



# CS106A: Programming Methodology

# Mehran Sahami

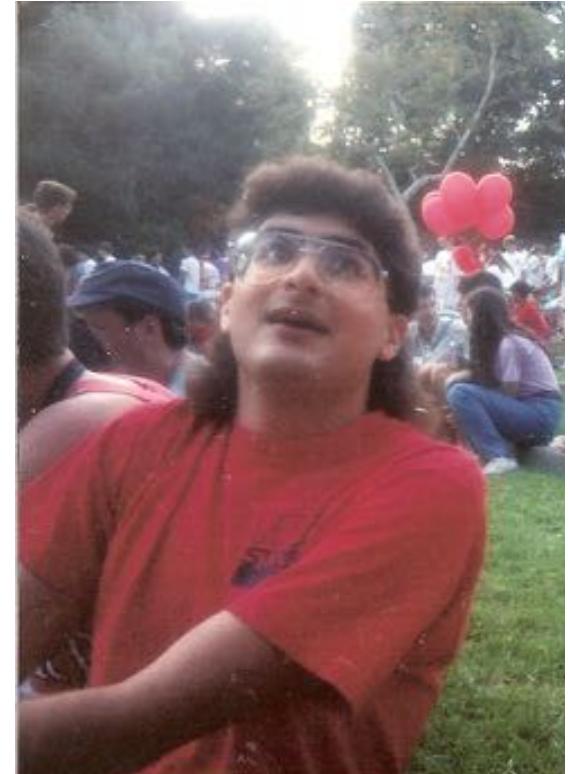


- Childhood: Iran
- High School: San Diego
- Stanford University Ph.D. in Machine Learning  
(Before Machine Learning was cool)
- Spent a decade in tech industry before coming back as professor
  - Love of teaching is why I came back



# 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...



# Chris Piech

My parents are interesting folks

I originally concentrated in graphics and worked at Pixar

- Childhood: Nairobi, Kenya
- High School: Kuala Lumpur, Malaysia
- Stanford University Ph.D. in Neural Networks
- Research lab on AI for Social Good

The problem I really want to solve is to make high quality education more accessible



# Chris Piech



12 years ago to this day, I was sitting in your seats

Piech and Sahami, CS106A, Stanford University



# Head TA: Brahm Capoor



Piech and Sahami, CS106A, Stanford University



# Section Leaders



Luciano  
Gonzalez



Maggie  
Davis



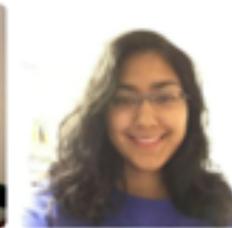
Marilyn  
Zhang



Meng  
Zhang



Nidhi  
Manoj



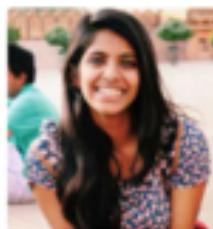
Niki  
Agrawal



Peter  
Maldonado



Rachel  
Gardner



Rhea  
Karuturi



Robbie  
Jones



Ruiqi  
Chen



Semir  
Shafi



Shanon  
Reckinger



Tessera  
Chin



Thariq  
Ridha



Vineet  
Kosaraju

\* Actually some of last year's section leaders

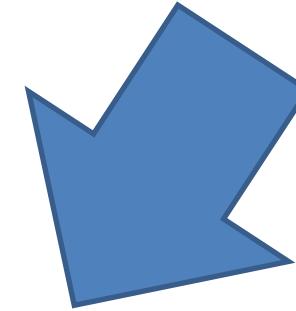
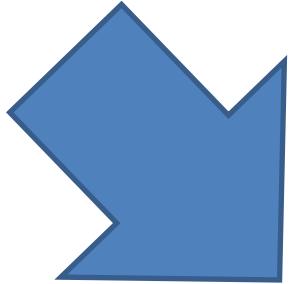
Piech and Sahami, CS106A, Stanford University



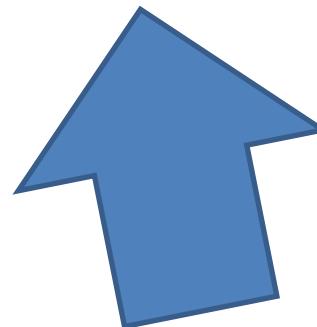
# Course mechanics

(This is a brief version.  
Please read the handout for full details).

# Course Website



<http://cs106a.stanford.edu>



# Prerequisite Test



# Getting To Know You

- Assignment #0 on website (“Who are you?”)

The screenshot shows a dark-themed website for CS106A. At the top, there is a navigation bar with tabs for "CS106A", "Lectures", "Assignments", "Handouts", and "Schedule". The "Assignments" tab is highlighted with a red box. Below the navigation bar, the Stanford University seal is displayed. The main title "CS106A: Programming Methodologies" is centered above course details: "Stanford University | Spring 2020" and "Monday, Wednesday, Friday | Live Lectures 1:30pm - 2:20pm PST".

- Some initial responses (~125 responses so far)
  - Class is 60% women, 39% men, 1% prefer not to say
  - 4 students are former/current members of the armed forces
  - 15 students have founded socially-focused ventures
  - 1 student attacked by kangaroo when they were younger
  - “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 1:30-2:20pm
  - 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.



Piech and Sahami, CS106A, Stanford University



# Office Hours



LaIR: evenings Sunday through Thursday  
(starting Sunday)

Piech and Sahami, 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

You are only competing against yourself.



# 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
- This quarter, especially, we can put much more focus on *learning* rather than grading



# What we will ask you to do

- 7 programming assignment 75%
  - Get more complicated as quarter progresses
- In-class diagnostic assessment (exam) 15%
- Section participation 10%
- Get 4 free “late days” (on assignments)
  - Each “late day” is a 24-hour period
  - Allows for turning in assignment late without penalty
  - After free late days are used, assignments penalized one grade bucket per day late
  - For extensions beyond free late days, contact Brahm (head TA)



# Optional Contest



Piech and Sahami, CS106A, Stanford University



# Online Text Books

A screenshot of a web browser window showing the "Karel the Robot" Python tutorial. The URL in the address bar is `compedu.stanford.edu/karel-reader/docs/python/en/intro.html`. The page title is "Karel the Robot". On the left, there is a sidebar with a navigation menu:

- 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

The main content area features the text "Karel the Robot" above an illustration of a robot head and a Python logo. Below this is the text "Learns Python". At the bottom, author information is provided: "Chris Piech and Eric Roberts", "Department of Computer Science", "Stanford University", and "January 2019". A blue button with the text "▶ Get Started" is located at the bottom right.



# 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

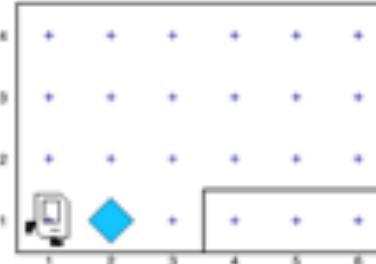
## Chapter 2: Programming Karel

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



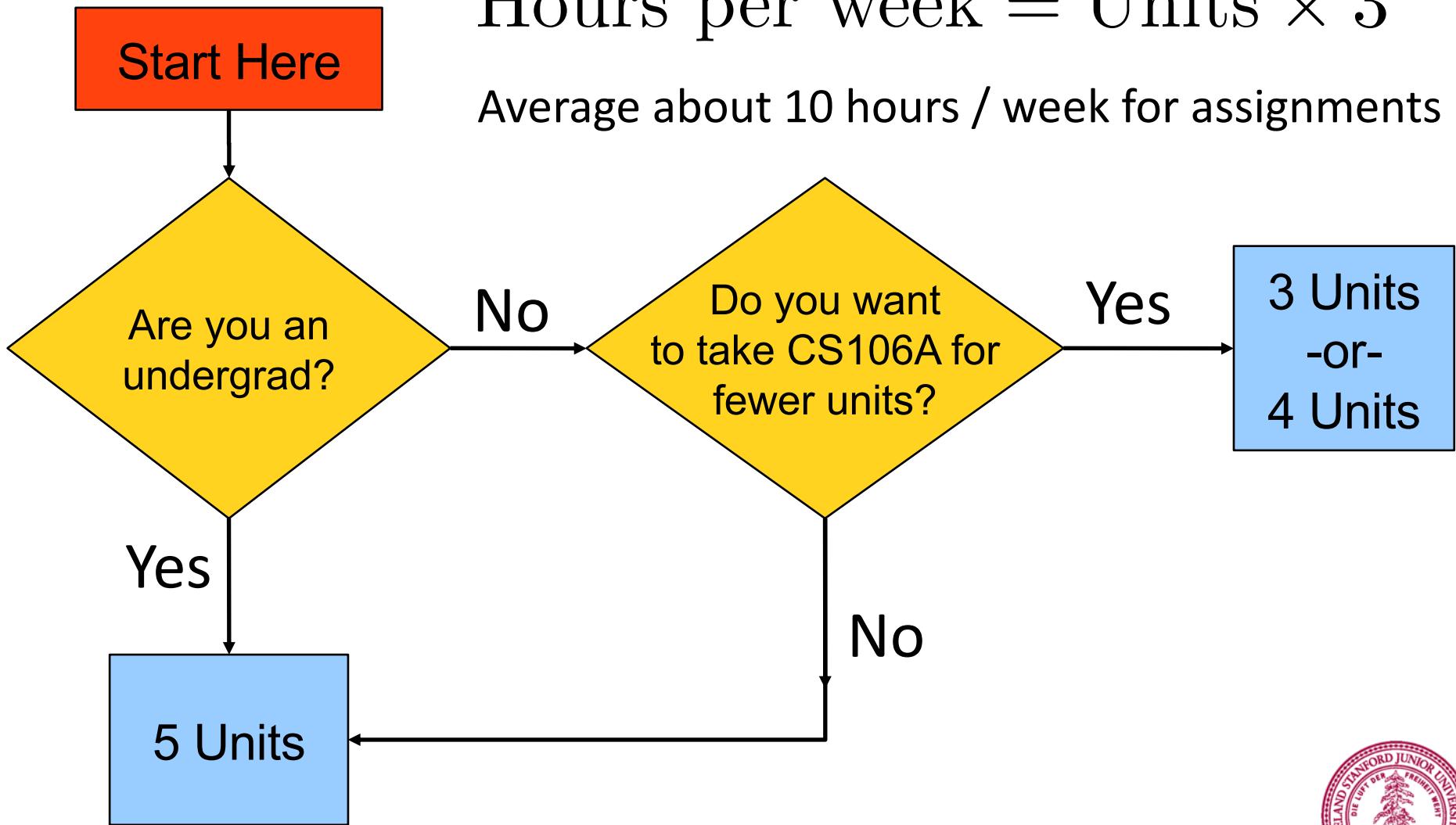
Press the "Run" button to execute the program. Programs are typically written in a special application called an **Integrated Development Environment** (IDE) and most Karel programs are written in an IDE called PyCharm. Like an IDE, this reader has the ability to execute programs in order to help you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



Are you in the right place?

# What is CS106A?

# 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 just won the Turing Award – the Nobel Prize of Computer Science

Piech and Sahami, CS106A, Stanford University



# Consumer Applications



# Autonomous Surgery



(c) 2012 Intuitive Surgical, Inc.



# Self-Driving Car



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

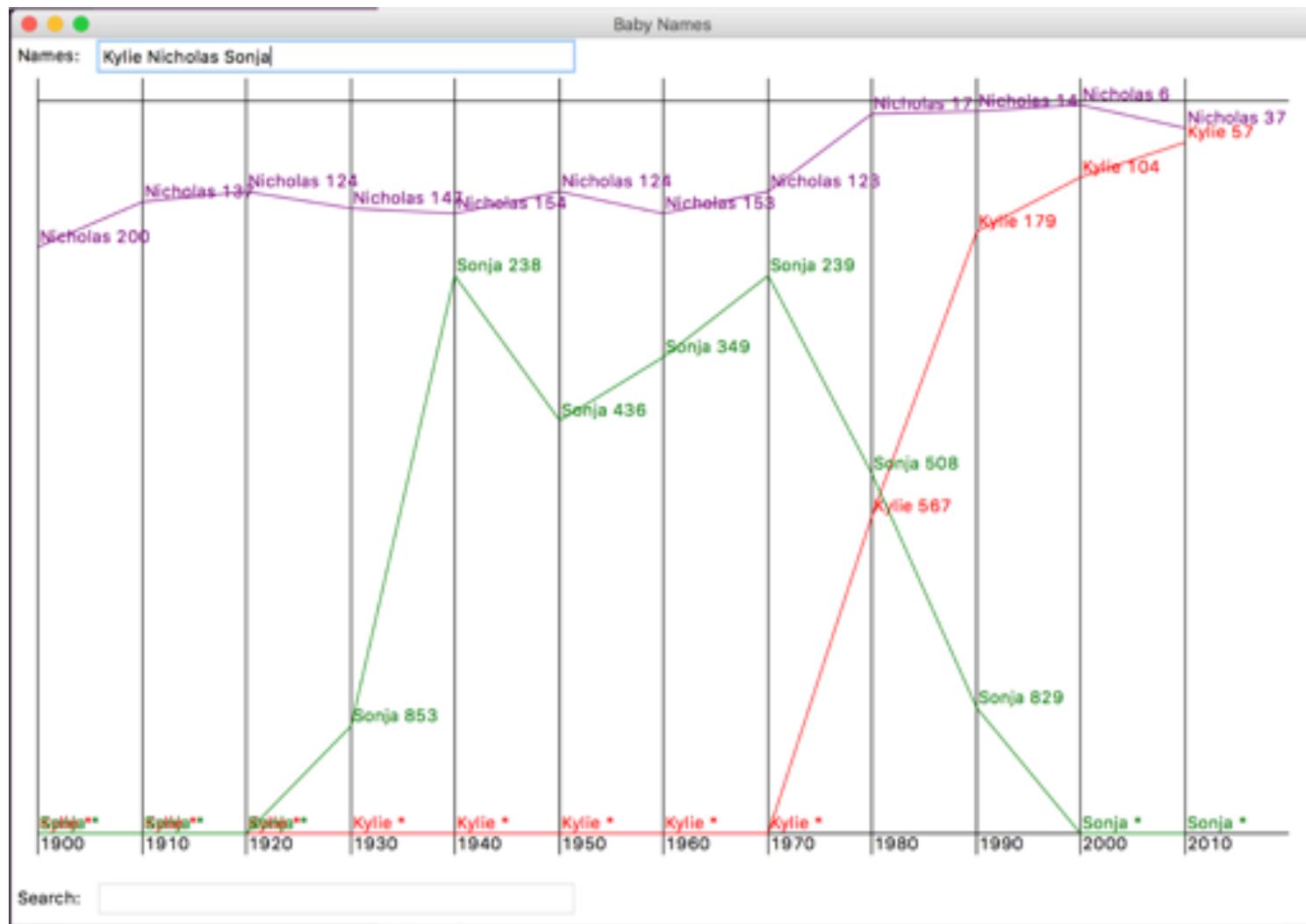
# Image Transformation



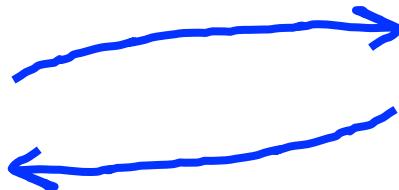
Piech and Sahami, CS106A, Stanford University



# Data Visualization



# Internet Applications



```
FacePamphletServer
Starting server on port 8000...
addProfile (name=Mehran)
=> success
addProfile (name=Chris)
=> success
addProfile (name=Chris)
=> Error: Database already contains Chris.
getStatus (name=Chris)
=> none
setStatus (name=Chris, status=teaching)
=> success
getStatus (name=Chris)
=> teaching
addFriend (name2=Mehran, name1=Chris)
=> success
getFriends (name=Chris)
=> [Mehran]
addProfile (name=Julie)
=> success
getImg (name=Julie)
=> none
getStatus (name=Julie)
=> none
getFriends (name=Julie)
=> []
setImg (img=JulieZ.jpg, name=Julie)
=> success
getImg (name=Julie)
=> JulieZ.jpg
getStatus (name=Julie)
=> none
getFriends (name=Julie)
=> []
addFriend (name2=Chris, name1=Julie)
=> success
getImg (name=Julie)
=> JulieZ.jpg
getStatus (name=Julie)
=> none
```



# Art of Computer Science



# Strive for Everyone to Succeed



# Lets Get Started



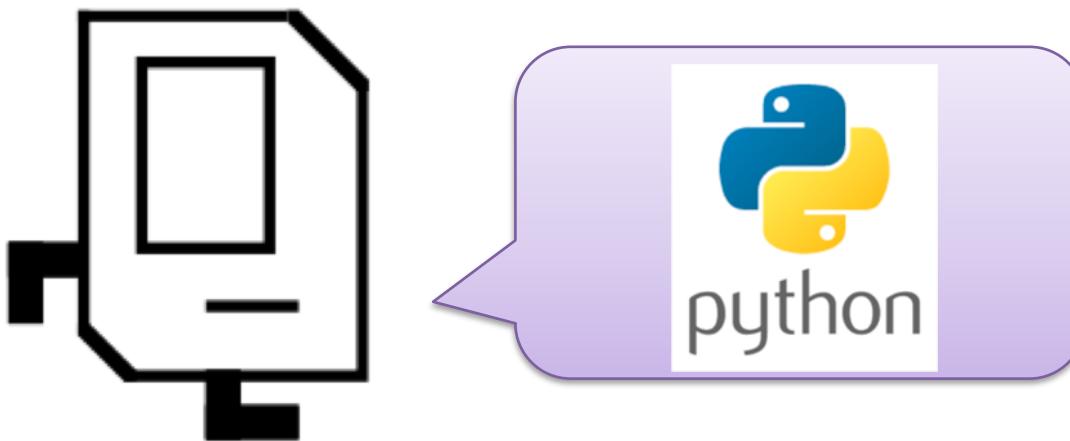
# Meet Karel the Robot



Good morning

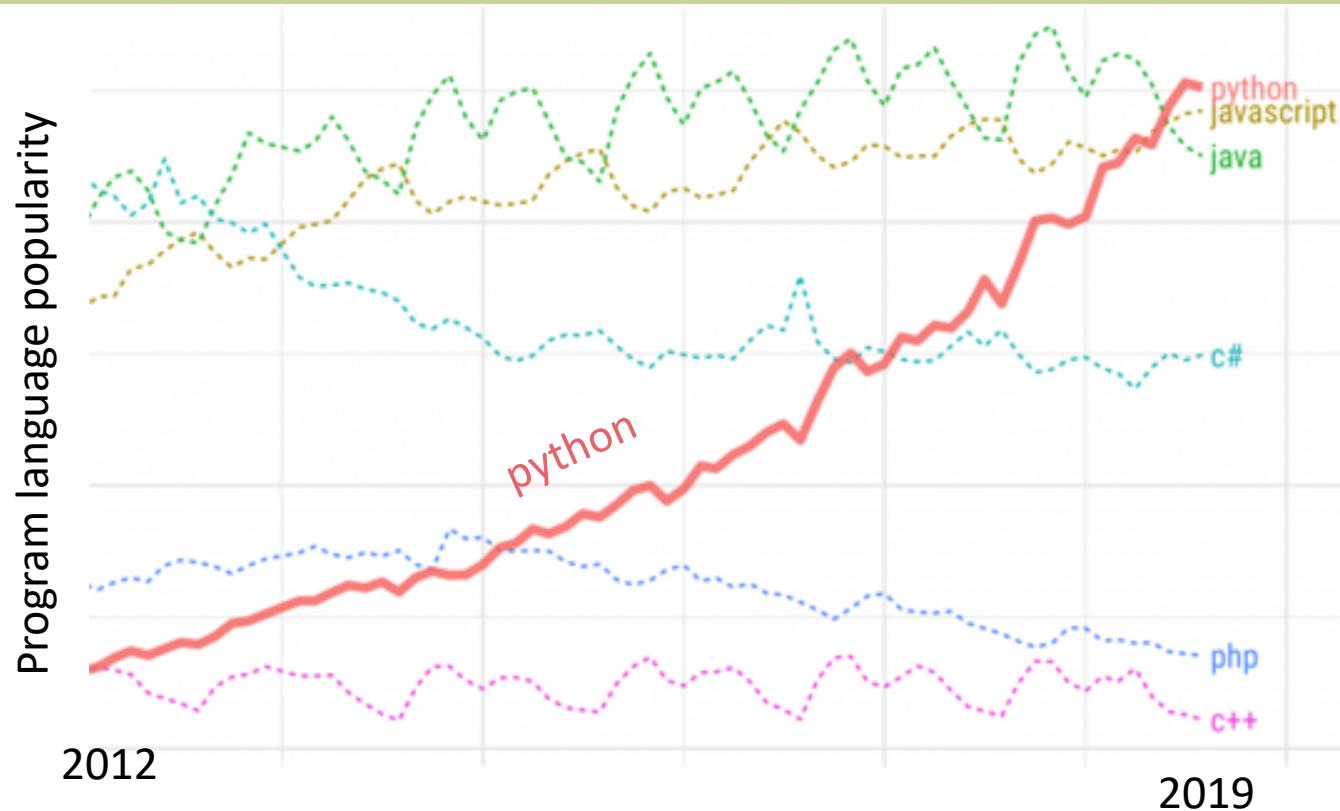


# Karel Speaks Python



# Why Python?

1



2



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

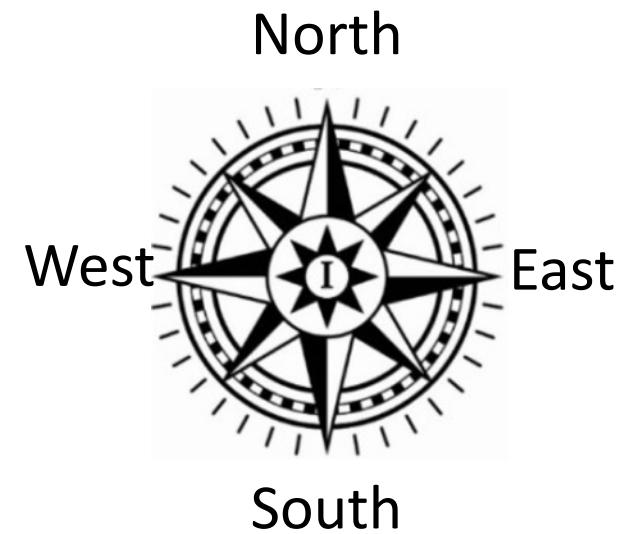


# Guido van Rossum

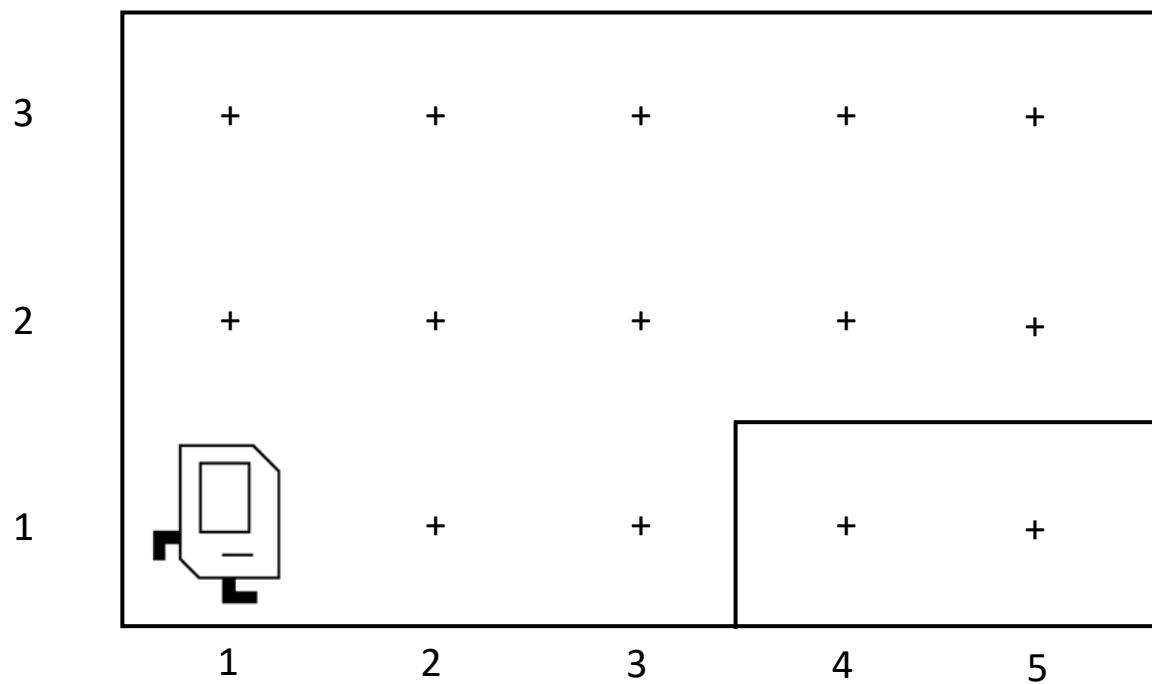


# Karel's World

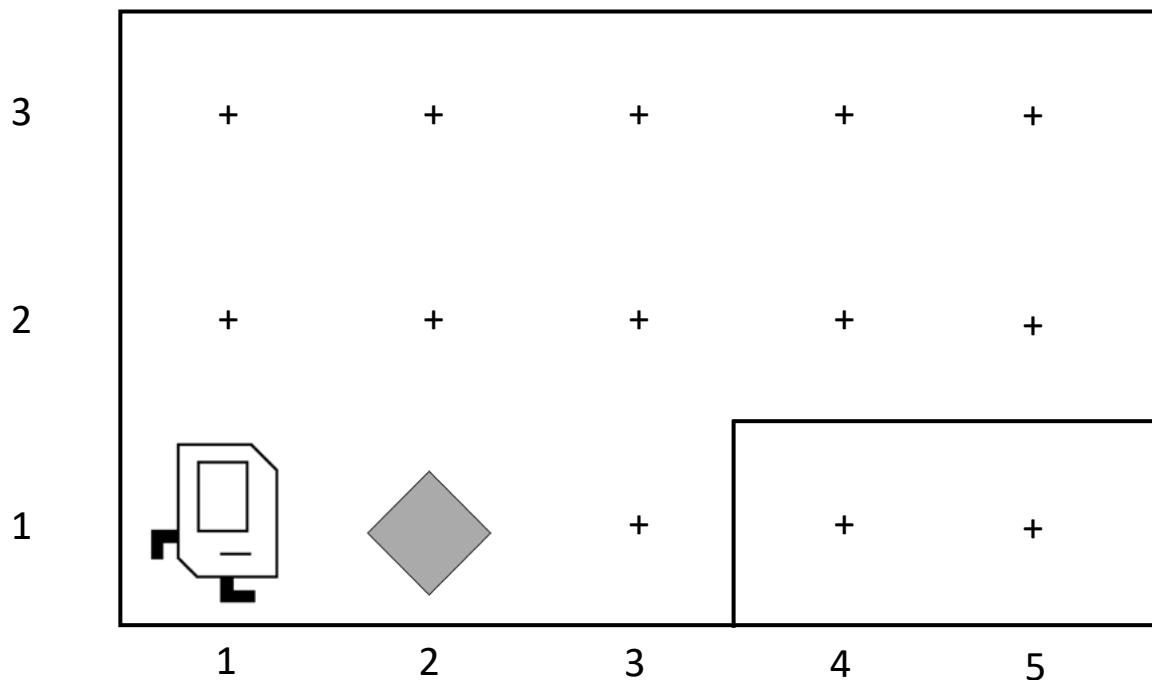
3	+	+	+	+	
2	+	+	+	+	
1		+	+	+	
	1	2	3	4	5



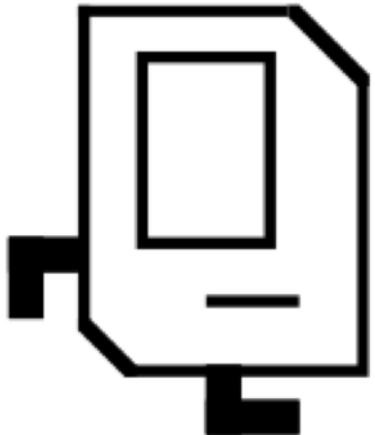
# Walls



# Beepers



# Knows Four Commands



`move( )`

`turn_left( )`

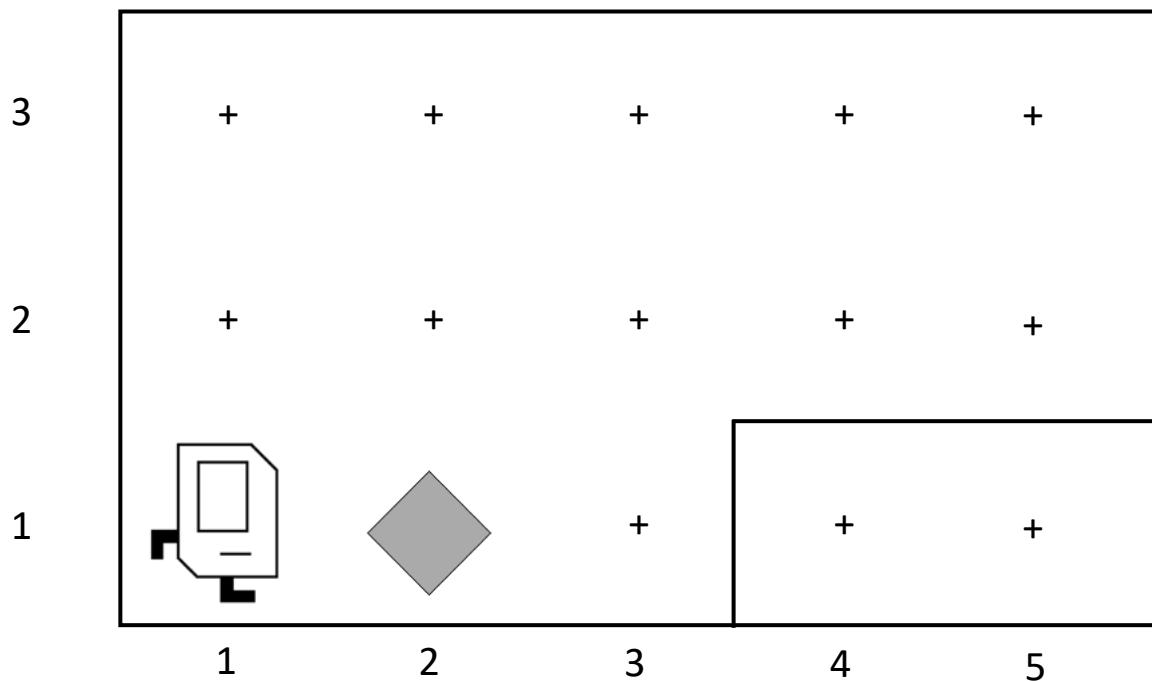
`put_beeper( )`

`pick_beeper( )`

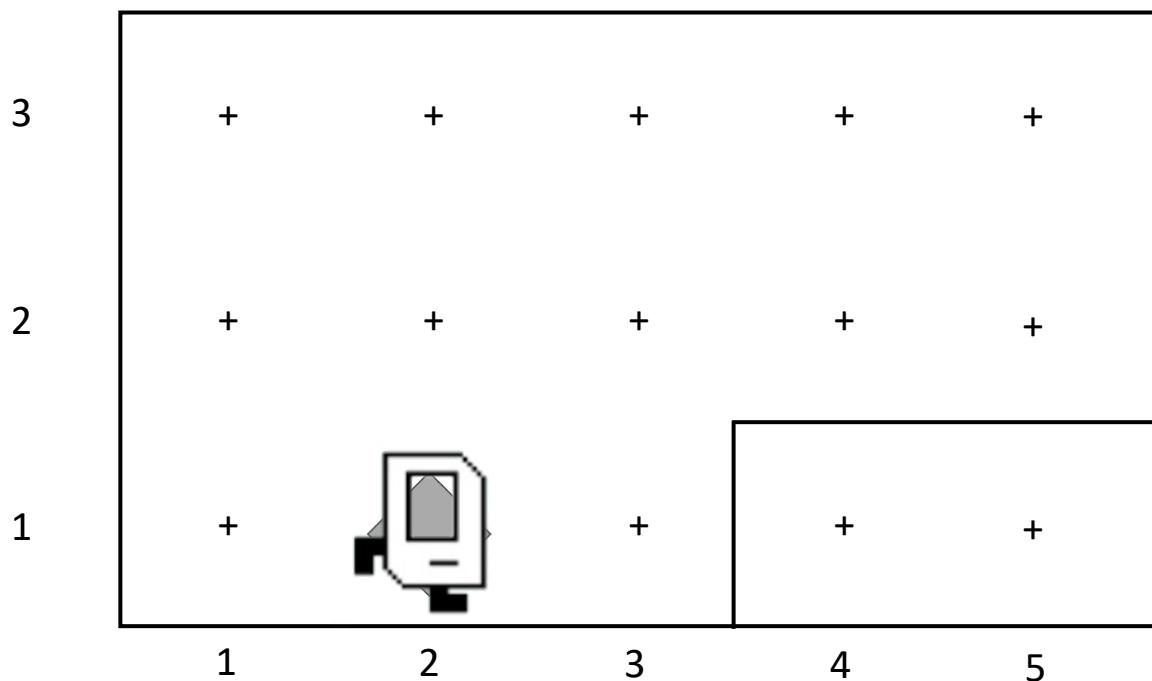


move( )

# move()

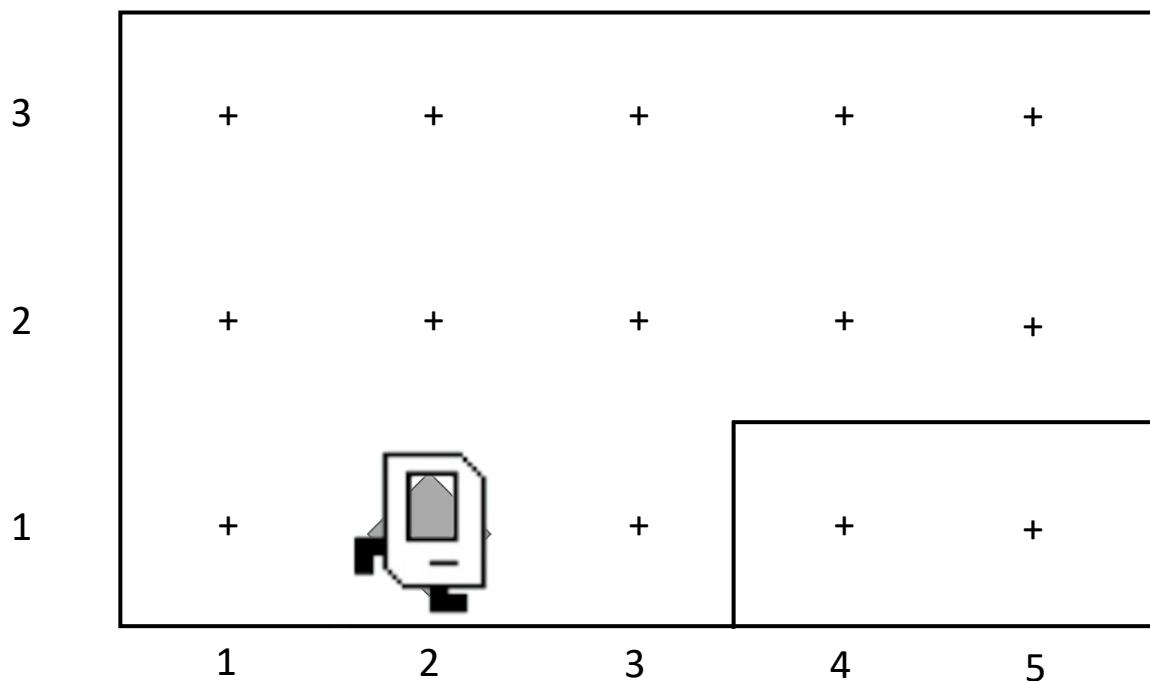


# move()

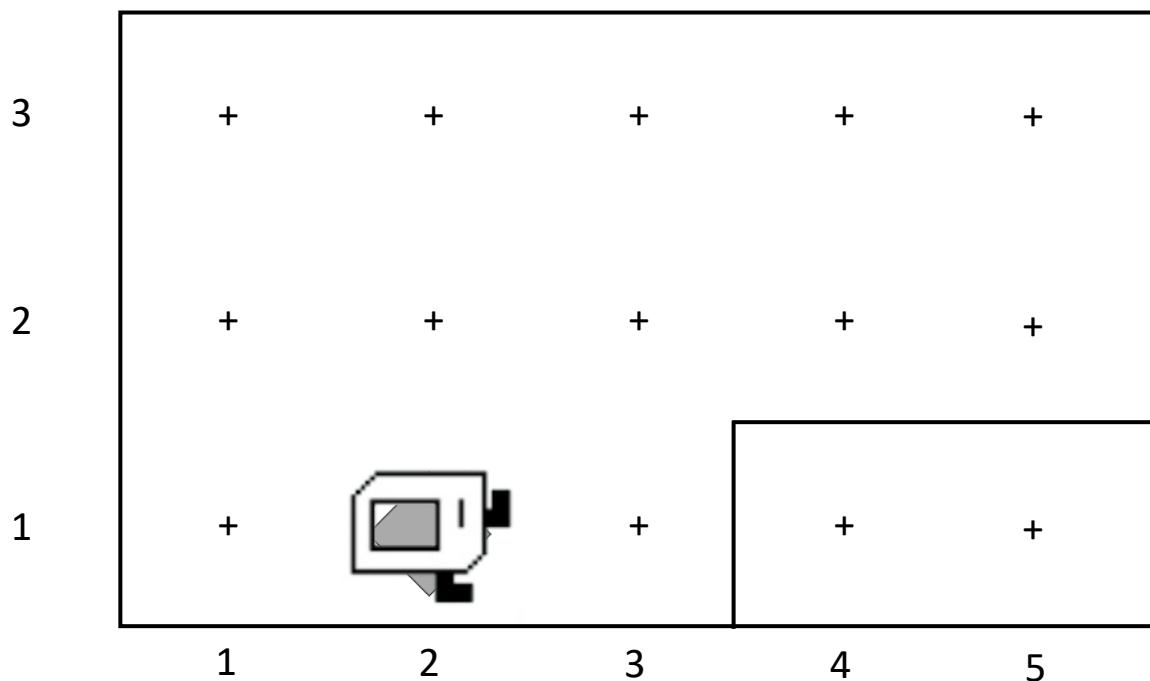


turn\_left()

# `turn_left()`

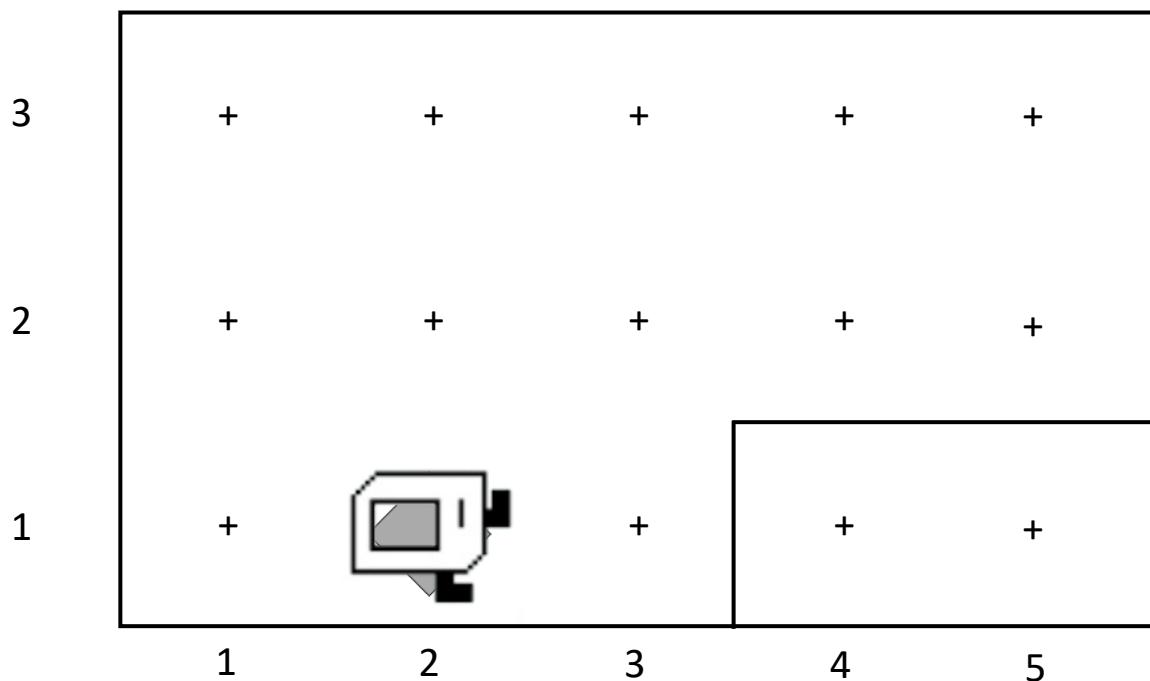


# `turn_left()`

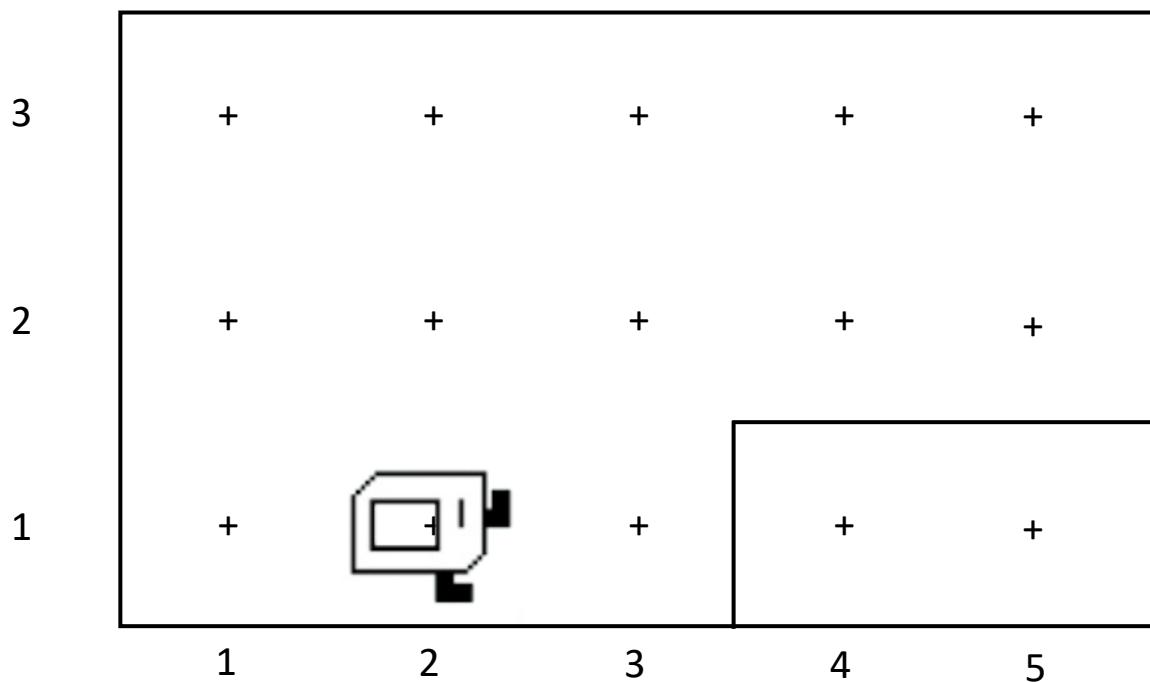


pick\_beeper( )

# `turn_left()`

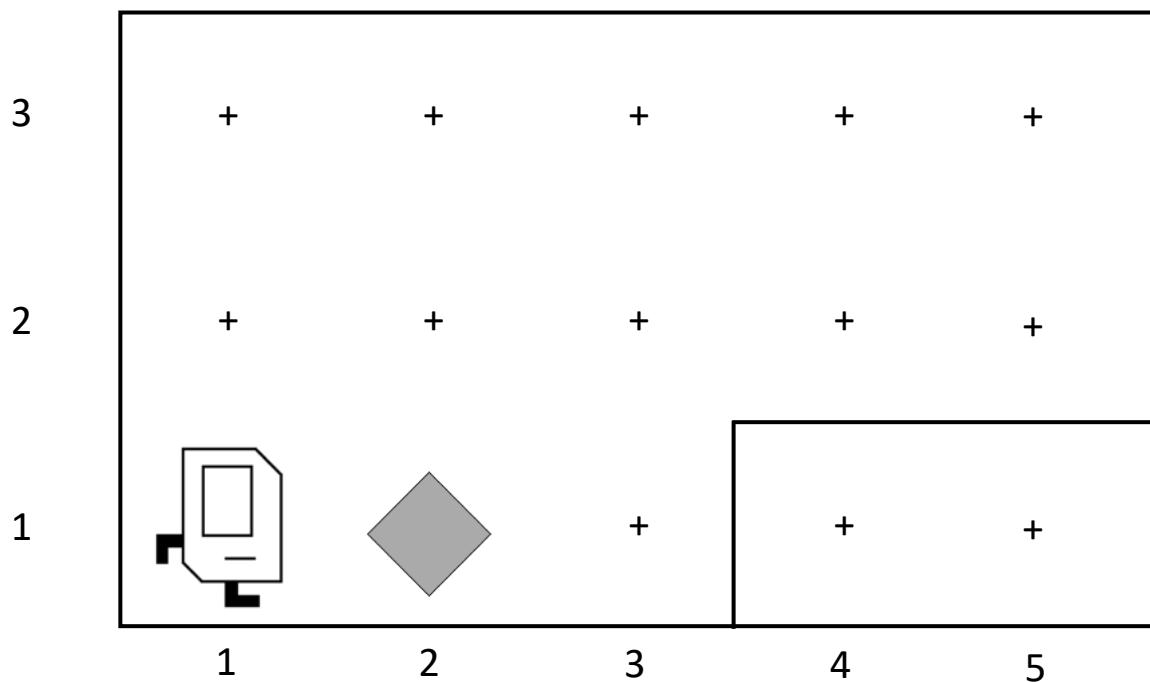


# **turn\_left()**

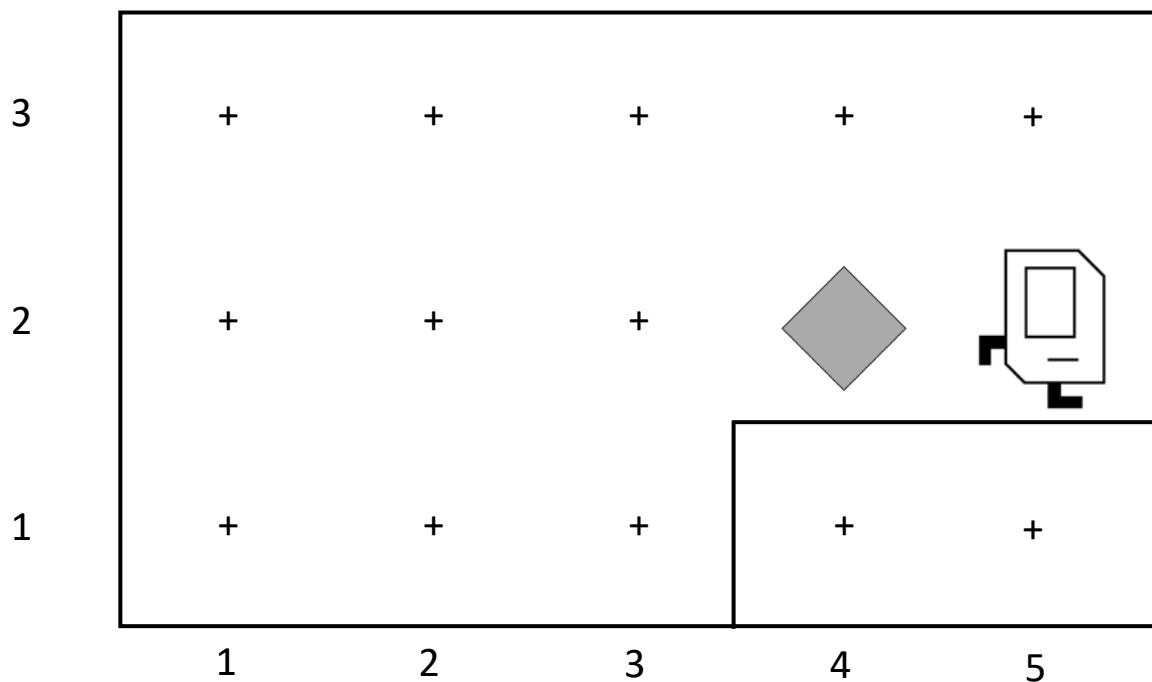


Make Sense?

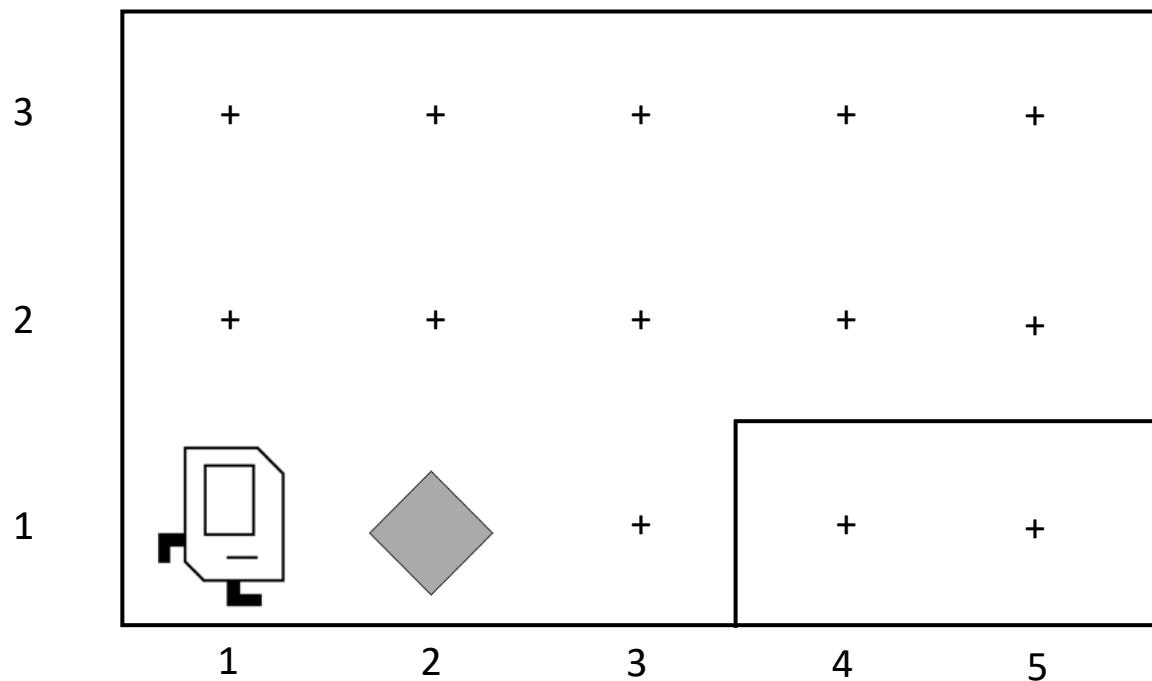
# First Challenge



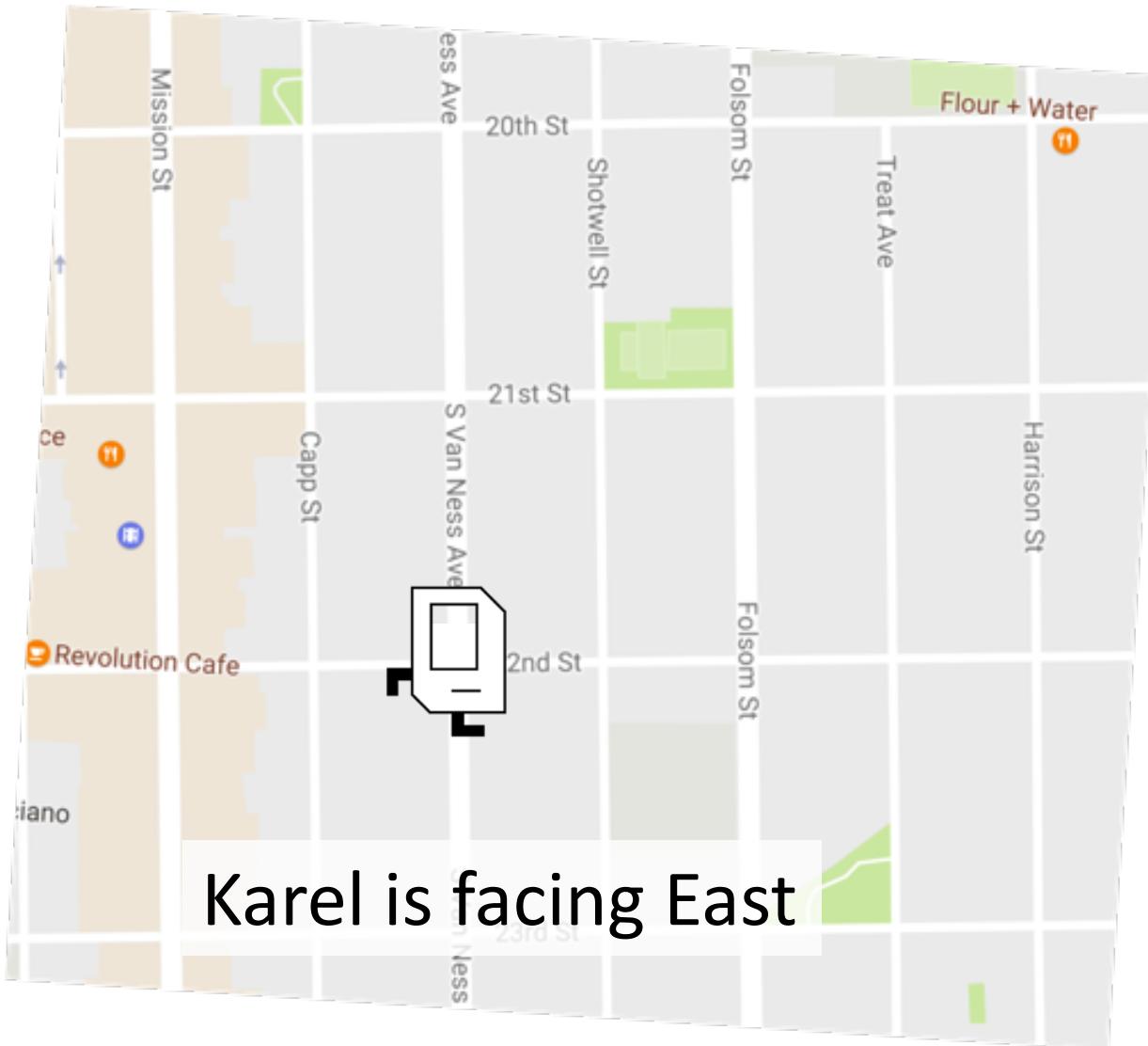
# First Challenge



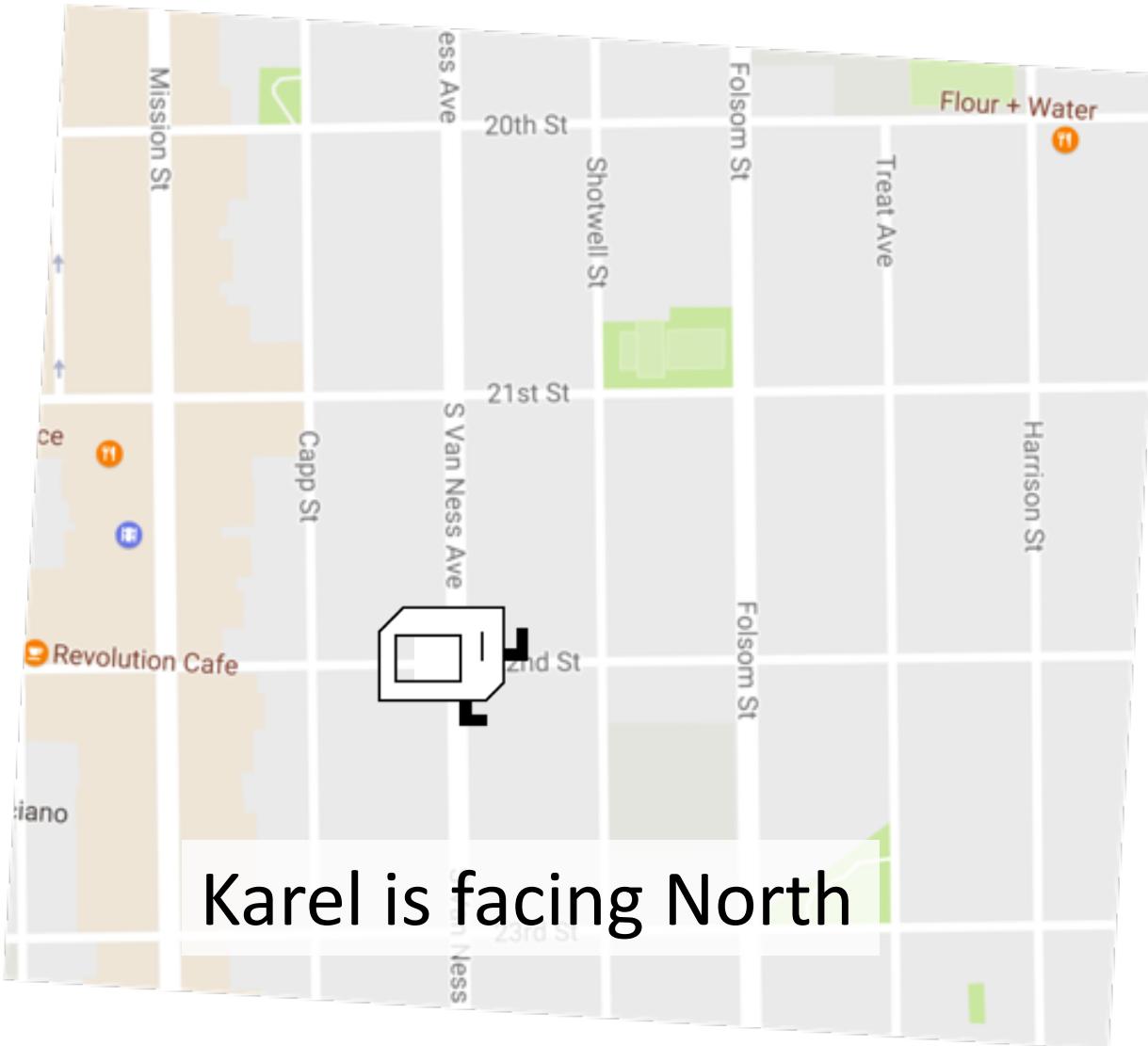
# Bird's Eye View



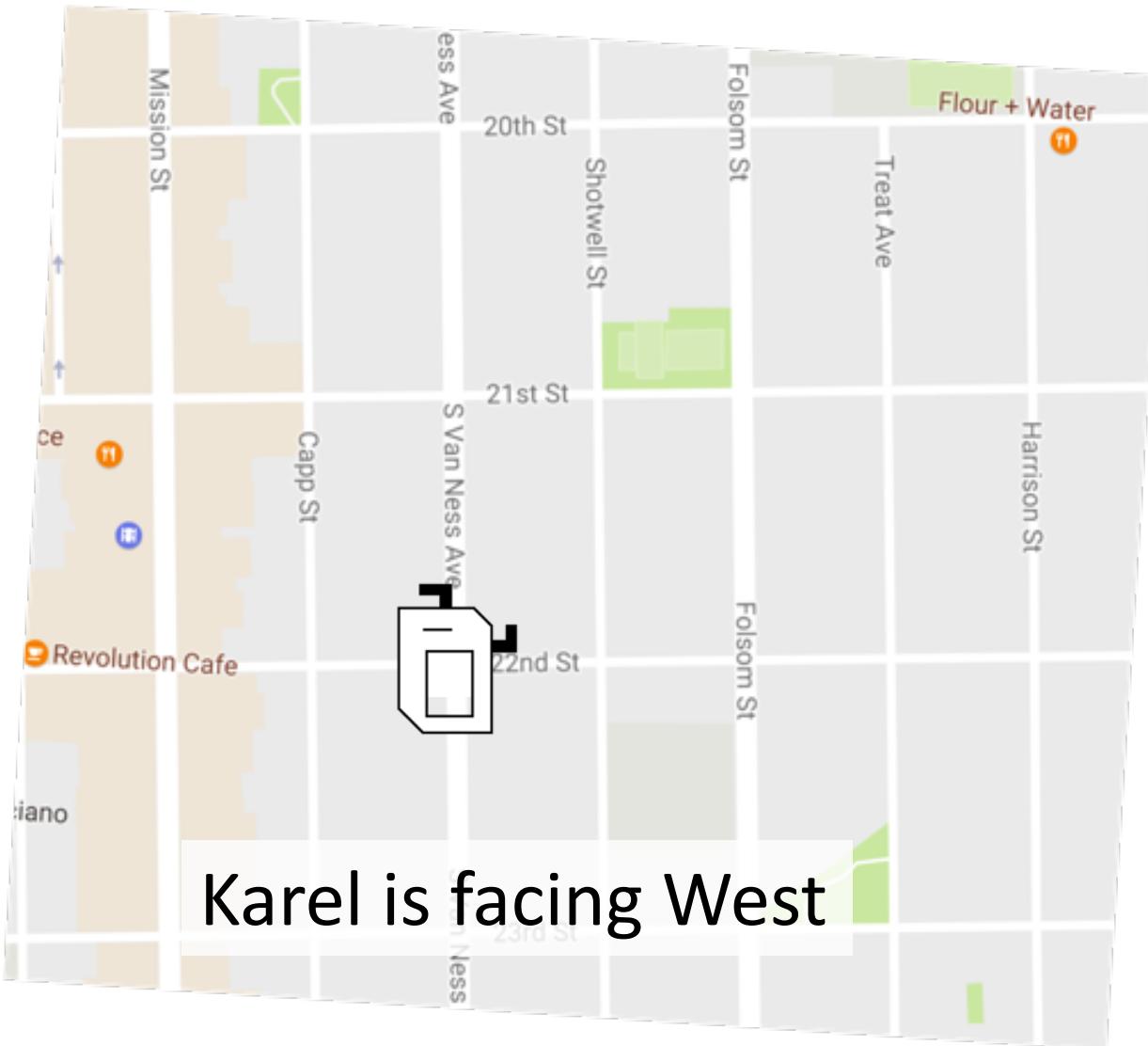
# Bird's Eye View



# Turn Left



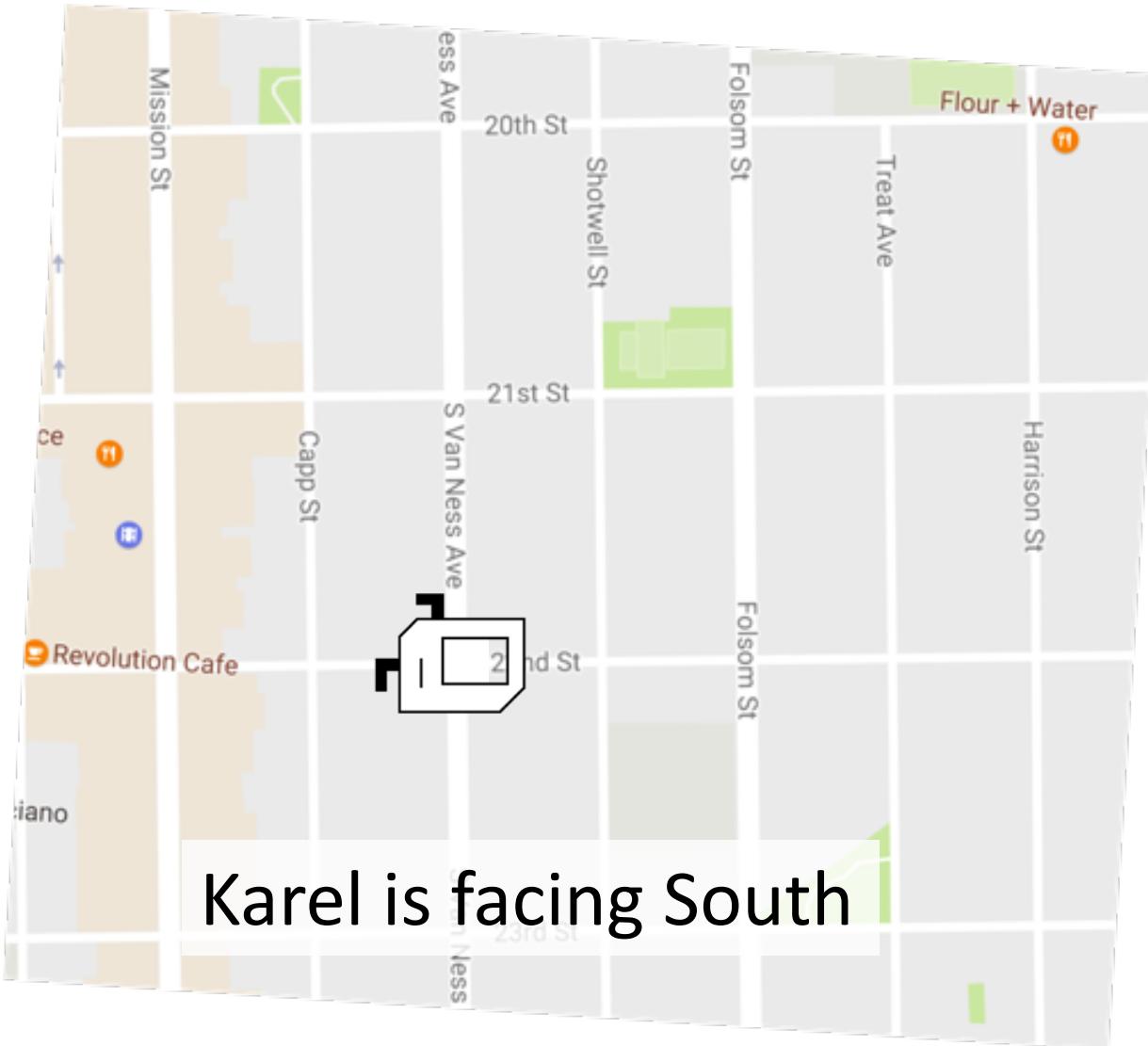
# Turn Left



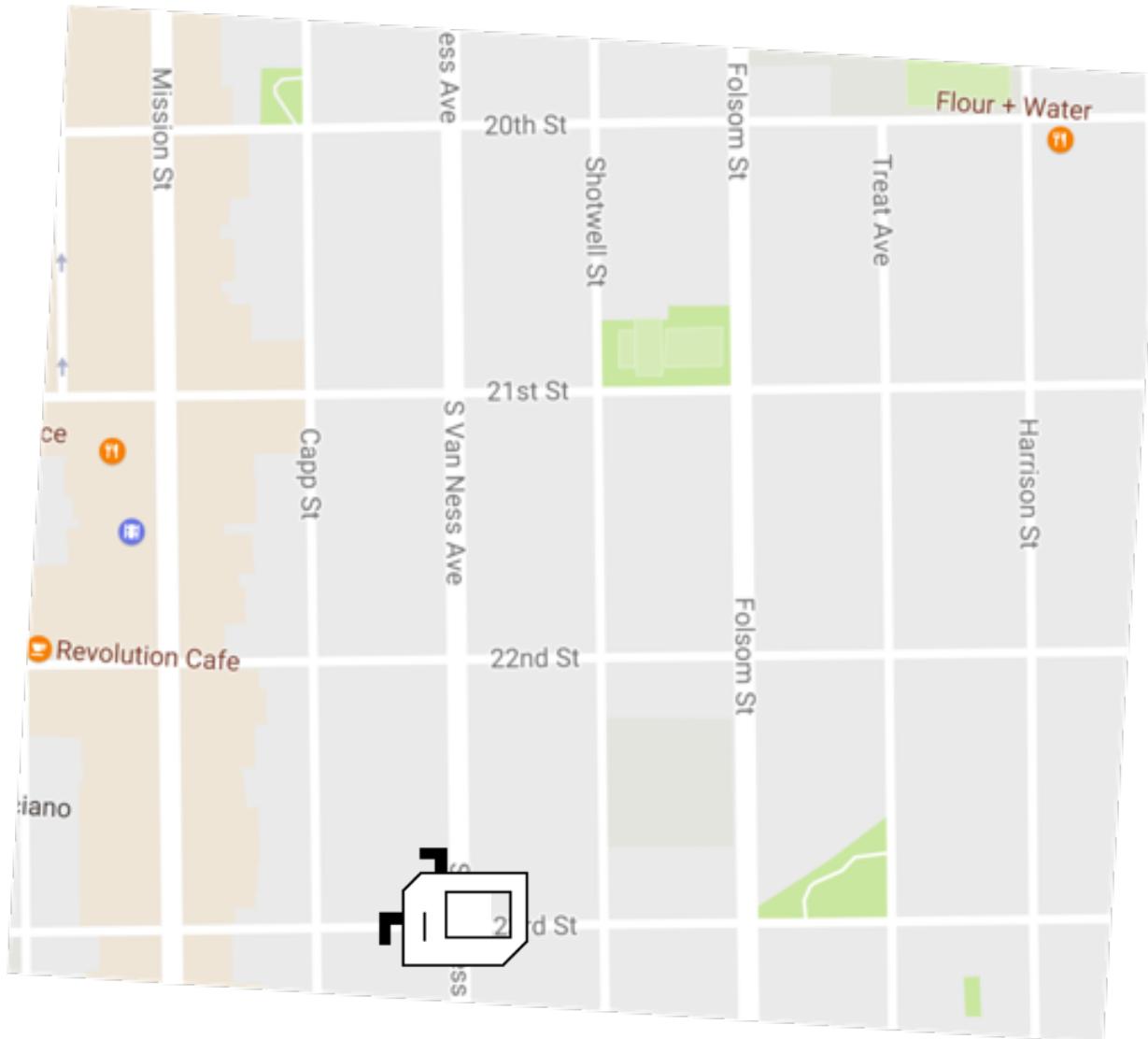
Piech and Sahami, CS106A, Stanford University



# Turn Left



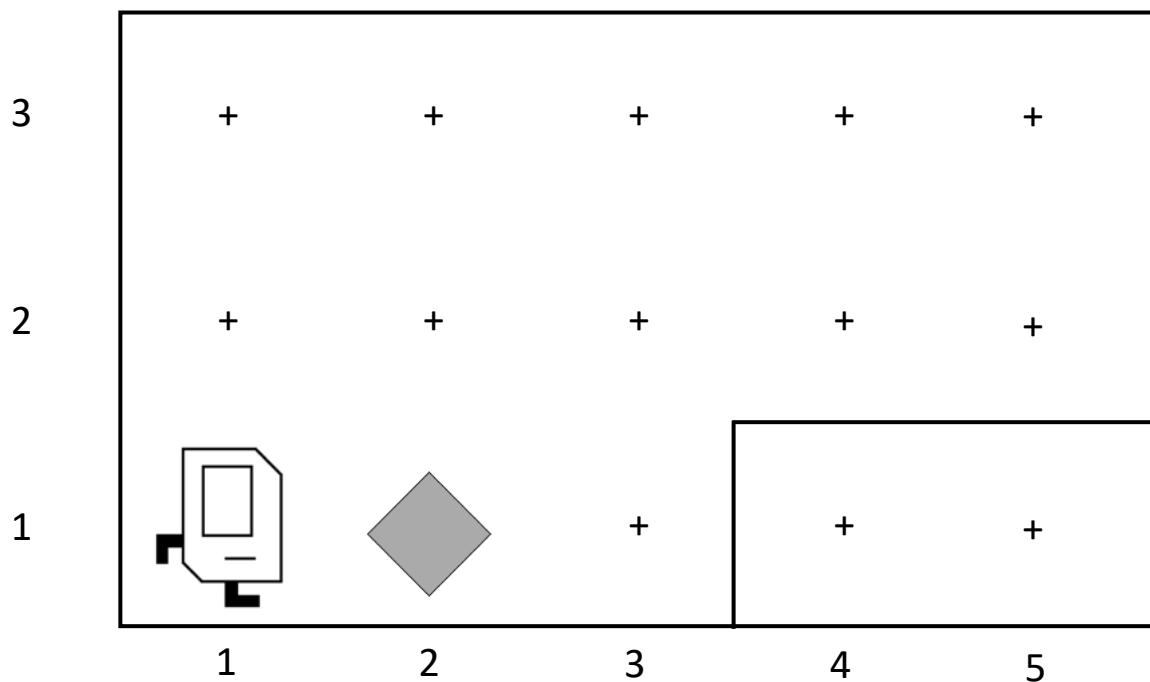
# Move



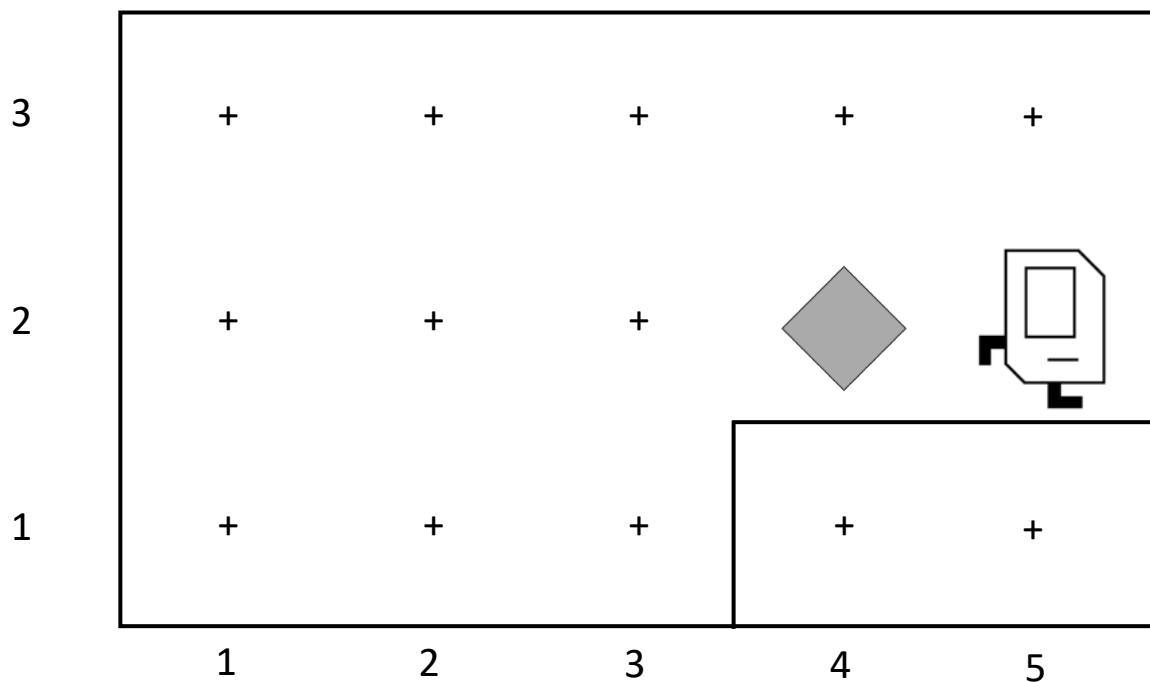
Piech and Sahami, CS106A, Stanford University



# First Challenge



# First Challenge





# Learn By Doing





The Python IDE  
for Professional  
Developers

DOWNLOAD

Full-fledged Professional or Free Community



# 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, run)

```
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, `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**



# 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**

```
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**

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

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



# Why Study CS?

# Joy of Building



# Interdisciplinary



Piech and Sahami, CS106A, Stanford University



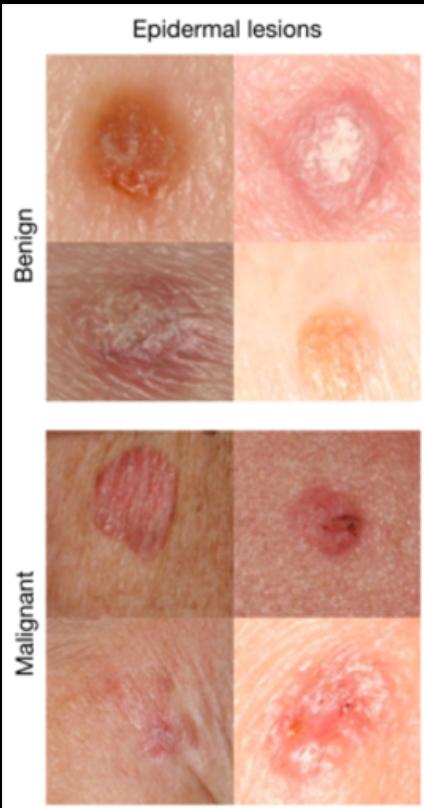
# Closest Thing To Magic



# Now is the Time



# Now is the Time



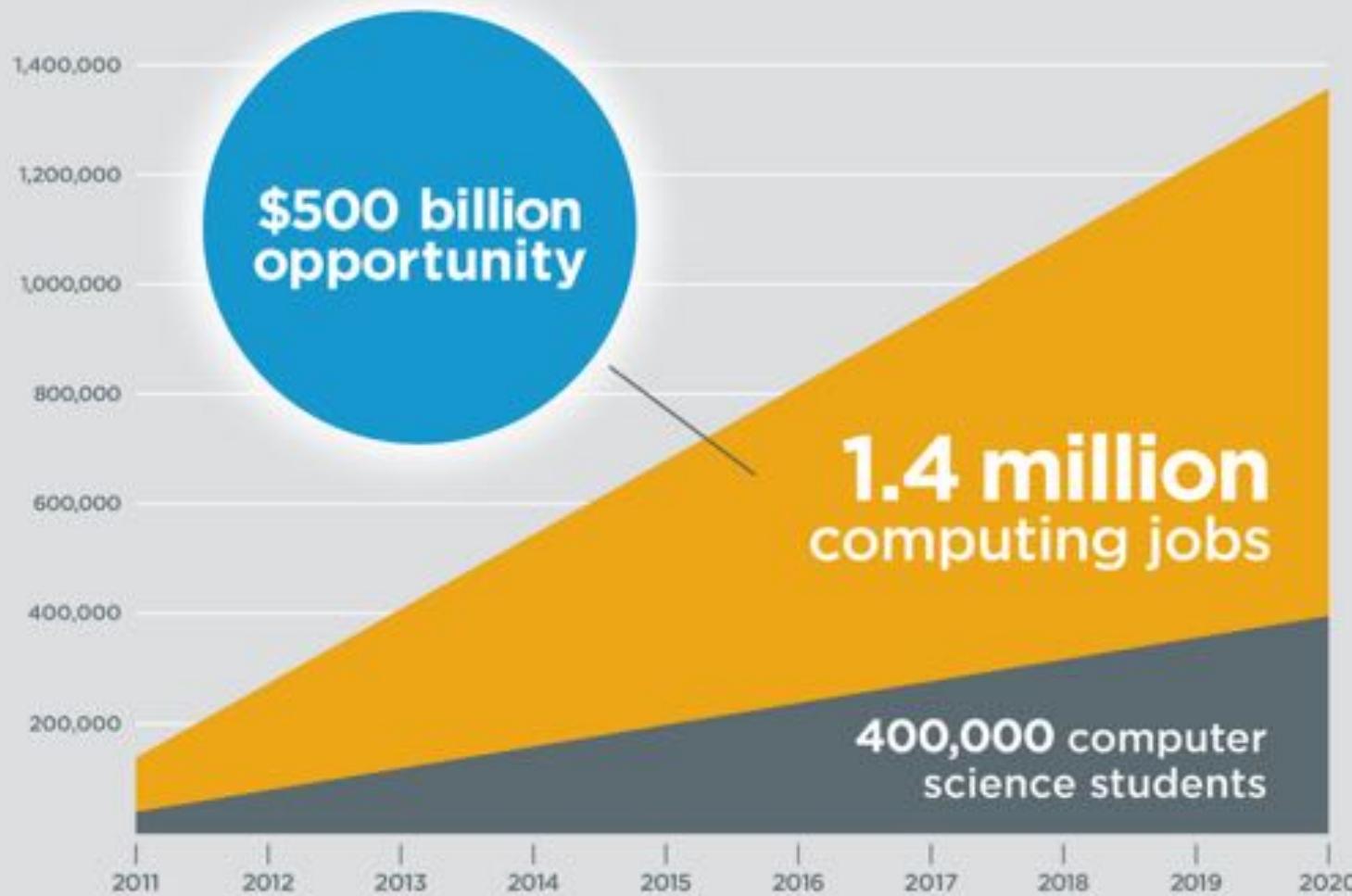
A machine learning algorithm performs **better than** the best dermatologists.

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.

# Oh and Its Useful

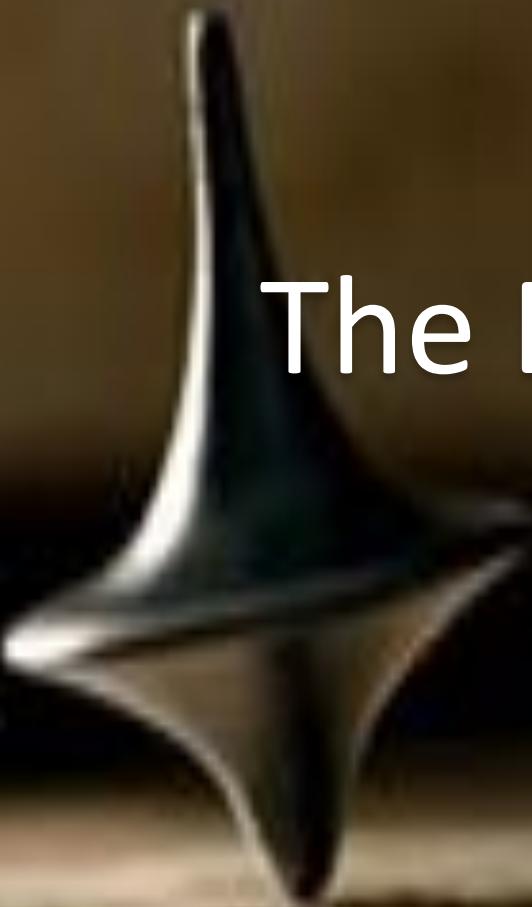
1,000,000 more jobs than students by 2020



# **Everyone is Welcome**



The End

A silver-colored spinning top is positioned vertically in the center of the frame. It is resting on a light-colored wooden surface, which appears to be a table or floor. The background is a dark, solid color.

The End?