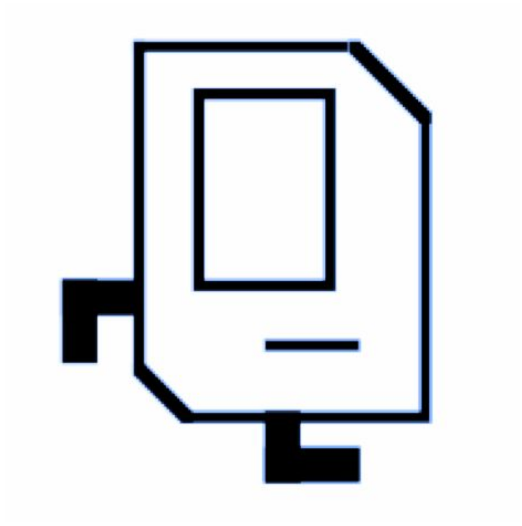
Section 1

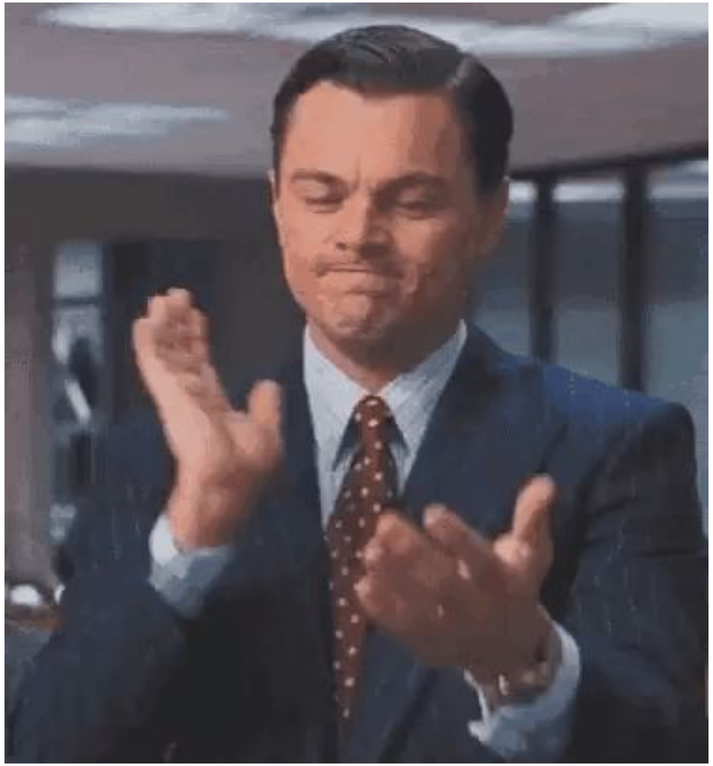
April 22, 2021



Katherine Michel, Section Leader, Stanford Code in Place, 2021

Welcome!

Congrats!



Katherine Michel, Section Leader, Stanford Code in Place, 2021

Today’s Agenda

- Get to know each other
- Recap: control flow, helper functions, decomposition
- Work on the Hospital Karel problem
- Questions/comments/discussion

Housekeeping

- One Ed for all of the students
- One Ed for the section
- Privacy
- If we run over, no pressure to stay
- This is not about perfect answer/finishing
- I’m here to answer your questions!

Wharton
UNIVERSITY of PENNSYLVANIA

pinax

About Me

DEFNA
DJANGO EVENTS
FOUNDATION
NORTH AMERICA

in_place(2020)

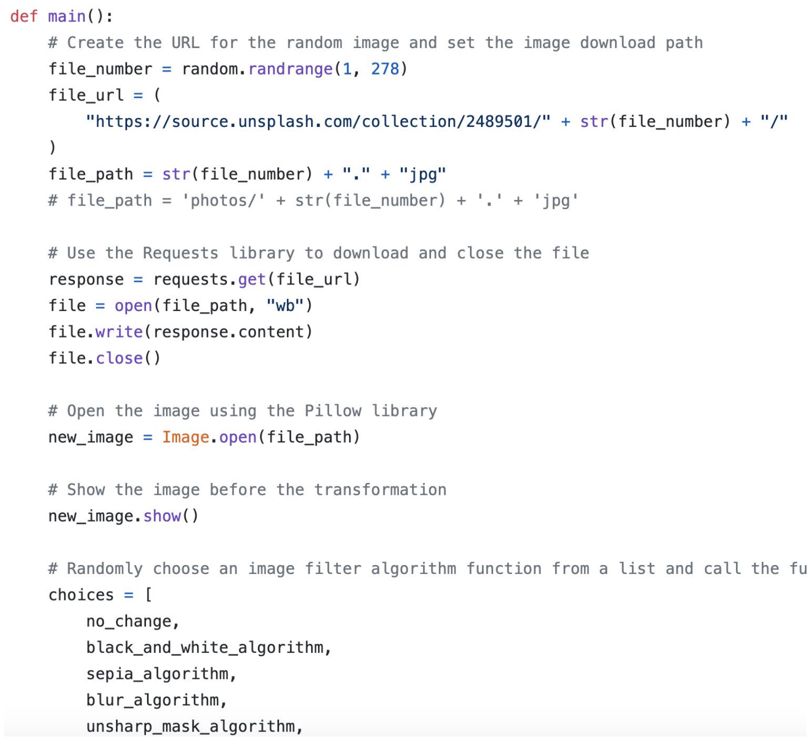
My Motivation for Being a Student Last Year

- In my work, I was reading code, but not writing it
- I wanted to start making apps from scratch
- I did not have an intuitive understanding of control flow

My Motivation for Being a Section Leader This Year

- Have the enjoyment of helping you learn
- I have a goal of doing more group teaching
- This is a great opportunity to get experience
- Reinforce what I learned last year

“Artwork” by Me and Python



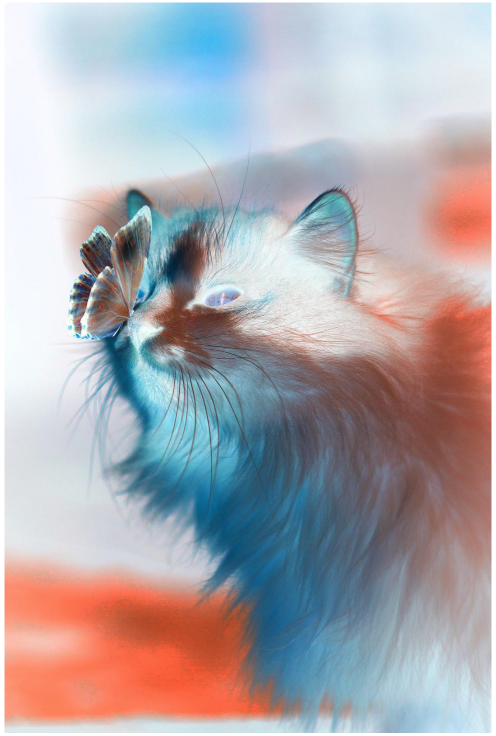
```python
def main():
    # Create the URL for the random image and set the image download path
    file_number = random.randrange(1, 278)
    file_url = (
        "https://source.unsplash.com/collection/2489501/" + str(file_number) + "/"
    )
    file_path = str(file_number) + "." + "jpg"
    # file_path = 'photos/' + str(file_number) + '.' + 'jpg'

    # Use the Requests library to download and close the file
    response = requests.get(file_url)
    file = open(file_path, "wb")
    file.write(response.content)
    file.close()

    # Open the image using the Pillow library
    new_image = Image.open(file_path)

    # Show the image before the transformation
    new_image.show()

    # Randomly choose an image filter algorithm function from a list and call the fu
    choices = [
        no_change,
        black_and_white_algorithm,
        sepia_algorithm,
        blur_algorithm,
        unsharp_mask_algorithm,
```



Katherine Michel, Section Leader, Stanford Code in Place, 2021

About You: Ice Breaker!

- Breakout rooms
- Introduce yourselves
- Why are you here?
- Is there anything in particular that you’d like to do with Python? (Web dev, automation, Data Science)

Control Flow

- Question: Why do we need it?

Control Flow

- Question: Why do we need it?
- A computer is not very sophisticated… it needs to be literally told what choice to make and when and how to loop to get things done

Recap: Control Flow

- What is an if statement?
- What is an if/else statement?
- What is a while loop?
- What is a for loop?

if Statement

```
def main():
    if front_is_clear():
        move()
```

if Statement

```
def main():
    if front_is_clear():
        move()
```

If a condition is met, something happens

if/else Statement

```
def main():
    if front_is_clear():
        move()
    else:
        turn_left()
```

if/else Statement

```
def main():
    if front_is_clear():
        move()
    else:
        turn_left()
```

If a condition is met, something happens
If the condition is not met, something else happens

while Loop

```
def main():
    while front_is_clear():
        move()
```

while Loop

```
def main():
    while front_is_clear():
        move()
```

The loop executes until the condition is no longer true

for Loop

```
def main():
    for i in range(2):
        move()
```

for Loop

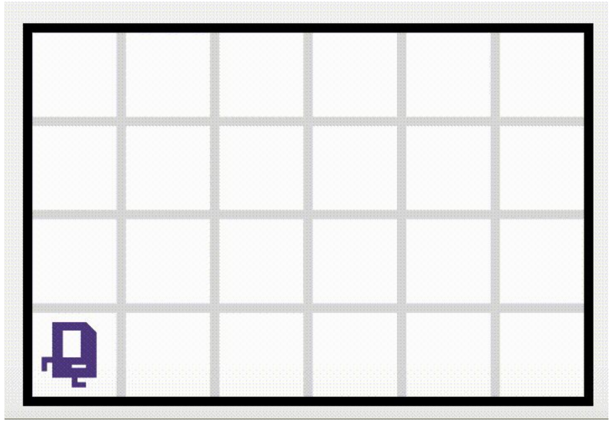
```
def main():
    for i in range(2):
        move()
```

The loop executes a predetermined number of times
In this case, it will execute 2 times

What I Wish I Had Known About Control Flow

- An if statement is not a loop… so it only executes one time… in this case, one move

```
def main():
    if front_is_clear():
        move()
```

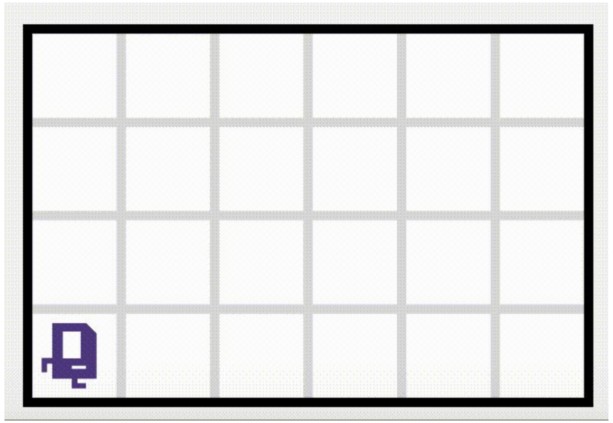


Katherine Michel, Section Leader, Stanford Code in Place, 2021

What I Wish I Had Known About Control Flow

- A while statement is a loop… so it executes more than one time (otherwise if statement)

```
def main():
    while front_is_clear():
        move()
```

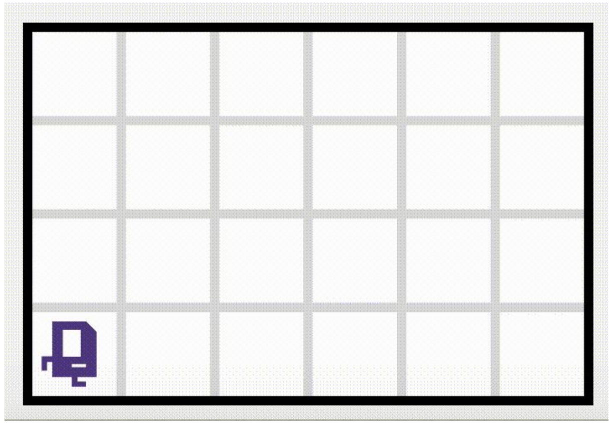


Katherine Michel, Section Leader, Stanford Code in Place, 2021

What I Wish I Had Known About Control Flow

- An if statement inside of a while statement is still not a loop… the while statement is executing it over and over

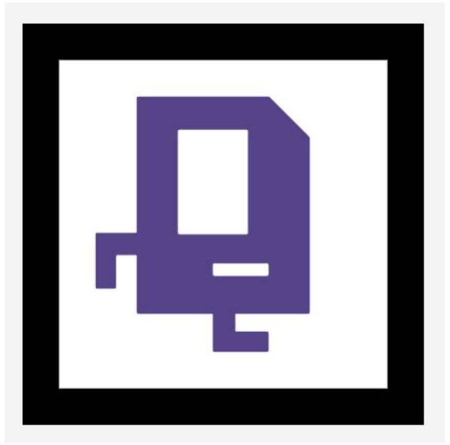
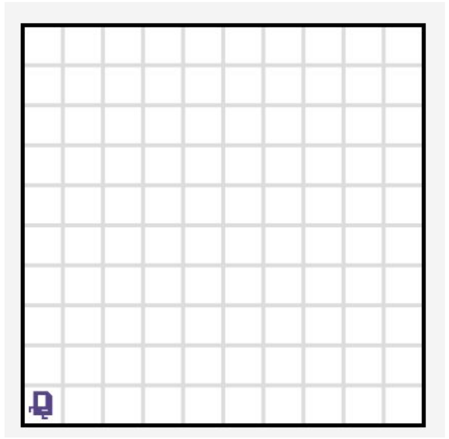
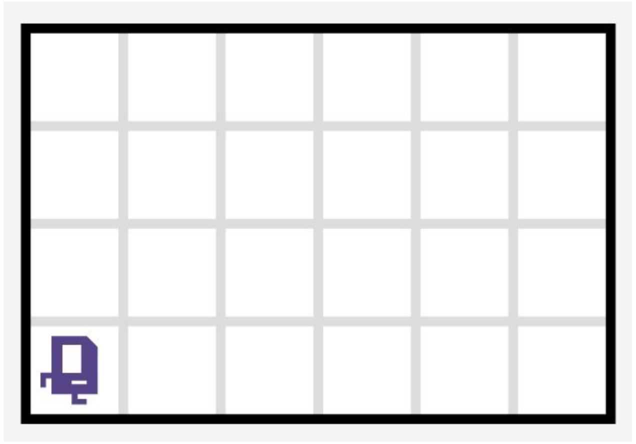
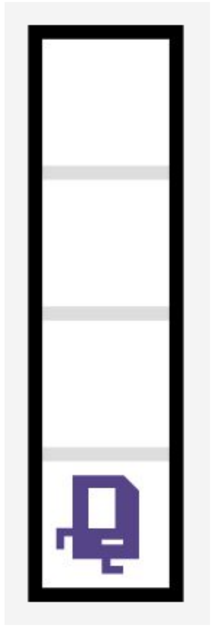
```
def main():
   while front_is_clear():
        if no_beepers_present():
            put_beeper()
            move()
```



Katherine Michel, Section Leader, Stanford Code in Place, 2021

What I Wish I Had Known About Control Flow: Major Gotcha!

- If you use a for loop when the number of iterations is unknown, you are likely to break a test case… might need a while loop instead!

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Recap: Helper Function

main function →

```
def main():
    while front_is_clear():
        if beepers_present():
            turn_right()
```

← Helper function is called from the main function… program jumps to it, executes it, then jumps back

Helper function →

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

Karel Commands

Base Karel commnds:

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

Karel program structures:

```
# Comments can be included in any part
# of a program. They start with a #
# and include the rest of the line.

def main() :
    code to execute

declarations of other functions
```

Names of the conditions:

```
front_is_clear()        front_is_blocked()
beepers_present()       no_beepers_present()
beepers_in_bag()        no_beepers_in_bag()
left_is_clear()         left_is_blocked()
right_is_clear()        right_is_blocked()
facing_north()          not_facing_north()
facing_south()          not_facing_south()
facing_east()           not_facing_east()
facing_west()           not_facing_west()
```

Conditions:

```
if condition:
    code run if condition passes

if condition:
    code block for "yes"
else:
    code block for "no"
```

Loops:

```
for i in range( count):
    code to repeat

while condition:
    code to repeat
```

Function Declaration:

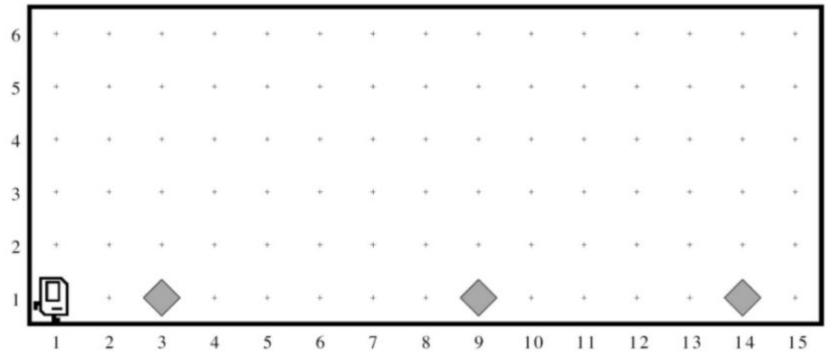
```
def  name():
    code in the body of the function.
```

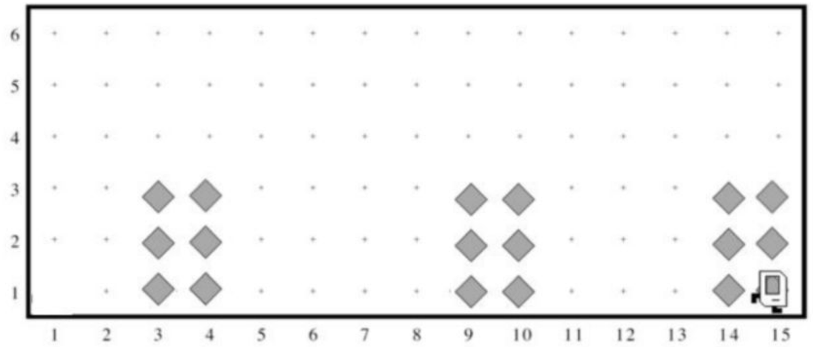
Extra Karel Commands:

```
paint_corner(COLOR_NAME)
corner_color_is(COLOR_NAME)
```

Katherine Michel, Section Leader, Stanford Code in Place, 2021

Hospital Karel





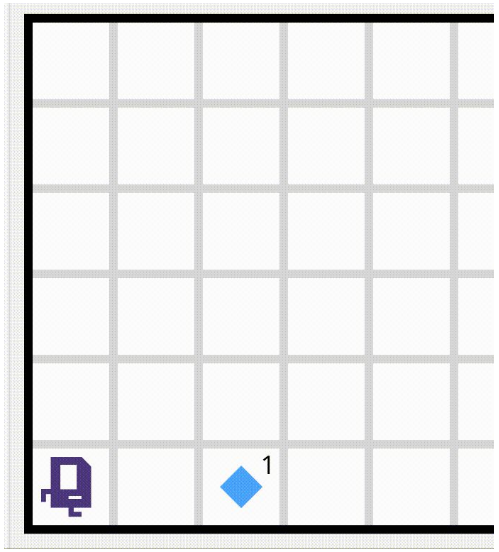
Katherine Michel, Section Leader, Stanford Code in Place, 2021

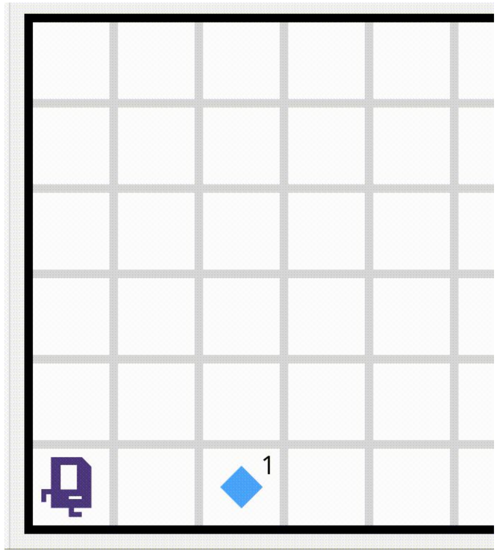
Decomposition Pro-Tips

Recap: Decomposition

"The process of breaking a program down into smaller pieces is called **decomposition**, and the component parts of a large problem are called **subproblems**." - Lecture 1, Decomposition Chapter

Karel leaves the first beeper in place… builds part of one wall and all of the other

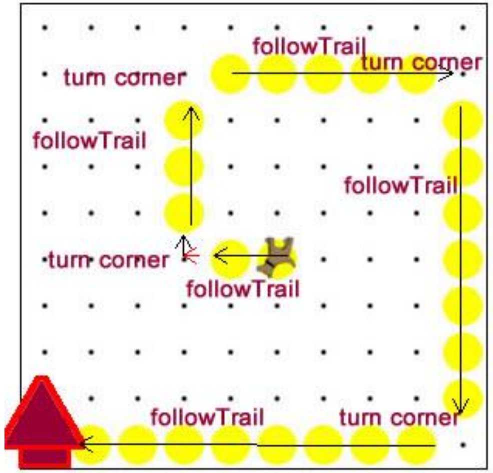


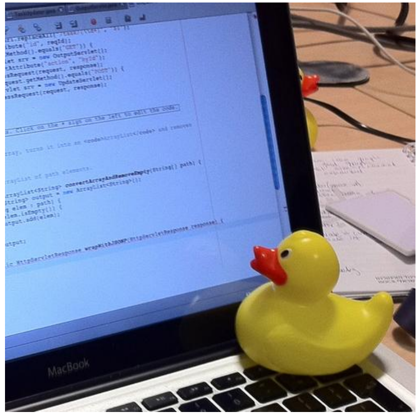


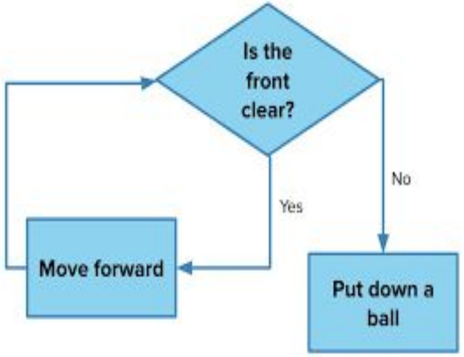
Karel picks up the first beeper… then builds two full walls… this enables more function reuse

Problem Solving Strategies

- Pseudocode/move Karel in Karel playground
- Rubber duck debugging (say code out loud)
- Draw it out (paint program)/create a diagram
- Take a walk







Closing… Thoughts, Questions, Discussion?