Welcome to this CoGrammar Q&A: Sequences, Functions and Debugging

The session will start shortly...

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

CoGrammar

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

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*.

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

Polls

CoGrammar

Poll

1. **Which feature of Python sequences do you find most useful or fascinating?**
    A. Indexing and Slicing: Accessing and modifying parts of a sequence.
    B. Mutable vs Immutable: Lists can change, but tuples can't.
    C. Comprehensions: Creating sequences in a simple, readable way.
    D. Negative Indexing: Accessing elements from the end of a sequence.
    E. Iterators: Efficiently looping through sequences.

CoGrammar

Poll

2. **Which concept in Python functions do you find most challenging or fascinating?**

   A. Scope: How variables work inside and outside functions.

   B. Higher-Order Functions: Functions that use or return other functions.

   C. Recursion: Functions that call themselves.

   D. Lambda Functions: Quick, anonymous functions.

   E. Decorators: Changing a function's behaviour without altering its code.

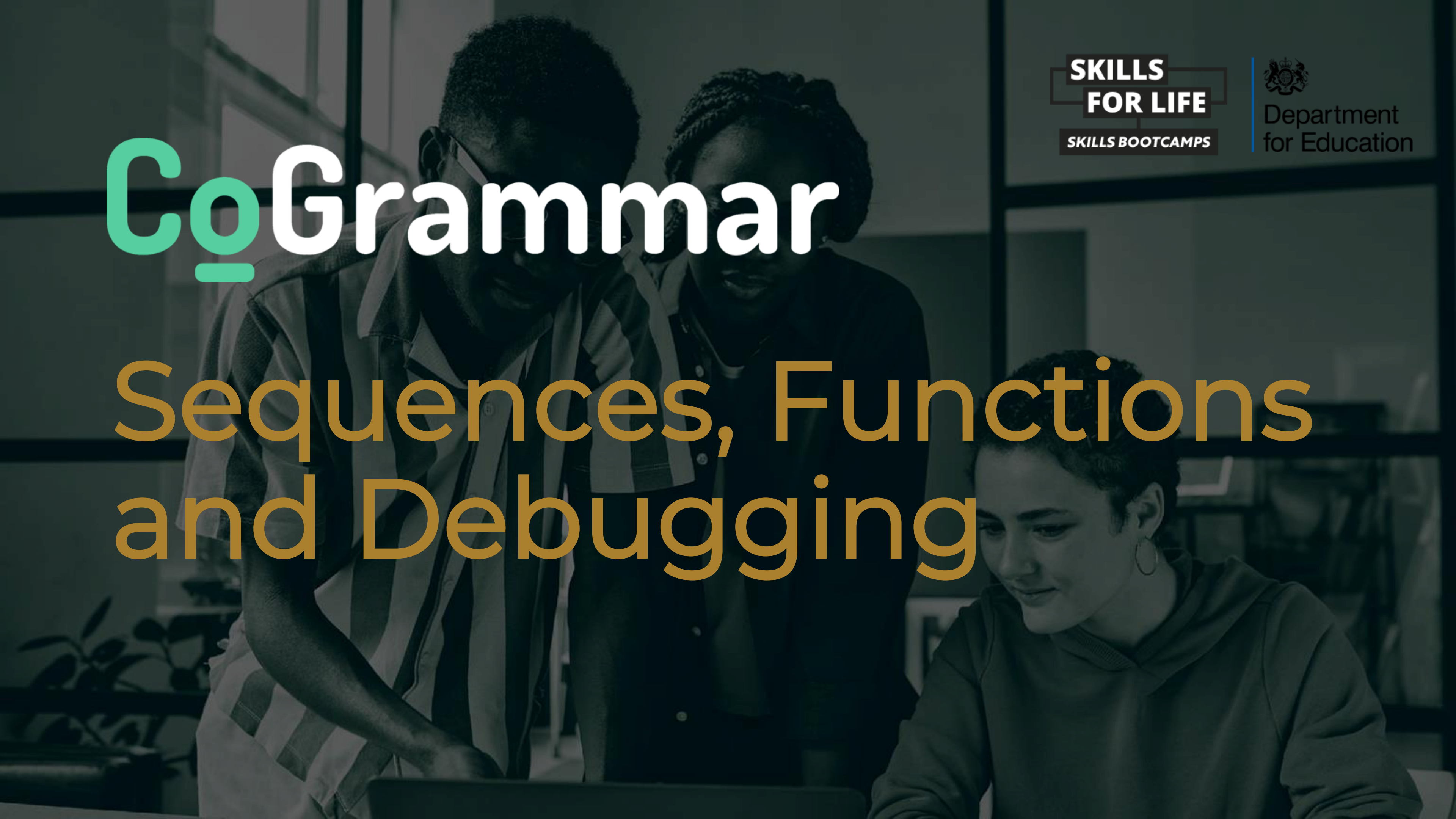
CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Sequences, Functions and Debugging



Learning Outcomes

- Perform basic string operations such as concatenation, slicing, and formatting.
- Use built-in string methods to manipulate and analyse text data.
- Perform basic list operations such as indexing, slicing, appending, and removing elements.
- Use list methods and functions to manipulate and process list data.
- Perform basic dictionary operations such as adding, updating, and removing key-value pairs.

CoGrammar

Learning Outcomes

- Use dictionary methods to access and manipulate data efficiently.
- Perform basic operations on 2D lists such as accessing, modifying, and iterating over elements.
- Define and call functions with parameters and return values.
- Implement functions to modularise and organise code effectively.
- Apply scope rules to avoid common errors related to variable access and modification.
- Interpret stack traces to debug and identify the source of errors in their code.

CoGrammar

Handling Strings

CoGrammar

What is String Handling?

- String handling involves working with sequences of characters (letters, numbers, symbols) to manage and manipulate text data.
- It’s essential for tasks like text processing, data parsing, input validation, and creating formatted output.

String Handling Recap

- In Python, strings are created by enclosing characters in single (' ') or double (" ") quotes.
- Concatenation: Combine strings using the + operator.
- String Formatting: Use methods like format() or f-strings for more flexible formatting.
- Indexing: Access individual characters using square brackets [] with an index number.

CoGrammar

String Handling Recap

- Python also provides useful built-in methods for string manipulation:
    - len(): Get the string’s length.
    - upper(), lower(): Convert to uppercase or lowercase.
    - strip(): Remove leading and trailing whitespace.
    - split(): Split a string into a list based on a delimiter.
    - join(): Combine list elements into a string with a specified delimiter.

Lists & Dictionaries

CoGrammar

What are Lists?

- Lists are a key data structure in Python, used for storing and manipulating data.
- They are versatile and essential for tasks ranging from simple storage to complex data analysis.
- Mastering list manipulation is crucial for effective Python programming.

Manipulating Lists

- Creating Lists: Use square brackets [] to create a list with items separated by commas.
- Accessing Elements: Access items using zero-based indexing (starting from 0).
- Slicing: Extract a portion of a list by specifying start and end indices.
- Modifying Lists: Lists are mutable, so their elements can be changed after creation.

Manipulating Lists cont.

- **Adding Elements:** You can append new elements to the end of a list using the append() method or insert them at a specific position using the insert() method.
- **Removing Elements:** You can remove elements from a list using methods like remove(), pop(), or del.
- **List Methods:** Python provides many built-in methods for working with lists, such as sort(), reverse(), count(), and index().

CoGrammar

Dictionaries

- A dictionary stores pairs of keys and their values.
- You access values using their corresponding keys.
- Dictionaries are great for representing structured data, like user info or settings.

CoGrammar

Manipulating Dictionaries

- Accessing Elements: Get values using their keys.
- Modifying Dictionaries: You can change values, add new pairs, or remove existing ones.
- Dictionary Methods: Common methods include keys(), values(), items(), get(), pop(), and update().

Functions & Scope

CoGrammar

Functions Recap

- In Python, a function is a reusable block of code that performs a specific task.
- There are two main types of functions: built-in functions and user-defined functions.
- **Built-in functions:** Functions provided by Python, like print(), len(), and input(), which you can use without defining them yourself.
- **User-defined functions:** Functions you create using the def keyword, giving them a name, parameters, and code to perform specific tasks.

CoGrammar

Functions Recap

- Further, we also get functions that can be imported.
- **Standard library functions:** These modules are part of Python's built-in distribution, and you can import them to use the provided functions, like import math for math functions.
- **Third-Party Libraries:** These libraries need to be installed before they can be imported and used, like tabulate or numpy for numerical operations.
  - Download and then use pip install before importing the module.
  - Visit pypi.org and search for the function you need.

CoGrammar

Scope

Scope determines where and how you can access variables in your code:

- **Global Scope**: Variables defined outside functions can be accessed anywhere in the code, including inside functions.
- **Local Scope**: Variables defined inside a function can only be accessed within that function.

Stack Traces & Debugging

CoGrammar

Stack Traces

- A stack trace is like a map that shows the sequence of function calls leading to an error in your Python code.
- It helps you trace what went wrong, starting from the main program and going through the functions or methods involved.

Debugging

- Stack traces are invaluable for debugging, as they pinpoint the cause and location of errors in your code.
- Debugging is the process of identifying and fixing these errors, much like solving a mystery.
- Debugging is a crucial skill for programmers, ensuring their software works correctly and reliably.

CoGrammar

Let’s take a short break

CoGrammar

Demo Time!

CoGrammar

Conclusion and Recap

In this lesson, we explored the importance and application of functions and sequences in Python programming.

- Importance of Functions:

  Purpose: Encapsulate code into reusable blocks.

  Benefits: Enhance code organisation, maintainability, and error management.

- Importance of Sequences:

  Types: Lists, tuples, and strings.

  Uses: Store and manipulate ordered collections of data.

Conclusion and Recap

- Combining functions with sequences allows for flexible and dynamic data handling.
- Key operations include searching, sorting, and transforming data.

Resources

- Official Python Documentation - Functions:
  - https://docs.python.org/3/tutorial/controlflow.html#defining-functions
- Online Tutorials:
  - https://realpython.com/defining-your-own-python-function/
  - https://www.w3schools.com/python/python_functions.asp
- Additional Reading: "Automate the Boring Stuff with Python" by Al Sweigart
- Python Official Documentation - Data Structures:
  - https://docs.python.org/3/tutorial/datastructures.html
- Online Tutorials:
  - https://realpython.com/python-sequences/

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending