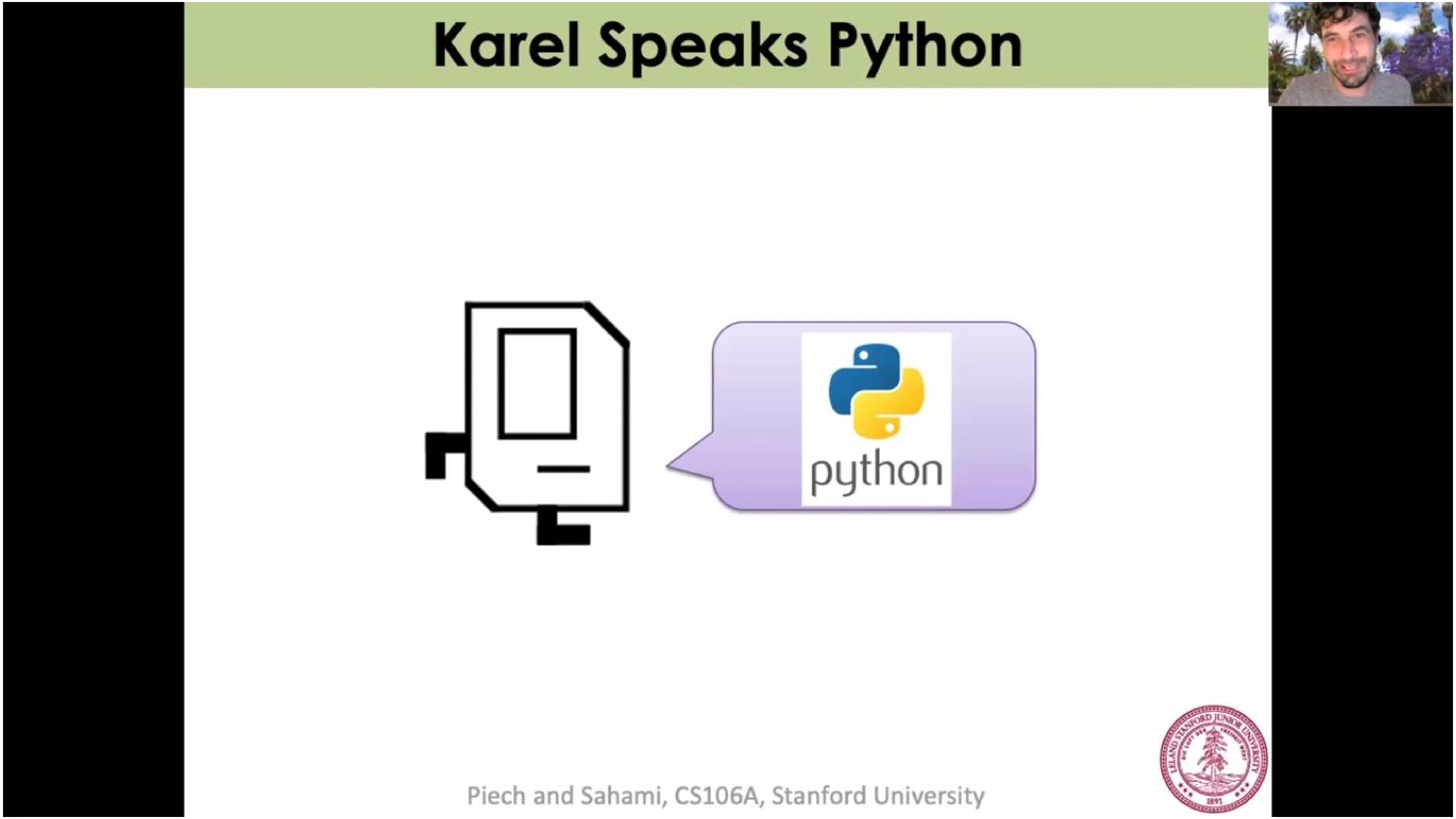
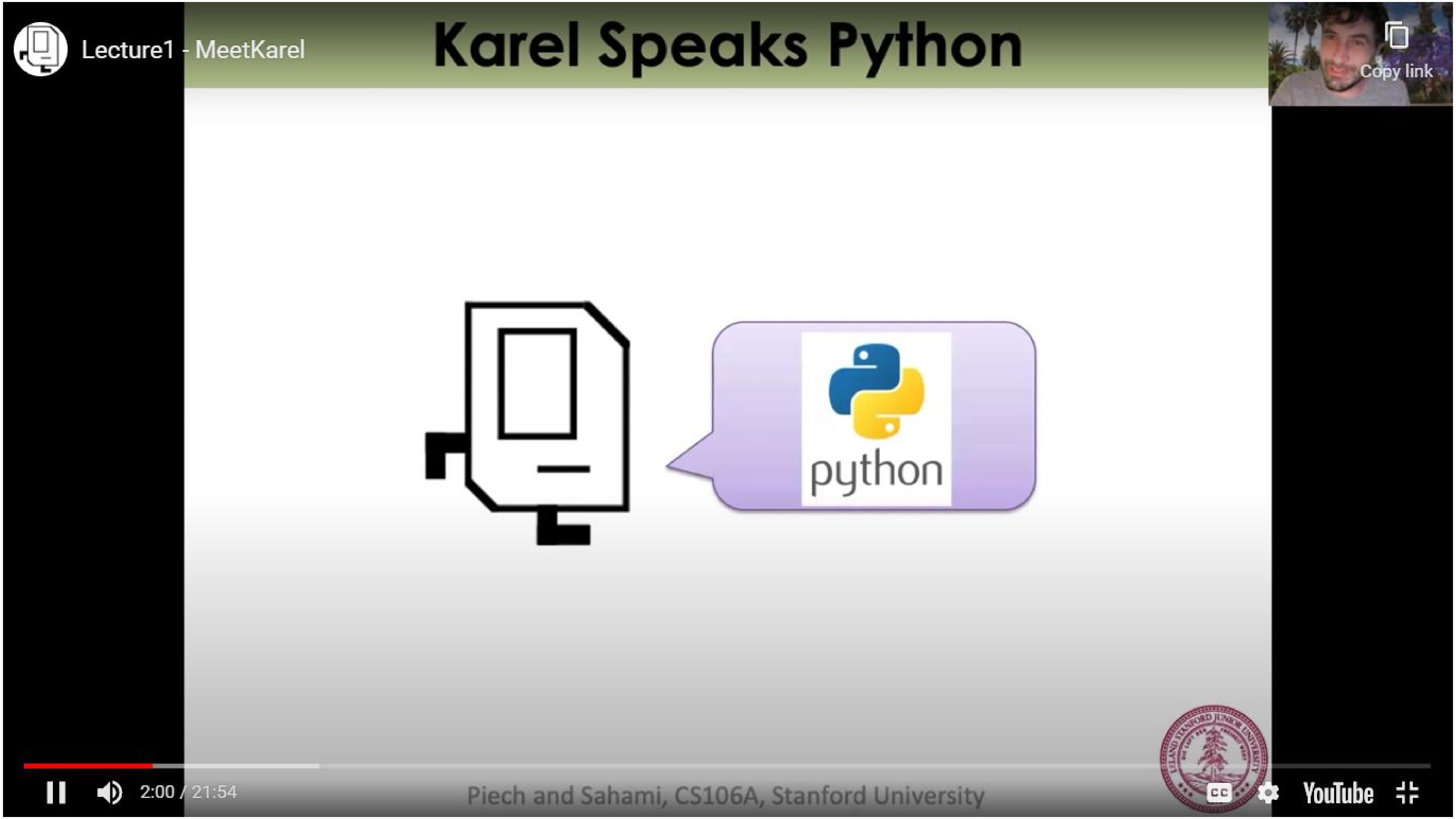
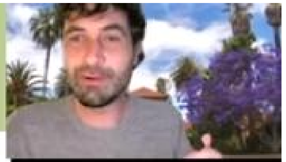


Karel Speaks Python



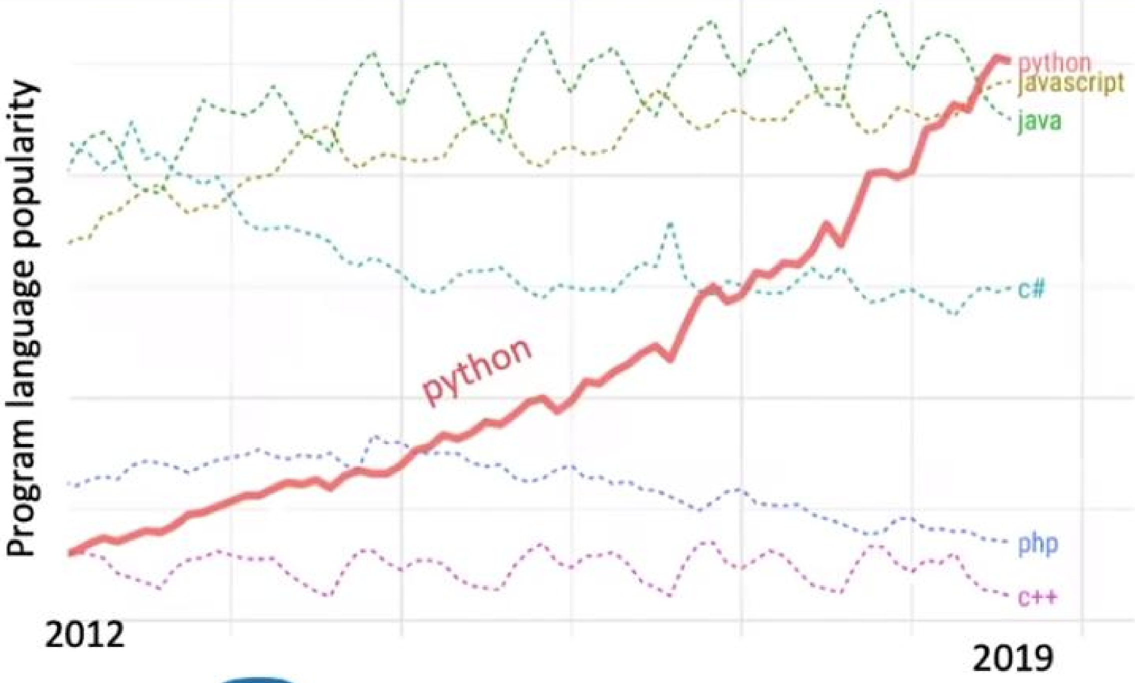
Karel Speaks Python

Piech and Sahami, CS106A, Stanford University



Why Python?

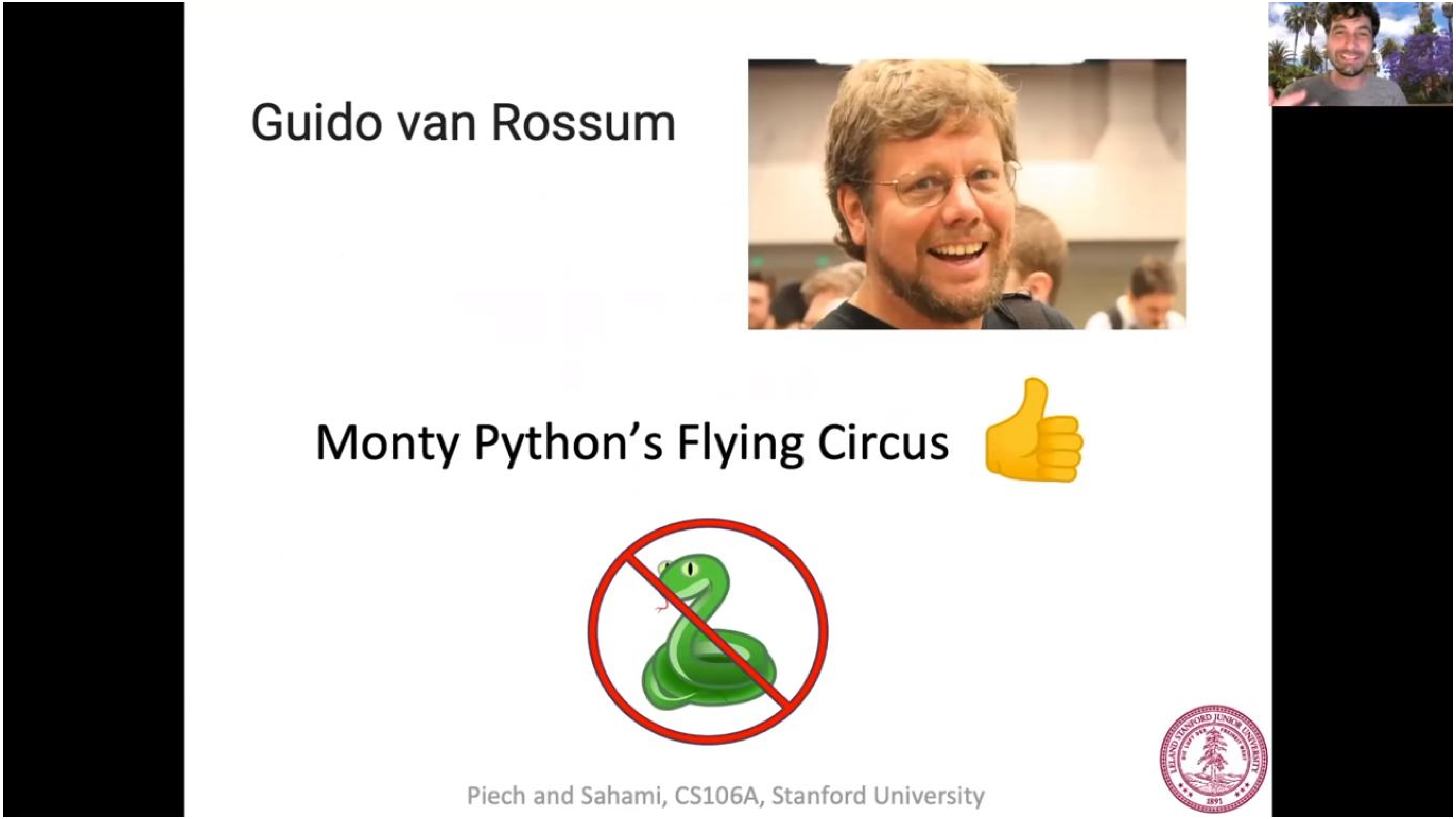
1



2

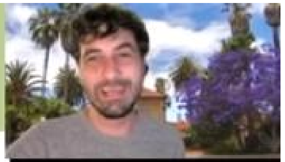
python

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

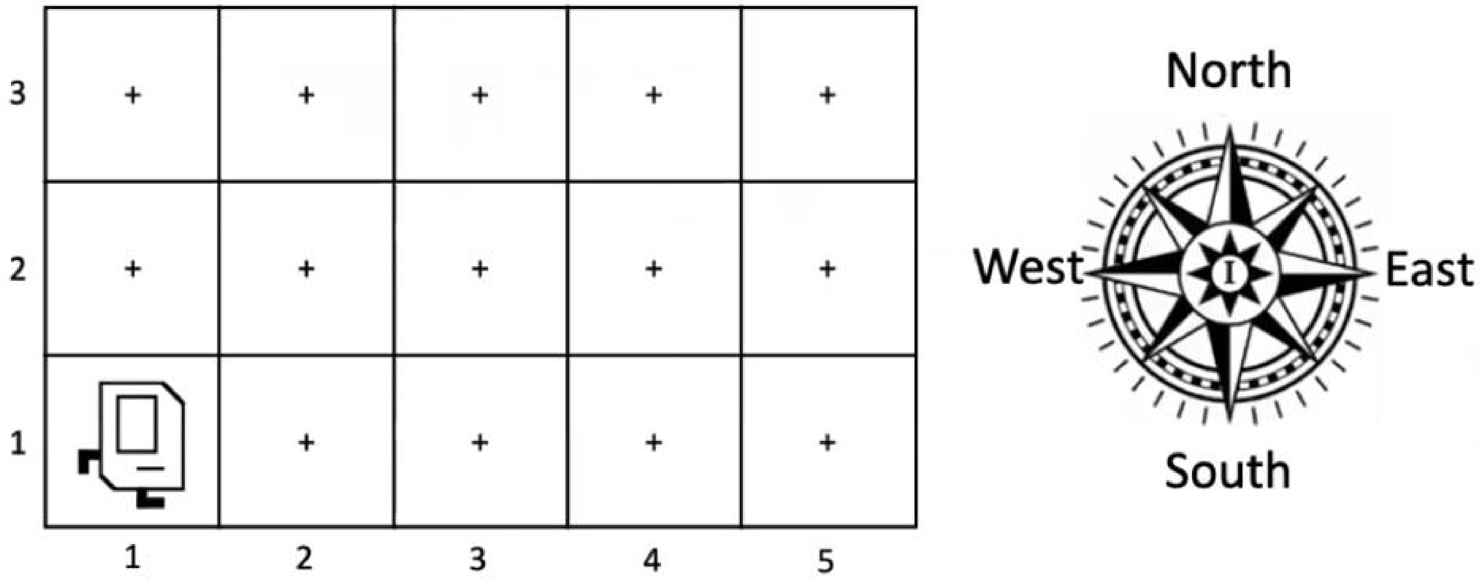


Guido van Rossum

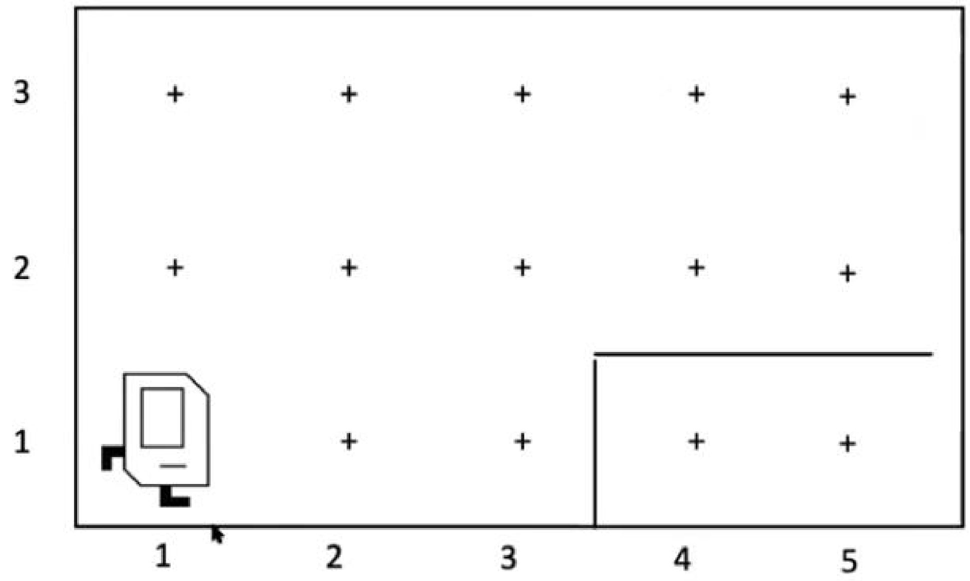
Monty Python's Flying Circus 👍

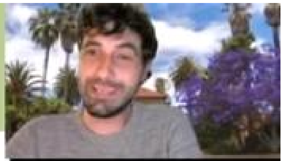


Karel's World

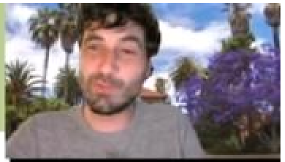


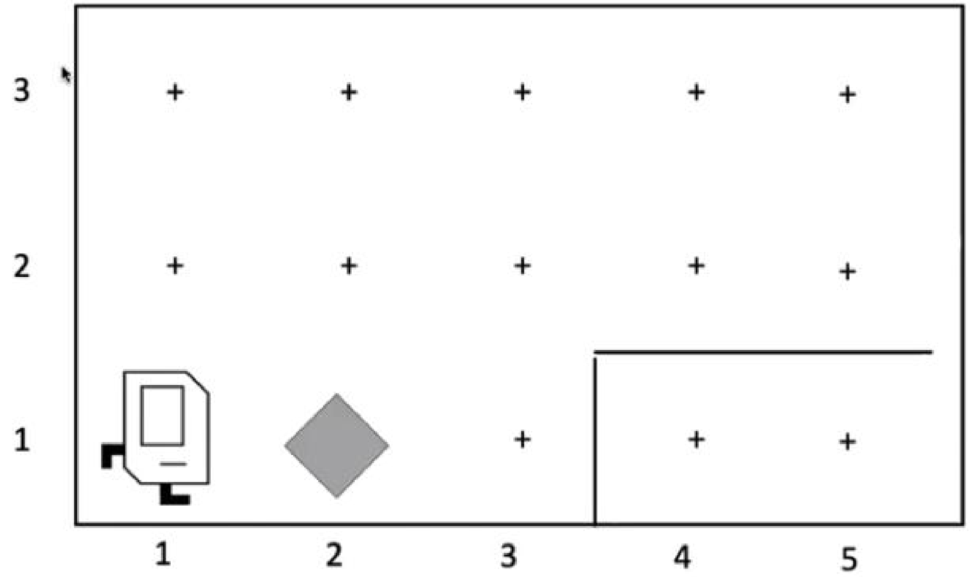
Walls

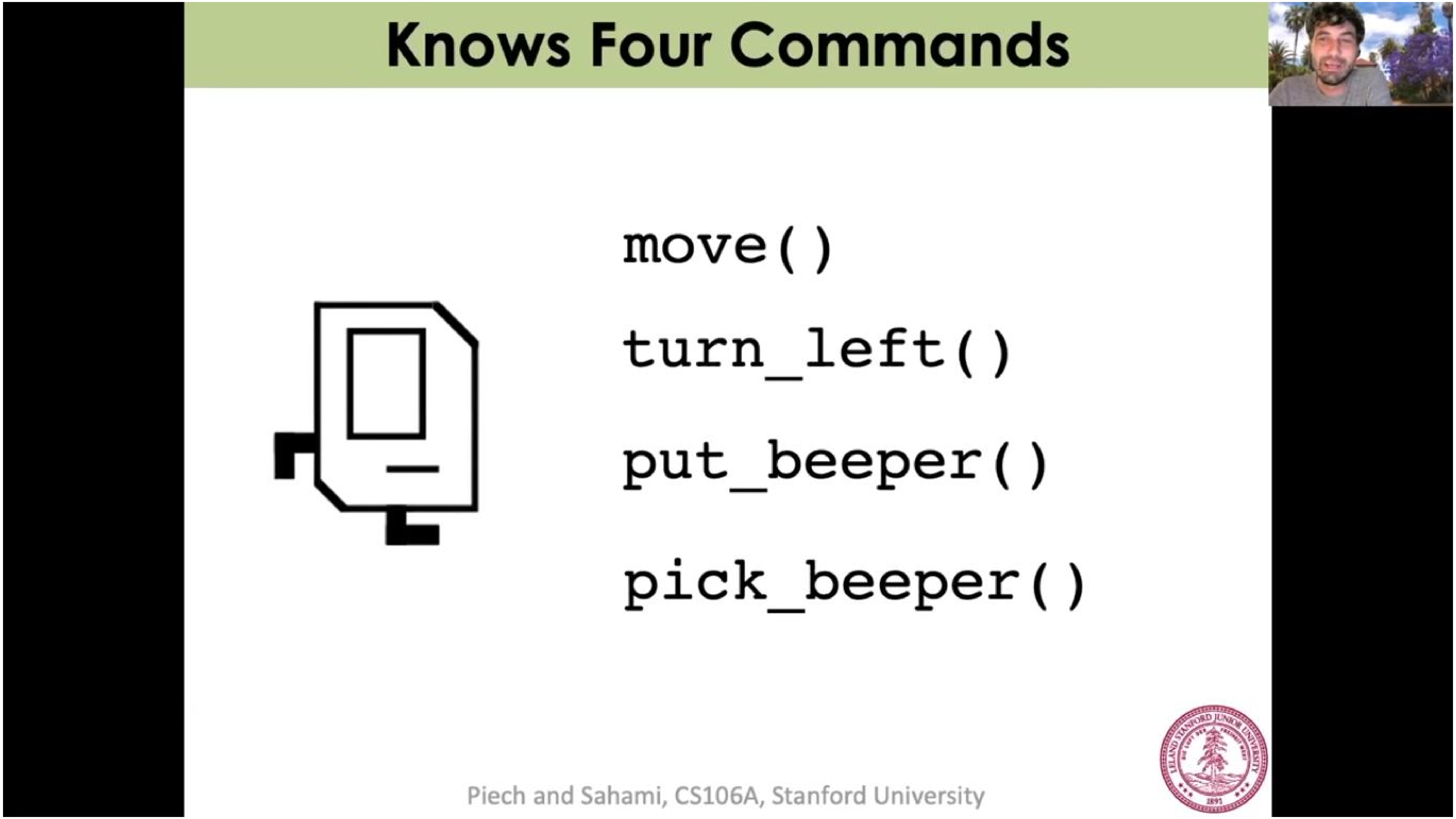




Beepers

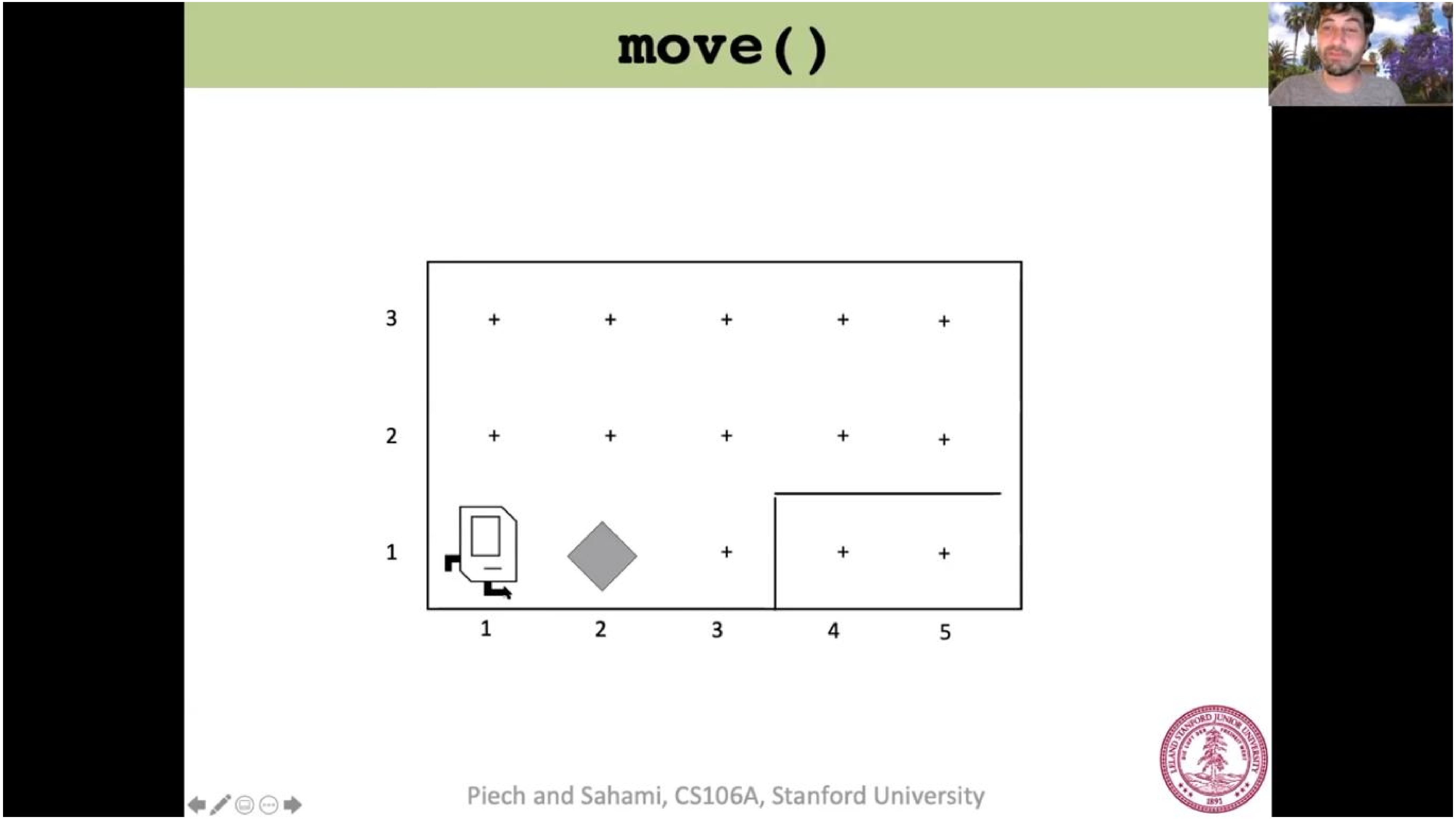






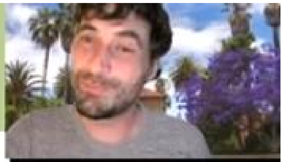
Knows Four Commands

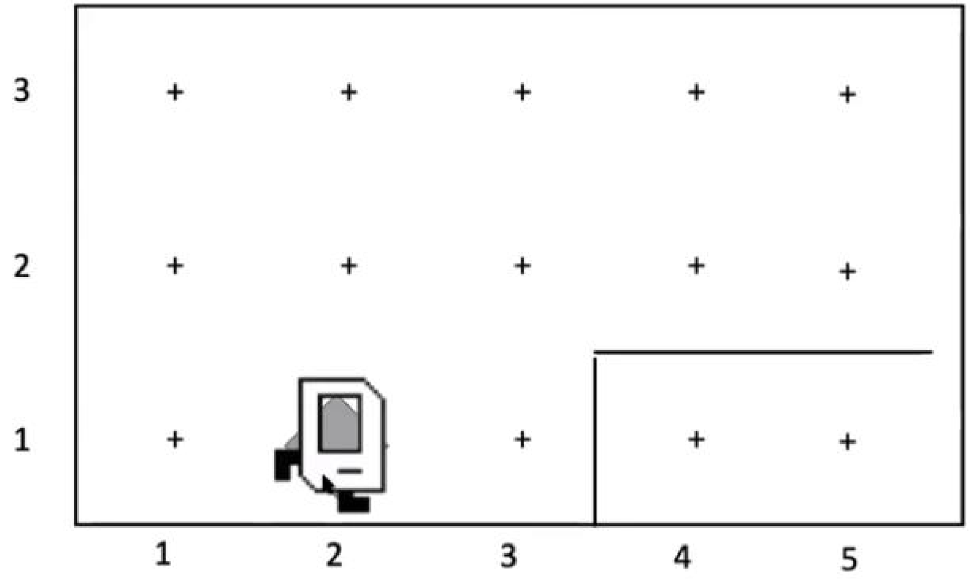
```
move()

turn_left()

put_beeper()

pick_beeper()
```



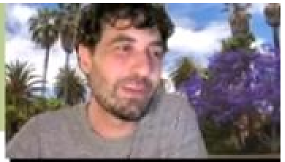
move()

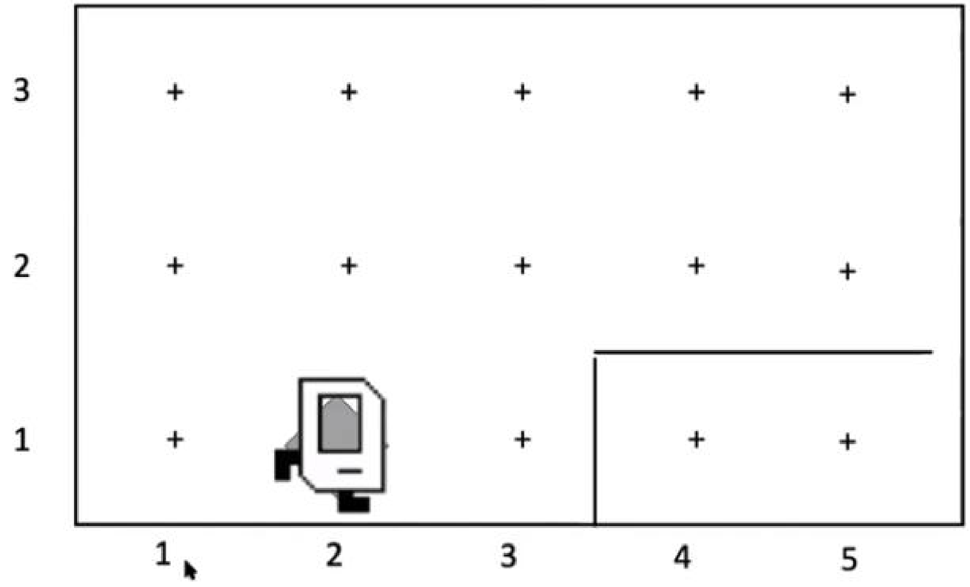
move()

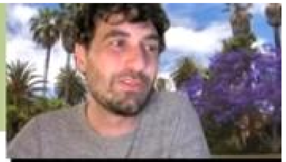




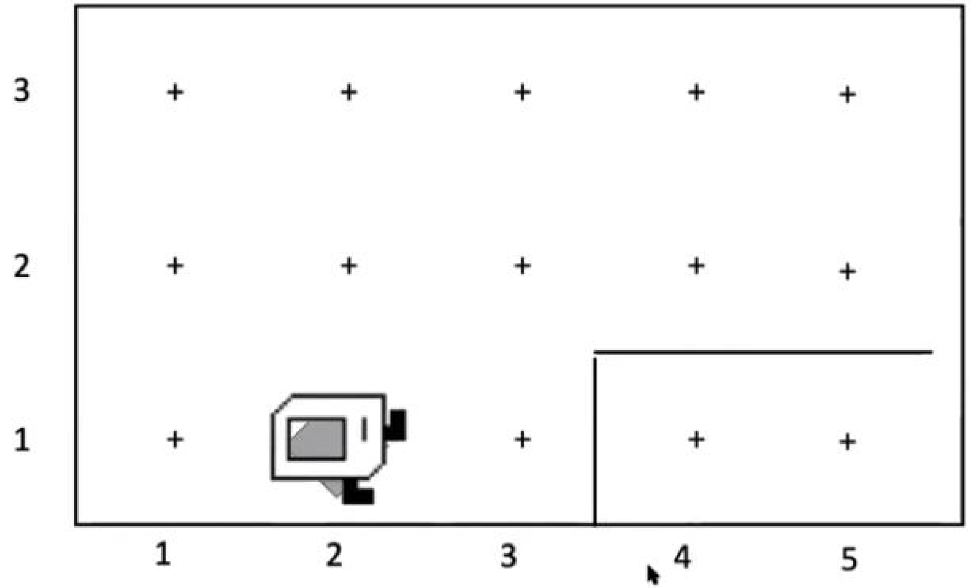
turn_left()

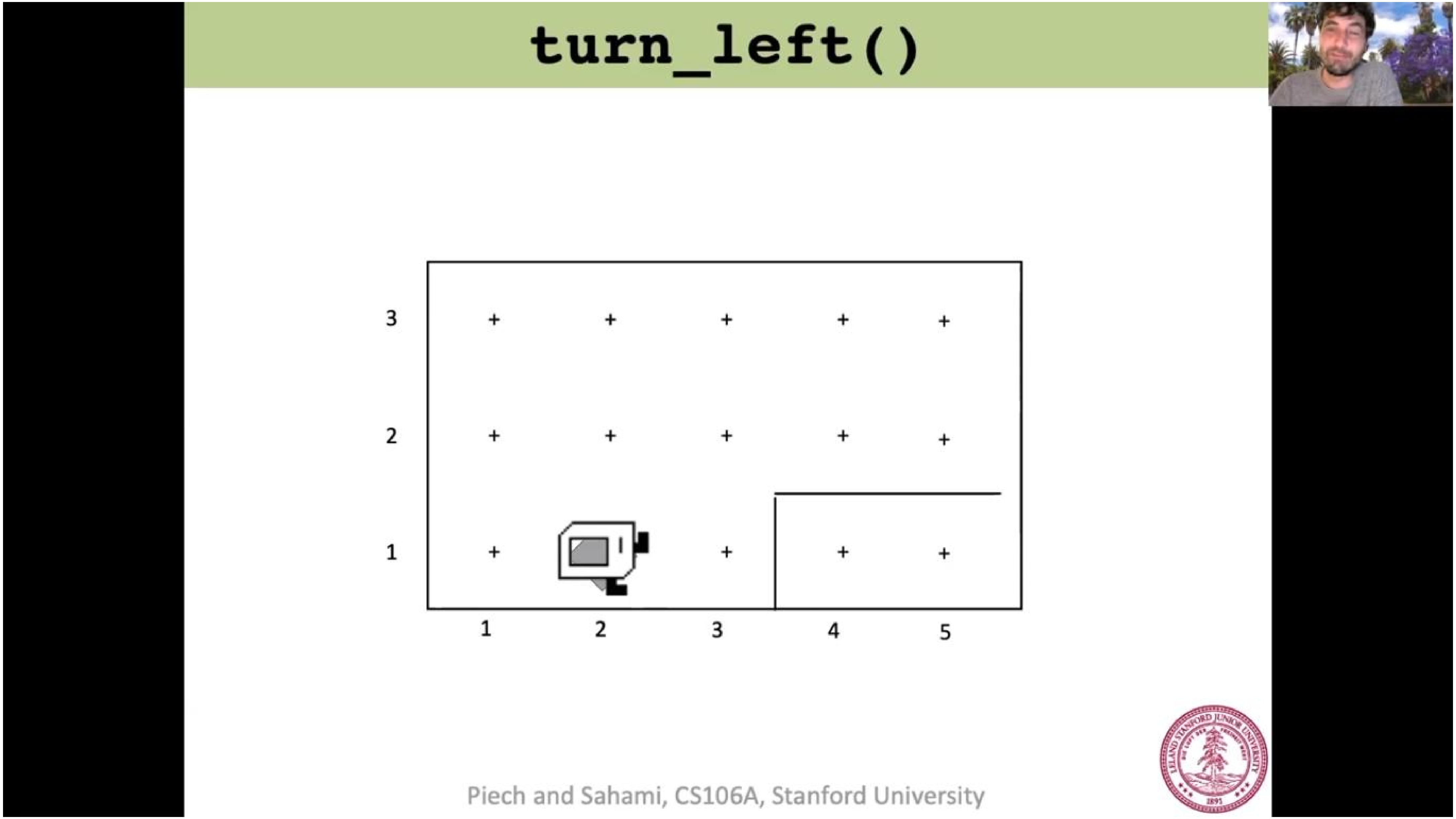




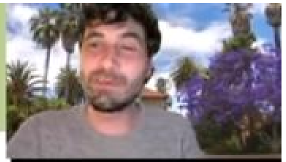


turn_left()

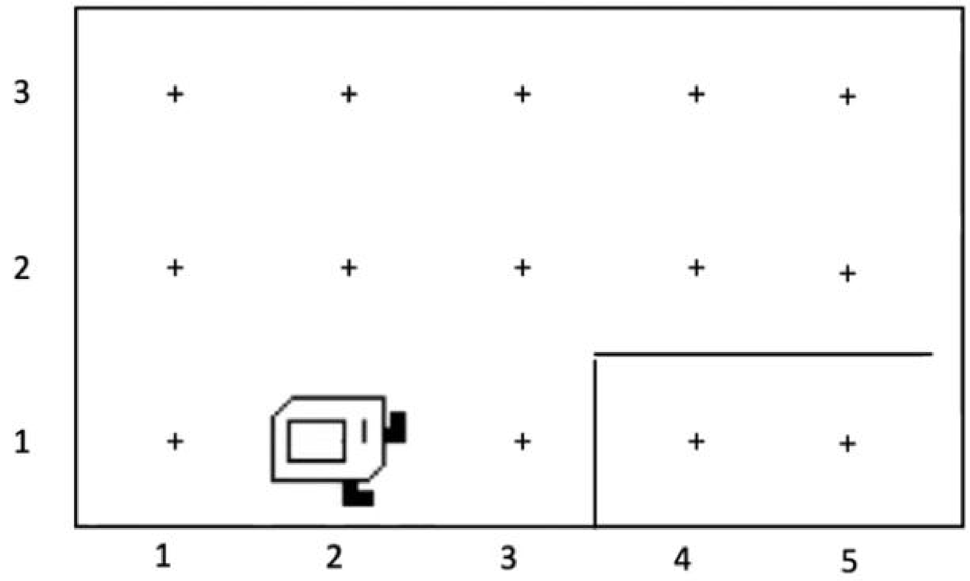




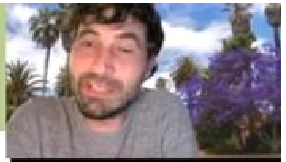
turn_left()

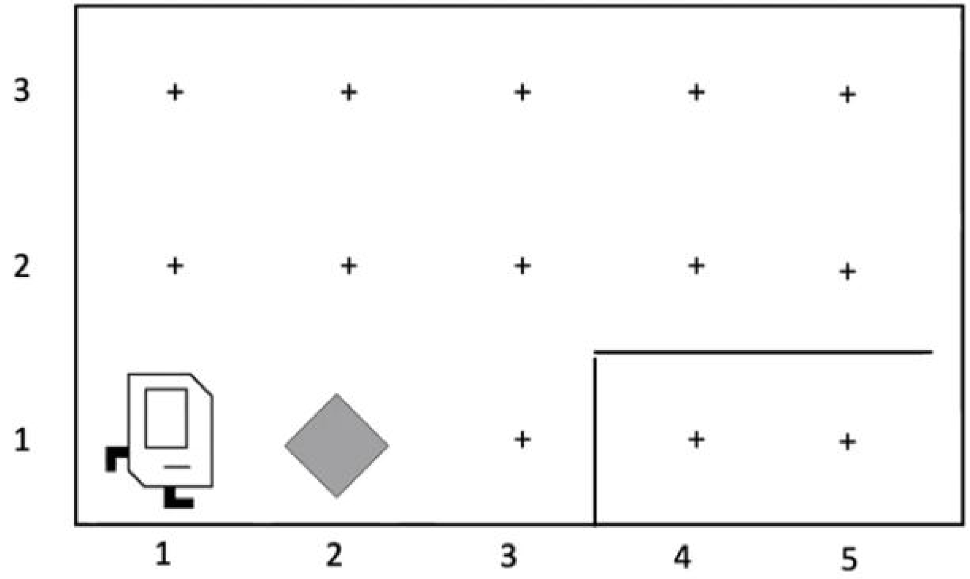


turn_left()

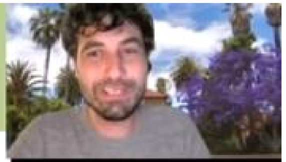


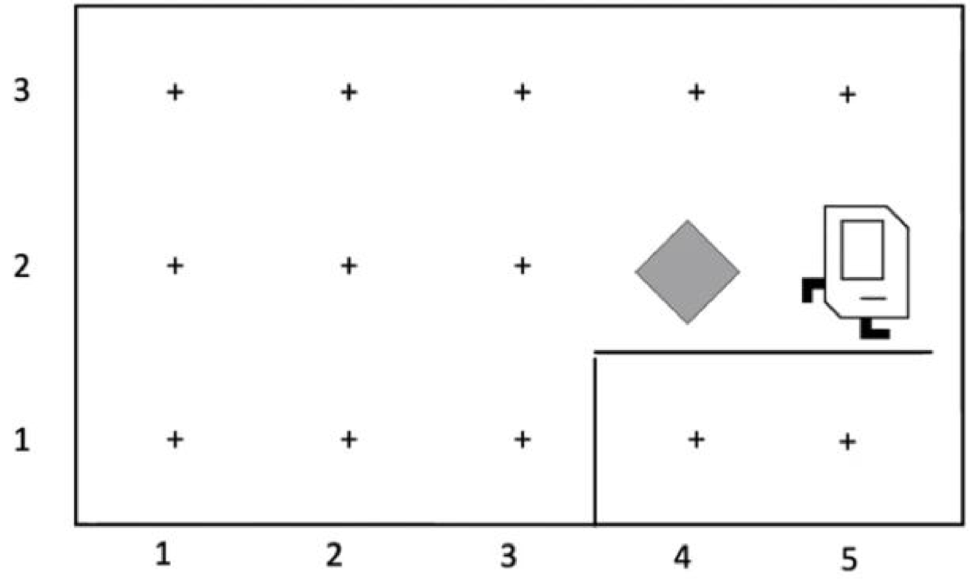
First Challenge



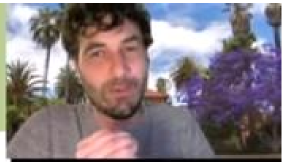


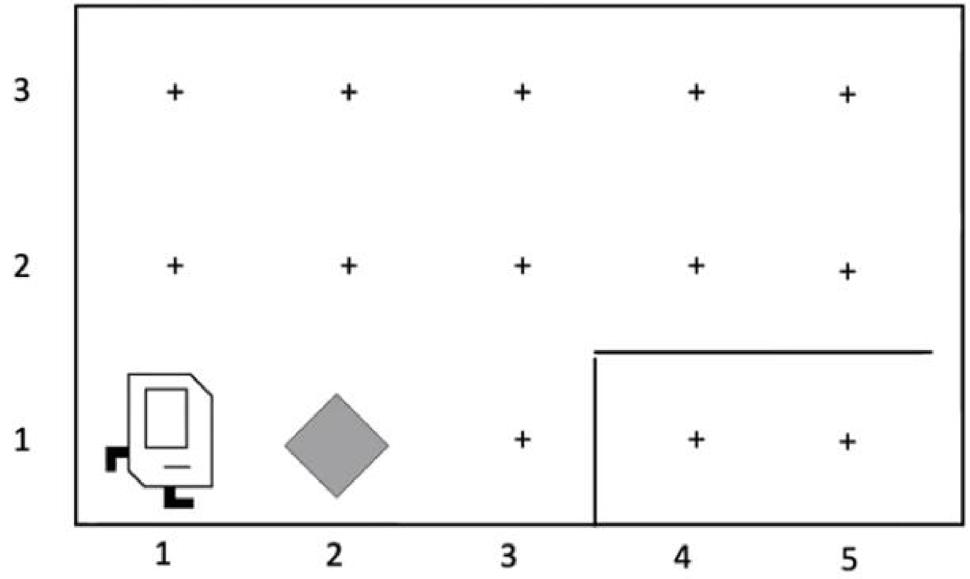
First Challenge

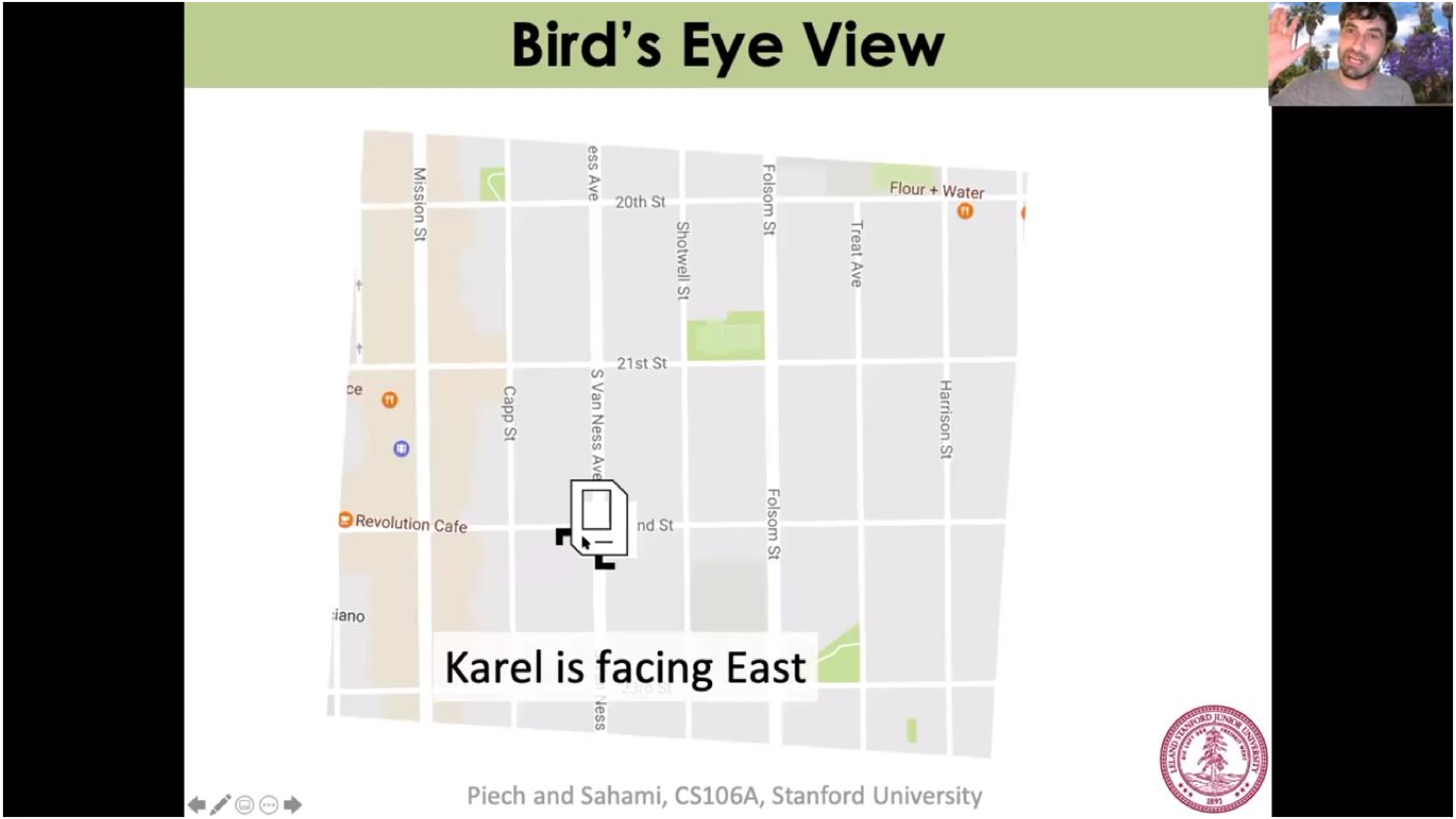




Bird’s Eye View

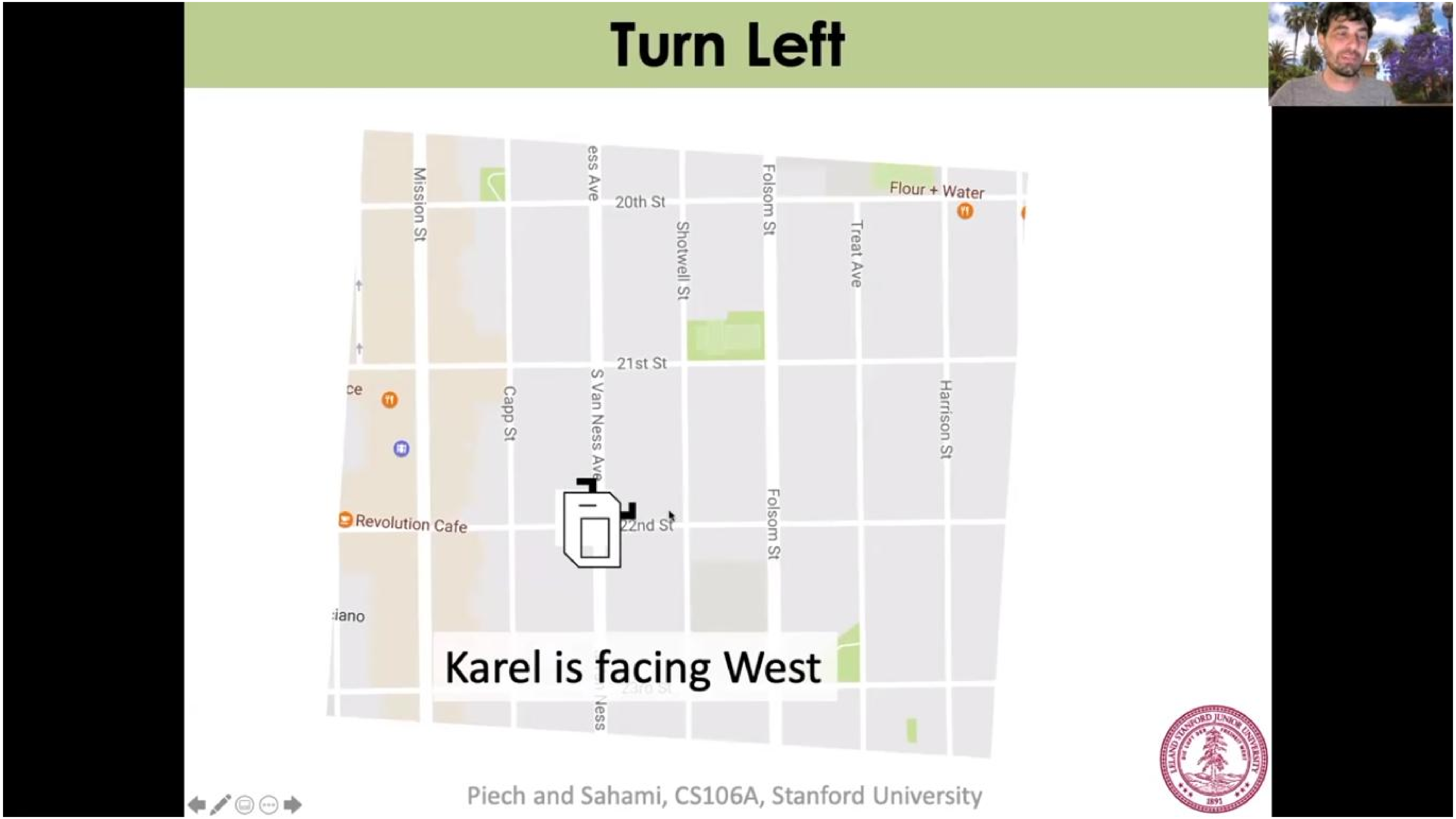






Bird's Eye View

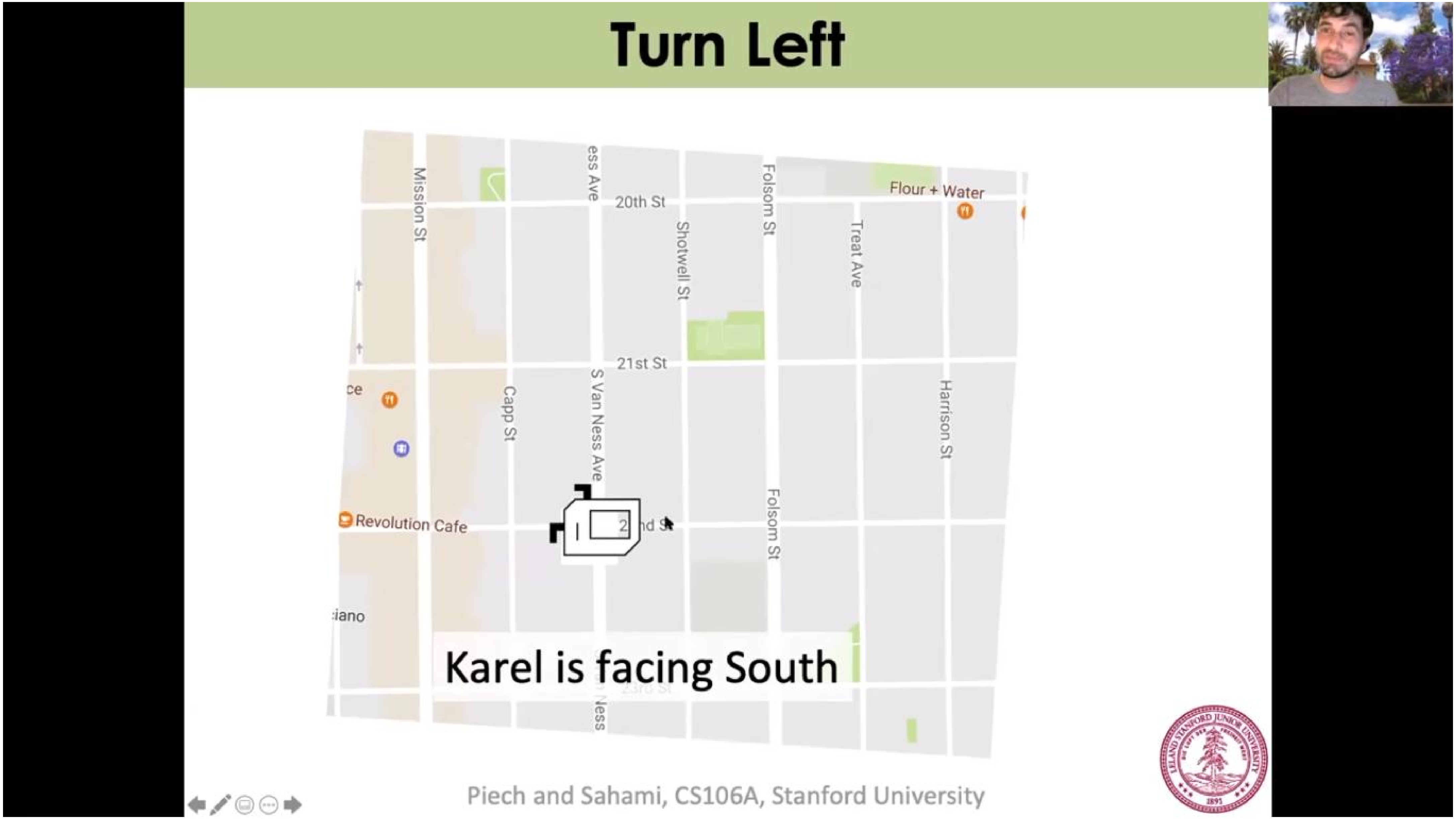
Karel is facing East



Turn Left

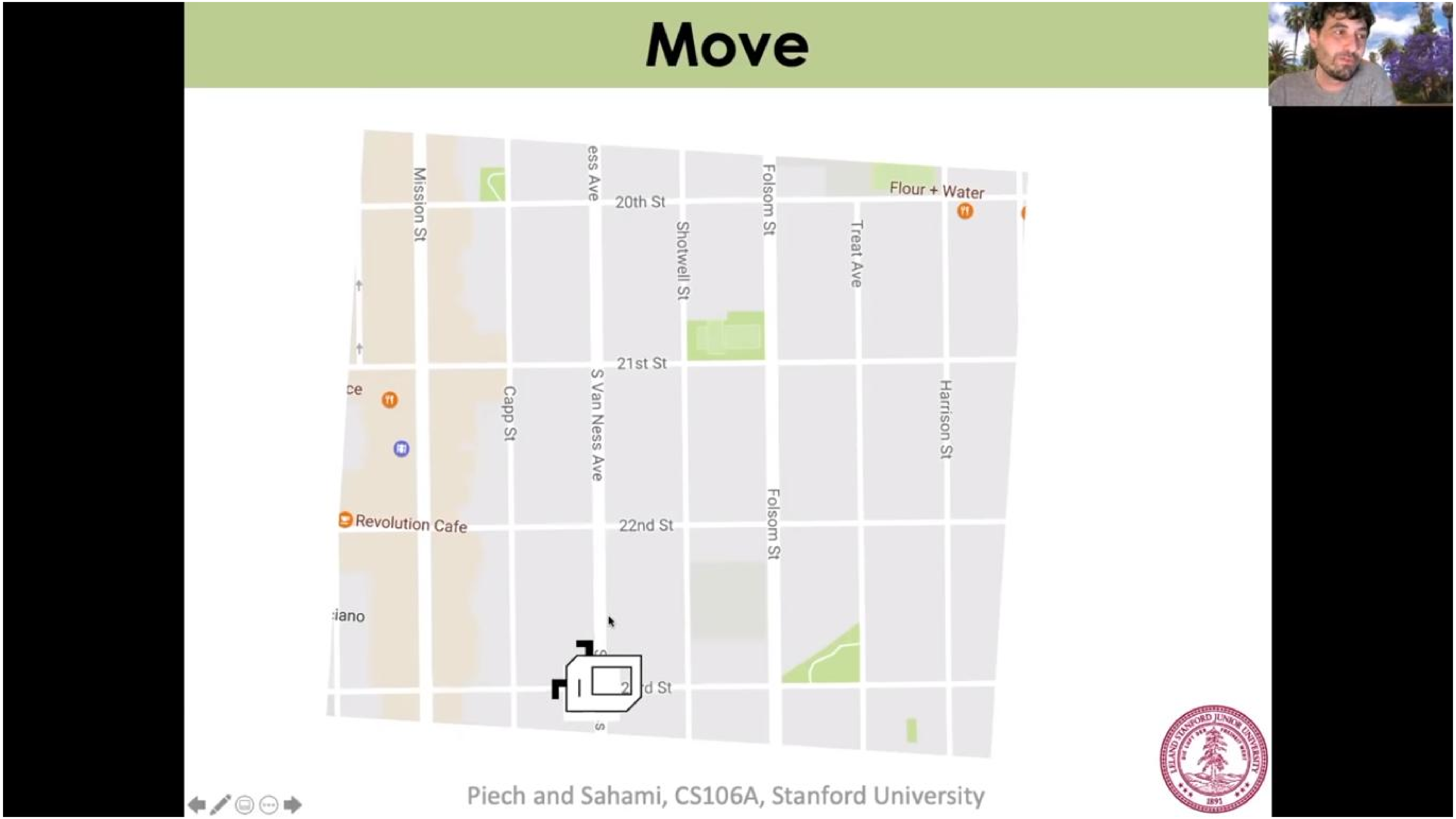
Karel is facing West

Piech and Sahami, CS106A, Stanford University



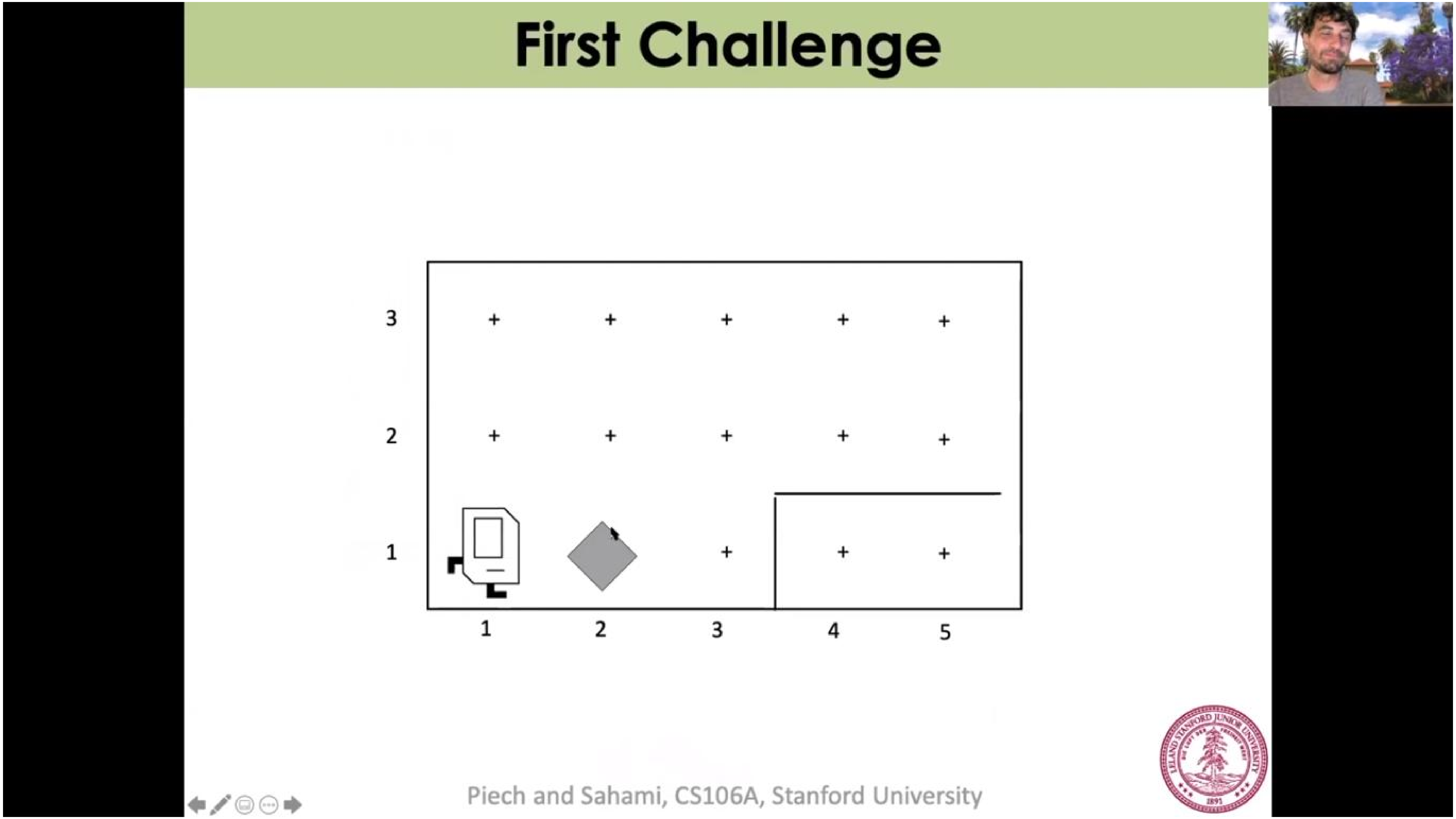
Turn Left

Karel is facing South

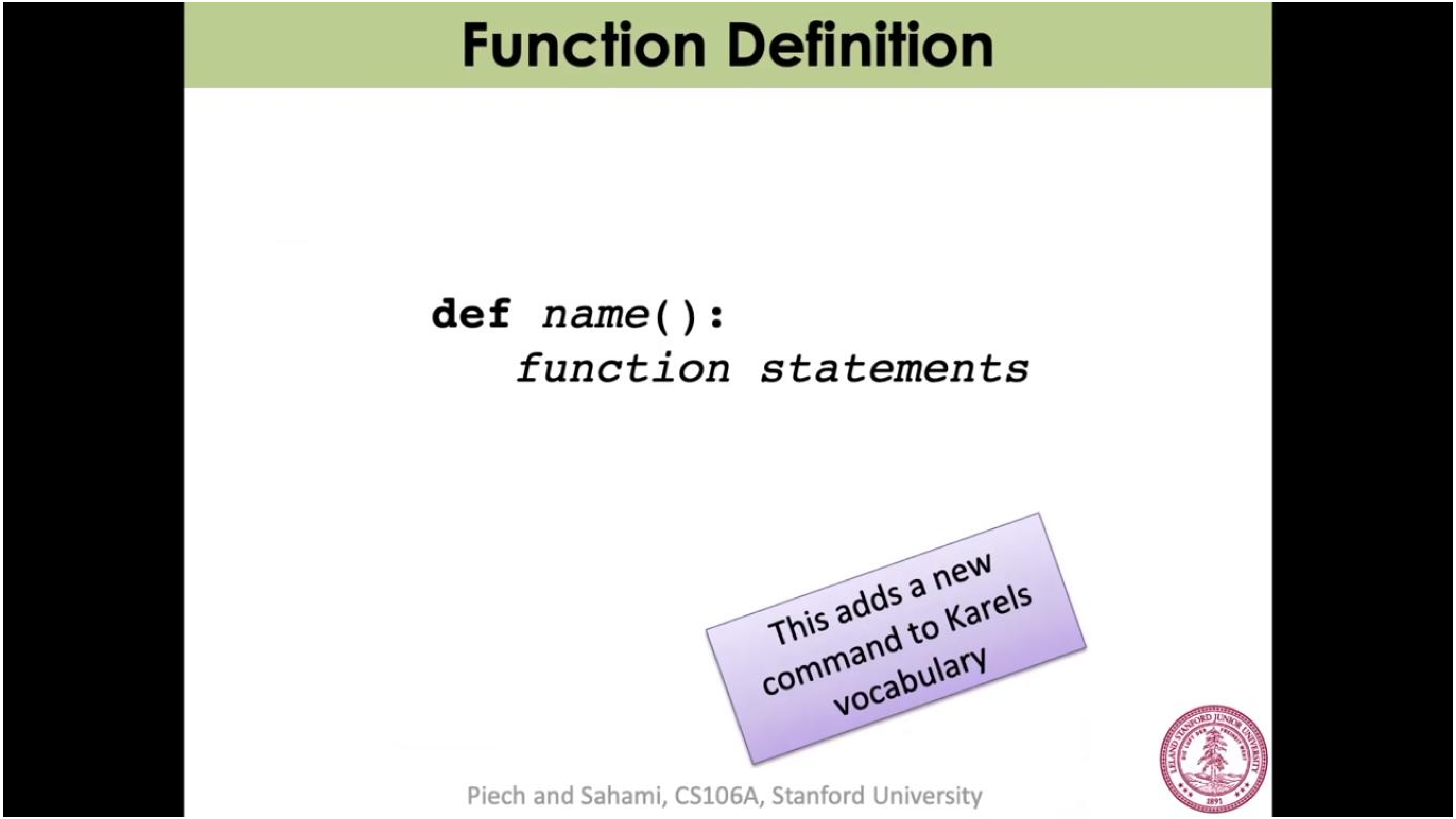


Move

Piech and Sahami, CS106A, Stanford University



First Challenge

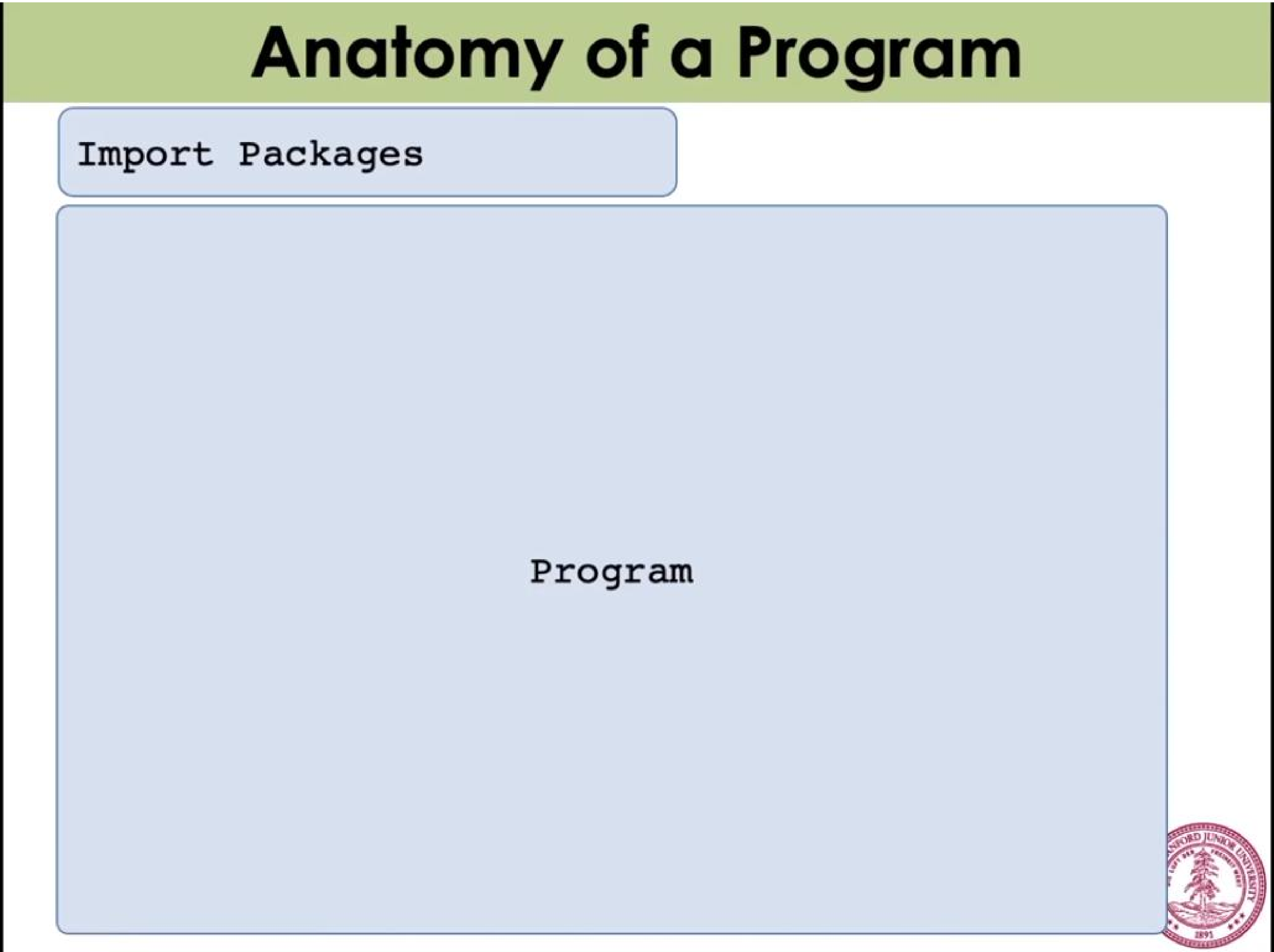


Function Definition

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

This adds a new command to Karels vocabulary

Piech and Sahami, CS106A, Stanford University

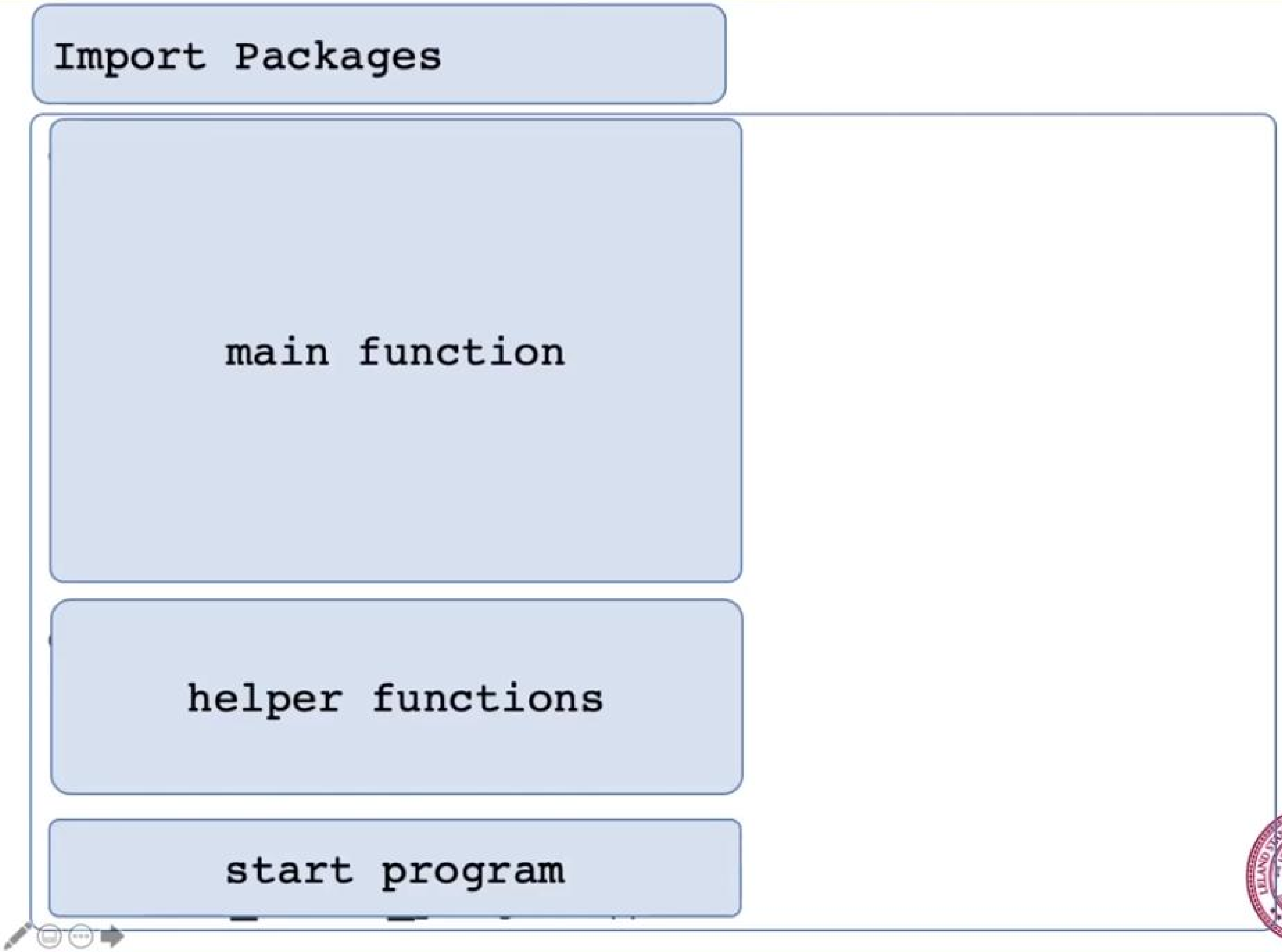


Anatomy of a Program

Import Packages

Program

Anatomy of a Program

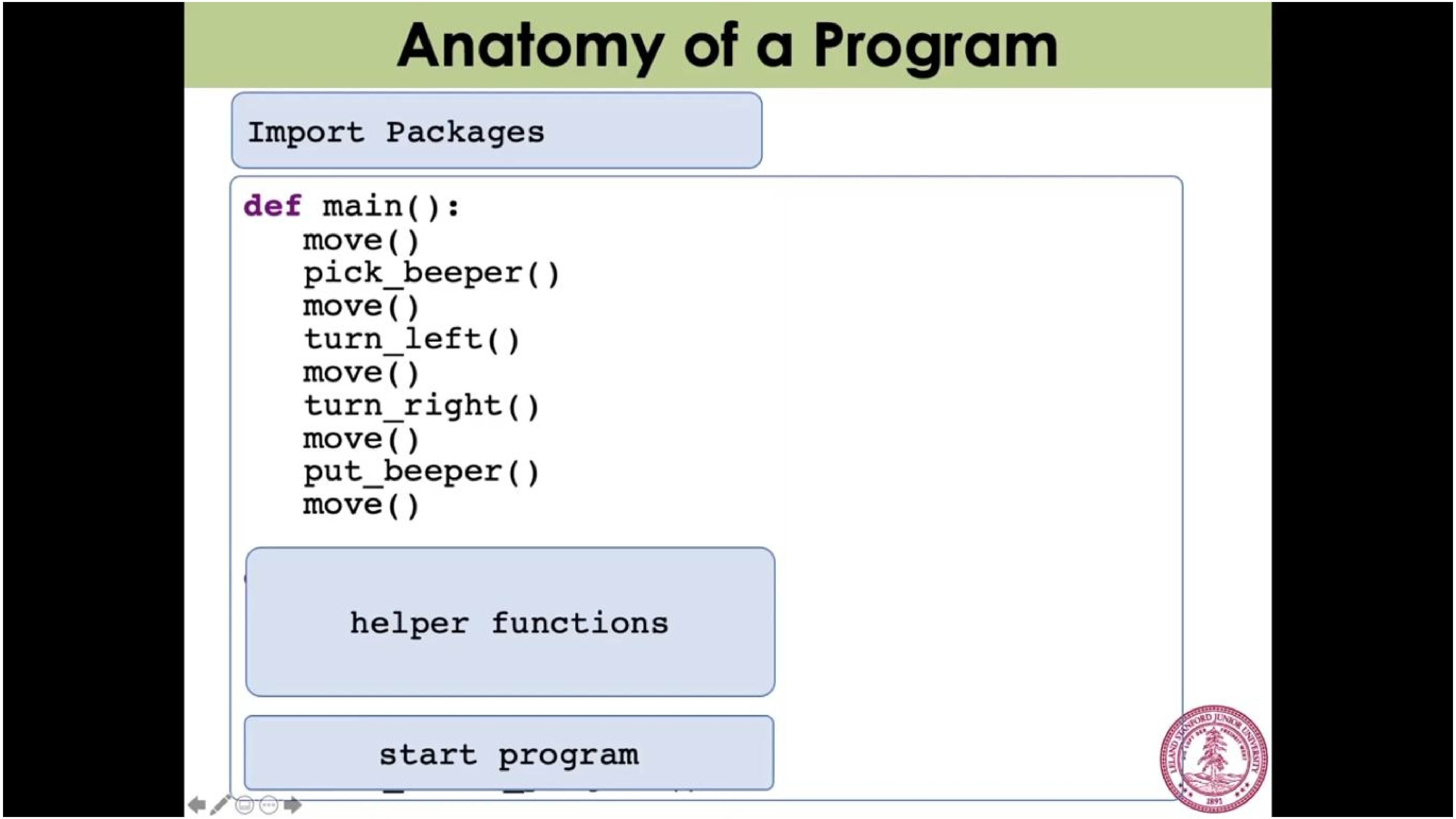


Import Packages

main function

helper functions

start program



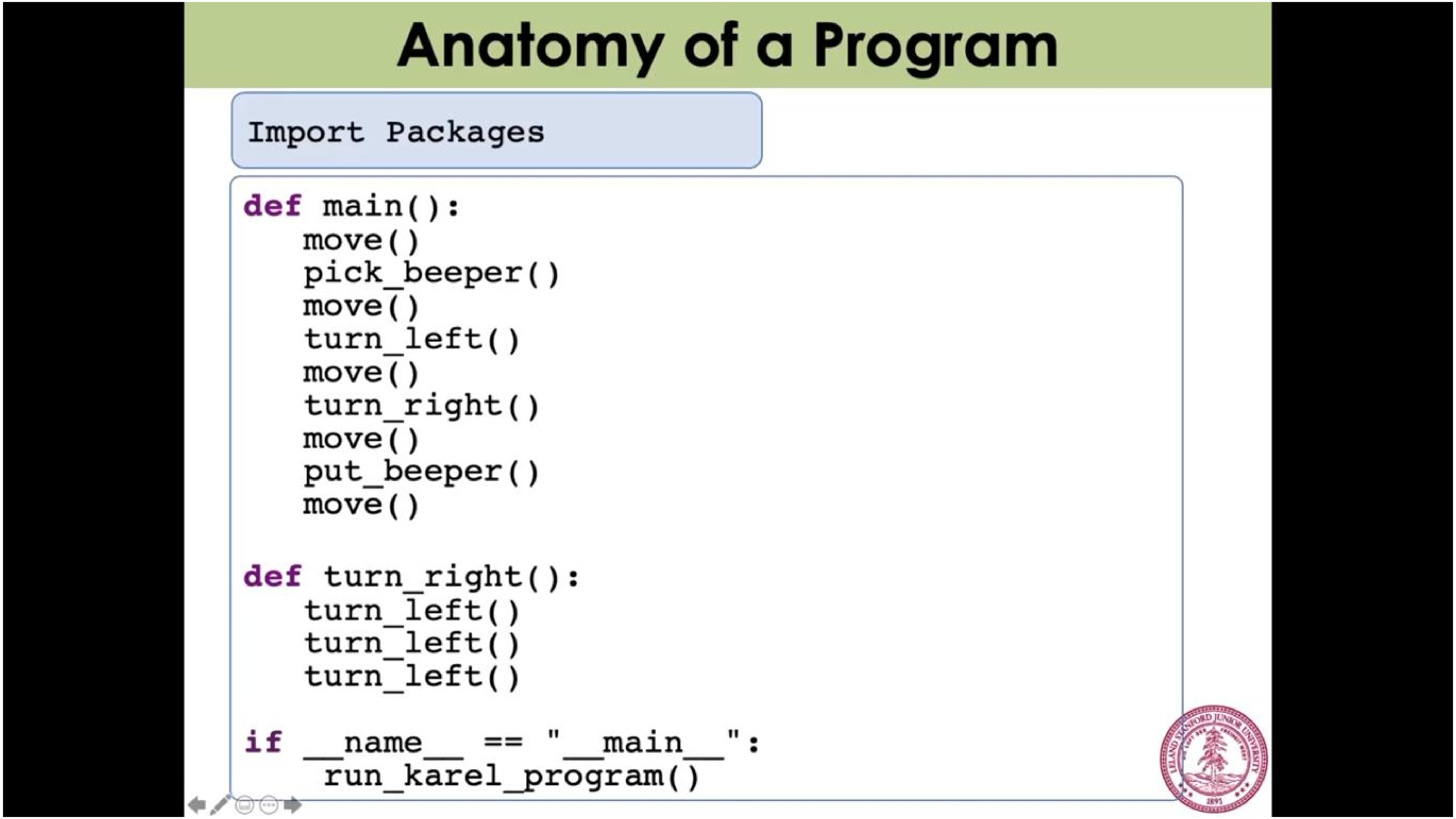
Anatomy of a Program

Import Packages

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

helper functions

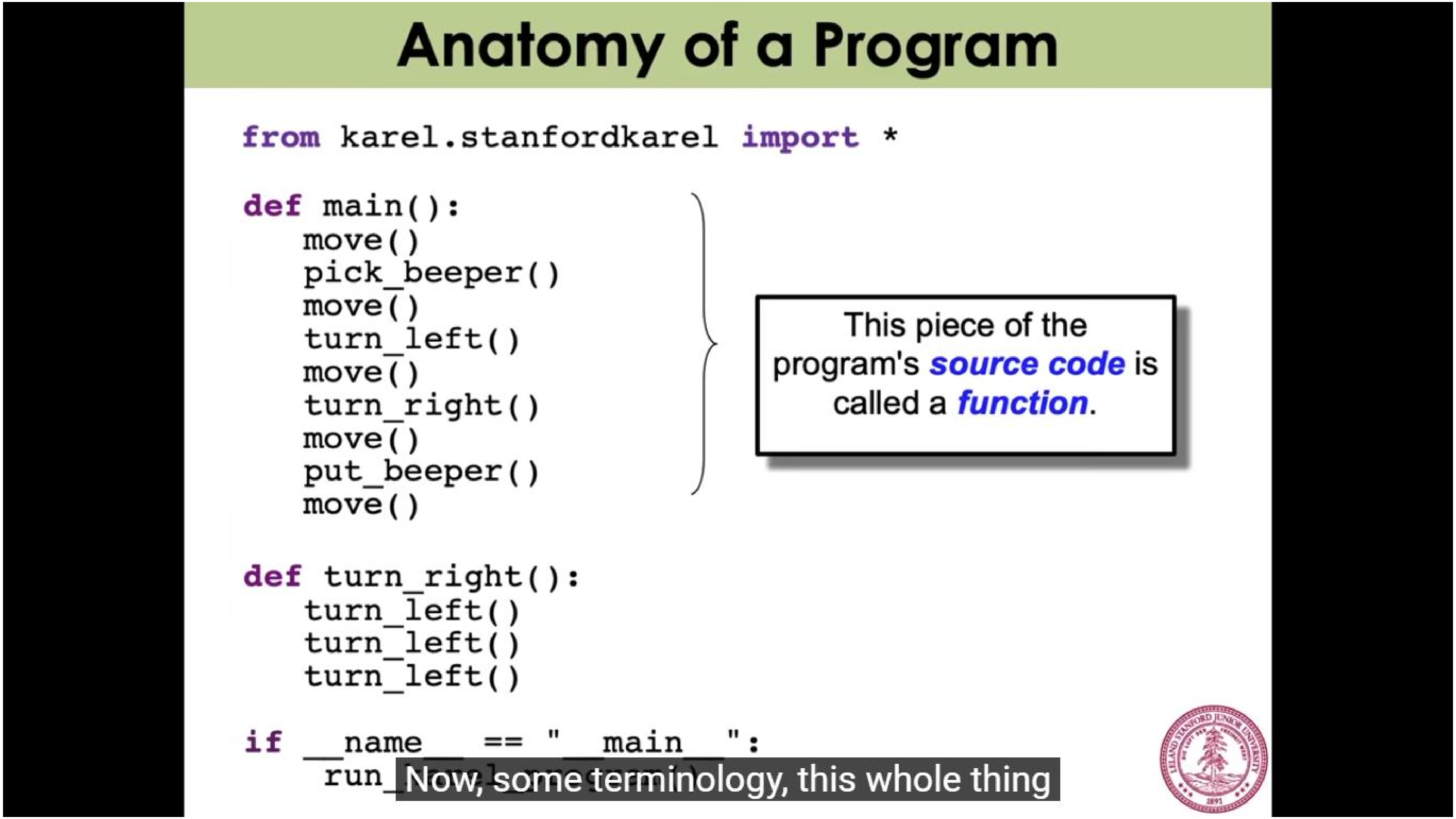
start program



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()

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

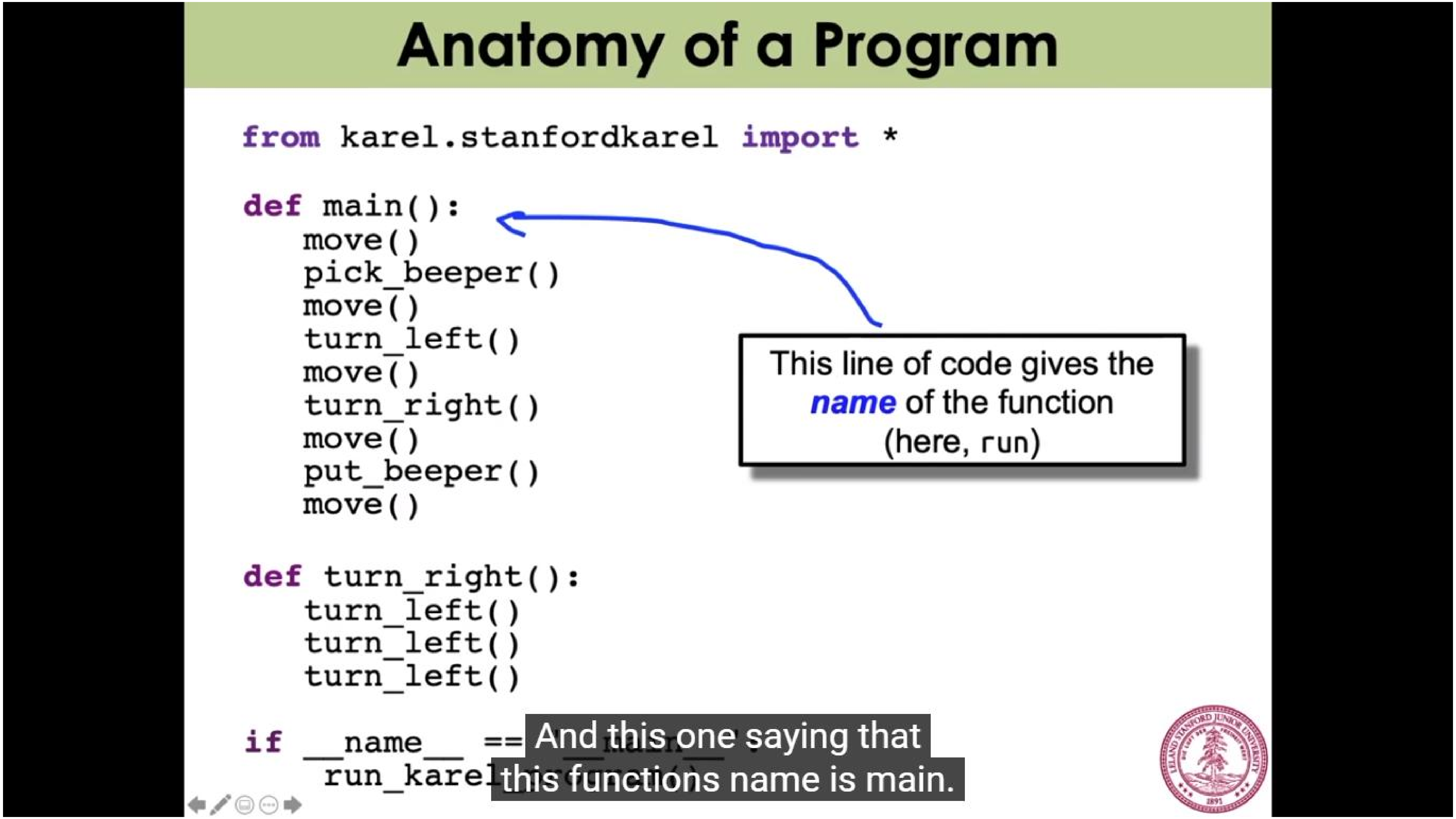


Anatomy of a Program

```
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_
```

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

Now, some terminology, this whole thing

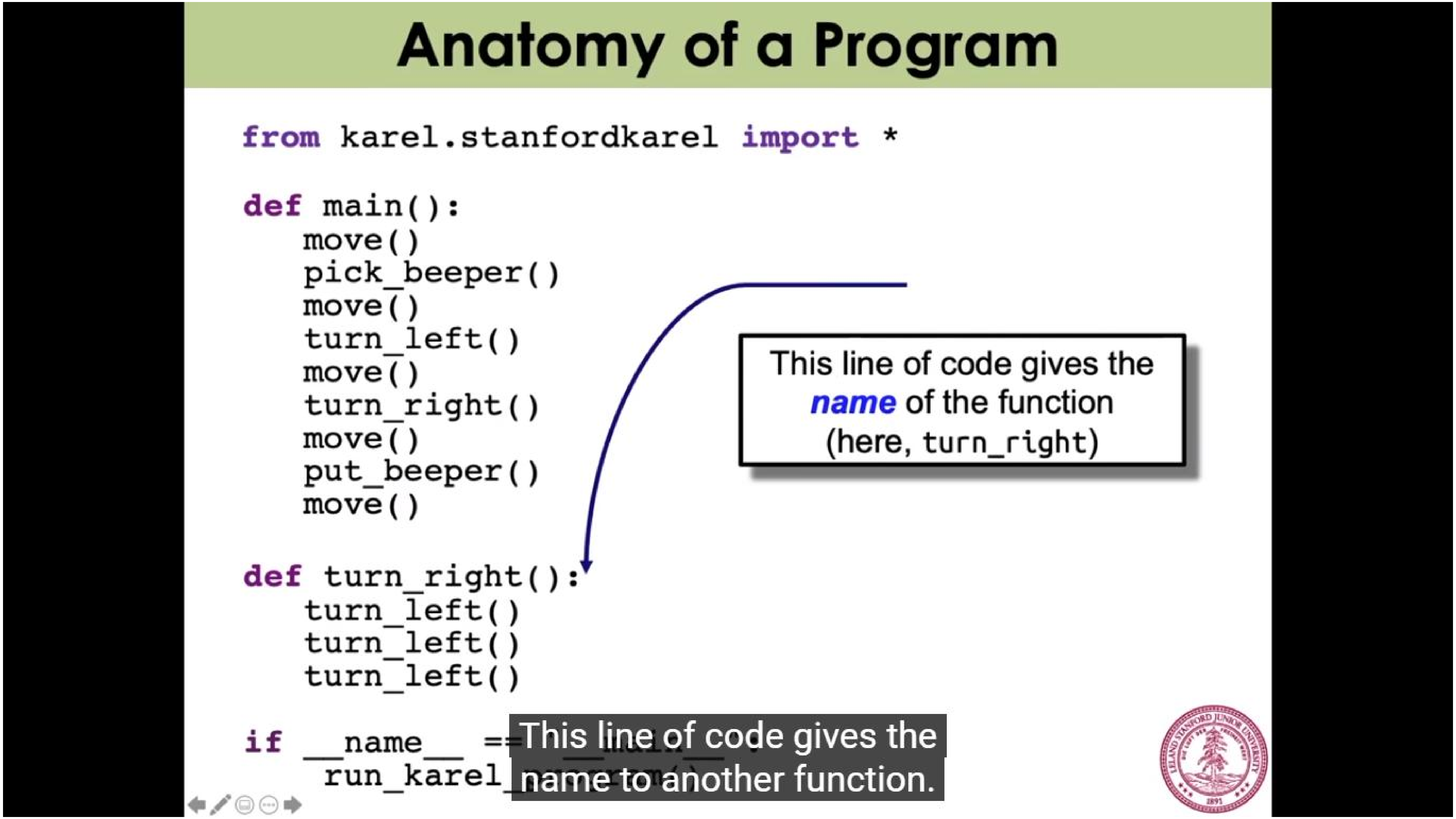


Anatomy of a Program

```
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__ ==
    run_karel
```

This line of code gives the *name* of the function (here, `run`)

And this one saying that this functions name is main.

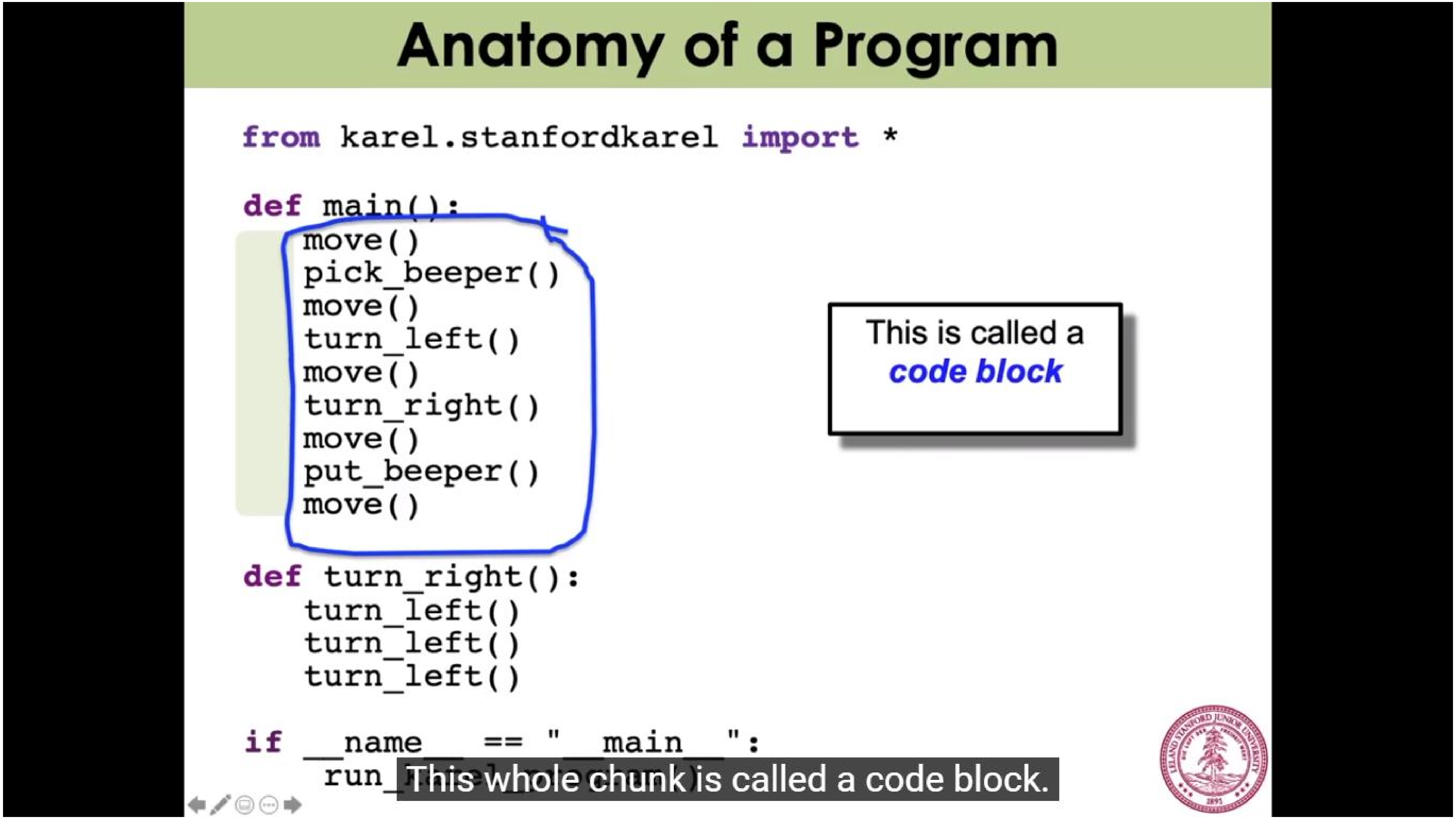


Anatomy of a Program

```
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, turn_right)

This line of code gives the name to another function.

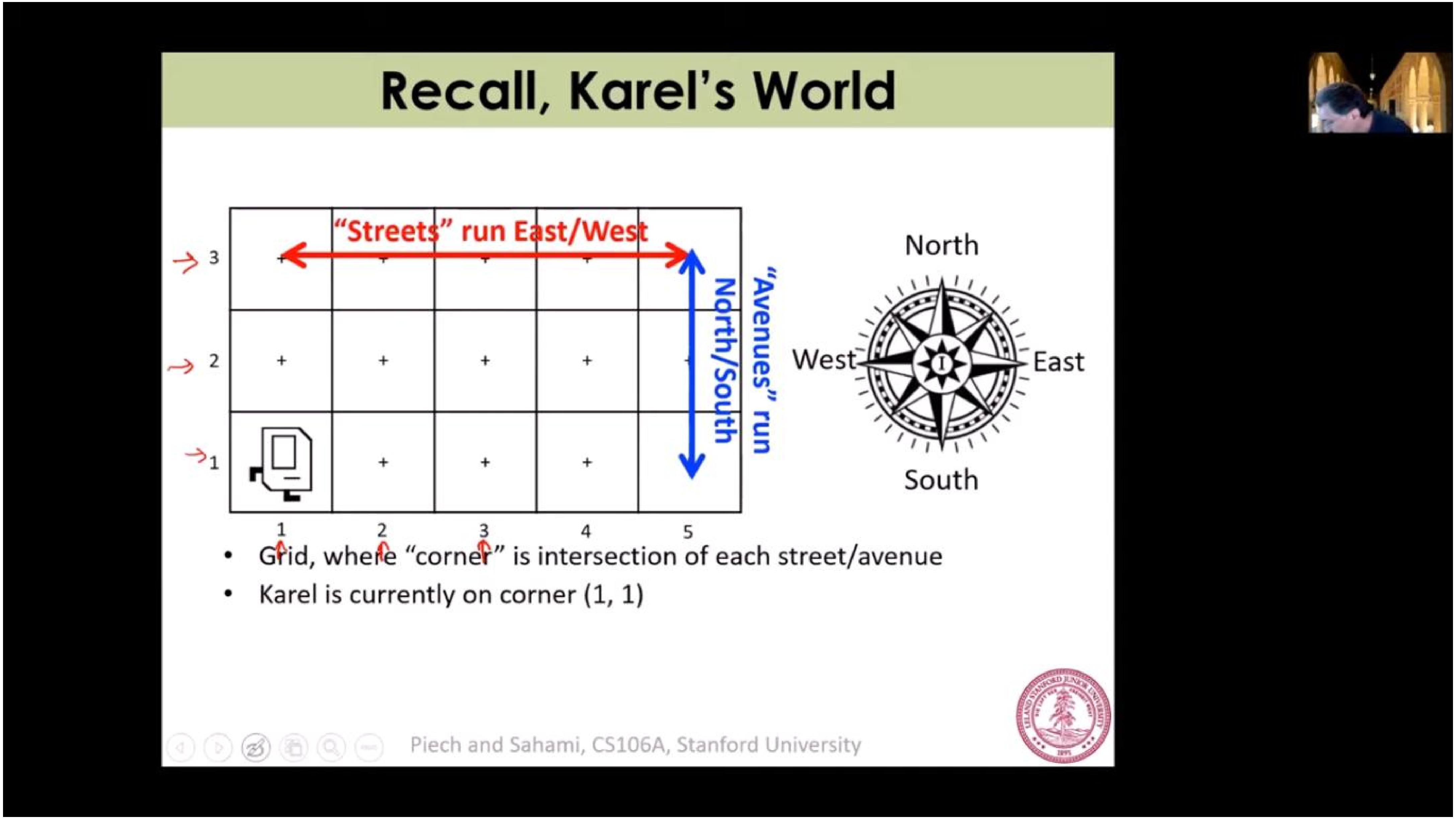


Anatomy of a Program

```
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_
```

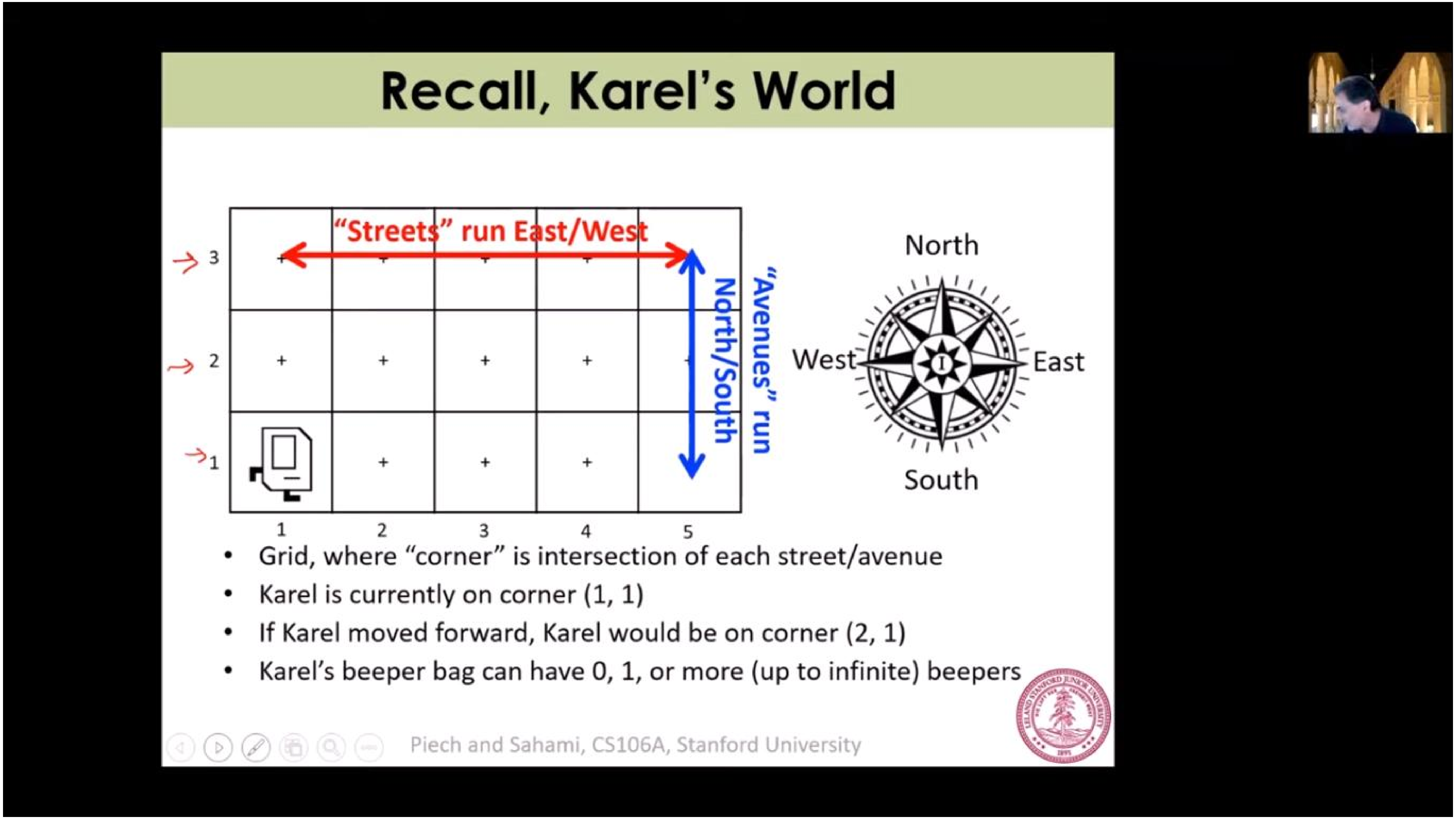
This is called a ***code block***

This whole chunk is called a code block.



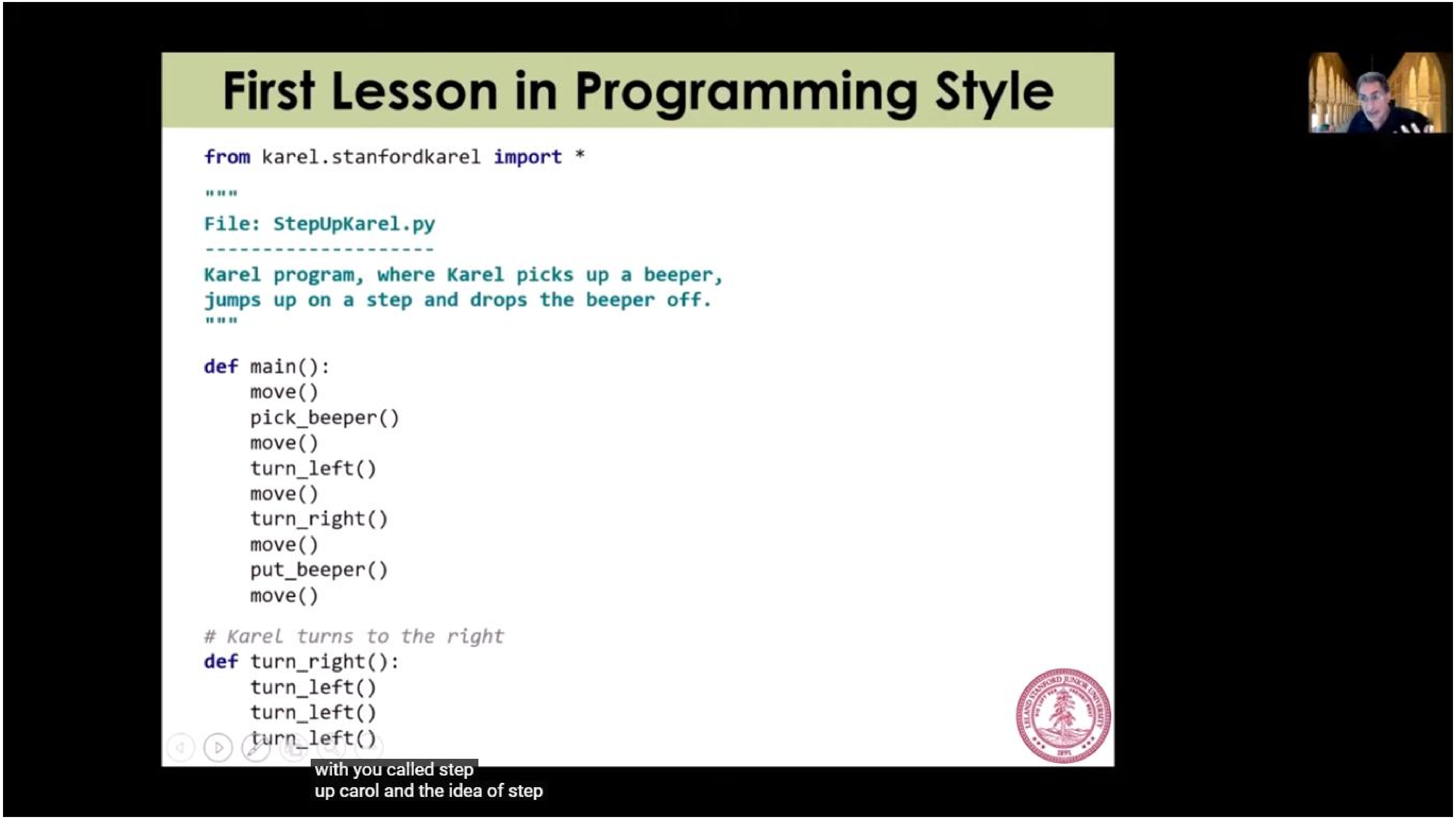
Recall, Karel's World

- Grid, where "corner" is intersection of each street/avenue
- Karel is currently on corner (1, 1)



Recall, Karel's World

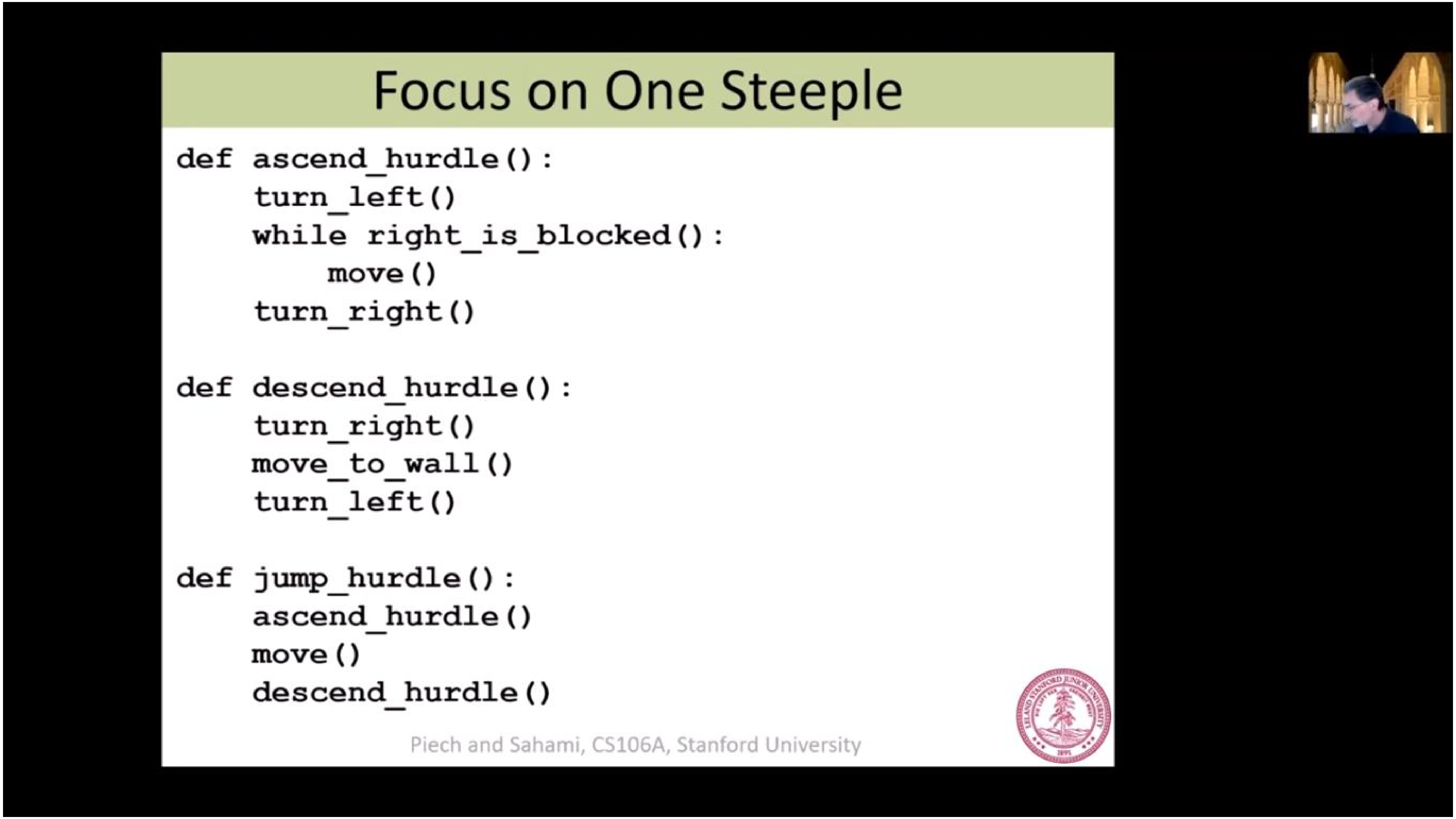
- Grid, where "corner" is intersection of each street/avenue
- Karel is currently on corner (1, 1)
- If Karel moved forward, Karel would be on corner (2, 1)
- Karel's beeper bag can have 0, 1, or more (up to infinite) beepers



First Lesson in Programming Style

```
from karel.stanfordkarel import *

"""
File: StepUpKarel.py
--------------------
Karel program, where Karel picks up a beeper,
jumps up on a step and drops the beeper off.
"""

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

# Karel turns to the right
def turn_right():
    turn_left()
    turn_left()
    turn_left()
```

with you called step
up carol and the idea of step



Focus on One Steeple

```
def ascend_hurdle():
    turn_left()
    while right_is_blocked():
        move()
    turn_right()

def descend_hurdle():
    turn_right()
    move_to_wall()
    turn_left()

def jump_hurdle():
    ascend_hurdle()
    move()
    descend_hurdle()
```

A Whole Program:
SteepChaseKarel.py