

The UMPIRE Strategy

TIP102

Week 1

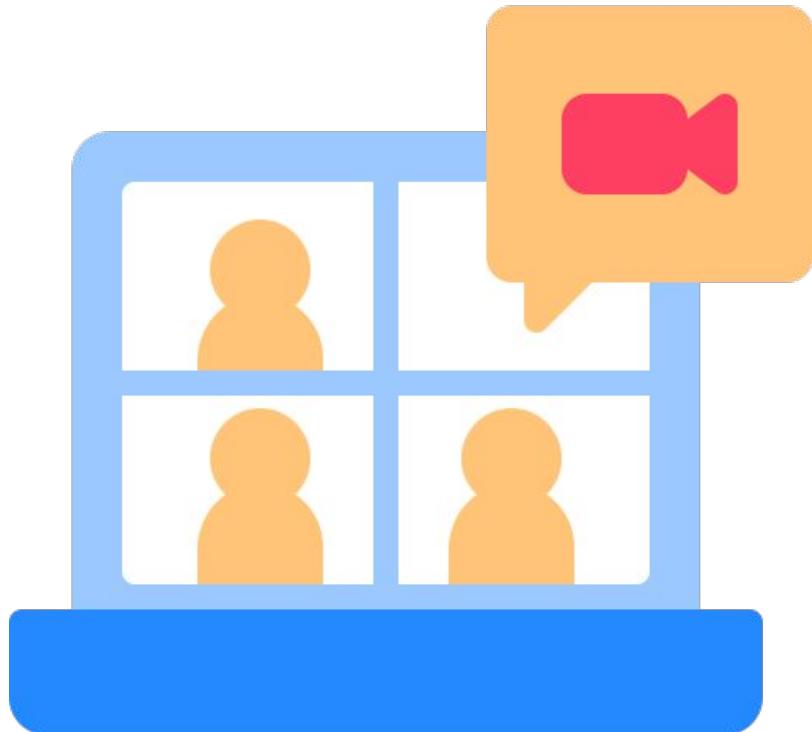
Agenda

1	Welcome	0:00 - 0:15
2	The UMPIRE Strategy	0:15 - 0:45
3	BREAK	0:45 - 0:50
4	Collaborative Problem Solving	0:50 - 1:50
5	Wrap Up	1:50 - 2:00

15 minutes

Welcome! 

Turn on your cameras !



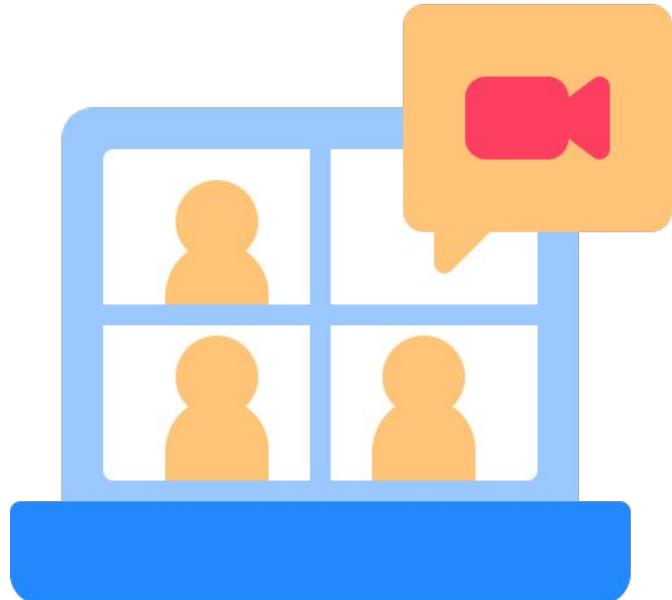


Announcements & Reminders

Don't Forget Cameras On!

You will be removed from today's session if you are not on camera and are unresponsive.

You will be unable to return to today's session if you are removed.





Icebreaker

5 minutes

- What is one goal you have this Summer?
 - Not accepted: get a job/internship 😊

Let's Create a Vision Board in Padlet!

<https://padlet.com/allisonchan6/goal-showcase-1jc3zogo161qttz5>



Your Feedback



Glow

- + BREAK OUT
ROOOOOMMM!!!!
- + Meeting new people
- + Python
- + Working on and solving problems



Grows

- Unclear breakout room instructions
- Repeat of orientation
- UMPIRE was not explicitly introduced
- More time in breakout room
- Python
- More TF presence

Choosing Your Breakout Room

1. We'll share the link to the entry ticket so you can indicate what unit you are on.
2. We'll create breakout groups based on the unit you selected.
3. When it's Collaborative Problem Solving time, we'll share the room assignments and demonstrate how to get to your room.



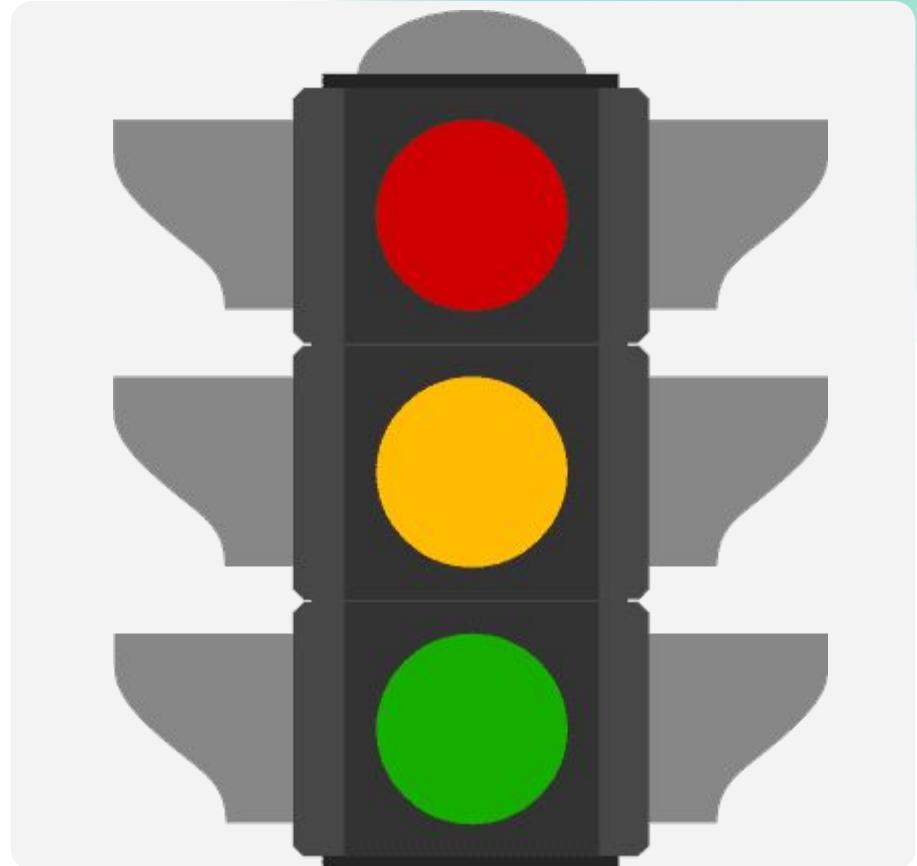
Traffic Lights

Periodically, I'll ask you to enter your traffic light in the chat

GREEN – Keep going!

YELLOW – Proceed Slowly

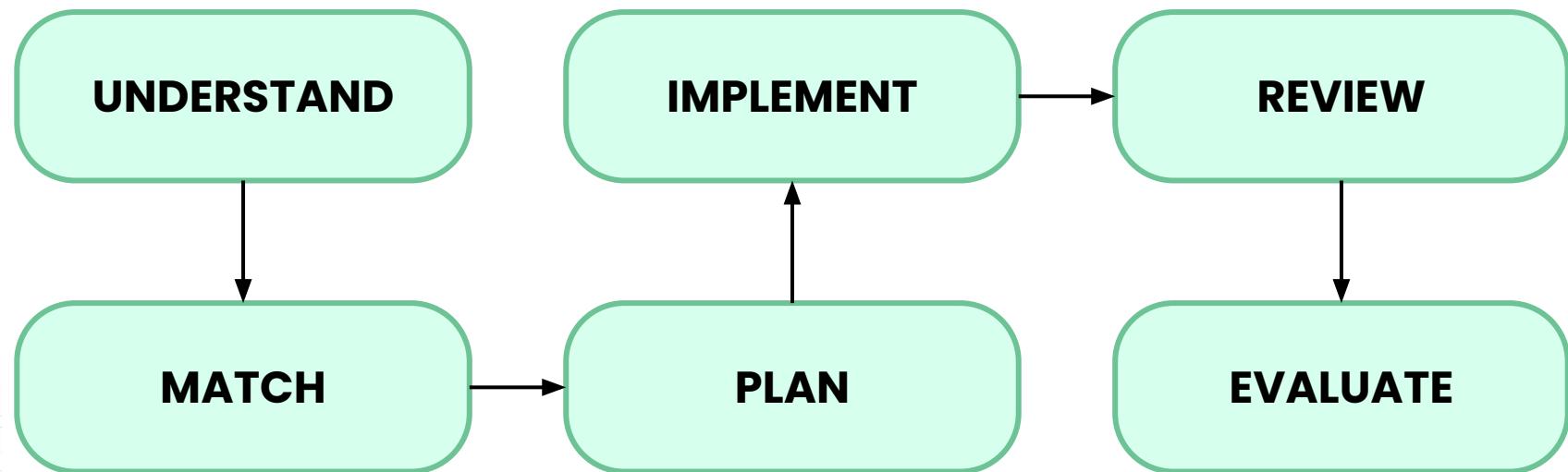
RED – Stop!



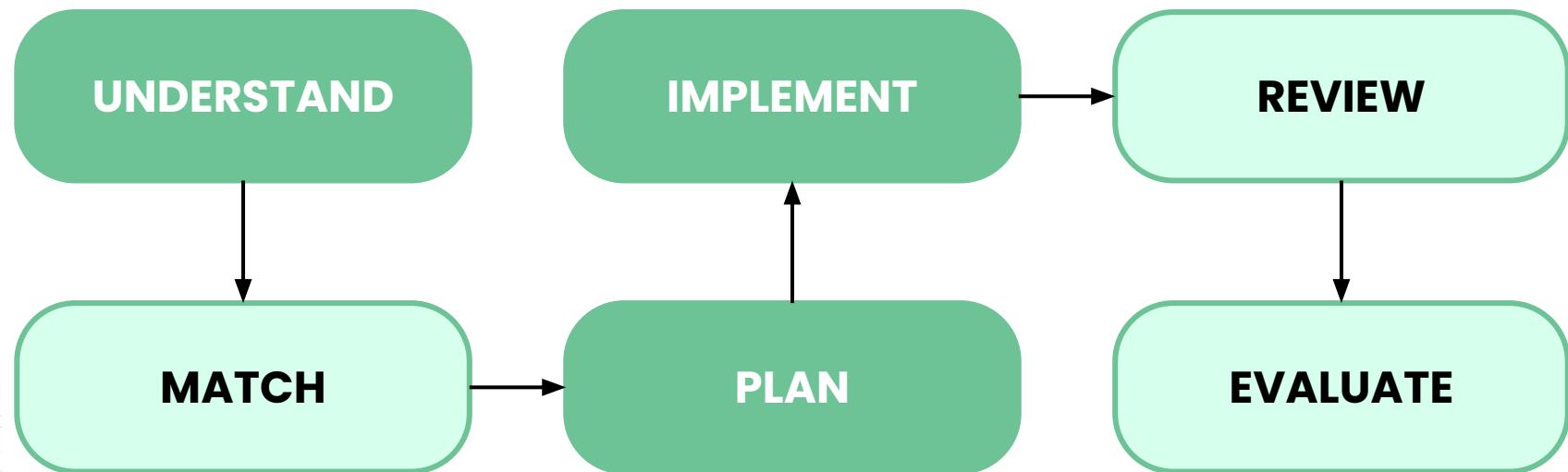
30 minutes

The UMPIRE Strategy

UMPIRE



UMPIRE



Understand the Problem

- * Understand what the interviewer is asking with clarifying questions
- * Generate basic sample input and output
- * Explore more unusual edge cases
- * Explore tradeoffs (memory, performance, code simplicity)



Plan the Solution

- * Describe your overall approach in one or two sentences
- * Write down the steps in English
- * Each step should be simple and clear



Implement the Solution

- * Translate your plan into code.
 - A clear plan should be relatively easy to translate. If it's hard to translate, the plan probably isn't good enough.
- * Look up any basic Python that you need.

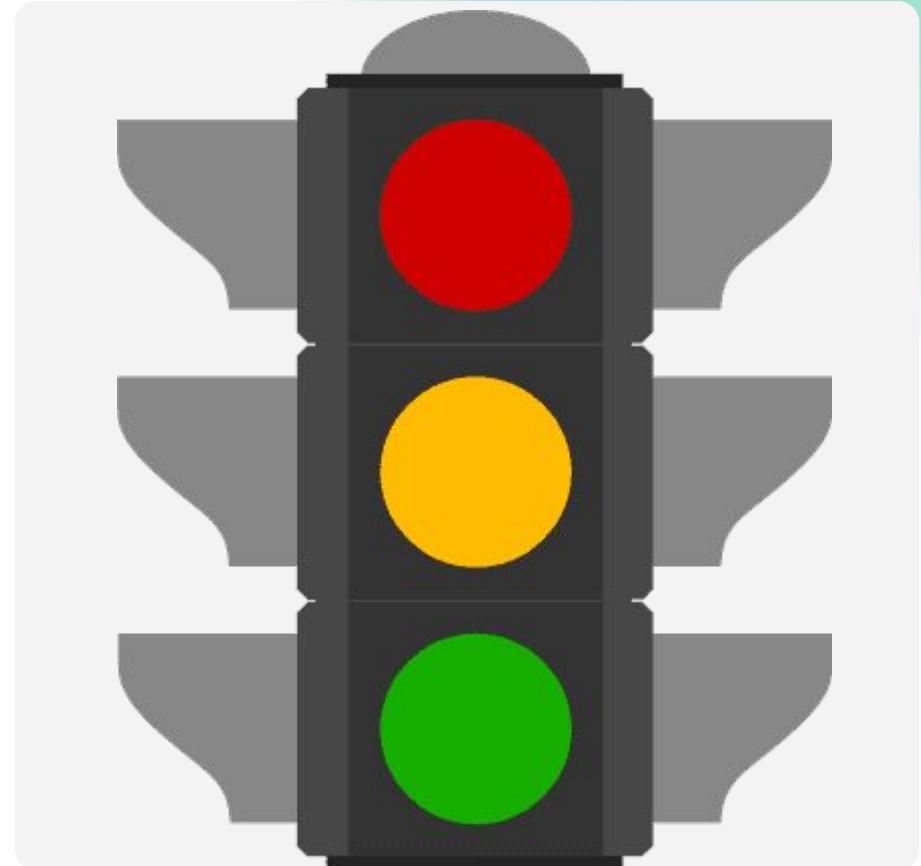


Traffic Lights

GREEN – Keep going!

YELLOW – Proceed Slowly

RED – Stop!

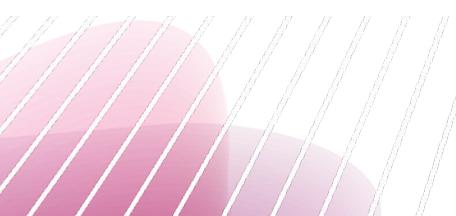




Instructor Demo

20 minutes

Let's see how we can use UPI to help us solve a problem.



Problem 7: NaNaNa Batman!

Write a function `nanana_batman()` that accepts an integer `x` and prints the string "nanana batman!" where "na" is repeated `x` times. Do not use the `*` operator.

```
def nanana_batman(x):  
    pass
```

Input: `x = 6`

Output: "nananananana batman!"

Input: `x = 0`

Output: "batman!"



```
1 """
2 Write a function nanana_batman() that accepts an integer x and prints the string
3 "nanana batman!" where "na" is repeated x times. Do not use the * operator.
4
5 UNDERSTAND:
6 1. input & output
7 2. edge cases
8 3. run time
9 4. anything you're not allowed to use? e.g. reverse string
10 5. happy case
11
12 input: integer value
13 output: not returning (void), BUT we printing out a string
14
15 edge cases:
16 - no input, null, None
17 - negative
18 - a very large number
19 - float
20 - 0
21 => only positive whole numbers including 0
22
23 run and space constraints? no
24
25 - can't use the * operator
26
27 happy case:
28 5 --> "nanananana batman!"
29 0 --> "batman!"
30
31
32 PLAN:
33 1. create a variable to hold our answer
34 2. iterate through a loop x (input) times (while, need a counter)
35 | 2a. add "na" to our answer
36 | 2b. inc counter
37 3. check if answer len > 0
38 | 3a. if yes, add " batman!"
39 | 3b. else, add "batman!"
40 4. print answer
41
42 """
43
44 def nanana_batman(x):
45     result = "" #step 1
46     i = 0
47     while i < x: #step 2
48         result += "na"
49         i += 1
50     if len(result) > 0: #step 3
51         result += " batman!"
52     else:
53         result += "batman!"
54     print(result) #step 4
```

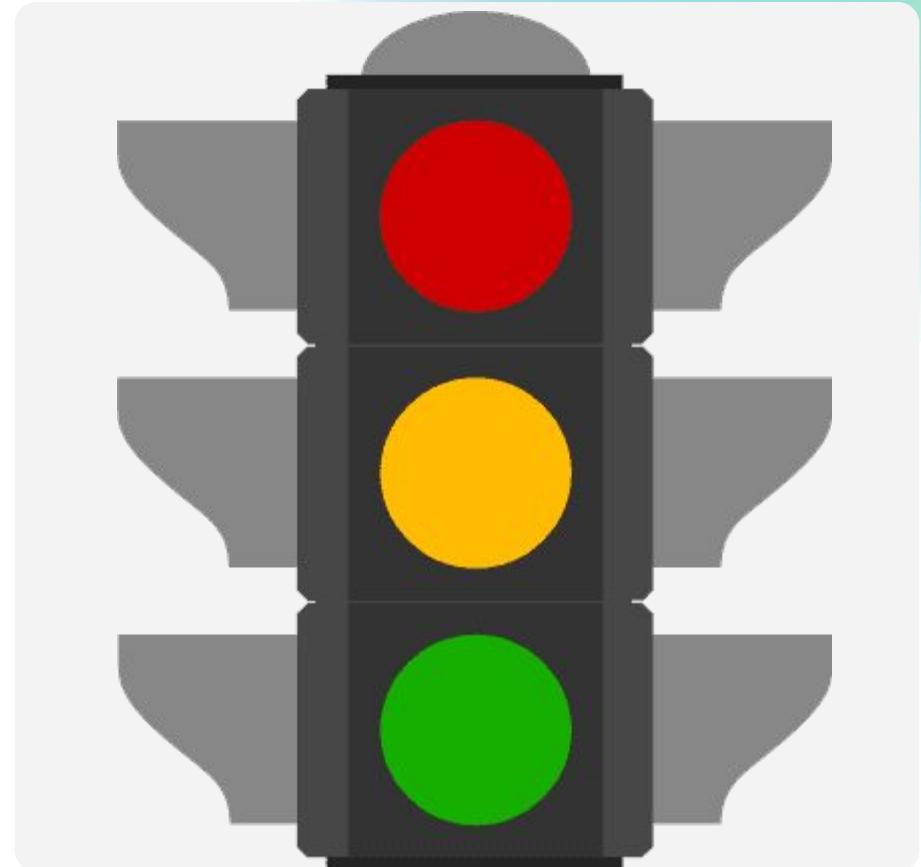


Traffic Lights

GREEN – Keep going!

YELLOW – Proceed Slowly

RED – Stop!



Break Time!

Take 5 mins to step away from the computer. Feel free to turn off your camera and return promptly after 5 mins.



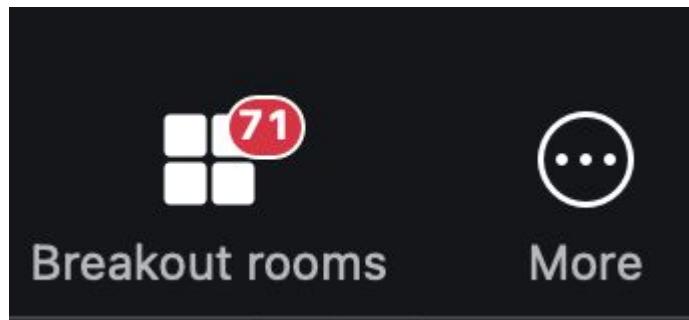
05:00

60 minutes

Collaborative Problem Solving

Choosing Your Breakout Room

1. We'll share the room assignments.
2. We'll demonstrate how to find your room assignment and navigate to your room.





Breakout Rooms

50 minutes

1. Navigate to your breakout room.
2. Introduce yourselves!
3. Work on the problems for your selected unit.
 - a. Pair programming
 - b. Driver
 - c. Interview setting

10 minutes

Wrap Up



One Good Thing

Let's reflect on today's session. Maybe you got a question answered, overcame a specific coding challenge, or learned how to do something completely new!

In the chat, share at least one good thing you've learned or experienced during the session today.

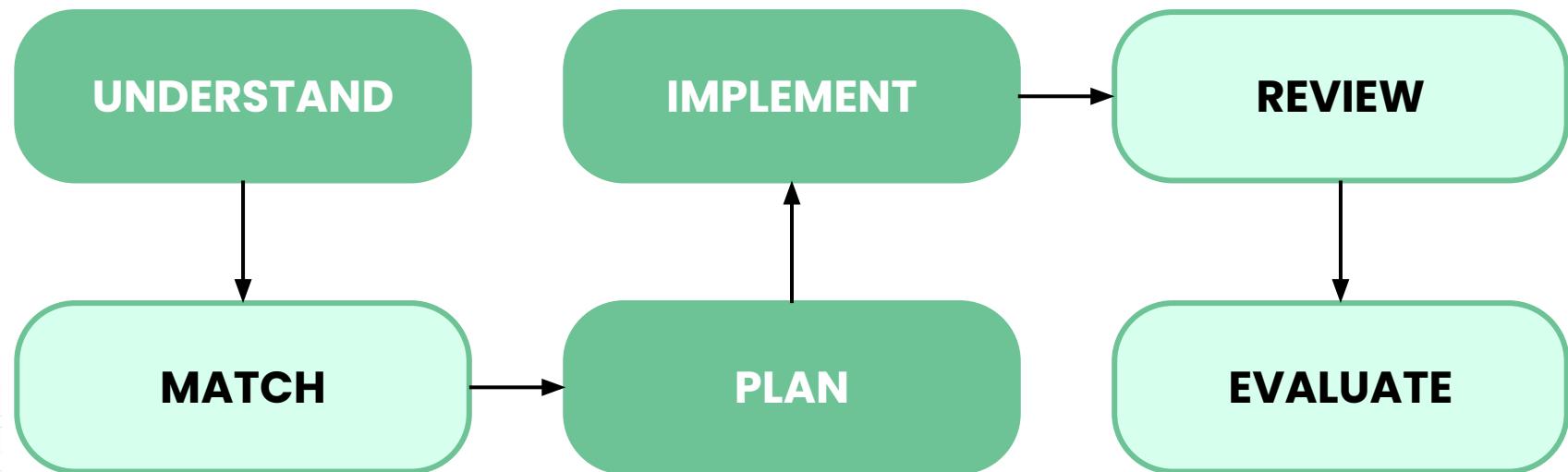
Let's celebrate together!



Session Survey Time!



UMPIRE



UMPIRE

The **U-M-P** section above should take 50-70% of your interview time !

These steps are the most crucial in having BOTH you AND the interviewer to be in agreement. It will show that you know how to approach a problem from all angles. Nothing should be assumed.

Many people skip directly to implementation step which shys away the opportunity to demonstrate problem solving skills. While writing code is important, HOW you think through the problem is what truly sets you apart.



Final Reminders

- Finish the problem sets for additional practice
- Take the weekly HackerRank assessment by Sunday @ midnight

Thank you for a GREAT first week :D