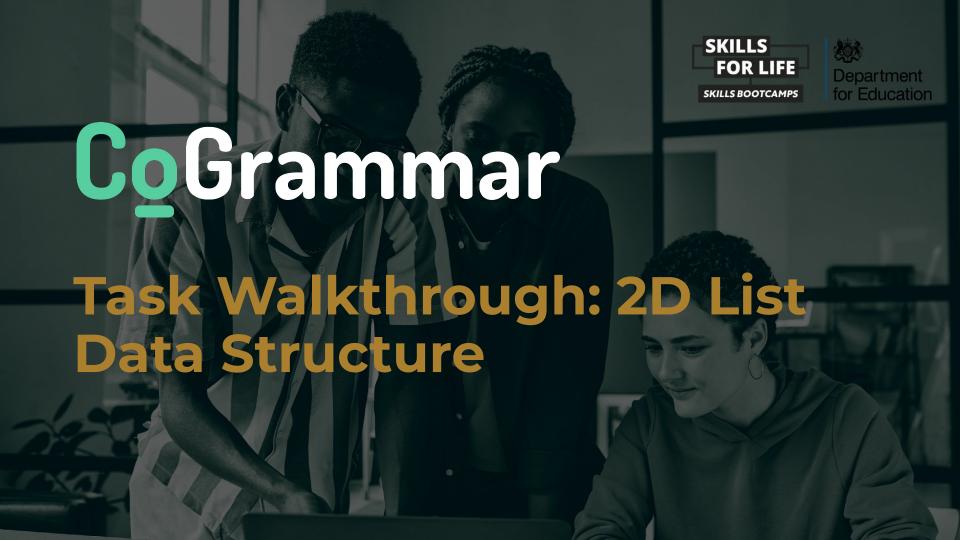
Welcome to this CoGrammar Task Walkthrough: Task 12

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.





Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
 (Fundamental British Values: Mutual Respect and Tolerance)
- No question is daft or silly ask them!
- There are **Q&A sessions** throughout this session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: <u>Questions</u>



Software Engineering Session Housekeeping cont.

- For all non-academic questions, please submit a query:
 www.hyperiondev.com/support
- Report a safeguarding incident:
 <u>www.hyperiondev.com/safeguardreporting</u>
- We would love your **feedback** on lectures: **Feedback on Lectures**

Enhancing Accessibility: Activate Browser Captions

Why Enable Browser Captions?

- Captions provide real-time text for spoken content, ensuring inclusivity.
- Ideal for individuals in noisy or quiet environments or for those with hearing impairments.

How to Activate Captions:

1. YouTube or Video Players:

Look for the CC (Closed Captions) icon and click to enable.

2. Browser Settings:

- Google Chrome: Go to Settings > Accessibility > Live Captions and toggle ON.
- Edge: Enable captions in Settings > Accessibility.



Skills Bootcamp Progression Overview

Criterion 1 - Initial Requirements

Specific achievements within the first two weeks of the program.

To meet this criterion, students need to, by no later than 01 December 2024:

- **Guided Learning Hours** (GLH): Attend a minimum of 7-8 GLH per week (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- Task Completion: Successfully complete the first 4 of the assigned tasks.

✓ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks within the first half of the program.

To meet this criterion, students should, by no later than 12 January 2025:

- Guided Learning Hours (GLH): Complete at least 60 GLH.
- Task Completion: Successfully complete the first 13 of the assigned tasks.



Skills Bootcamp Progression Overview

 \mathbf{V} Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- Guided Learning Hours (GLH): Complete the total minimum required GLH, by the support end date.
- Task Completion: Complete all mandatory tasks, including any necessary resubmissions, by the end of the bootcamp, 09 March 2025.

Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- Record an Interview Invite: Students are required to record proof of invitation to an interview by 30 March 2025.
 - South Holland Students are required to proof and interview by 17 March 2025.
- **Record a Final Job Outcome :** Within 12 weeks post-graduation, students are required to record a job outcome.



Walkthrough





D Lists - fundamental characteristics

- 2D lists are nested list structures meaning a list where each nested item within the list is another list.
- The structure can be expanded into 3 or even 4D nested structures!
- The syntax is similar to a standard list with the difference being duplicated sets of brackets for each nested list.
- Nested lists are commonly used in software engineering to store large amounts of data in runtime memory.



Auto-graded task

Now it's time to see whether you're ready to apply what you've learned to some coding of your own! This is a challenging task, but worth persisting through as you'll gain valuable experience with 2D lists and nested loops.

- 1. Create a file named minesweeper.py.
- 2. Create a function that takes a grid of # and -, where each hash (#) represents a mine and each dash (-) represents a mine-free spot.
- Return a grid where each dash is replaced by a digit, indicating the number of mines immediately adjacent to the spot, i.e., horizontally, vertically, and diagonally.

Example of an input:

```
[ ["-", "-", "-", "#", "#"],
        ["-", "#", "-", "-", "-"],
        ["-", "#", "#", "-", "-"],
        ["-", "#", "#", "-", "-"]]
```

Example of the expected output:

```
[ [1, 1, 2, "#", "#"],
        [1, "#", 3, 3, 2],
        [2, 4, "#", 2, 0],
        [1, "#", "#", 2, 0],
        [1, 2, 2, 1, 0] ]
```



Steps to follow

- Use a nested for loop to navigate through each nested list position.
- Use indexing to access each adjacent position to your current position.
- Use validating conditions to avoid incurring indexing errors.



Navigating Nested Structures Trace Tables

- Navigating a nested structure can be confusing since you will need to implement a nested for loop.
- Trace tables are invaluable tools if you are struggling to visualize the logical steps with each iteration.
- Trace table provide full transparency of each value through each iteration.



Trace table example

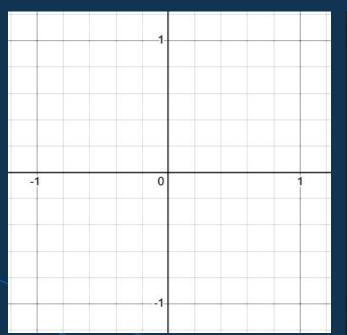
matrix = [Step	i	j	Matrix[i][j]	Operation/Comment
	# Column 0, 1, 2	0	0	0	1	i = 0, j = 0, print Matrix[0][0]
	[1, 2, 3], # Row 0	1	0	1	2	j increments, Matrix[0][1]
	[4, 5, 6], # Row 1	2	0	2	3	j increments, Matrix[0][2]
	[7, 8, 9], # Row 2	3	1	0	4	i increments, j = 0, Matrix[1][0]
]		4	1	1	5	j increments, Matrix[1][1]
		5	1	2	6	j increments, Matrix[1][2]
for	i in range(3):	6	2	0	7	i increments, j = 0, Matrix[2][0]
	for j in range(3):	7	2	1	8	j increments, Matrix[2][1]
C	<pre>print(matrix[i][j])</pre>		2	2	9	j increments, Matrix[2][2]

Coordinates - visualizing a graph

- Once you wrap your head around navigating the nested structure how can we modify our current position to view any adjacent positions?
- This can be challenging for beginners. A good analogy is using a simple X, Y graph.
- As the X axis moves up (north) the value of x decreases. Moving down (south) causes the value of x to increase.
- Similarly moving left (west) on the y axis causes y to decrease and moving left (east) causes y to increase.



Comparison: nested list vs x, y axis



```
nested_list = [
   # Column 0, 1, 2
    [1, 2, 3], # Row 0
    [4, 5, 6], # Row 1
    [7, 8, 9], # Row 2
```

Finding your grid boundaries

- Each nested structure has index limits. A grid index will always start at 0 and end at len(grid).
- When using indexing within a nested structure it is important to consider indexing errors that may affect your runtime.
- Consider your index limits and implement validating if conditions to ignore any index values that exceed your grid limits.



Questions and Answers



Thank you for attending







