Welcome to this CoGrammar Tutorial:

Problem-Solving Practice (Pseudocode, Flowcharts, Data Structures)

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: **Questions**

Software Engineering Session Housekeeping cont.

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

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

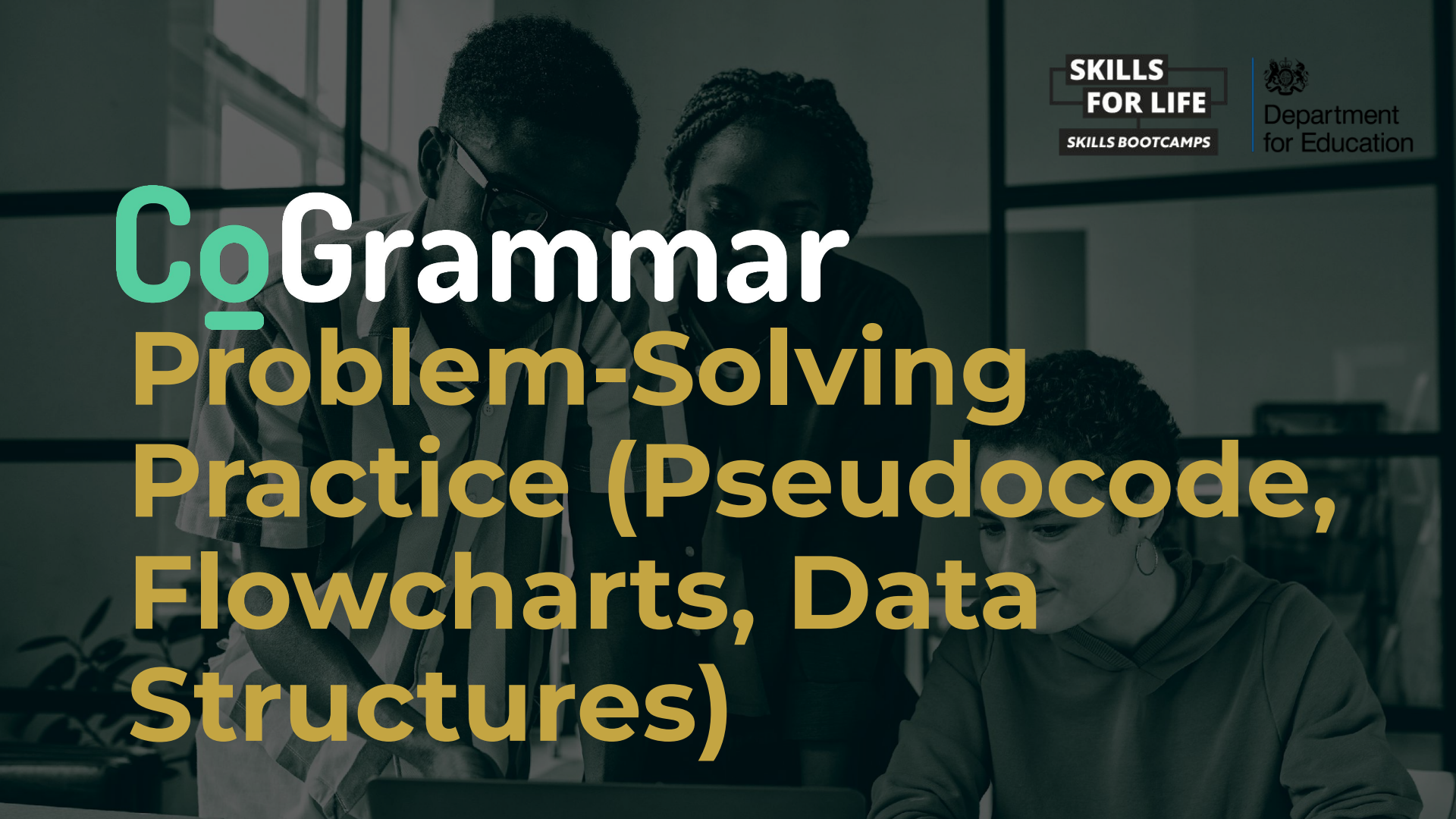
Enhancing Accessibility: Activate Browser Captions

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 FOR LIFE
SKILLS BOOTCAMPS
Department for Education
CoGrammar
Problem-Solving Practice (Pseudocode, Flowcharts, Data Structures)

Learning Objectives & Outcomes

- Define flowcharts and their purpose in problem-solving.
- Identify the basic symbols used in flowcharting.
- Create simple flowcharts to represent algorithms.
- Explain the concept of data structures.
- Recognize the importance of data structures in efficient problem-solving.
- Identify common data structures: arrays, dictionaries, stacks, and queues.

Quote

Bad programmers worry about the code. Good programmers worry about data structures and their relationships....

Linus Torvalds

Introduction

CoGrammar

Introduction

- **Pseudocode** as a planning tool for algorithms.
- **Data Structures** for organizing and managing data efficiently.
- **Problem Solving** techniques involving breaking problems into manageable steps and designing structured solutions.

Basic Data Structures

CoGrammar

What Are Data Structures?

- What Are Data Structures?
  - Definition: Organized ways to *store* and *manage* data efficiently.
  - Purpose:us to access, organize, and modify information easily.
- Why Data Structures Matter?
  - Enables efficient handling of data.
  - Optimizes resource usage in applications.
  - Forms the foundation for advanced algorithms.

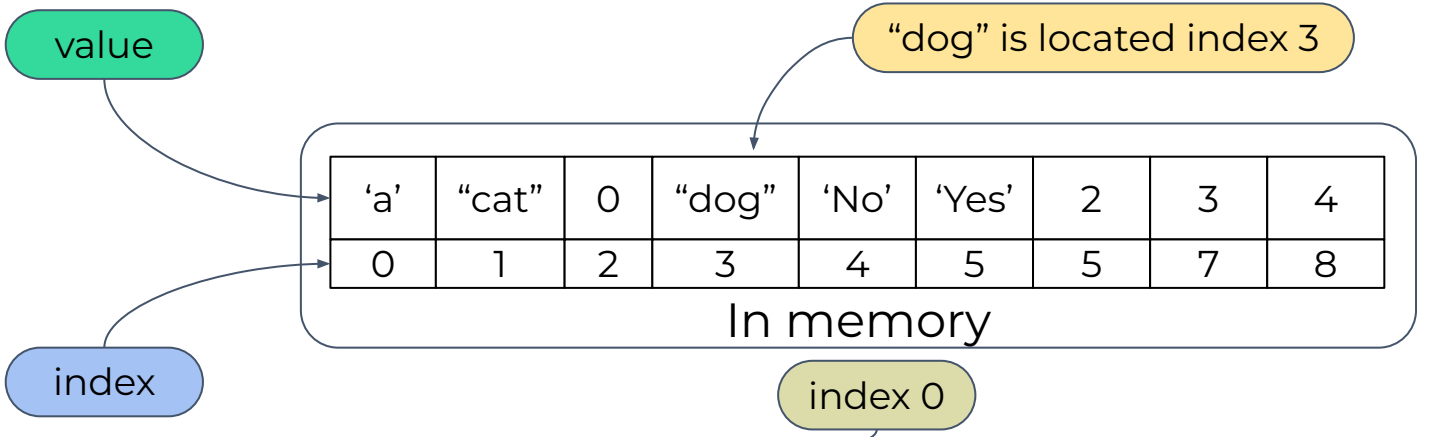
Basic Data Structures

CoGrammar

Arrays/Lists

- Definition: An ordered collection of elements.
- Key Features:
  - Indexing: Direct access to elements using indices (e.g., arr[0] for the first element).
  - Contiguity: Stored in adjacent memory, enabling fast access.
  - Efficiency: Best for scenarios needing frequent reads or updates at specific indices.
- Use Case: Suitable for applications requiring sequential data storage and indexed access.

Arrays/Lists



list_items = [‘a’, “cat”, 0, “dog”, ‘No’, ‘Yes’, 2, 3, 4]

index 8

Retrieval: list_items[3] is “dog”

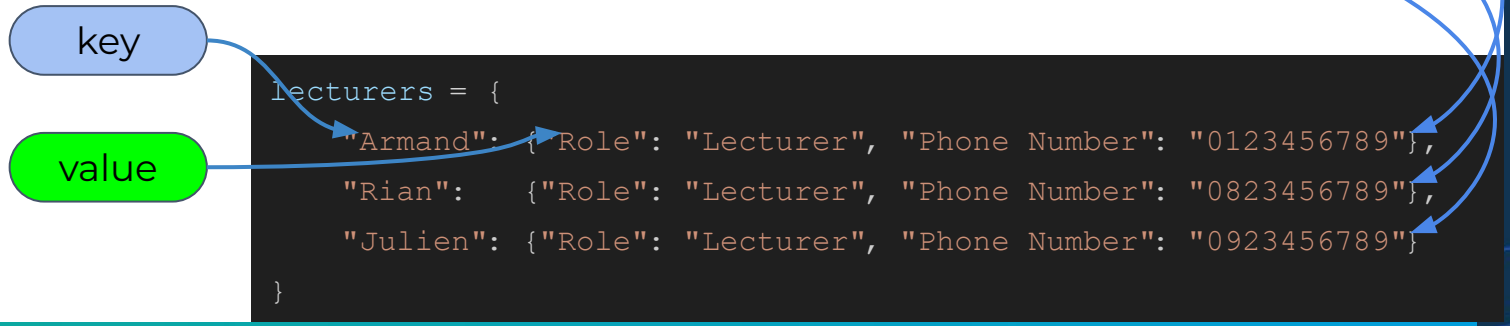
Retrieval: list_items[1] is “cat”

Key Data Structures: Dictionaries

- Definition: Sometimes called **hashmaps**, is a collection of key-value pairs for efficient data lookup and storage
- Key Features:
  - Keys: Unique identifiers for accessing values
    - Example: 'name' in {**'name'**: 'Alice'}.
  - Values: Data associated with each key
    - Example: 'Alice' in {'name': **'Alice'**}.
  - Flexibility: Allows different types of keys and values.
  - Efficiency: Provides fast lookups using.
- Example Use Case:
  - You look up a word (the key) to find its meaning (the value).

Dictionaries/Hashmaps

Name	Role	Phone Number
Armand	Lecturer	0123456789
Rian	Lecturer	0823456789
Julien	Lecturer	0923456789



Stacks

- Definition: A data structure following the **Last In, First Out (LIFO)** principle, where the **last** element added is the first to be removed.
- Key Features:
  - Operations:
    - Push: Adds an element to the top.
    - Pop: Removes the top element.
    - Peek: Retrieves the top element without removing it.
  - Access: Restricted to the top element only.
  - Efficiency: Operations are quick for both push and pop.

Queues

- Definition: A data structure following the **First In, First Out (FIFO)** principle, where the **first** element added is the first to be removed.
- Key Features:
  - Operations:
    - Enqueue: Adds an element to the rear.
    - Dequeue: Removes the front element.
    - Peek: Retrieves the front element without removing it.
  - Access: Restricted to the front and rear elements.
  - Efficiency: Operations are quick for enqueue and dequeue.

Operations that Shape Data Structures

CoGrammar

Insertion

- Purpose: Adding new elements to the data structure.
- Examples:
  - Operations:
    - Arrays: `arr.append(6)`.
    - Stacks: `stack.push(5)`.
    - Queues: `queue.enqueue(3)`.

Deletion

- Purpose: Removing elements from the data structure.
- Examples:
  - Operations:
    - Arrays: `arr.remove(6).`
    - Stacks: `stack.pop()`.
    - Queues: `queue.dequeue()`.

Update

- Purpose: Modifying existing elements present in the data structure.
- Examples:
  - Operations:
    - Arrays: `arr[2] = 10.`
    - Dictionaries: `dict['key'] = 'new_value'`.

Understanding and Writing Pseudocode

CoGrammar

Definition and Importance

- What is Pseudocode?
  - A simple, plain-language way to describe step-by-step solutions.
  - Focuses on logic, not coding or syntax.
- Why Use Pseudocode?
  - Clarity: Breaks problems into clear, manageable steps.
  - Communication: Helps explain ideas to others without using code.
  - Planning: Creates a roadmap for problem-solving.

Key Features of Pseudocode

- Core Characteristics:
  - **Simple Language:** Uses structured yet plain terms like `IF`, `WHILE`, and `FOR`.
  - **Focus on Flow:** Prioritizes the logical steps over programming syntax.
  - **Readable:** Easily understood by programmers and non-programmers alike.
- Best Practices:
  - Keep it concise but clear.
  - Use consistent indentation for readability.
  - Avoid implementation details specific to a programming language.

Example: Finding the Largest Number in a List

```
SET max = first number in list
FOR each number in list
    IF number > max THEN
        SET max = number
END FOR
PRINT max
```

Key Steps:

1. Initialize the `max` variable.
2. Iterate through the list.
3. Update `max` when a larger number is found.
4. Print the largest number.

Practical Exercises

CoGrammar

Find the Smallest Number in a List:

Find the Smallest Number in a List:

```
SET list_items = set of numbers
SET min = first number in list
FOR each number in list
    IF number < min THEN
        SET min = number
END FOR
PRINT min
```

Calculate the Sum of Numbers in a List

Calculate the Sum of Numbers in a List

```
SET summ = 0
FOR each number in list
        Summ = summ + number
END FOR
PRINT count
```

Reverse a List

Reverse a List

```
SET reversed_list = []
FOR i FROM length of list - 1 DOWN TO 0
    APPEND list[i] to reversed_list
END FOR
PRINT reversed_list
```

Count Even Numbers in a List

Count Even Numbers in a List

```
SET count = 0
FOR each number in list
    IF number MOD 2 == 0 THEN
        INCREMENT count by 1
END FOR
PRINT count
```

Count Occurrences of a Number in a List

Count Even Numbers in a List

```
INPUT target_number
SET count = 0
FOR each number in list
    IF number == target_number THEN
        INCREMENT count by 1
END FOR
PRINT count
```

CoGrammar

Remove Duplicates from a List

Remove Duplicates from a List

```
SET unique_list = []
FOR each number in list
    IF number NOT IN unique_list THEN
        APPEND number to unique_list
END FOR
PRINT unique_list
```

Lesson Conclusion and Recap

Recap the key concepts and techniques covered during the lesson.

- Pseudocode simplifies problem-solving and planning.
- Data structures improve efficiency in data management and algorithms.
- Practice reinforces understanding.

Resources

Resources

- Online Flowchart Tools:
  - draw.io
  - Lucidchart
- Data Structure Visualization Websites:
  - VisuAlgo
  - Printables - CS Unplugged

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar