



**MICOM**

**DEBUGGING R CODE**

MARISA EISENBERG  
JULIE (JULES) GILBERT

VISIT

**[HTTPS://JULESJULESJULES.QUARTO.PUB/R-WORKSHOPS-WITH-MICOM/](https://julesjulesjules.quarto.pub/r-workshops-with-micom/)**  
TO DOWNLOAD CODE & DATA FILES FOR TODAY



# INTRODUCTIONS

Let's go around the room and tell everyone your:

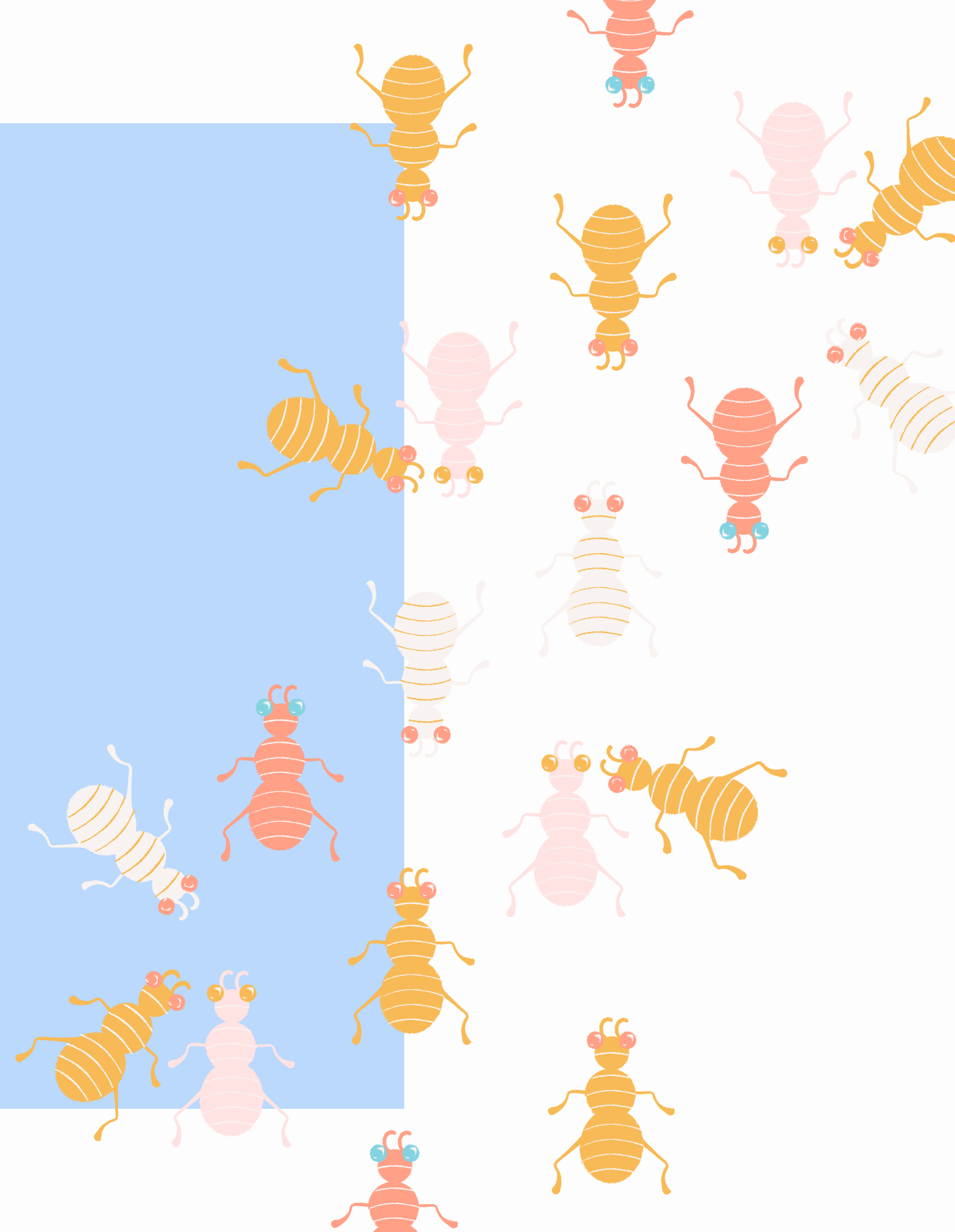
Name

Position/Location

If you were to describe yourself as an animal, what would you choose?

# HOW DO I DEBUG MY CODE?

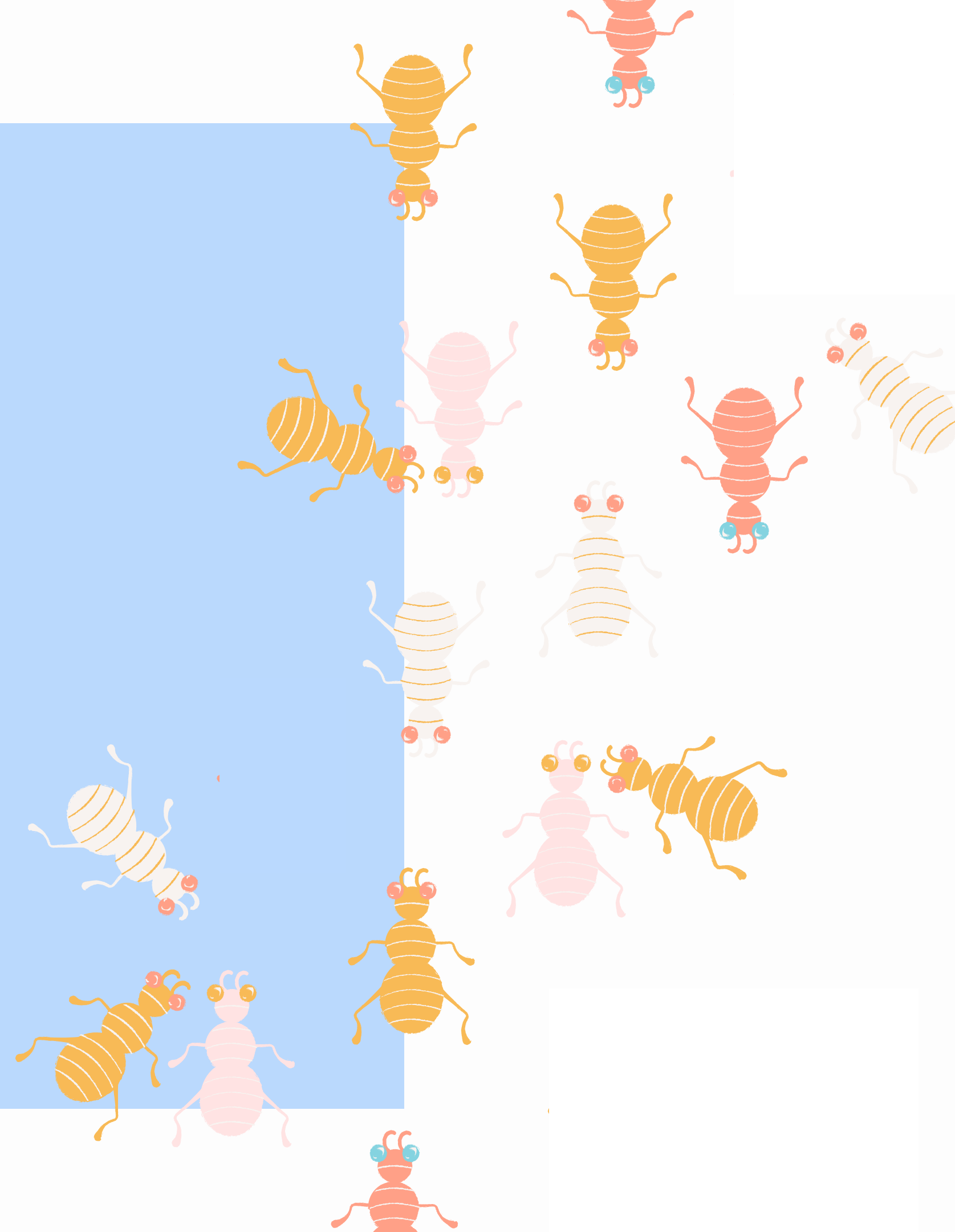
- Debugging - crucial skill in programming
- Many ways to do it
- Best way to learn is lots of practice!
- But here are some common methods & useful tips to get you started





# HOW DO I DEBUG MY CODE?

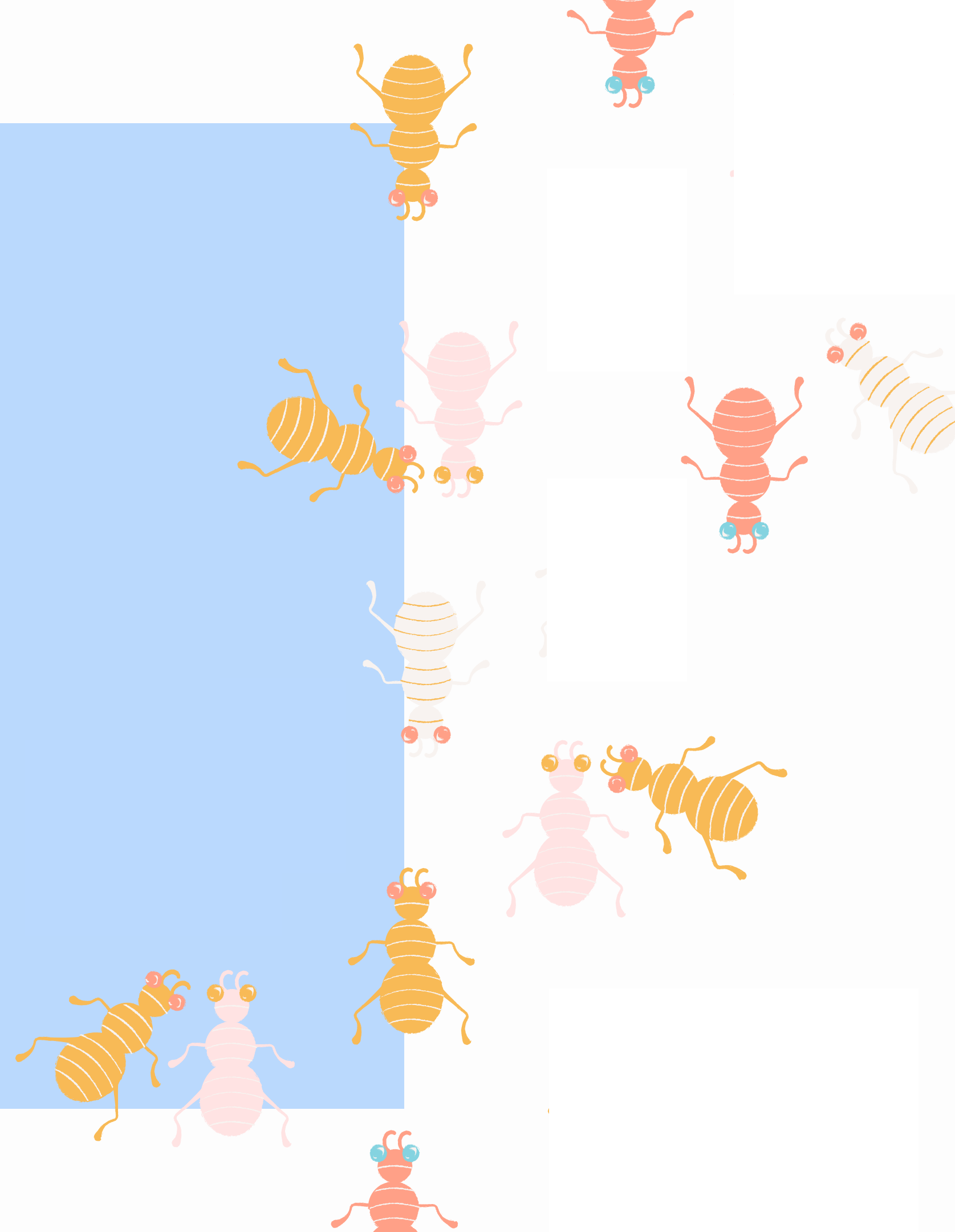
- The computer just does what you tell it—  
an error is usually in the code
- Questions to ask yourself:
  - What did you expect your code to do and why?
  - What did your code do instead and how do you know?





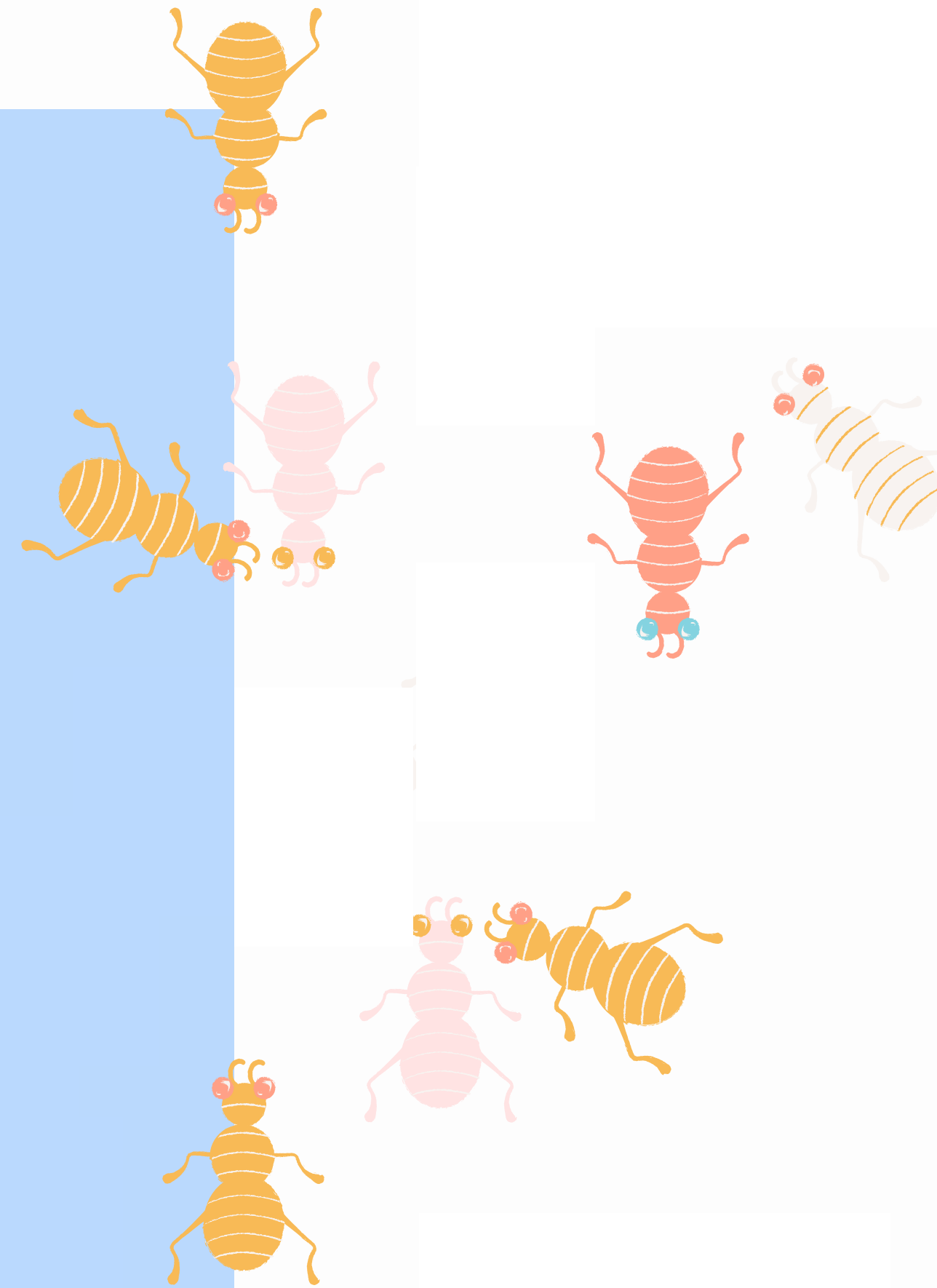
# HOW DO I DEBUG MY CODE?

- Use the error messages—if nothing else, they give you a line number
  - Don't forget to scroll all the way up to the first error!
- Find where the problem appears to be (note it might not actually be there!)



# HOW DO I DEBUG MY CODE?

- Follow through the steps of the code as though you are the computer
- Use print statements!
- Hypothesize what the problem could be, then test it



# SOME FUN: WHY ARE THEY CALLED BUGS?

92  
9/9

0800 Antan started  
1000 " stopped - antan ✓


1300 (032) MP-MC  $\left\{ \begin{array}{l} 1.2700 \cdot 9.037847025 \\ 1.982647000 \\ 2.130476415 \end{array} \right\}$   $\left\{ \begin{array}{l} 9.037846995 \text{ correct} \\ 4.615925059(-2) \end{array} \right\}$

(033) PRO 2 2.130476415  
correct 2.130676415

Relays 6-2 in 033 failed special speed test  
in relay .. 11,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

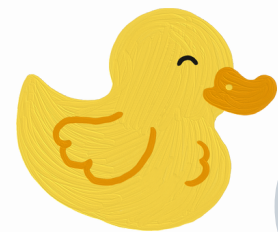
~~1630~~ 1630 Antan started.  
1700 closed down.

*Relay 2145  
Relay 3370*





# WE'RE GOING TO GET RIGHT INTO IT, BUT HERE ARE SOME BIG IDEAS



01

Documenting your code and data sources is very helpful for debugging efforts

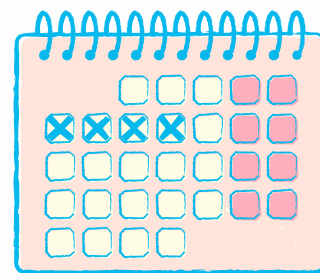


02

The classic: explain what's going on to a rubber duck (or a friend)

03

Errors can be stated by R and be clear or unclear, but practice helps!



04

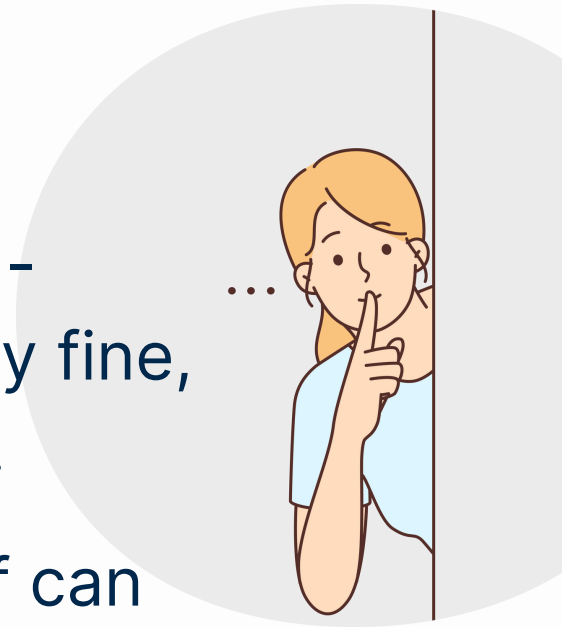
Silent errors are the real villains - often your code will run perfectly fine, but things may still be incorrect.

05

Adding checks to the code itself can prevent errors from propagating, particularly in processing pipelines

06

Print statements are your best friend!



# SOME DEBUGGING RESOURCES (FOR THE FUTURE)

01 <https://blog.codeunion.io/2014/09/03/teaching-novices-how-to-debug-code/>

02 <https://www.geeksforgeeks.org/how-to-debug-your-code-for-beginners/>

03 <https://www.codecademy.com/resources/blog/how-to-debug-your-code/>

