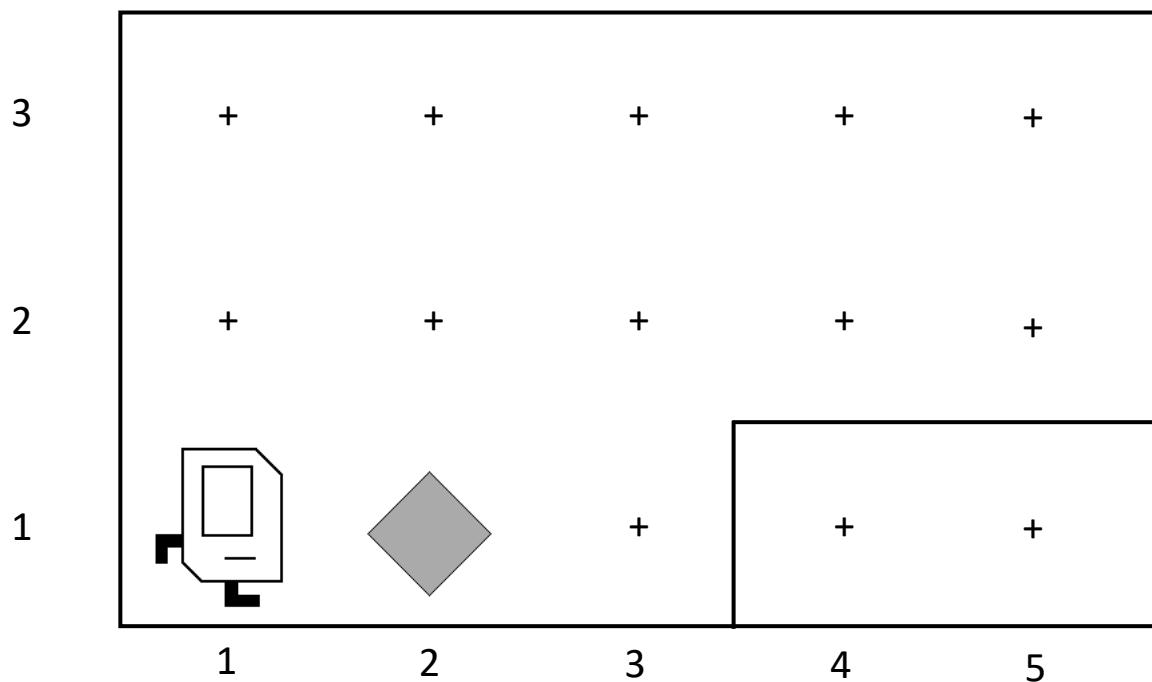
# Move



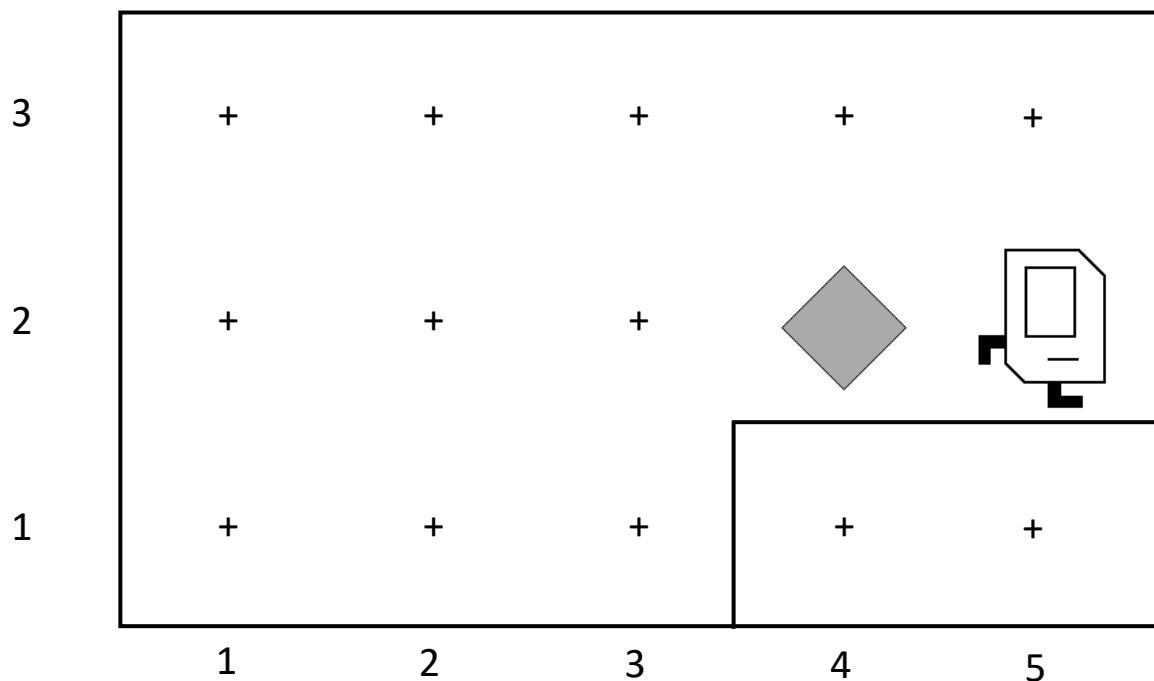
Sahami and Gregg, CS106A, Stanford University



# First Challenge



# First Challenge





Midjourney AI

# Learn By Doing



Where do you program???



The Python IDE  
for Professional  
Developers

[DOWNLOAD](#)

Full-fledged Professional or Free Community



Or:

Use our practice, online IDE

ide.stanford.edu

CS106A | Lecture 1

cs106a.netlify.app/lecture/1

Chris ▾

# Lesson 1: Welcome

APRIL 1, 2024

## Learning Goals

By the end of class you should know about the class, CS106A. You should also know basic Karel commands and be ready to write your first program. Learn about the art of writing beautiful code.

## Readings and Examples

Follow along with this Karel challenge from class:

[StepUp Karel](#)

Lecture covers these chapters from the Karel reader:

[Intro to Karel](#)

[Programming Karel](#)

[Defining New Functions](#)

Extra worked examples / challenges:

<https://ide.stanford.edu/cs106a/a/movebeeper>

<https://ide.stanford.edu/cs106a/a/obstacles>

<https://ide.stanford.edu/cs106a/a/archway>

<https://ide.stanford.edu/cs106a/a/rainbow>

<https://ide.stanford.edu/cs106a/a/warmup>

## Lecture Slides



<https://ide.stanford.edu/cs106a/a/stepup>

Stanford IDE

ide.stanford.edu/cs106a/a/stepup

IDE | Step Up

Practice

Show solution

Follow along the example from class. Have karel pick up the beeper and put it on the ledge, like so:

main.py

```
1 # This tells PyCharm who Karel is
2 # Every Karel file has a line just like it
3 from karel.stanfordkarel import *
4
5 # this program executes in a special function called main
6 def main():
7     move()
8     pick_beeper()
9     move()
10    turn_left()
11    move()
12    turn_right()
13    move()
14    put_beeper()
15    move()
16
17    def turn_right():
18        turn_left()
19        turn_left()
20        turn_left()
21
22
23    # This is "boilerplate" code which launches your code
24    # when you hit the run button
25    if __name__ == '__main__':
26        main()
```

World

World: Step Up

Terminal

```
% python main.py
%
```



# Function Definition

```
def name(parameters):  
    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()
```



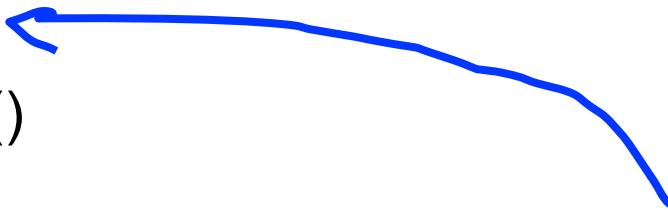
# Anatomy of a Program

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

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

```
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: **main** )



# 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()
```

```
def turn_right():
```

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

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

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



# 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



Sahami and Gregg, CS106A, Stanford University



# Closest Thing To Magic



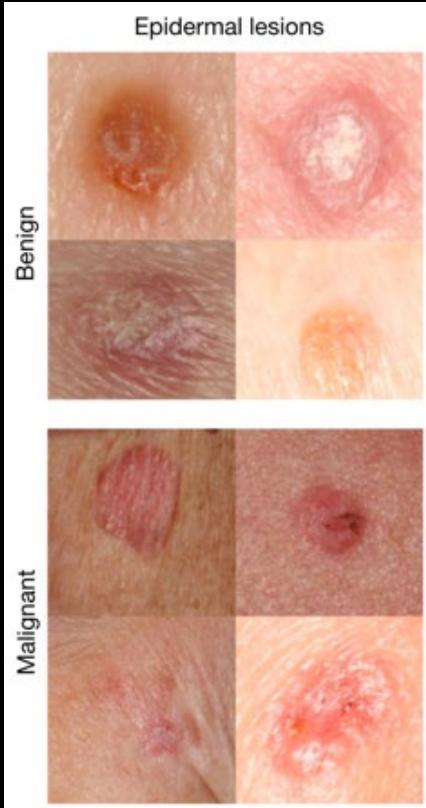
Midjourney AI



# Now is the Time



# Now is the Time



A machine learning algorithm performs **better than** the best dermatologists.

Developed this year, at Stanford.

Esteva, Andre, et al. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature* 542.7639 (2017): 115-118.

# **Everyone is Welcome**



The End



The End?