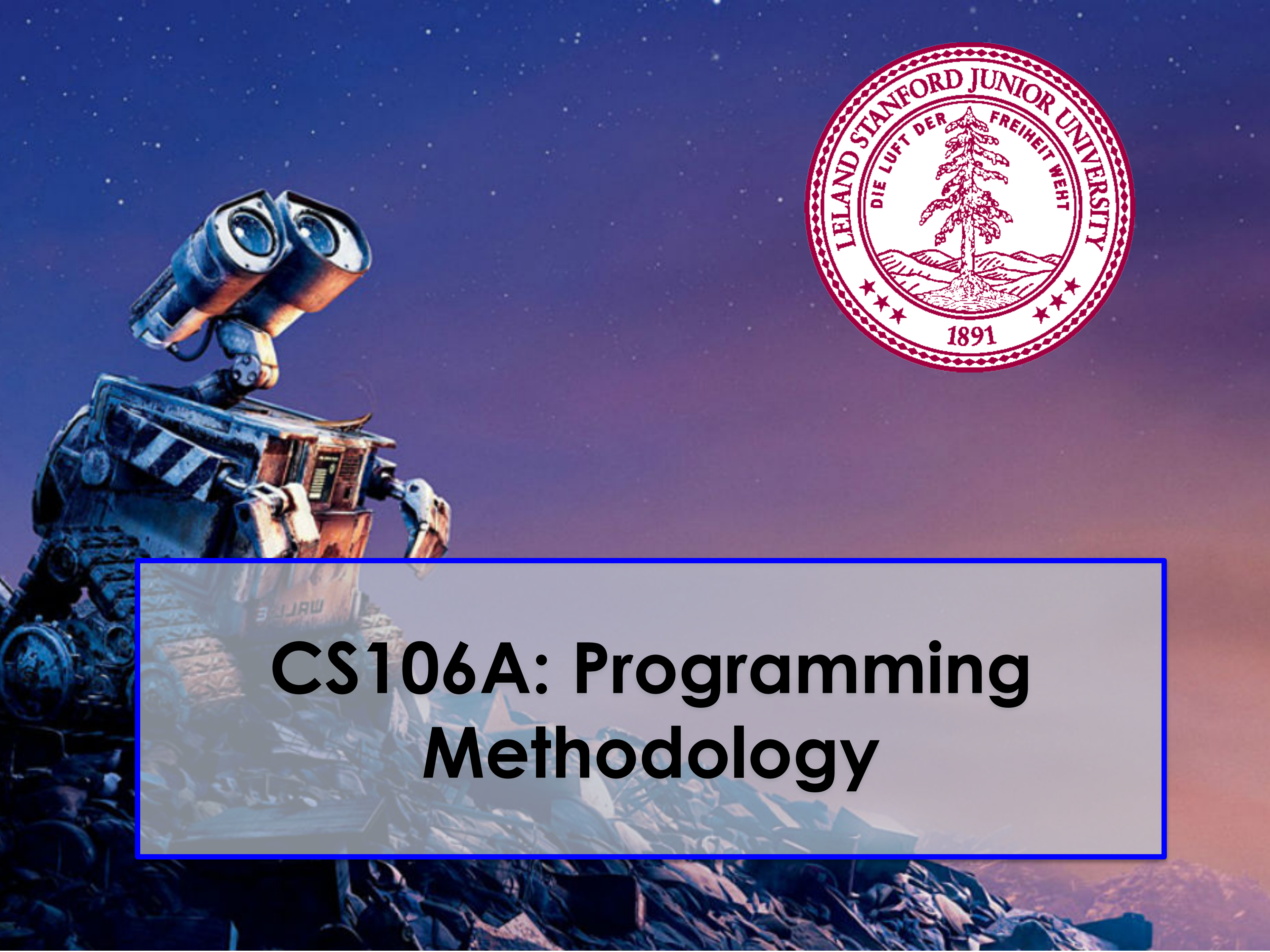


CS106A: Programming Methodology

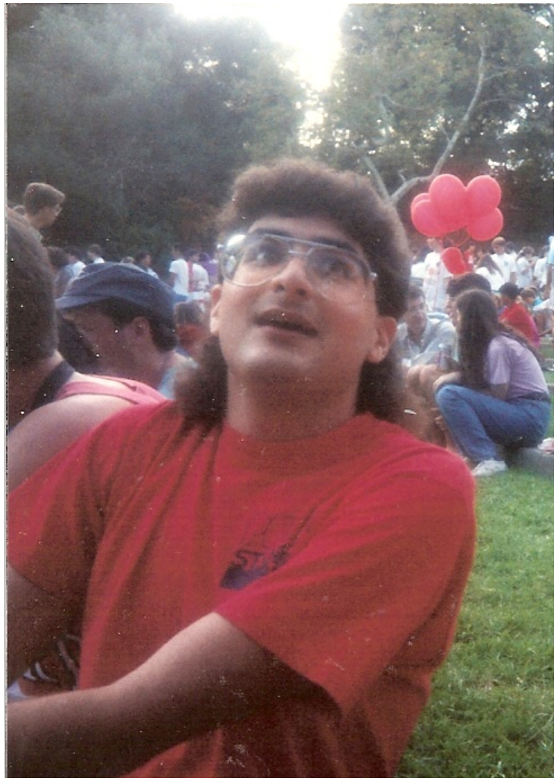
Mehran Sahami

- 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

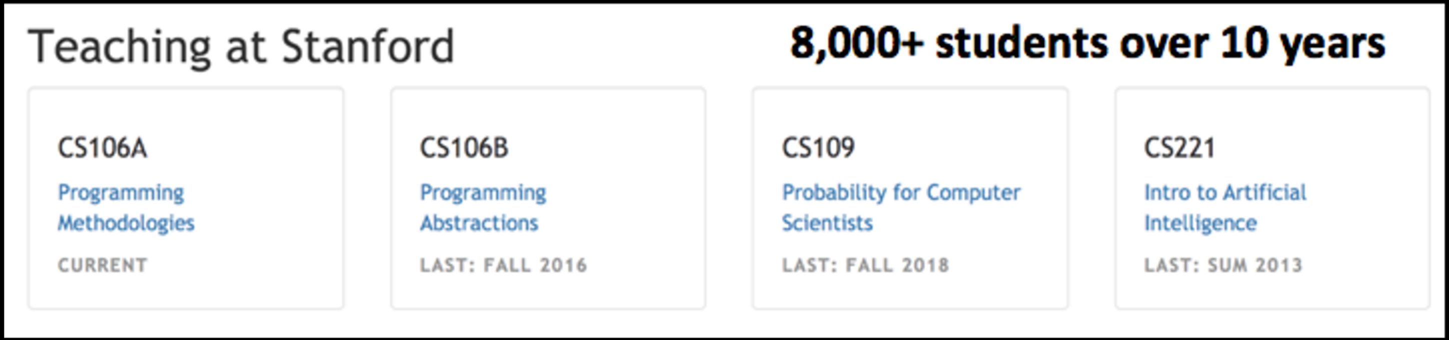
Mehran Sahami

- 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



Teaching at Stanford

8,000+ students over 10 years

CS106A
Programming Methodologies
CURRENT

CS106B
Programming Abstractions
LAST: FALL 2016

CS109
Probability for Computer Scientists
LAST: FALL 2018

CS221
Intro to Artificial Intelligence
LAST: SUM 2013

Created a research lab in:
Computational Education

Grew up in Nairobi, Kuala Lumpur before Stanford!

Chris Piech

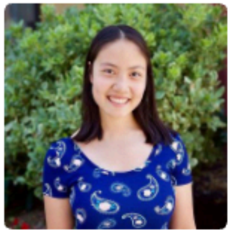
12 years ago to this day, I was sitting in your seats

Head TA: Juliette Woodrow

Section Leaders

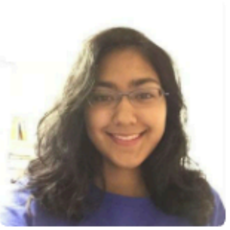
Luciano
Gonzalez ✉

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 ✉

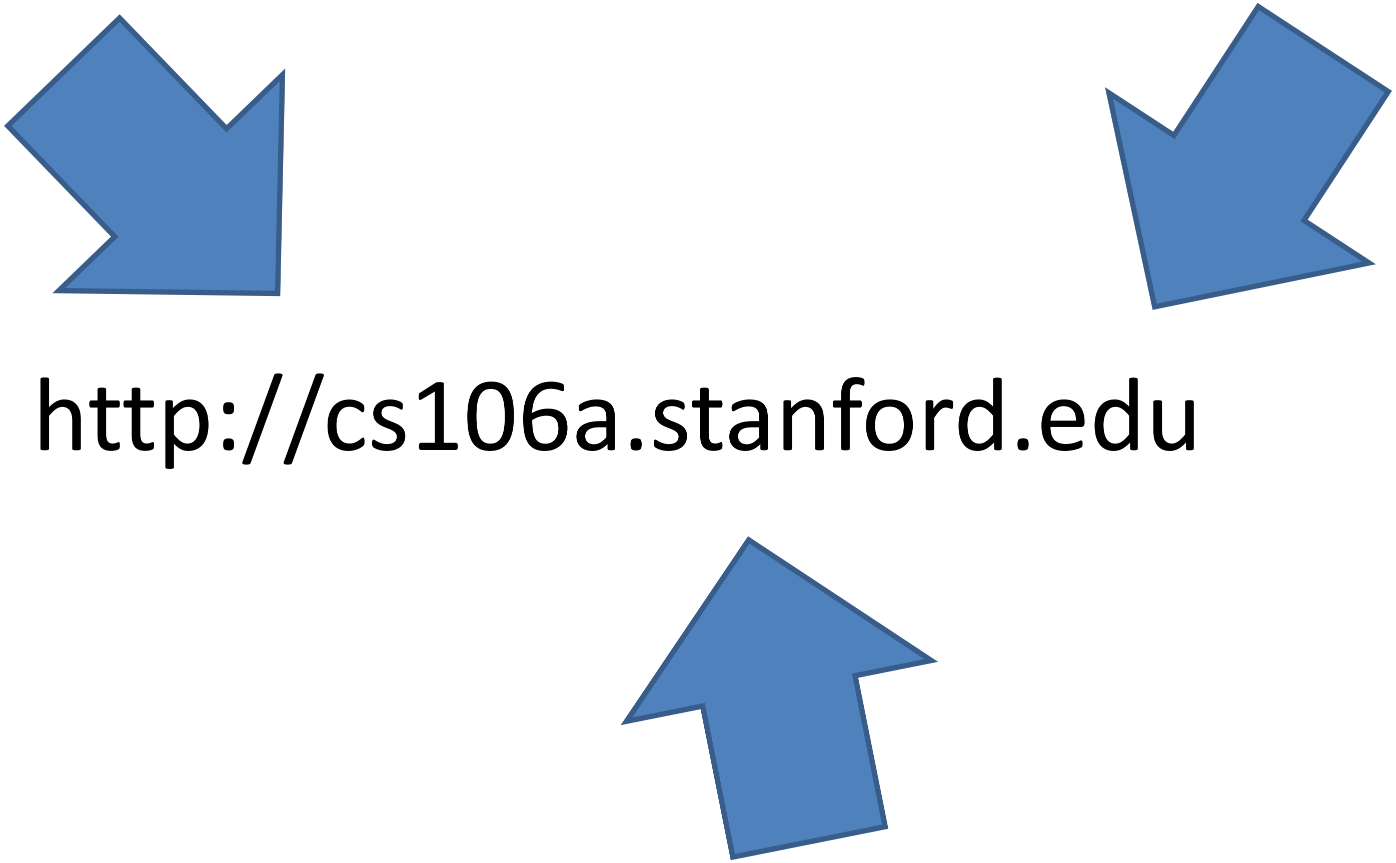
Vineet
Kosaraju ✉

* Actually some past section leaders

Course mechanics

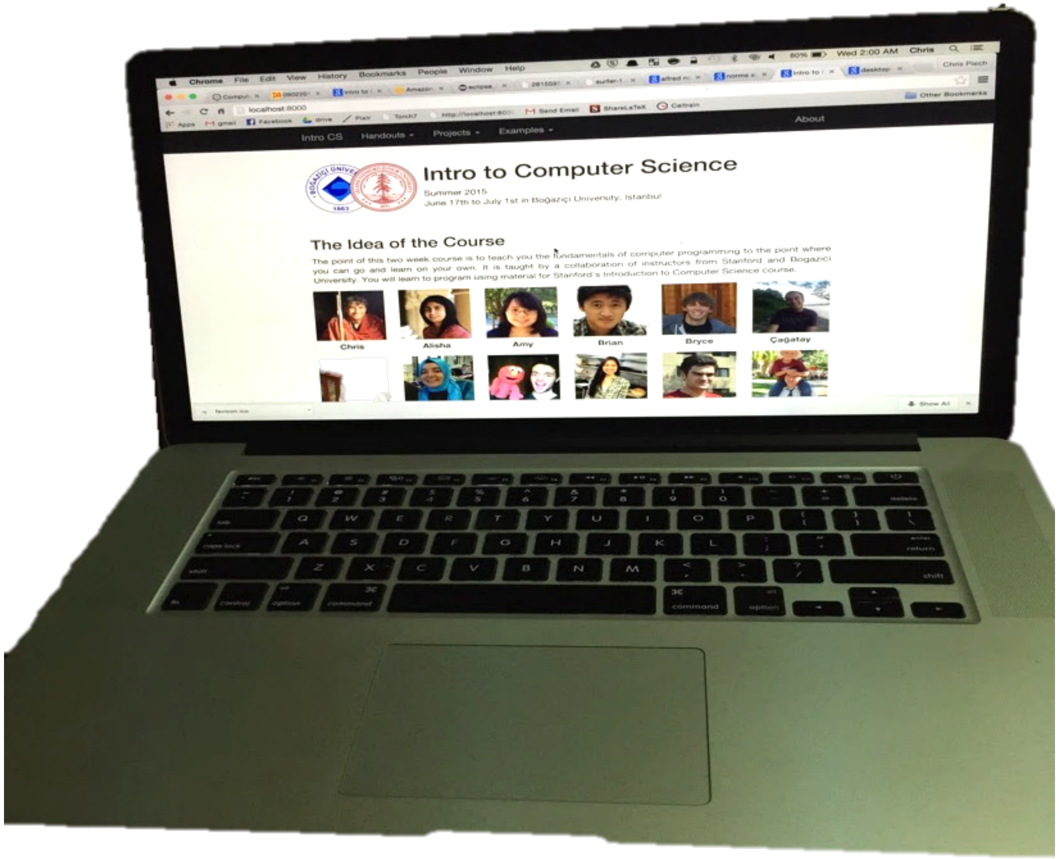
(This is a brief version.
Please read the handout for full details).

Course Website



http://cs106a.stanford.edu

Prerequisite Test



Intro to Computer Science

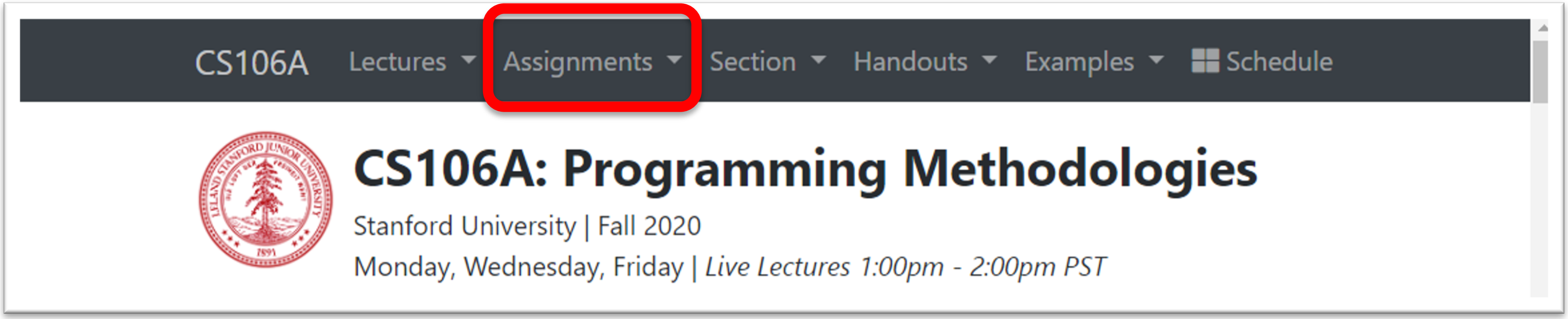
Summer 2015
June 17th to July 1st in Boğaziçi University, Istanbul

The Idea of the Course

The point of this two week course is to teach you the fundamentals of computer programming to the point where you can go and learn on your own. It is taught by a collaboration of instructors from Stanford and Bogazici University. You will learn to program using material for Stanford's Introduction to Computer Science course.

Getting To Know You

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



CS106A: Programming Methodologies

Stanford University | Fall 2020

Monday, Wednesday, Friday | *Live Lectures 1:00pm - 2:00pm PST*

"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 1:00-2:00pm
  - 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.

Office Hours

LaIR: evenings Sunday through Thursday (starting Sunday)

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

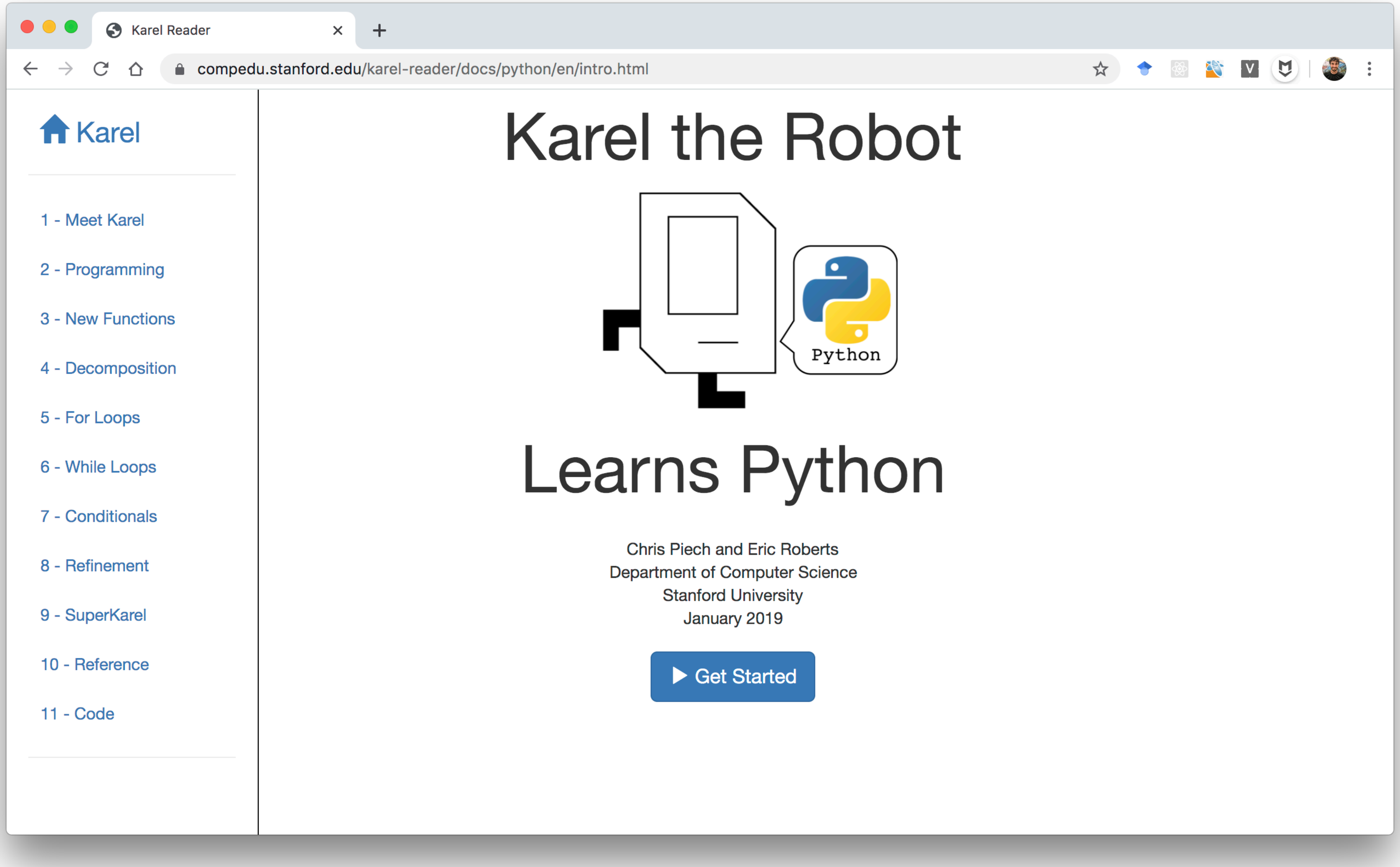
What we will ask you to do

- 7 programming assignment 70%
  - Get more complicated as quarter progresses
- In-class diagnostic assessment I (exam) 10%
- In-class diagnostic assessment II (exam) 10%
- 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 Juliette (head TA)

Optional Contest

Online Text Books



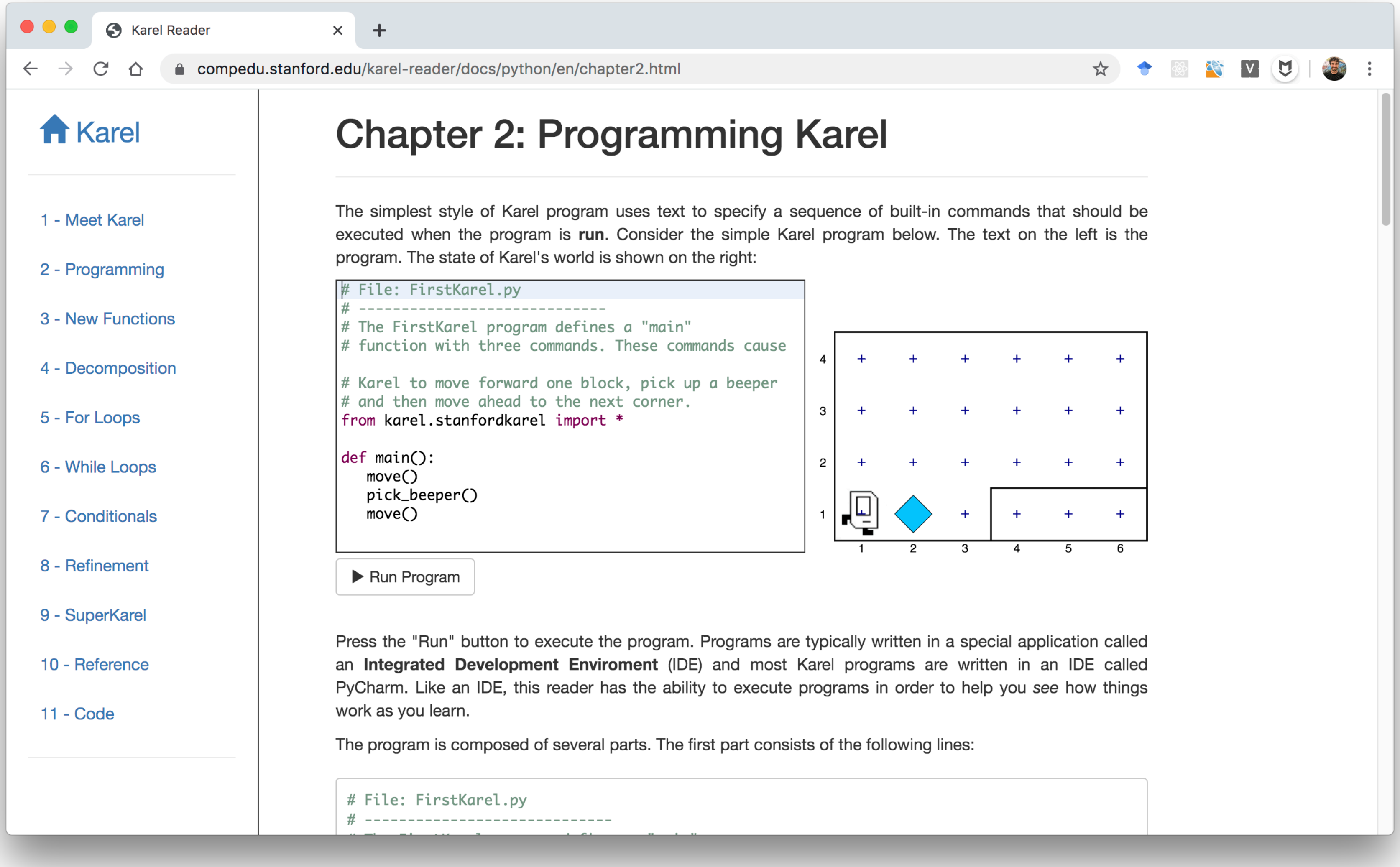
1 - Meet Karel
2 - Programming
3 - New Functions
4 - Decomposition
5 - For Loops
6 - While Loops
7 - Conditionals
8 - Refinement
9 - SuperKarel
10 - Reference
11 - Code

Karel the Robot

Learns Python

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

Online Karel Reader


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

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()
```

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.

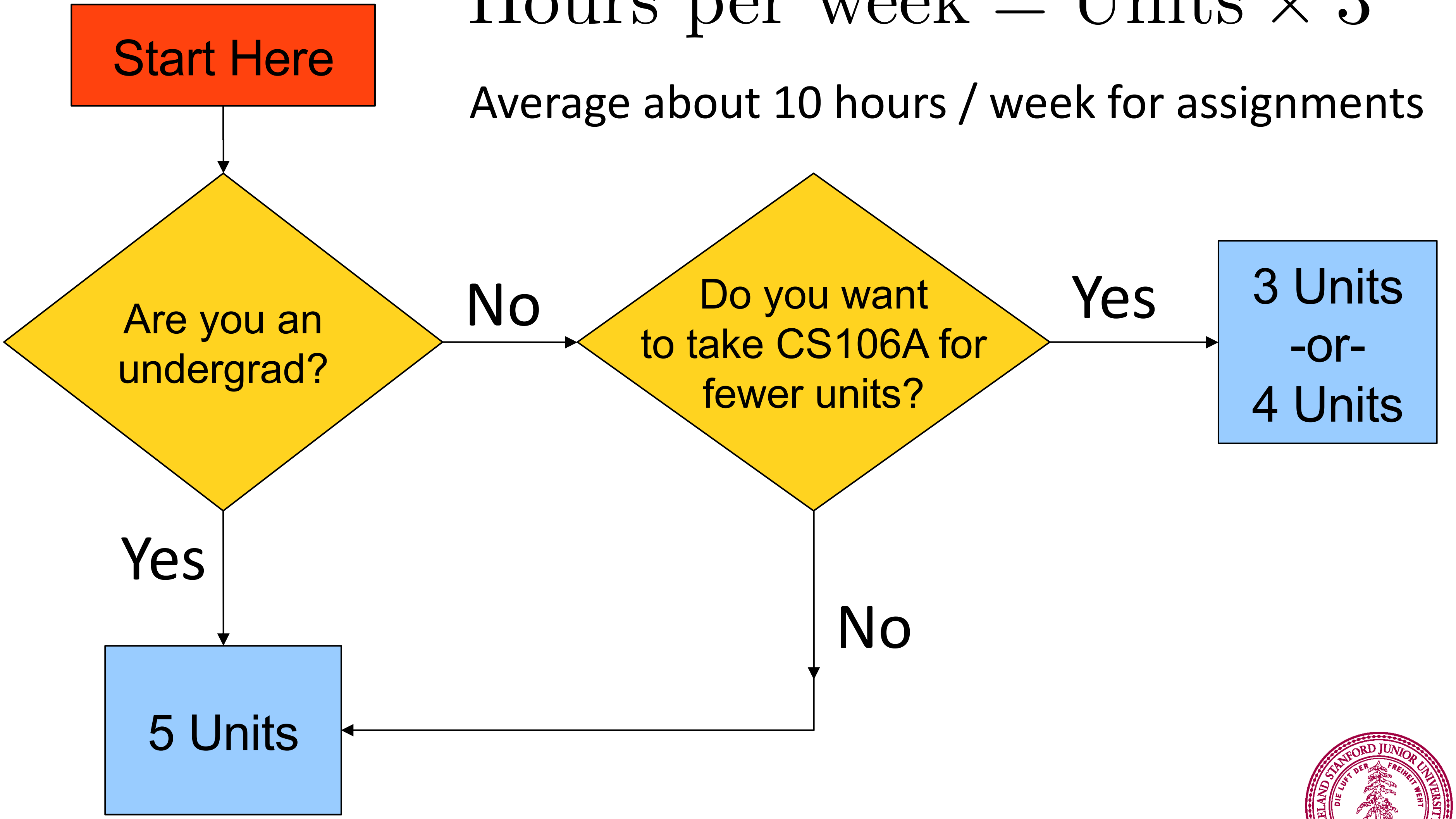
The program is composed of several parts. The first part consists of the following lines:

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

CS106A Units

$$\text{Hours per week} = \text{Units} \times 3$$

Average about 10 hours / week for assignments



Are you in the right place?

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 (+M)
- No AP, significant previous programming experience → CS106B (+M)
- *Extensive* prior experience and/or multiple prior CS classes → CS106B (+M) or CS107
- Just want to satisfy "Ways" and know that will be all you'll take → CS105 or 106A

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.

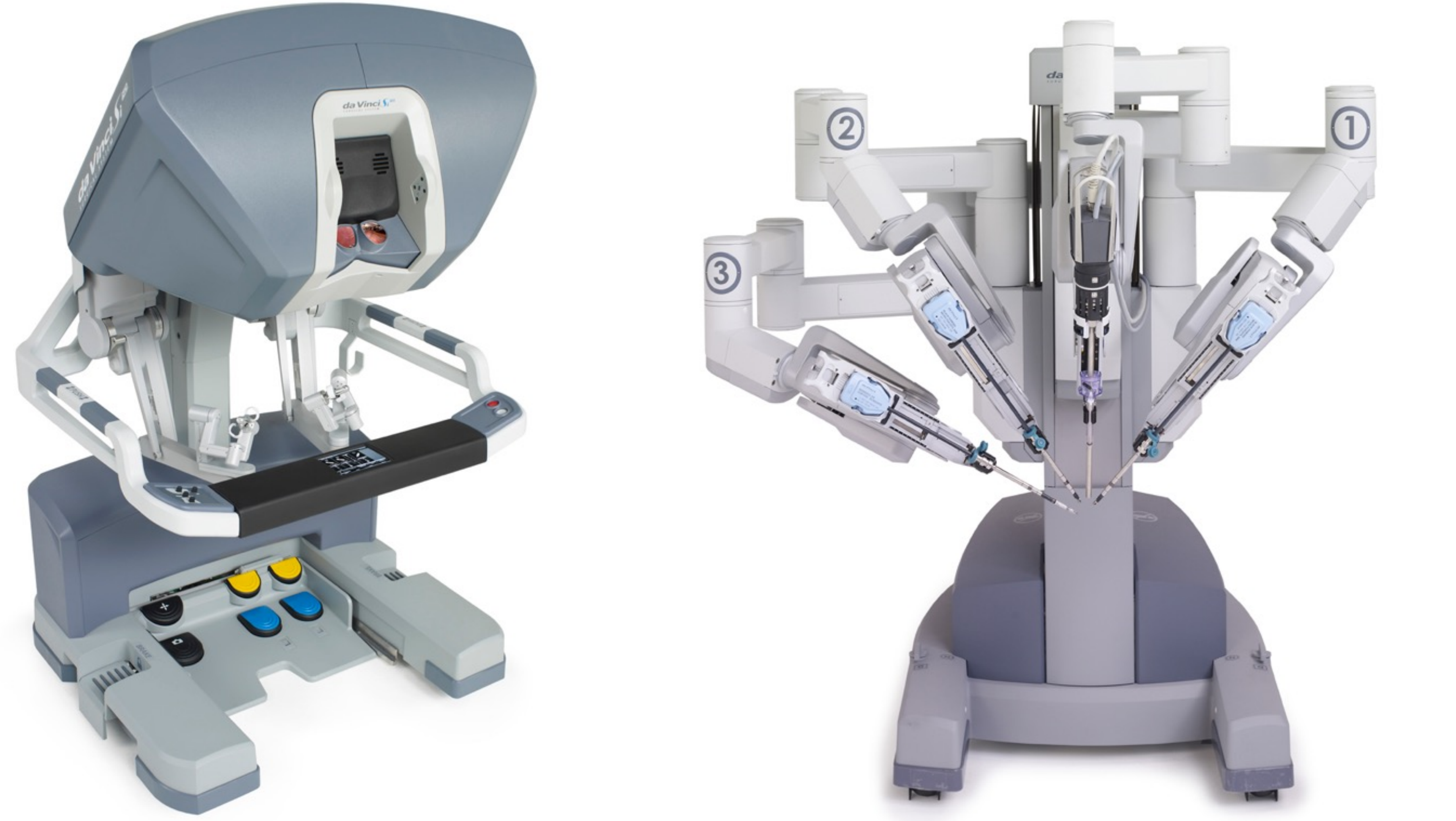
There are a lot of cool programs you may one day write

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

Consumer Applications

Computing in Medicine

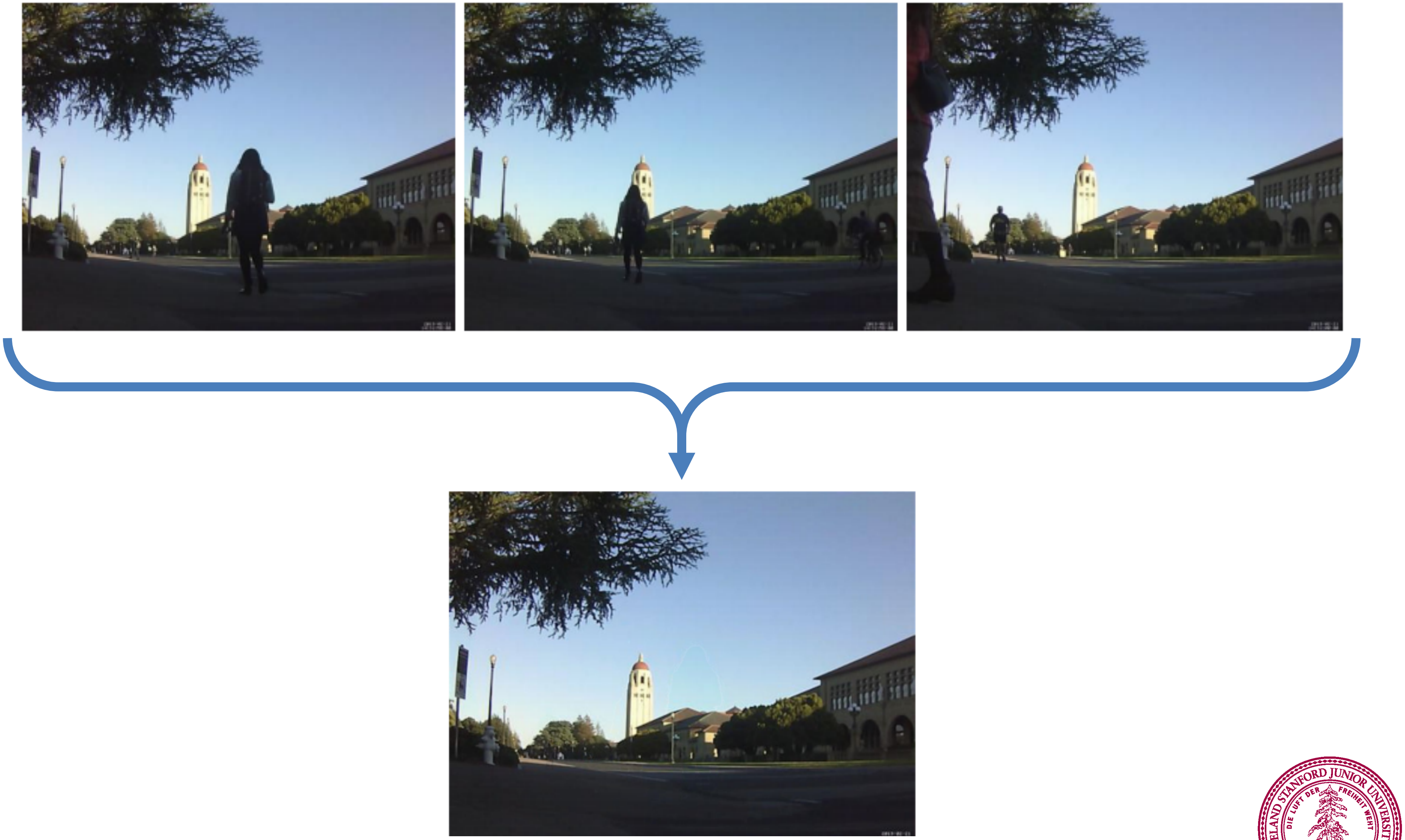


(c) 2012 Intuitive Surgical, Inc.

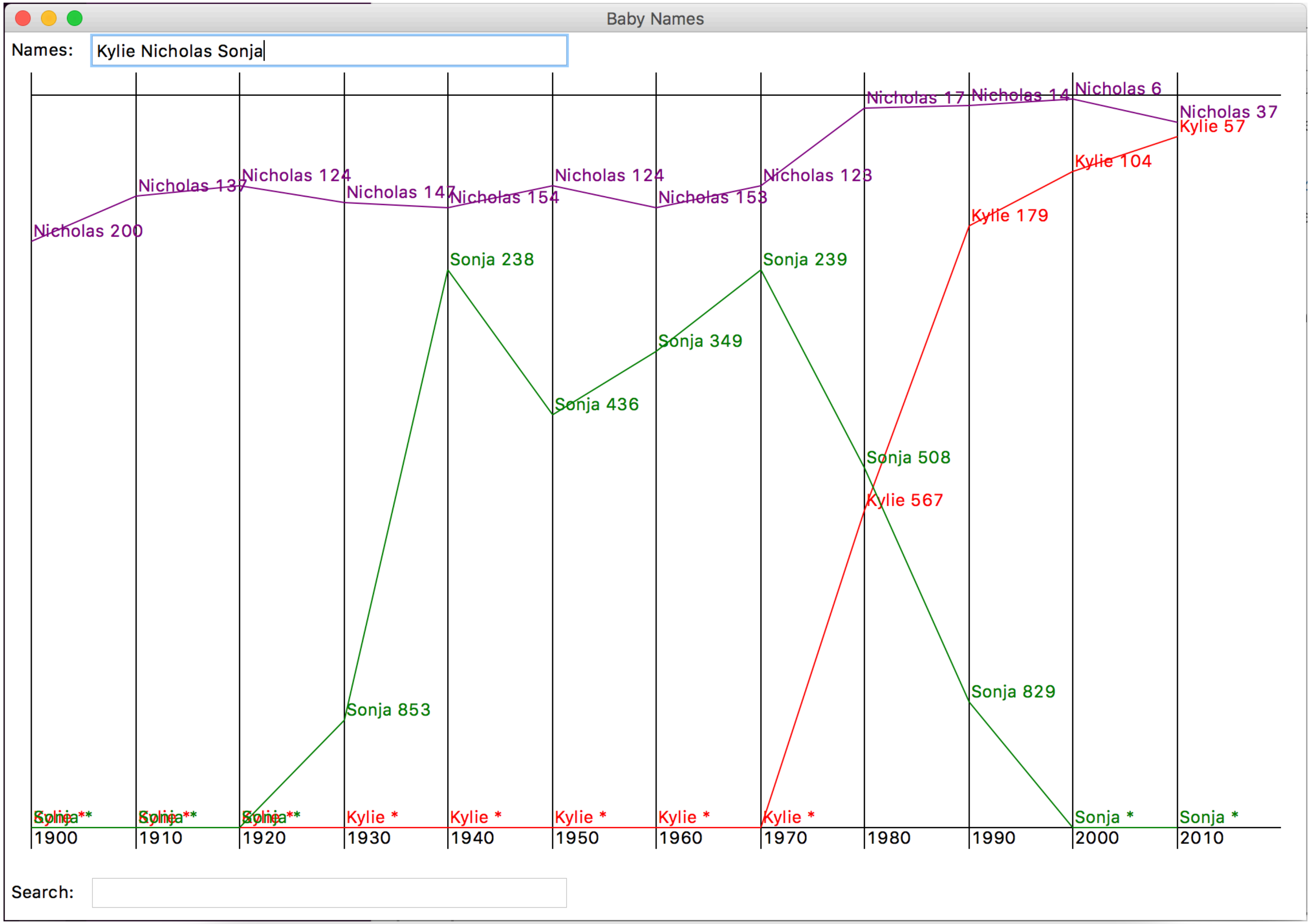
Self-Driving Car

If only we could program self-driving cars…

Image Transformation



Data Science

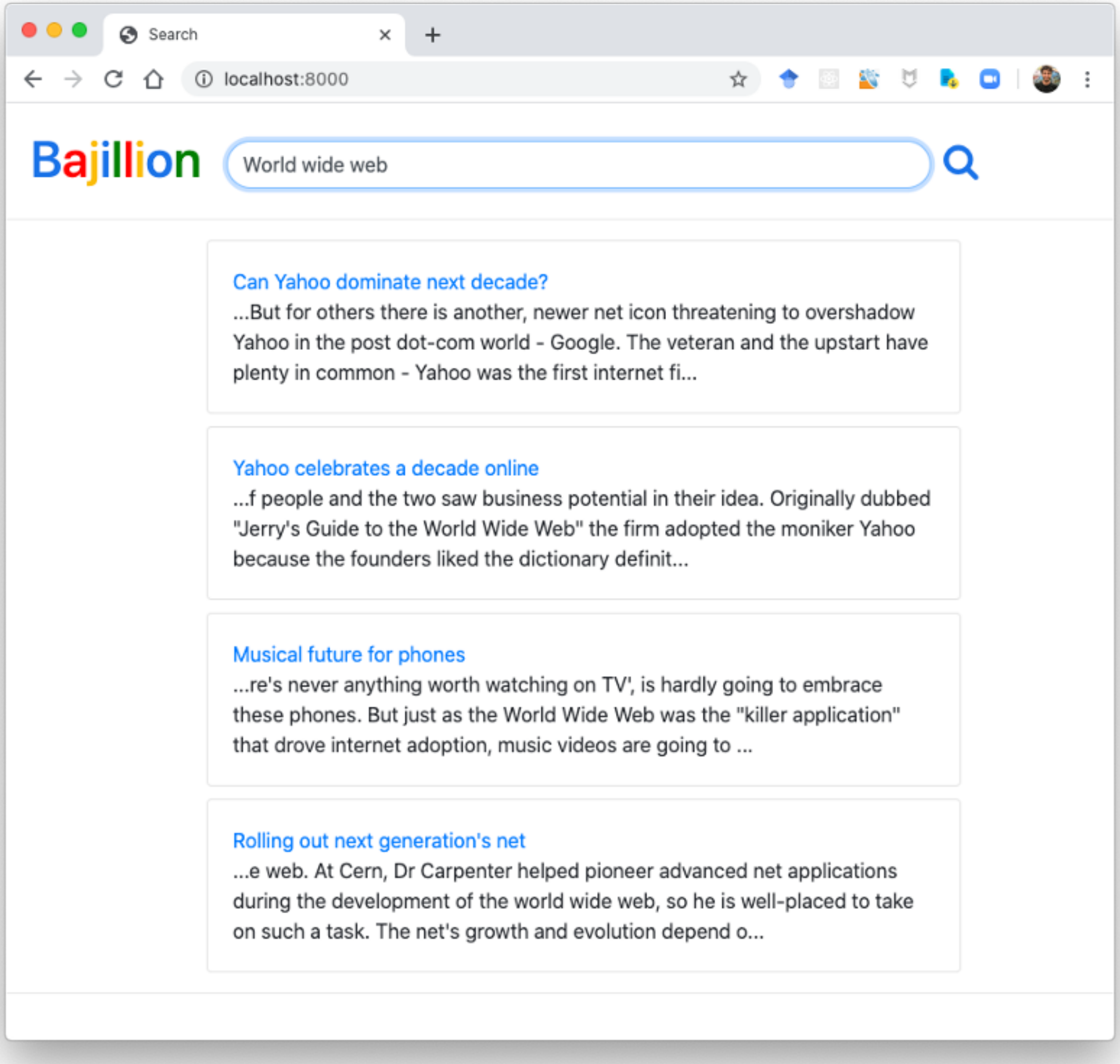


Baby Names

Names: Kylie Nicholas Sonja

Search:

Internet Applications



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...

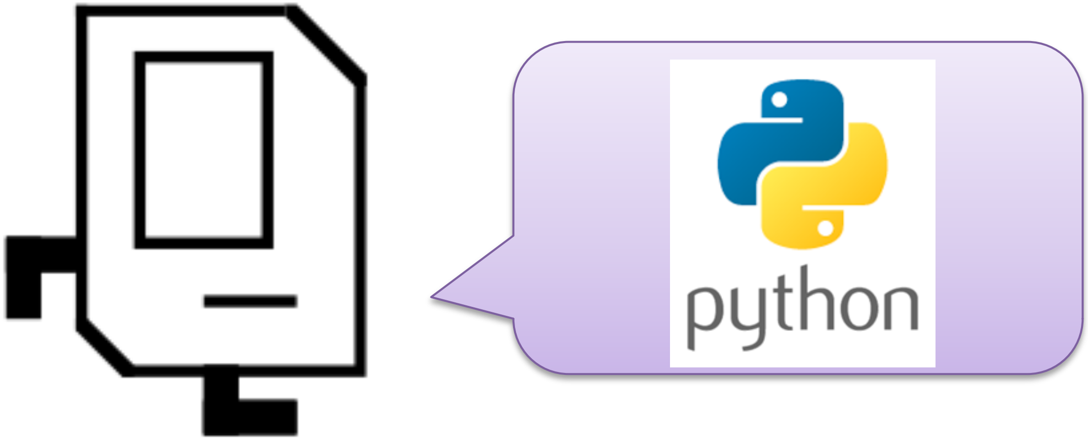
Strive for Everyone to Succeed

Lets Get Started

Meet Karel the Robot

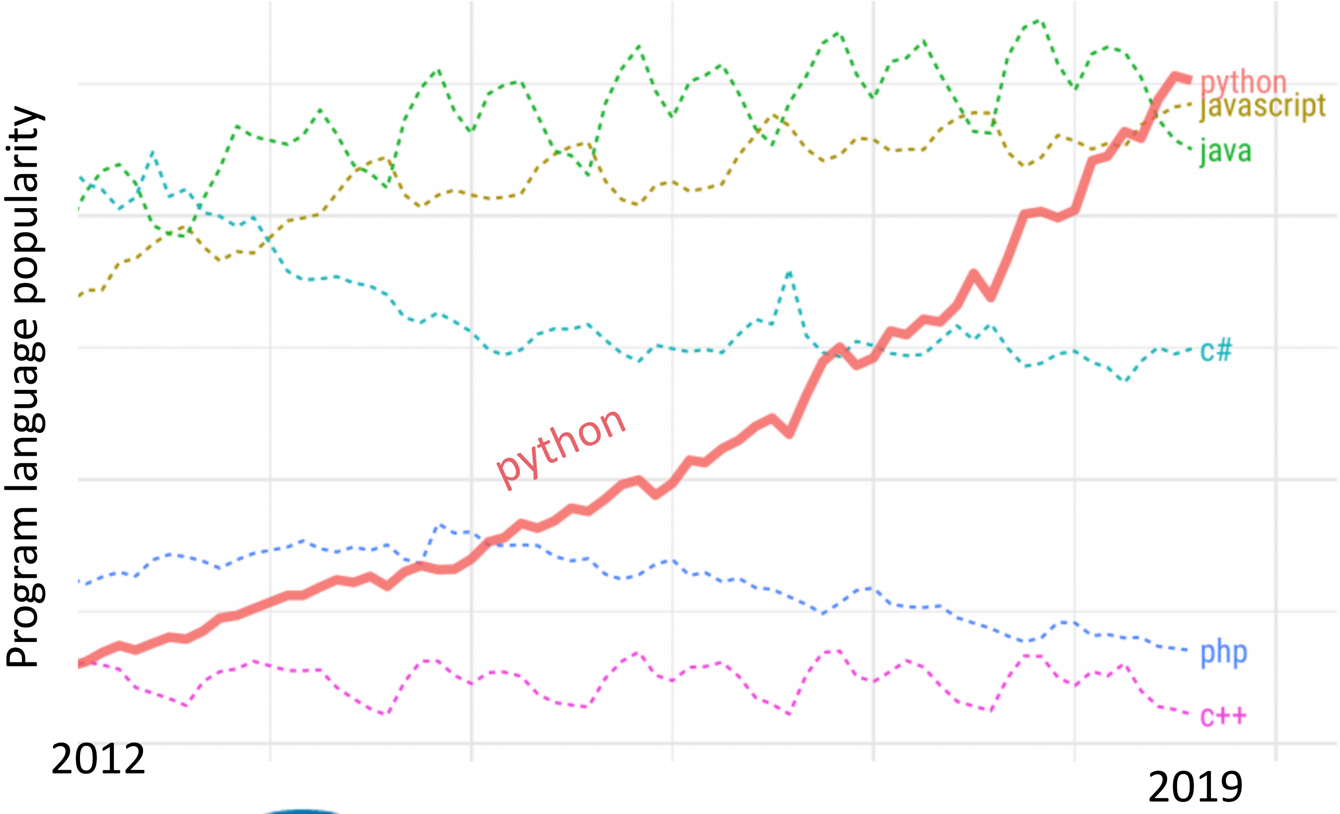
Good morning

Karel Speaks Python



Why Python?

1



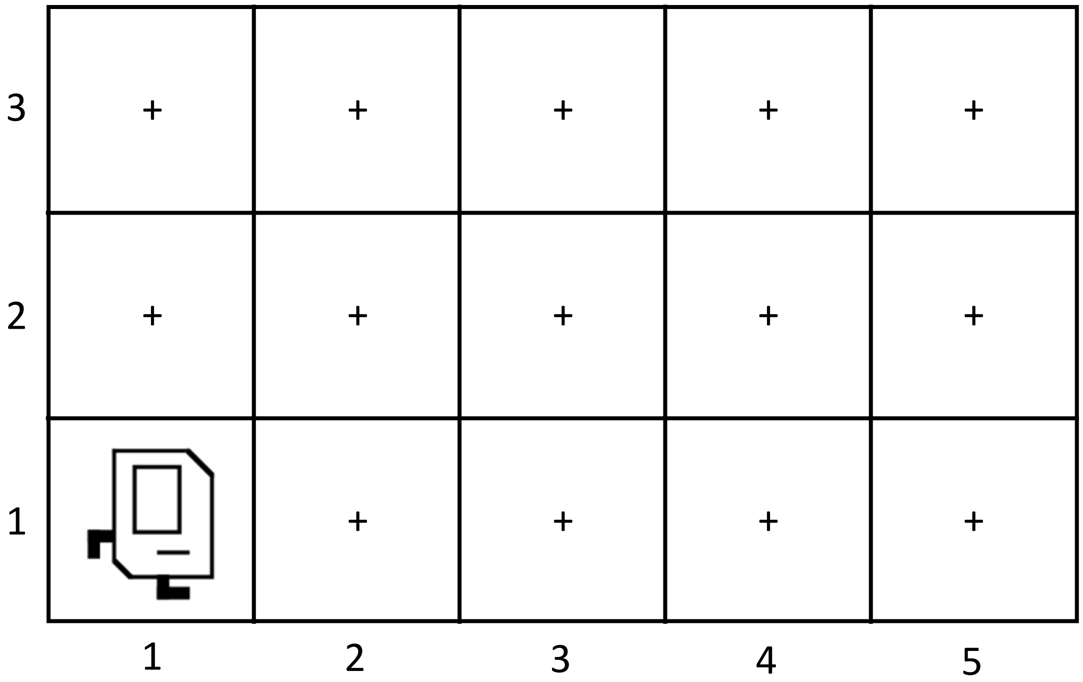
2

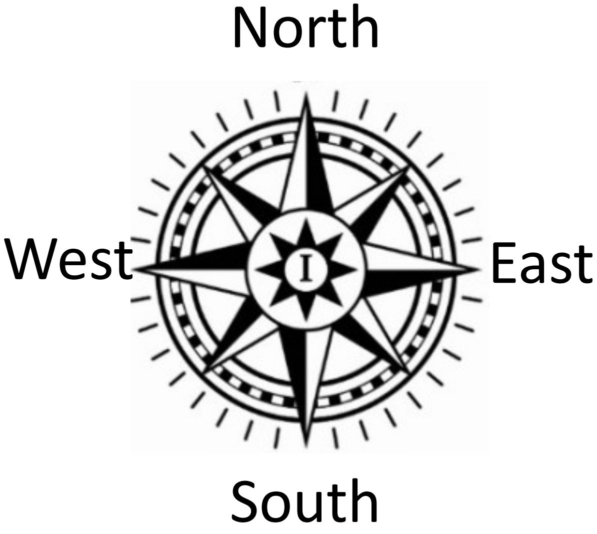
python

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

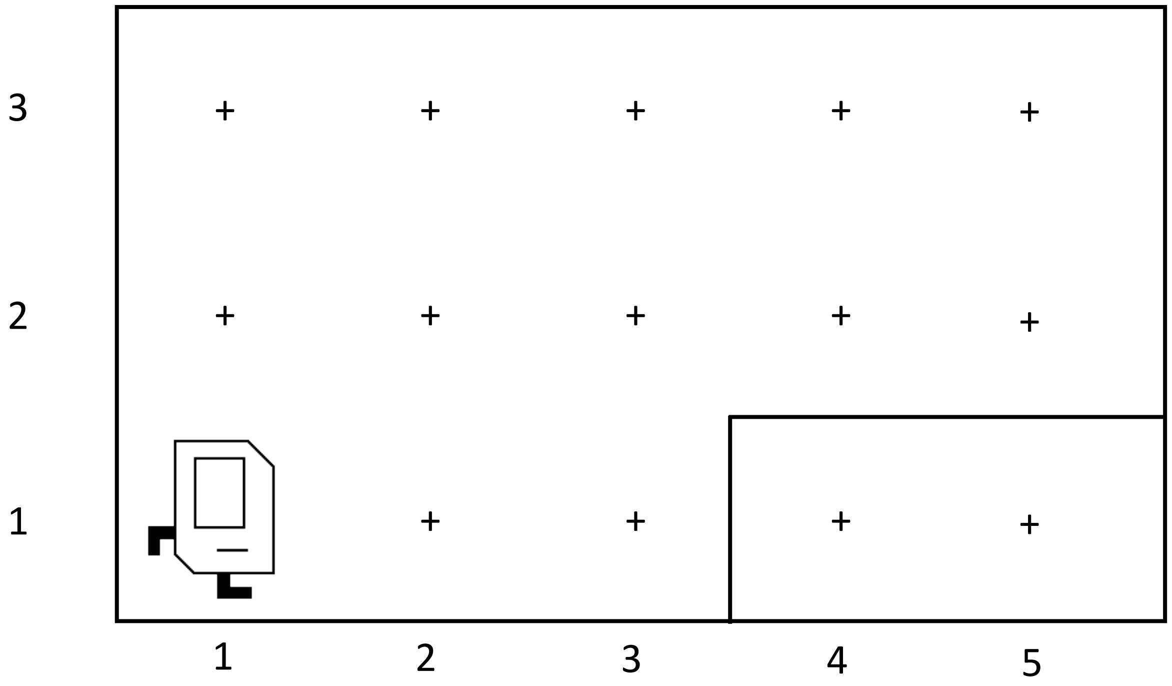
Guido van Rossum

Karel’s World

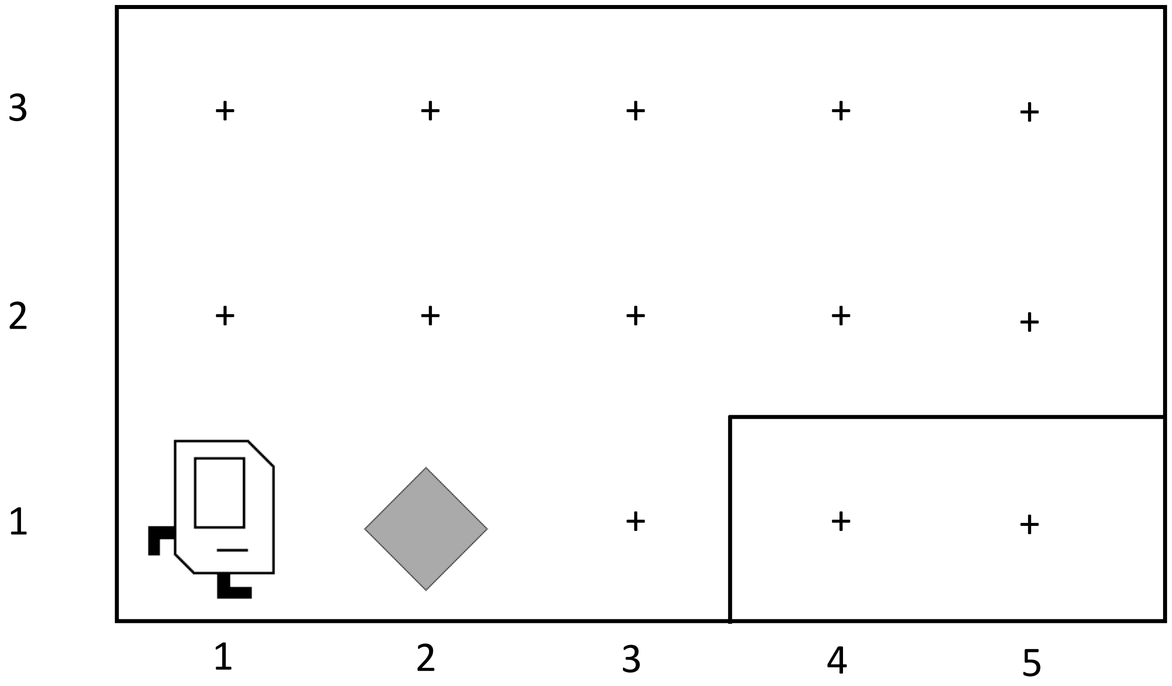




Walls



Beepers

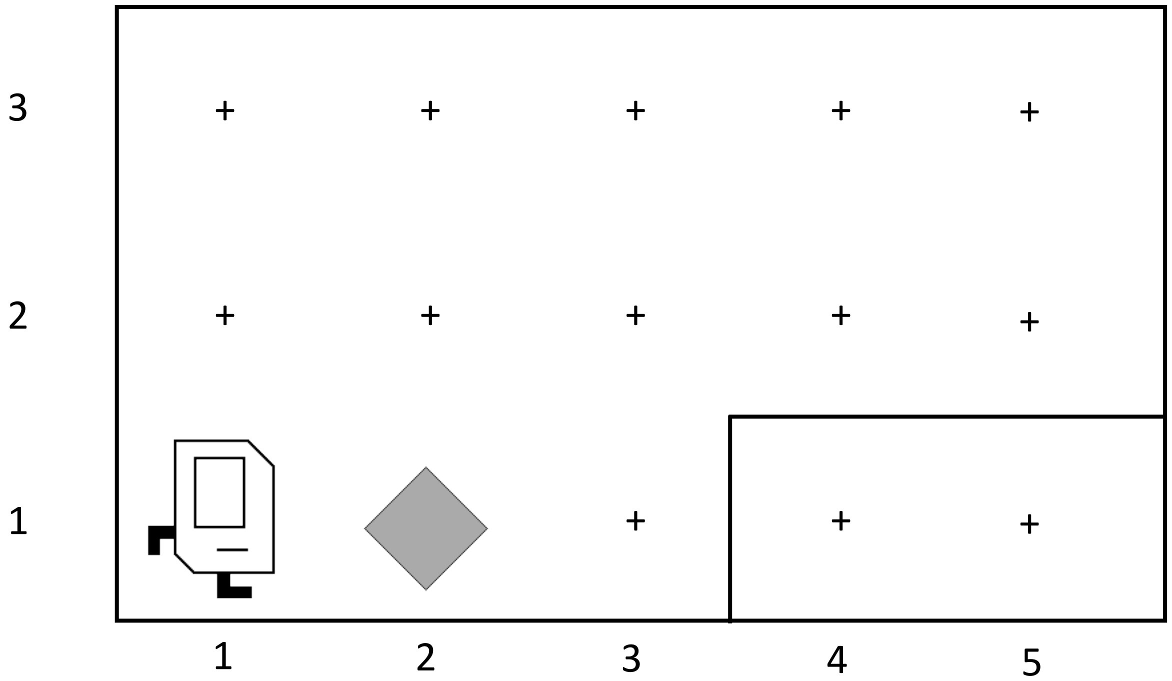


Knows Four Commands

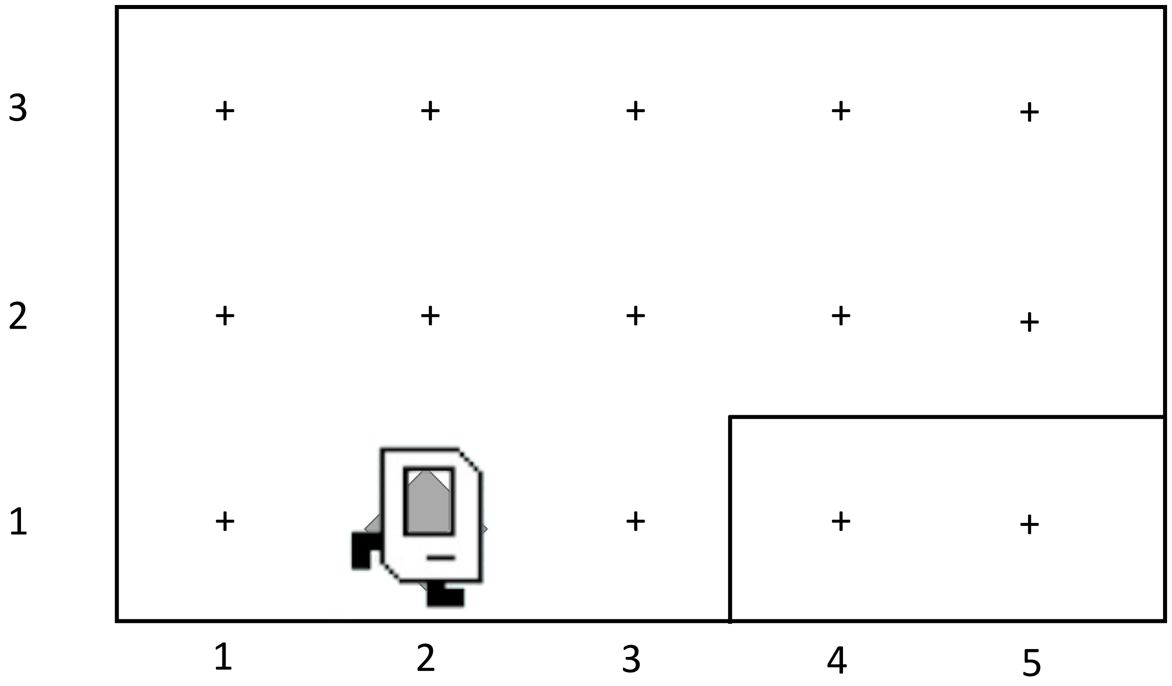
```
move()

turn_left()

put_beeper()

pick_beeper()
```

```
move()
```

`move()`

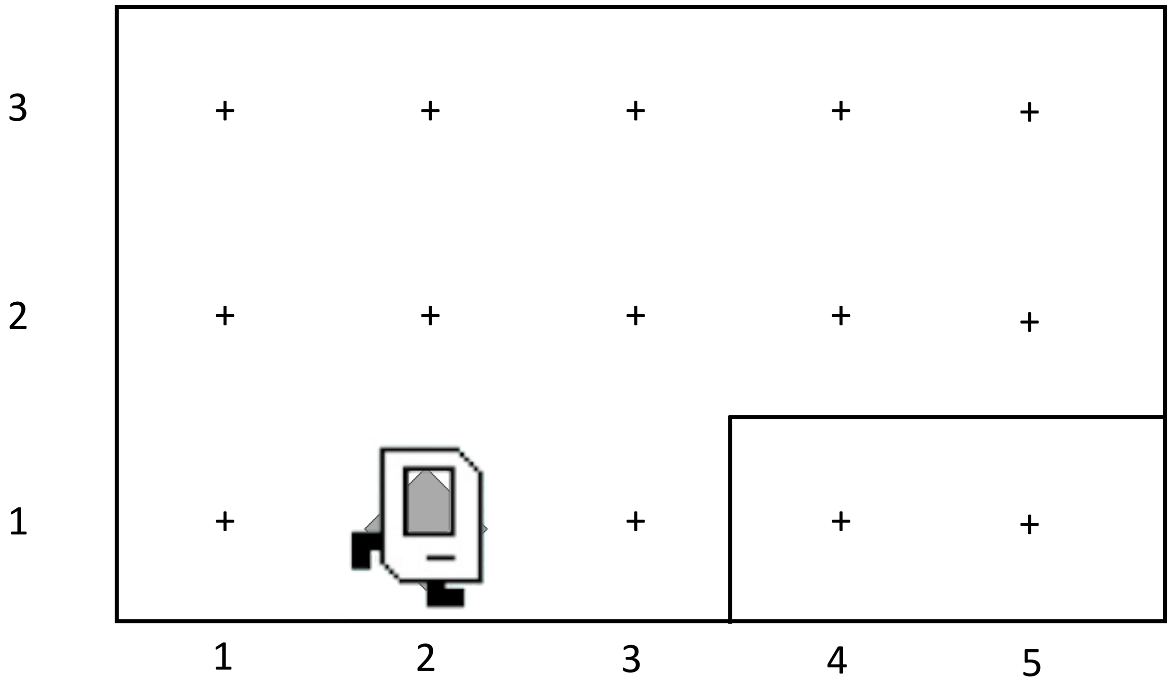


move()

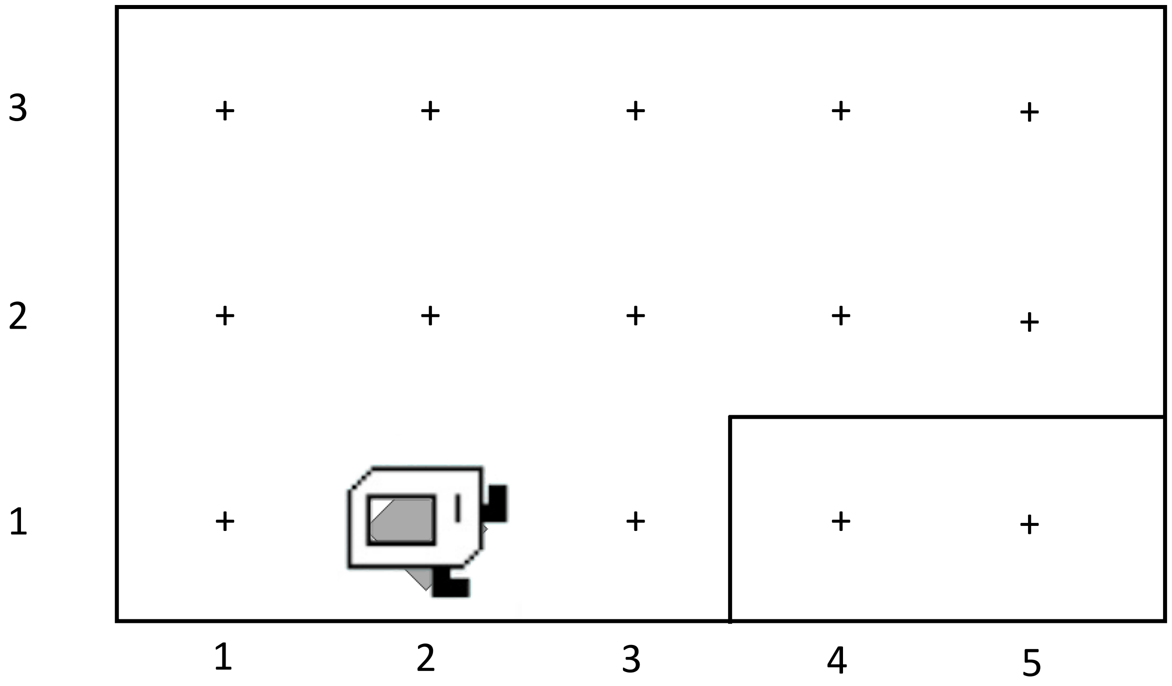


```
turn_left()
```

turn_left()

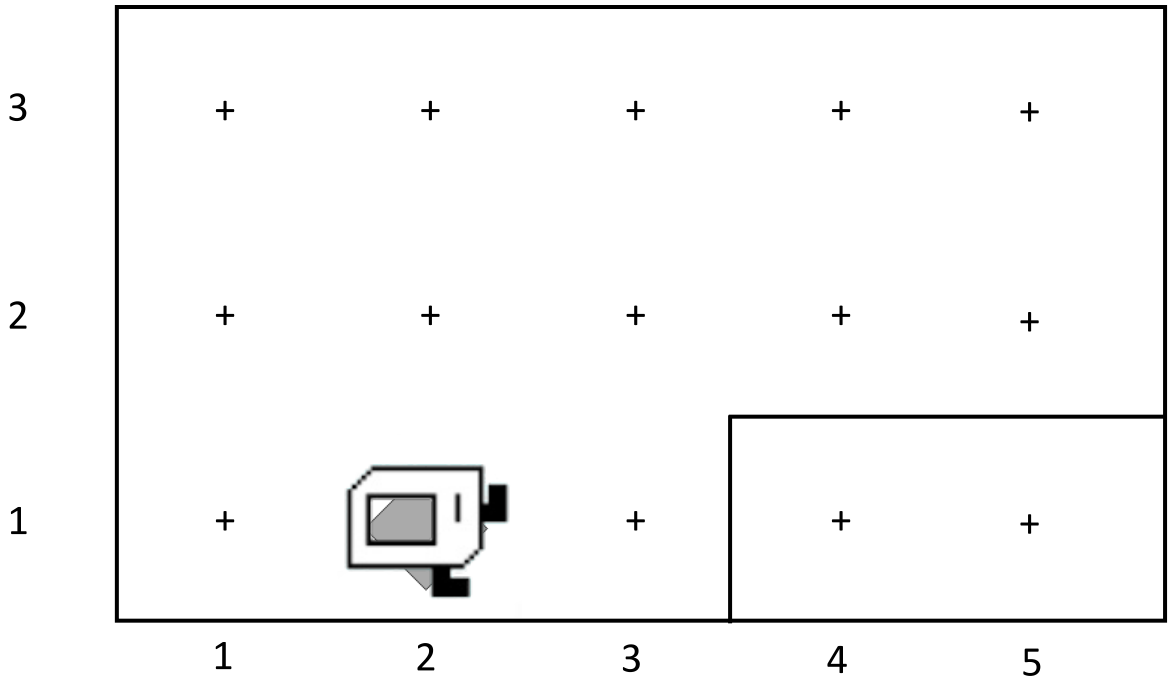


`turn_left()`

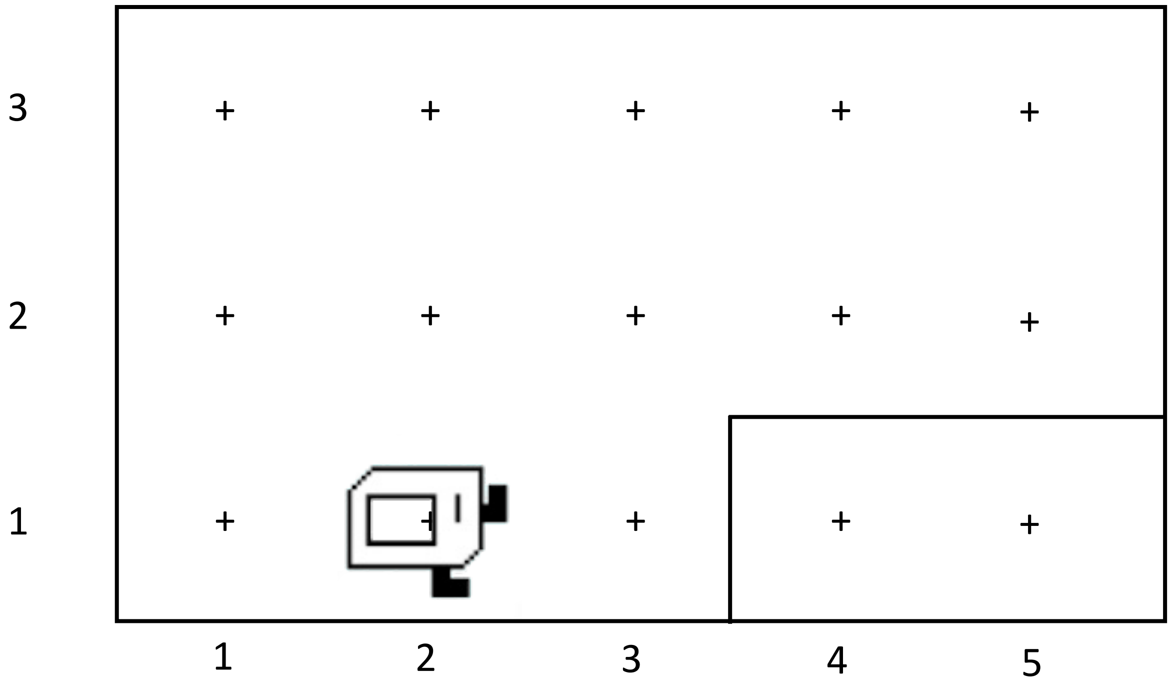


```
pick_beeper()
```

turn_left()

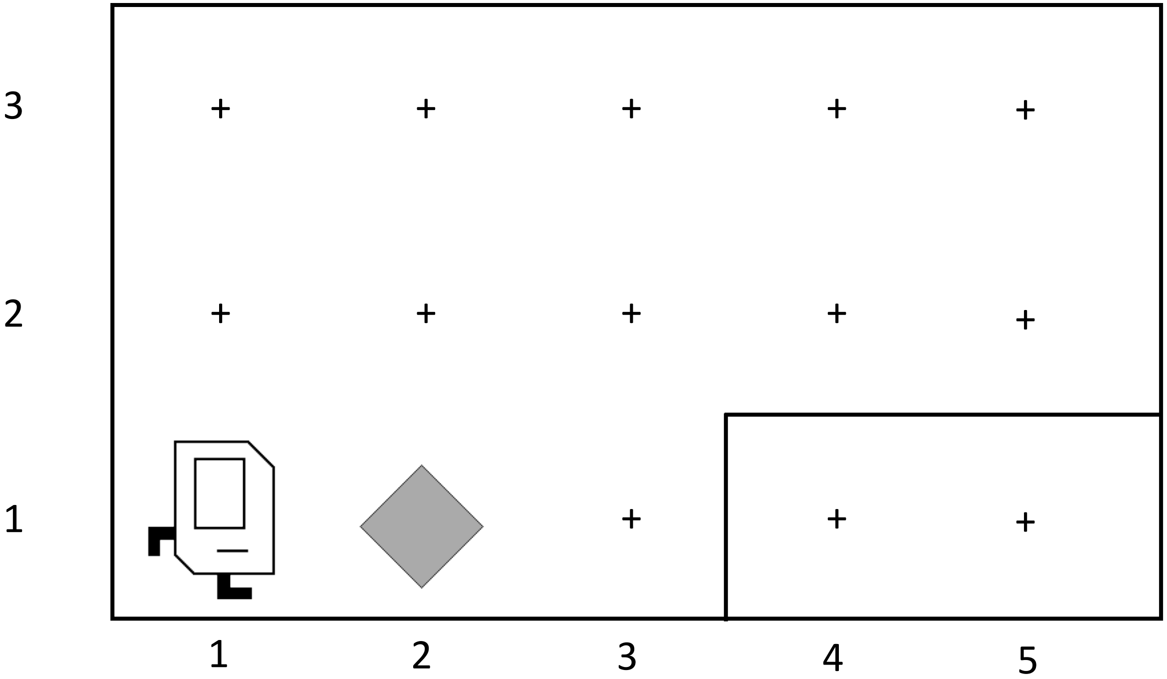


`turn_left()`

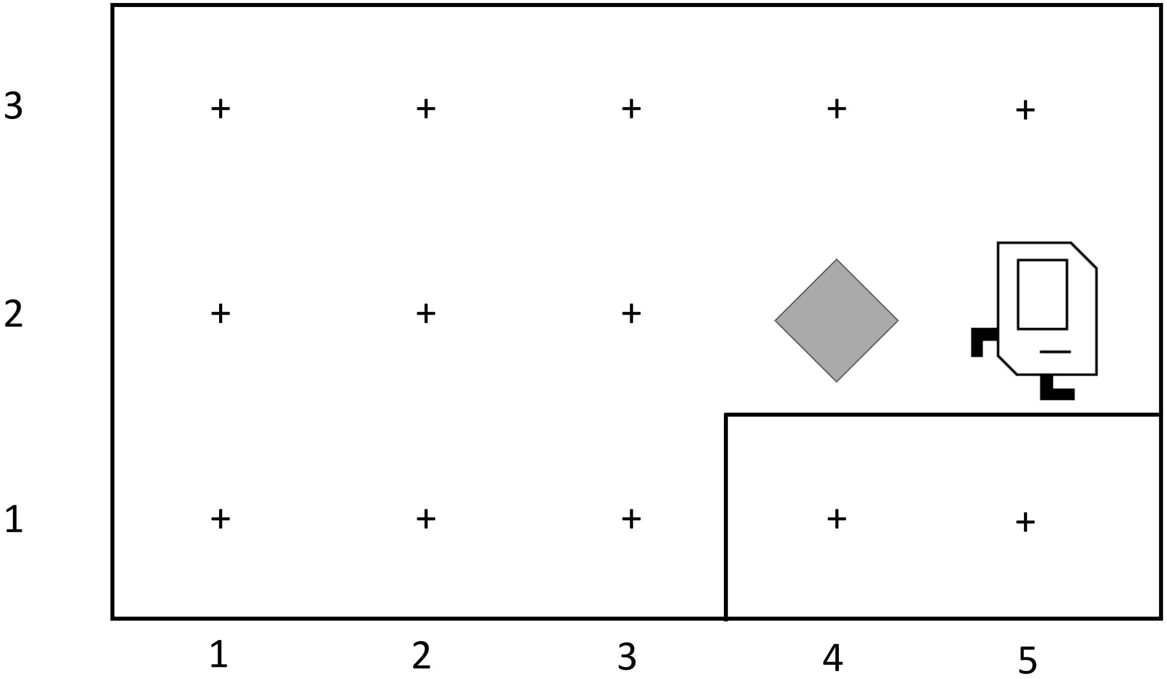


Make Sense?

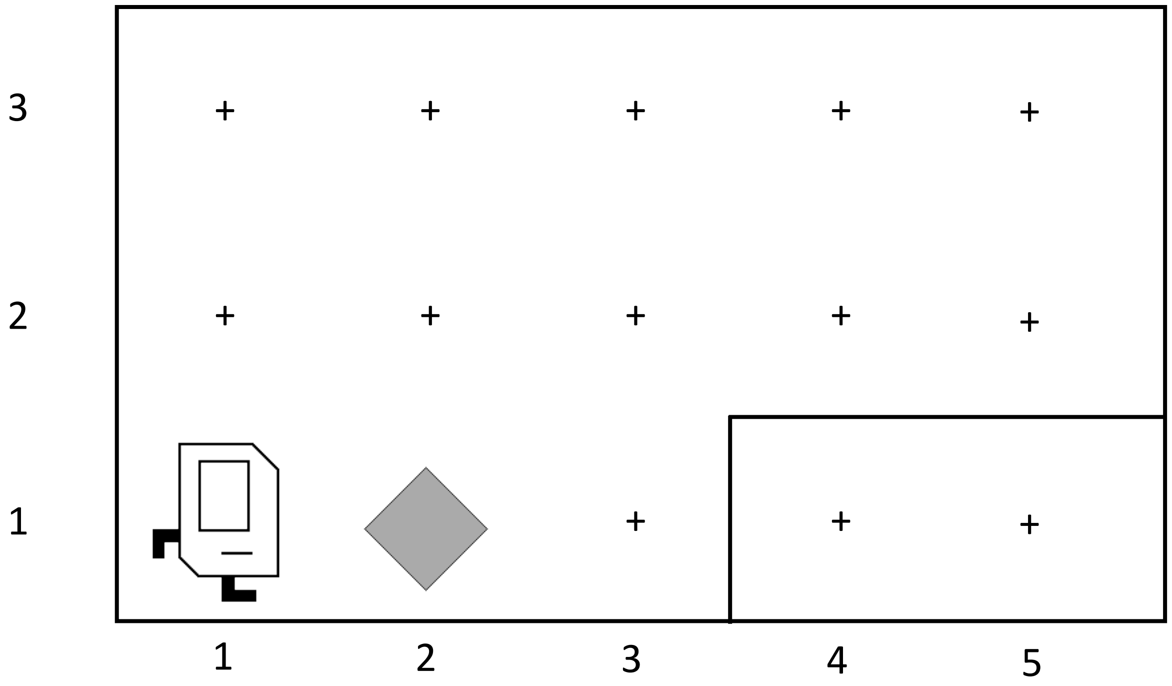
First Challenge



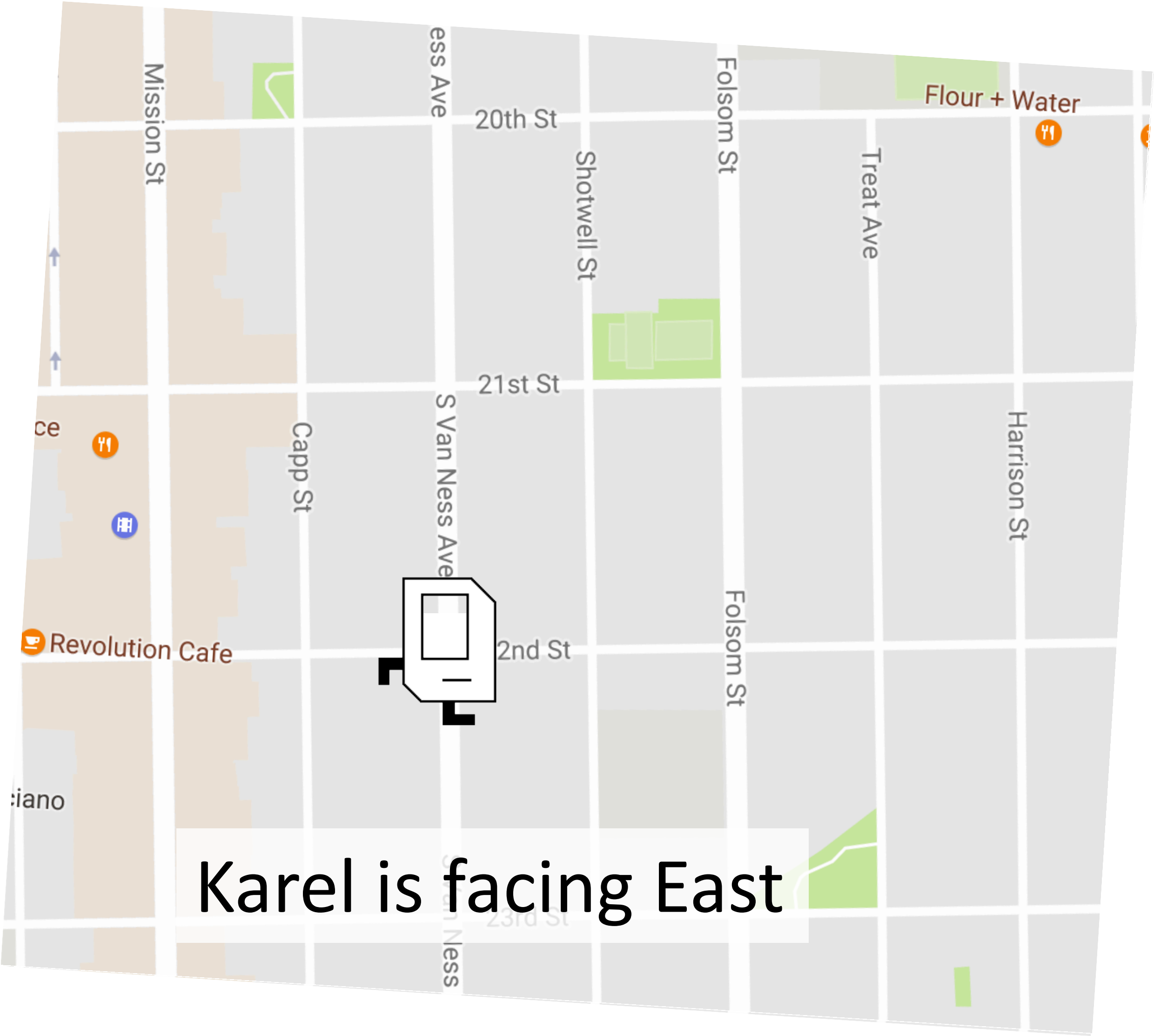
First Challenge



Bird’s Eye View

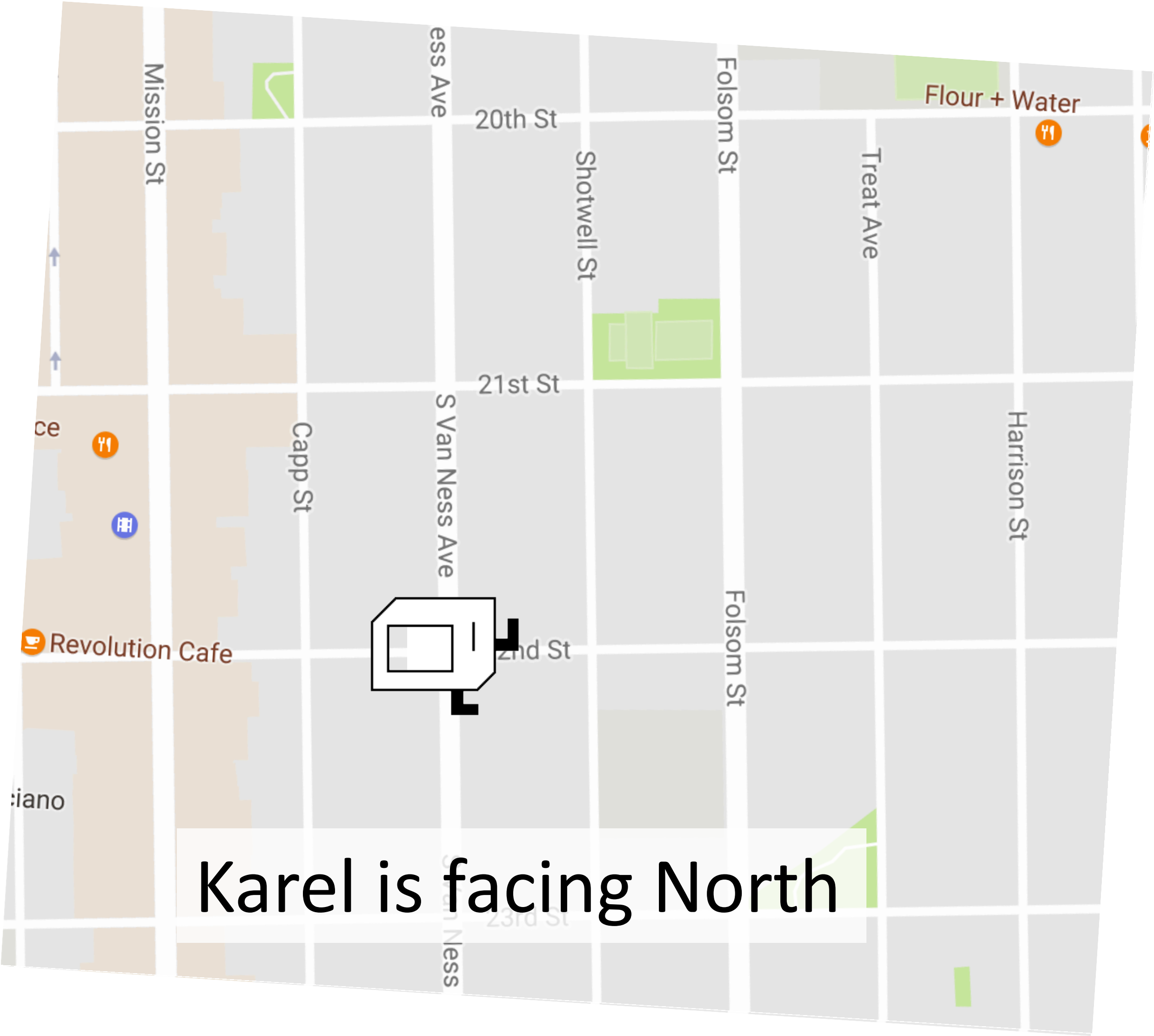


Bird's Eye View



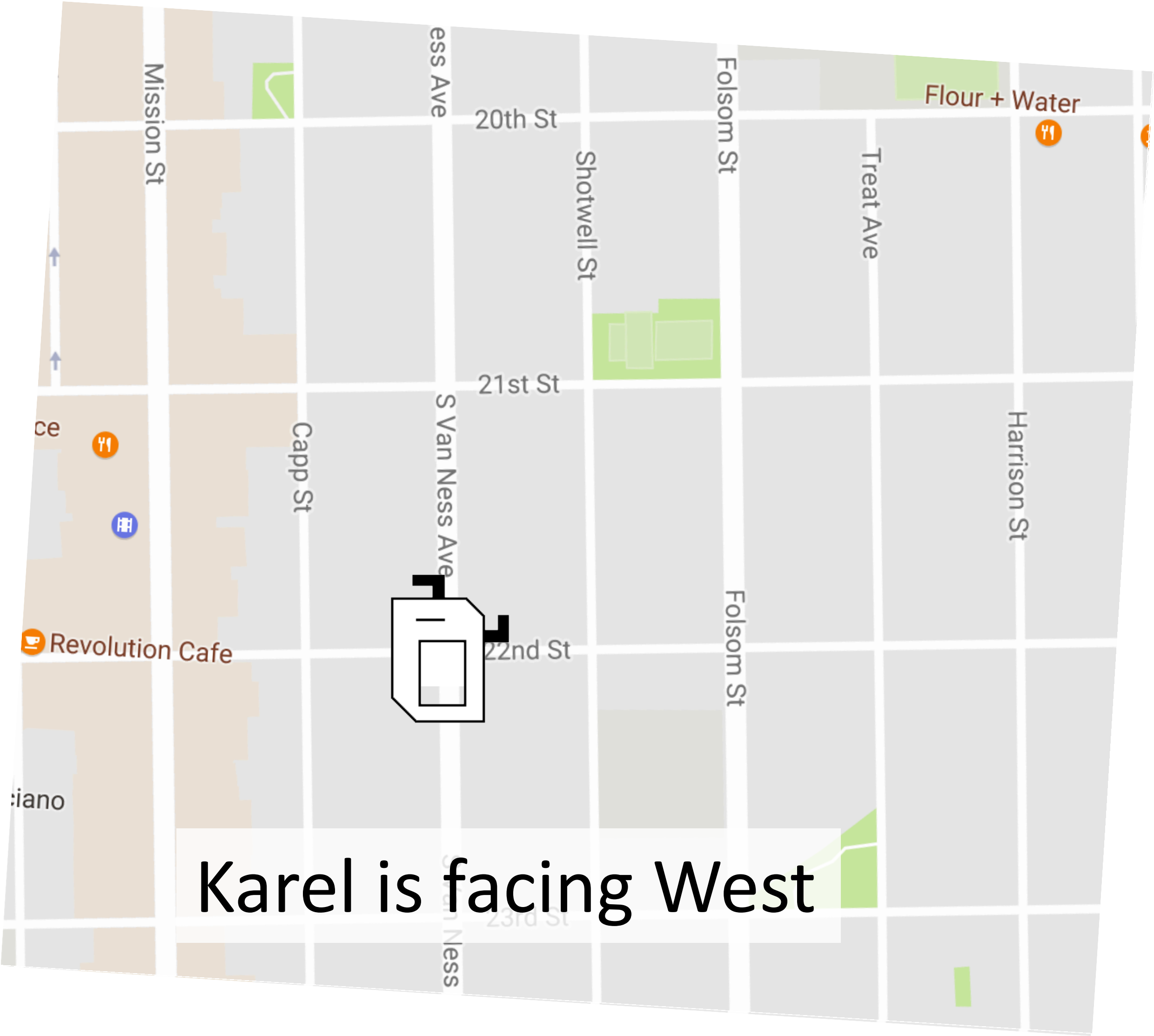
Karel is facing East

Turn Left



Karel is facing North

Turn Left



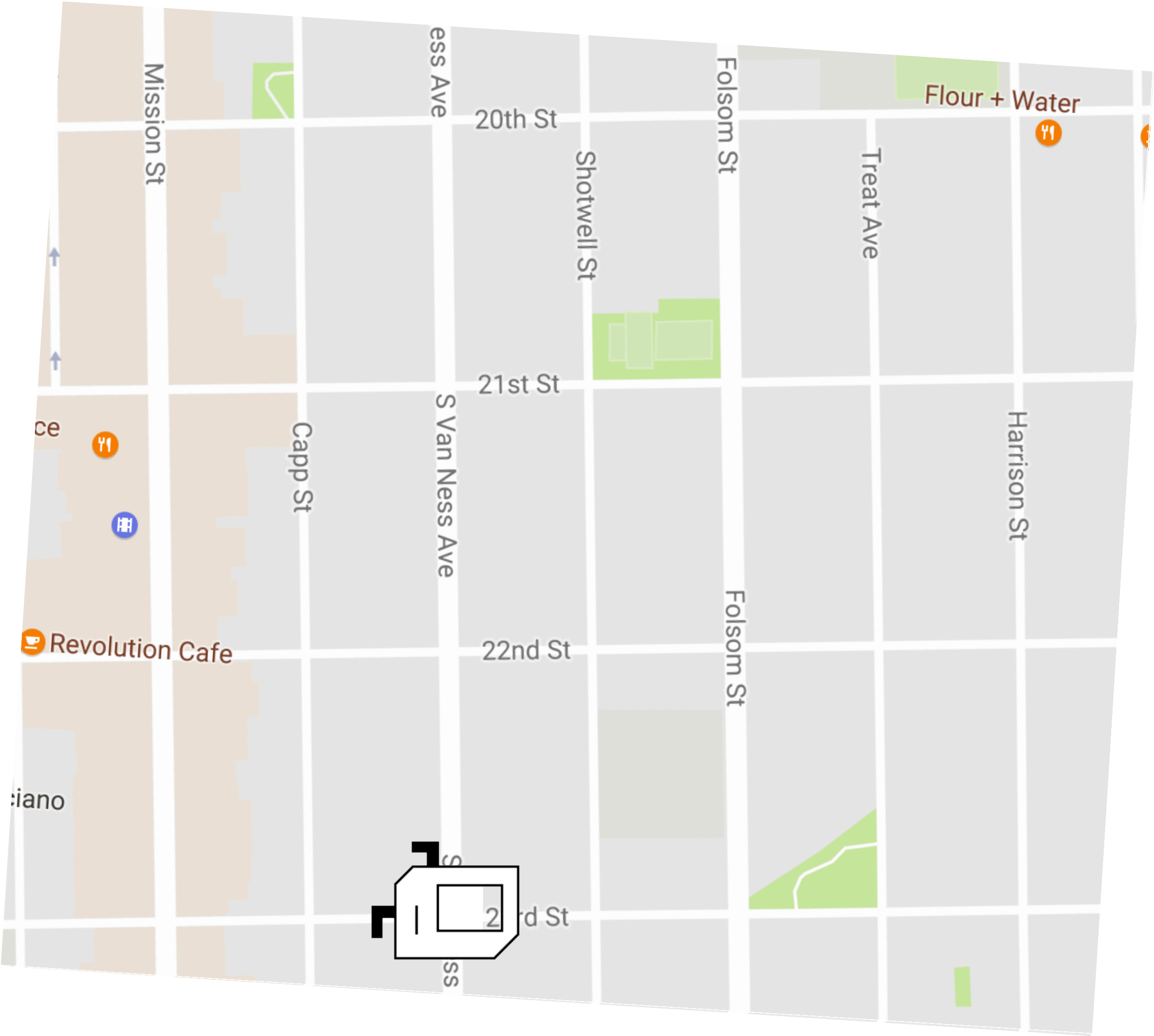
Karel is facing West

Turn Left

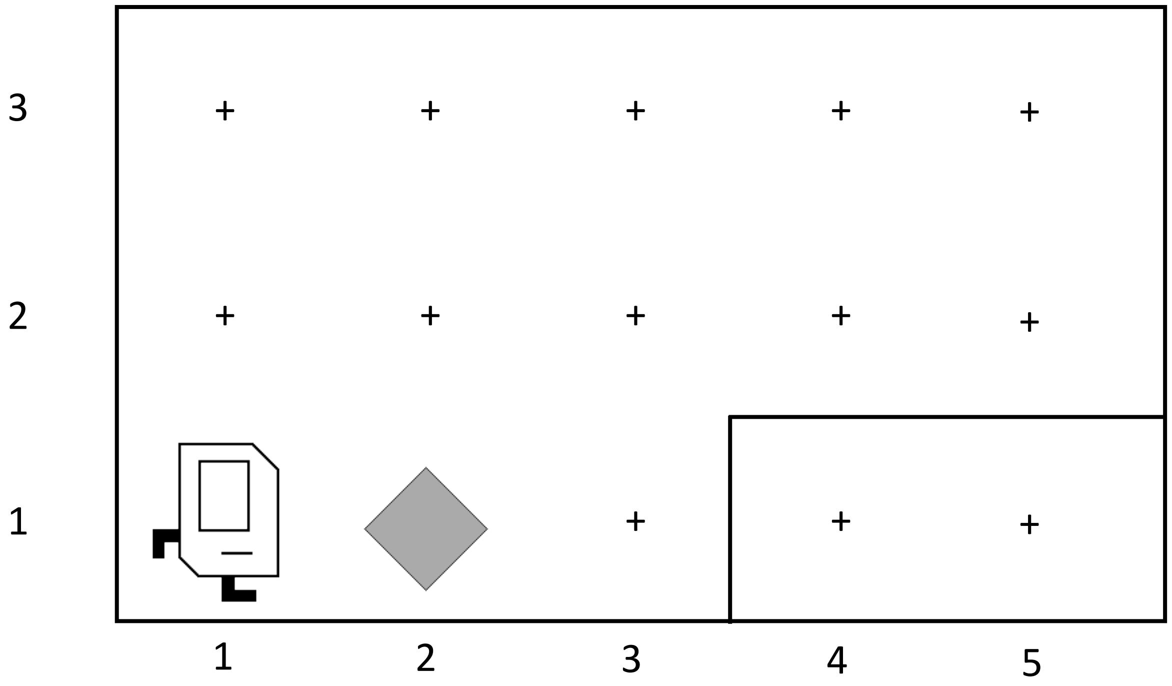
Mission St
ess Ave
20th St
Folsom St
Flour + Water
Shotwell St
Treat Ave
21st St
ce
Capp St
S Van Ness Ave
Harrison St
Revolution Cafe
2 nd St
Folsom St
:iano
23rd St
Ness

Karel is facing South

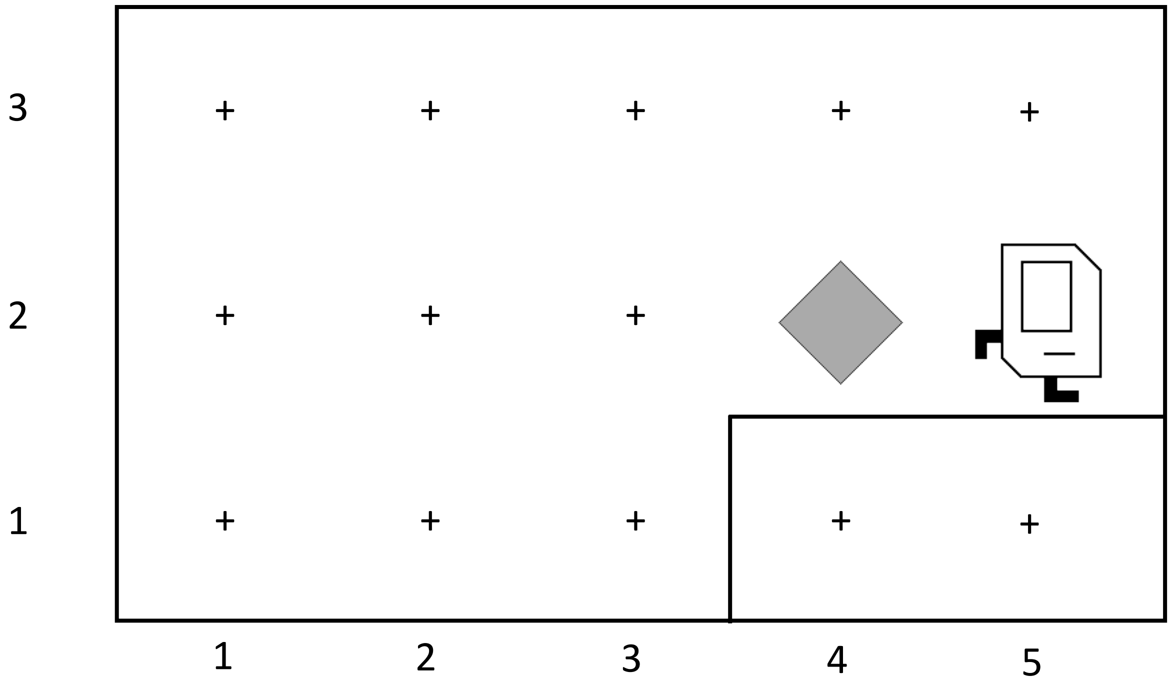
Move


Mission St
ess Ave
20th St
Folsom St
Flour + Water
Shotwell St
Treat Ave
21st St
S Van Ness Ave
Capp St
Harrison St
ce
Revolution Cafe
22nd St
Folsom St
iano
2 d St
ss

First Challenge



First Challenge



Learn By Doing

PyCharm

The Python IDE for Professional Developers

Full-fledged Professional or Free Community

Function Definition

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

This adds a new command to Karels vocabulary

Anatomy of a Program

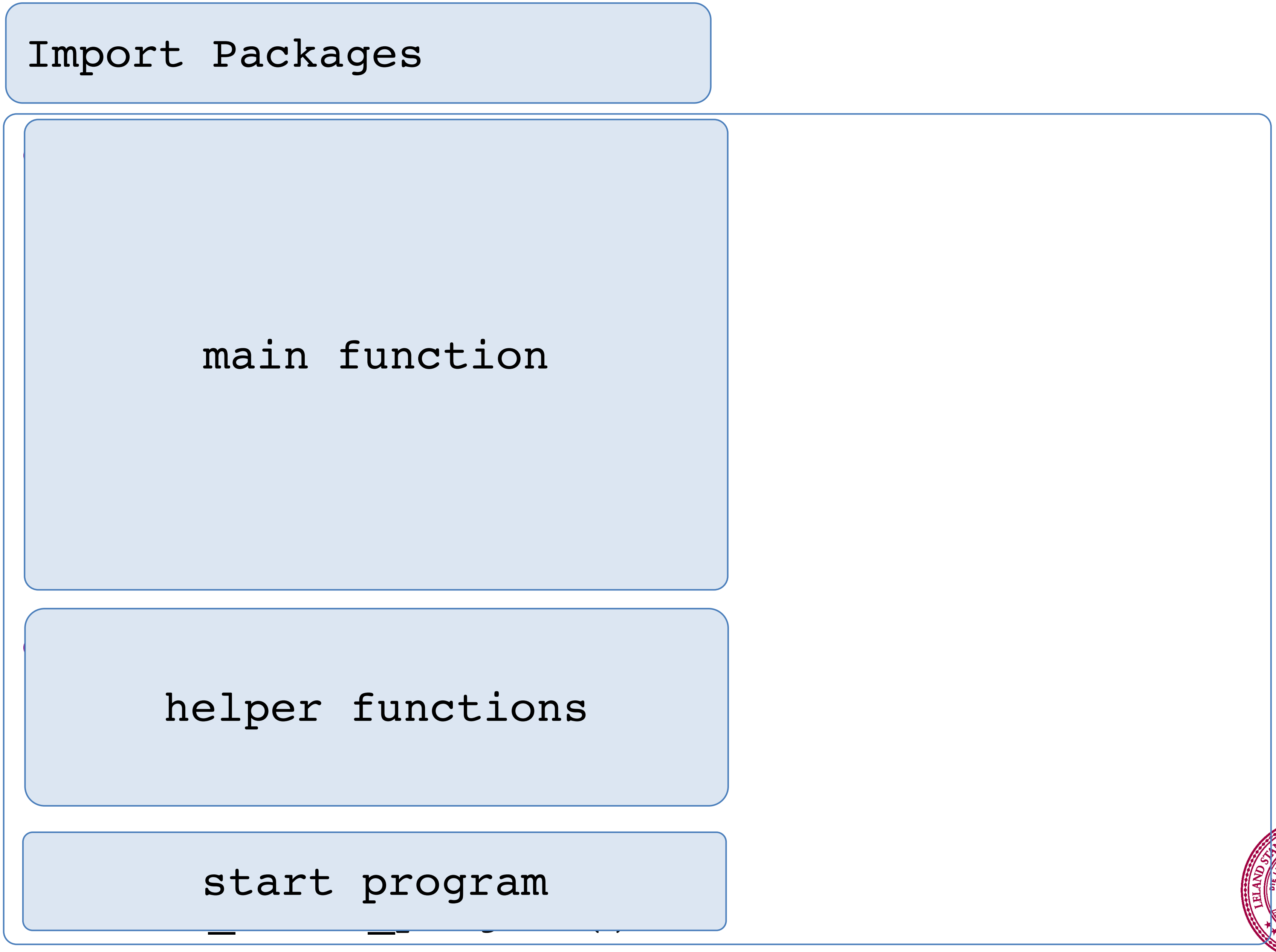
Import Packages

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()

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

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

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

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, run)

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()

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()

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

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

This is called a ***code block***

Why Study CS?

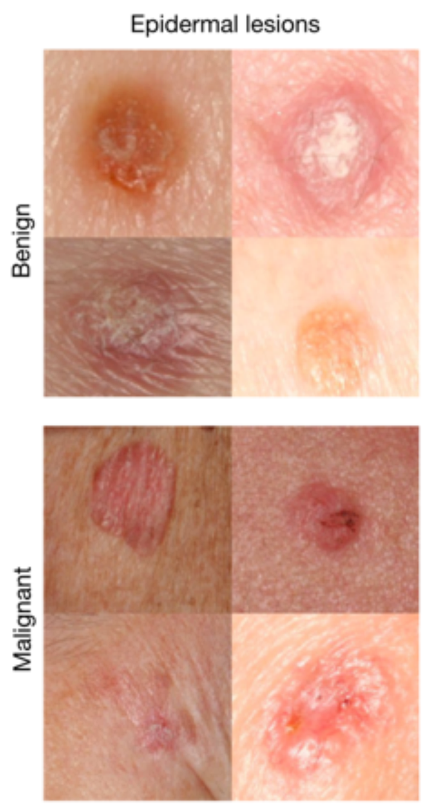
Joy of Building

Interdisciplinary

Closest Thing To Magic

Now is the Time

Now is the Time

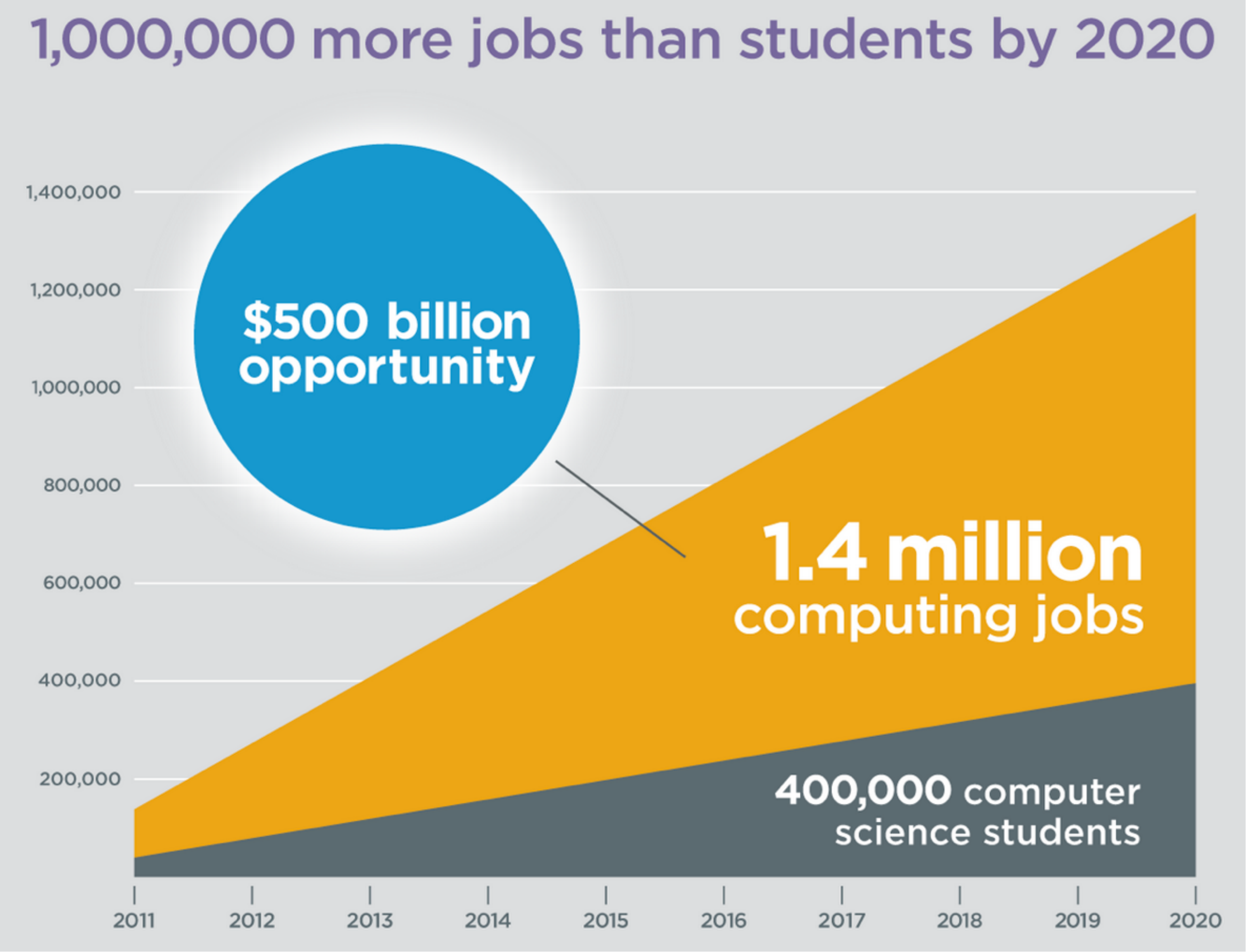


A machine learning algorithm performs **better than** the best dermatologists.

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



Code.org

Everyone is Welcome

The End

The End?