

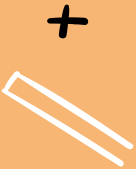



Welcome to Section!!!

Week One

Matt Smith

Original slides by Cameron Mohne and Maggie Lee



Introductions



A Little About Me

+



- ★ My name is Matt. I'll be your Section Leader for Code in Place, and everyone here will be your section-mates!
- ★ I'm from Canada but live in the Bay Area in California, USA.
- ★ In my free time I like to swim, make ceramics and play with my cat Spring.



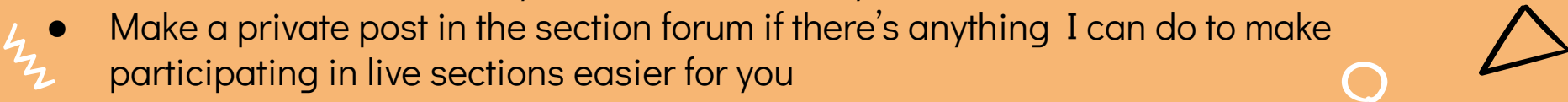


What About You All?

Go ahead and share:

1. **Your name**
2. **Where you're from**
3. **One thing you're looking forward to (it doesn't have to be from Code in Place)!**

How to participate in section:

- Unmute and speak directly to the group — this is not a large class where you need to ask to speak
 - Preferably keep cameras on if you are comfortable & able
 - You can instead share directly to everyone in the Zoom chat or you can message me on Zoom and I can read out your introduction for you
 - Make a private post in the section forum if there's anything I can do to make participating in live sections easier for you
- 



How to benefit from section

+

If you do the following, you'll get the maximum benefit from section time:

- Read through the lesson before section starts
 - You don't need to read the section handout or work on the section assignment before section – we will cover it all during section
 - Bring any questions you have about the lectures to section!
- Participate in the live section
- Ask questions in the section forum
 - You can ask questions privately to me or to the group
 - Only put your code for the assignments in private posts
 - You can also post questions in the course-wide forum publicly or privately (visible to SLs, TAs and instructors only)



Lecture Review



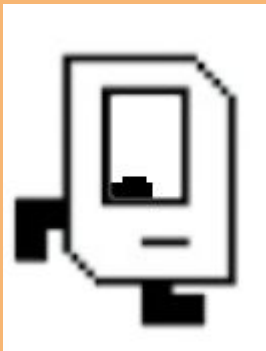


What We've Learned

Before we get into our sample problem for today, let's review a bit. We've learned:

- The basics about Karel, the magnificent and wonderful robot
- Functions, a way of breaking down big problems into smaller chunks
- Control Flow, loops and conditional statements which guide our programs

This is a **LOT** of content, especially if you are newer to CS!



make_dough()



shape_pasta()




cook_pasta()



For Loops

```
def main():  
    # repeats the body 99 times  
    for i in range(99):  
        # the "body"  
        put_beeper()
```



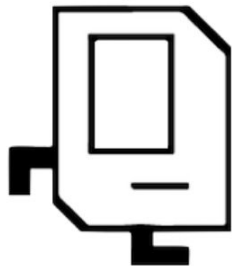
The background is a solid orange color. It is decorated with various white and black geometric shapes and lines. In the top left, there is a dashed line and a small white triangle. In the top center, there is a dashed line and a small white triangle. In the top right, there is a dashed line, a small white circle, and a black zigzag line. In the middle right, there are two parallel black lines and a small white triangle. In the bottom left, there is a black plus sign. In the bottom center, there is a black circle and a white triangle. In the bottom right, there is a black line and a white circle.

**Let's review and
refresh these
concepts a bit!**

Karel Overview



Hello, my name is Karel! Nice to meet you.



- Karel is a small, but mighty robot
- It has a few basic commands including: **move()**, **turn_left()**, **pick_beeper()**, and **put_beeper()**
- On its own, Karel has limited functionality, but with the help of our code, we can make great things happen!



Functions Overview



We can use functions to give a name to a chunk of code. When approaching a problem, it's useful to break down the problem into smaller pieces, and then solve each subproblem individually. Recall from lecture, where Chris and Mehran broke down the process of making pasta into three functions.

To make a function, start with the word **def** followed by the name of the function. The function's name should be lowercase, with words separated by underscores. Don't forget to indent the code:

```
def function_name():  
    # Function code goes here!!!
```

make_dough()



shape_pasta()



cook_pasta()




Control Flow Overview



Lastly, let's take a look at control flow! We have:

For-loop (definite loop):

- Do something a specific amount of times.
 - Do 10 push-ups
- 

While-loop (indefinite loop):

- Continuously do something as long as a condition is true.
 - While the cat is purring in your lap, pet the cat

If-statement:


- Do something only if a condition is true.
 - If the clothing looks good and is affordable, then buy it

For-Loop




An example for-loop that you may see and use with Karel:

```
def turn_right():  
    for i in range(3):  
        turn_left()
```



This loop is also called a *definite loop* because we know how many times it will run.




While-Loop




An example while-loop that you may see and use with Karel:

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



This loop is also called an *indefinite loop* because it will run until the associated condition becomes false (which may be never – be careful you don't get stuck in an infinite loop!)




If-Statement




An example if-statement that you may see and use with Karel:

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



An if-statement runs code inside of it only if the associated condition is true.



Any Questions?



Section Problem: Hospital Karel

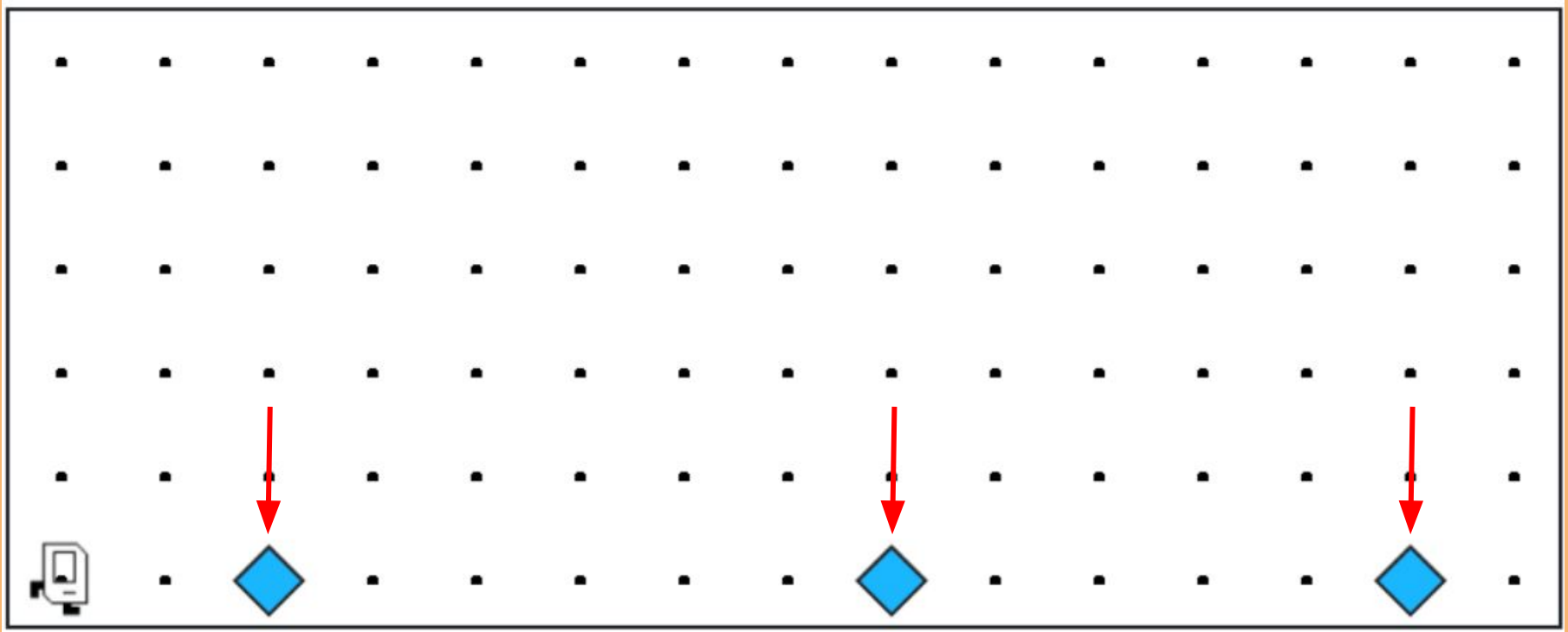


The background is a solid orange color. It is decorated with various white and black geometric shapes and lines. In the top left, there is a dashed white line and a solid white triangle. In the top center, there is a dashed white line and a solid white triangle. In the top right, there is a solid black zigzag line and a solid black circle. In the middle right, there are two parallel solid black lines and a solid black circle. In the bottom left, there is a solid black plus sign and a solid black circle. In the bottom center, there is a solid black plus sign and a solid black circle. In the bottom right, there is a solid black plus sign and a solid black circle.

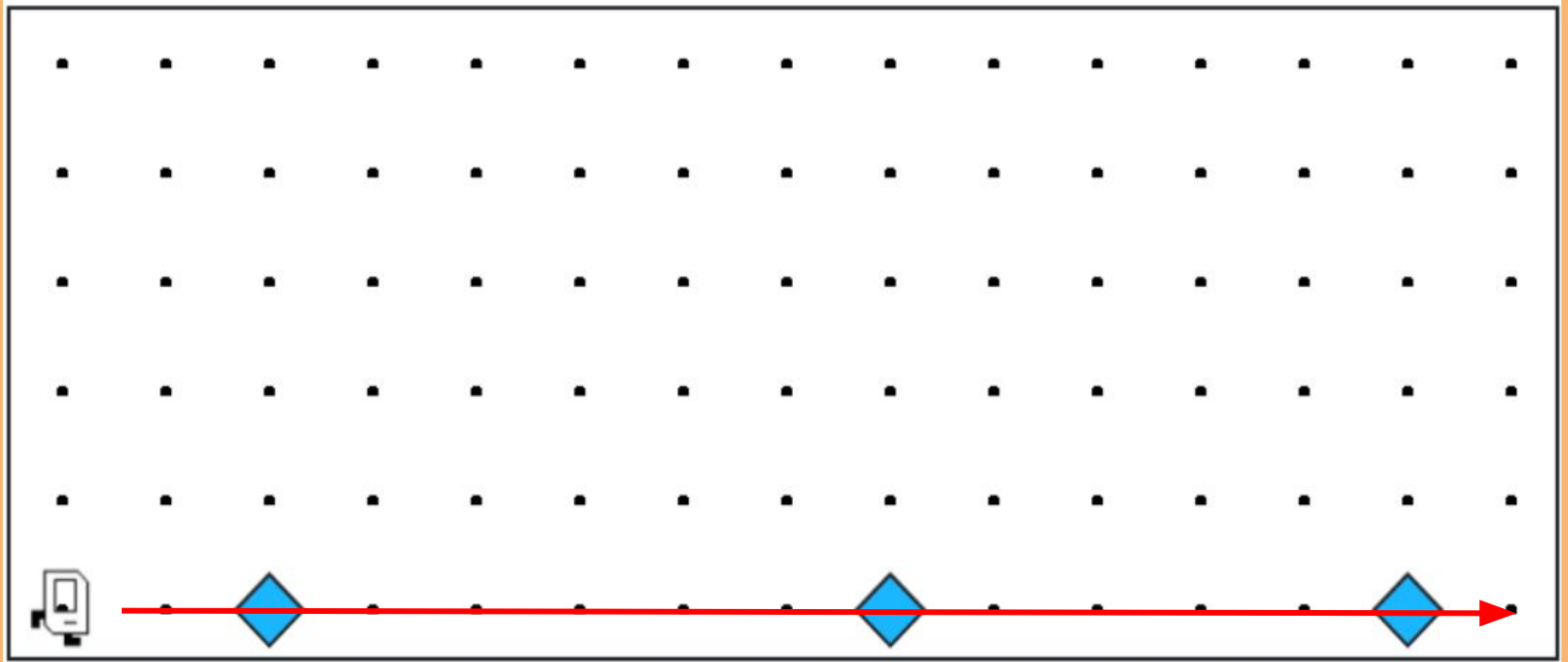
Setting Context

Countries around the world are dispatching hospital-building robots to make sure anyone who gets sick can be treated. They have decided to enlist Karel robots. Your job is to program those robots.

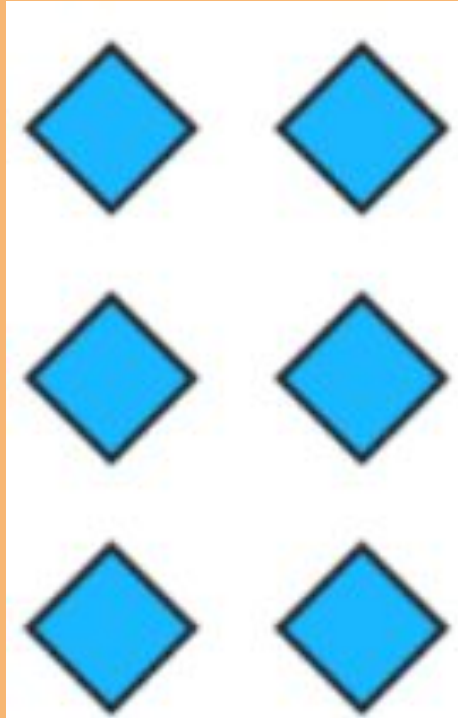
Each beeper in the figure represents a place a hospital should be built.



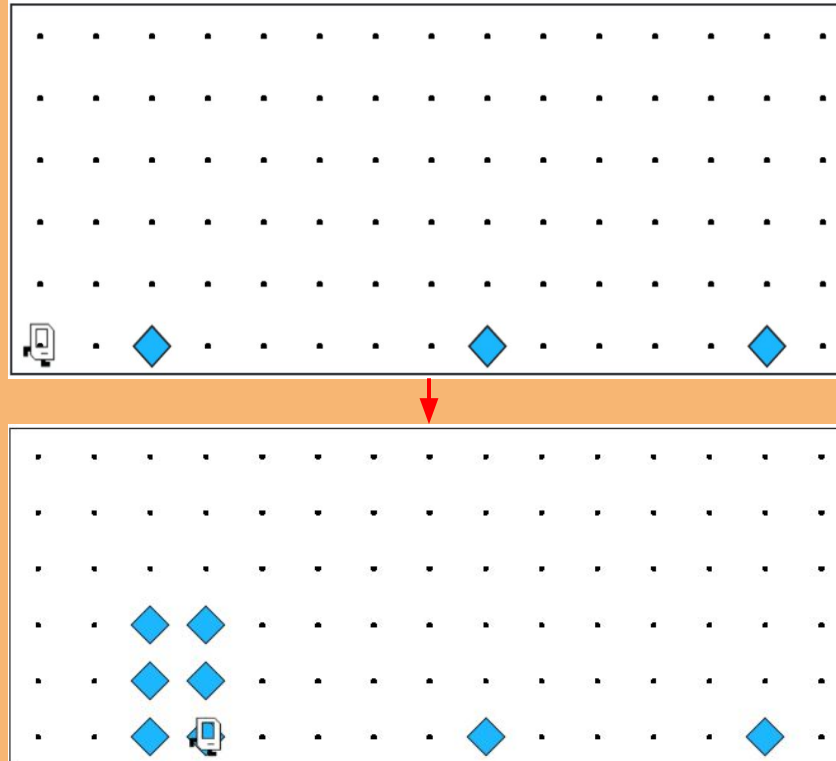
Karel's job is to walk along the row and build a new hospital in the places marked by each beeper.



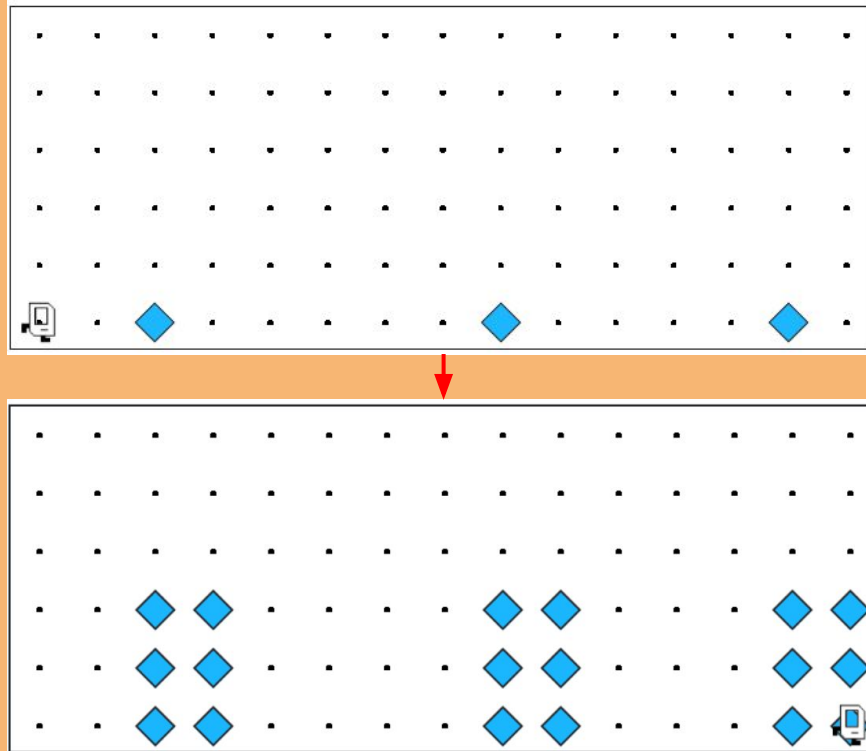
Hospitals look like this: a 3x2 rectangle of beepers!



The new hospital should have their corner at the point at which the pile of supplies was left.



At the end of the run, Karel should be at the end of the row having created a set of hospitals. For the initial conditions shown, the result would look like this:



The background is a solid orange color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a black plus sign in the bottom left, a white circle in the bottom center, a white triangle in the bottom center, a black plus sign in the bottom center, a black circle in the bottom center, and a white circle in the bottom right.

Questions Before We Begin?

The background is a solid pink color. It is decorated with various hand-drawn geometric shapes and symbols in white and black. These include: a dashed line in the top left; a white triangle in the top center; a black zigzag line in the top right; a white circle in the top right; two parallel black lines in the top right; a white triangle in the top right; a large black circle on the right edge; a white circle in the bottom center; a black plus sign in the bottom left; a white triangle in the bottom center; a dashed line in the bottom center; a black plus sign in the bottom left; and a white circle in the bottom right.

Let's get to work!