Welcome to this CoGrammar Task Walkthrough: Task 7

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

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.

CoGrammar

Skills Bootcamp Progression Overview

✓ 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.

CoGrammar

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

CoGrammar

Task Walkthrough: Handling Strings, Lists and Dictionaries

Learning Outcomes

- Apply practical string manipulation techniques such as modifying characters in a string.
- Implement dictionaries and lists together to perform calculations such as calculating expected profits.
- Explain the reasoning behind each code block and apply the logic to similar tasks.

Strings, Lists and Dictionaries

- Strings are sequences of characters that are enclosed within single or double quotes. Common operations we perform on strings are concatenation, slicing, and formatting.
- Lists are mutable, ordered collections of items which can be of any type. Lists allow for indexing, slicing, appending, and more.
- Dictionaries are collections of key-value pairs, where each key maps to a value. Dictionaries are unordered and are accessed using keys, not indices.

Strings, Lists and Dictionaries

- There are lots of useful methods we can use on strings, lists and dictionaries, such as:
  - String Methods: split(), join(), replace()
  - List Methods: append(), pop(), sort()
  - Dictionary Methods: get(), keys(), values()

Part 1 Walkthrough

CoGrammar

Follow these steps:

- Create a file called **alternative.py**.
- Write a program that reads in a string and makes each alternate **character** into an uppercase character and each other alternate character a lowercase character.

  E.g.: The string "`Hello World`" would become "`HeLlO WoRlD`"
- Now, try starting with the same string but making each alternative **word** lowercase and uppercase.

  E.g.: The string "`I am learning to code`" would become "`i AM learning TO code`".

  Tip: Using the `split()` and `join()` functions will help.

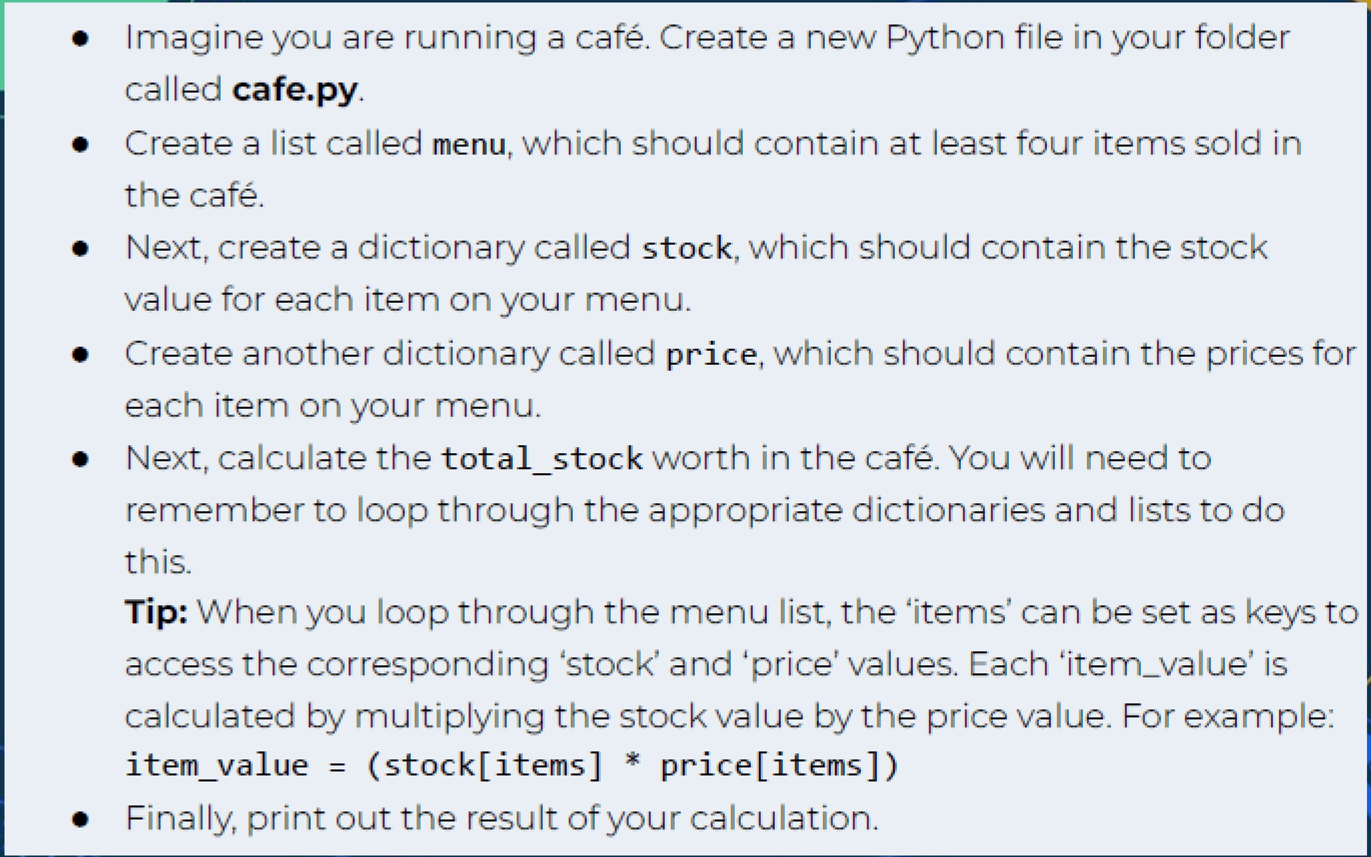
Be sure to place files for submission inside your **task folder** and click "**Request review**" on your dashboard.

alternative.py - Task Objective

The objective of this task is to demonstrate your understanding of string manipulation, for loops, and conditional logic in Python. You will:

- Use loops and conditional statements to dynamically modify a string based on alternating patterns.
- Apply logic to transform individual characters and words into alternating cases.
- Utilise string functions like split() and join() to manipulate and reconstruct text efficiently.

Part 2 Walkthrough



- Imagine you are running a café. Create a new Python file in your folder called **cafe.py**.
- Create a list called **`menu`**, which should contain at least four items sold in the café.
- Next, create a dictionary called **`stock`**, which should contain the stock value for each item on your menu.
- Create another dictionary called **`price`**, which should contain the prices for each item on your menu.
- Next, calculate the **`total_stock`** worth in the café. You will need to remember to loop through the appropriate dictionaries and lists to do this.

  Tip: When you loop through the menu list, the ‘items’ can be set as keys to access the corresponding ‘stock’ and ‘price’ values. Each ‘item_value’ is calculated by multiplying the stock value by the price value. For example:

  ```
  item_value = (stock[items] * price[items])
  ```
- Finally, print out the result of your calculation.

cafe.py - Task Objective

The objective of this task is to demonstrate your ability to work with Python lists, dictionaries, and loops to solve a practical problem. You will:

- Create and manage data structures like lists and dictionaries to represent menu items, stock, and prices in a café.
- Use loops to iterate through these data structures and perform calculations dynamically.
- Apply logic to calculate the total stock value of the café by combining data from multiple dictionaries.

This task is designed to enhance your problem-solving skills and deepen your understanding of data structures, iteration, and basic arithmetic operations in Python.

Questions and Answers

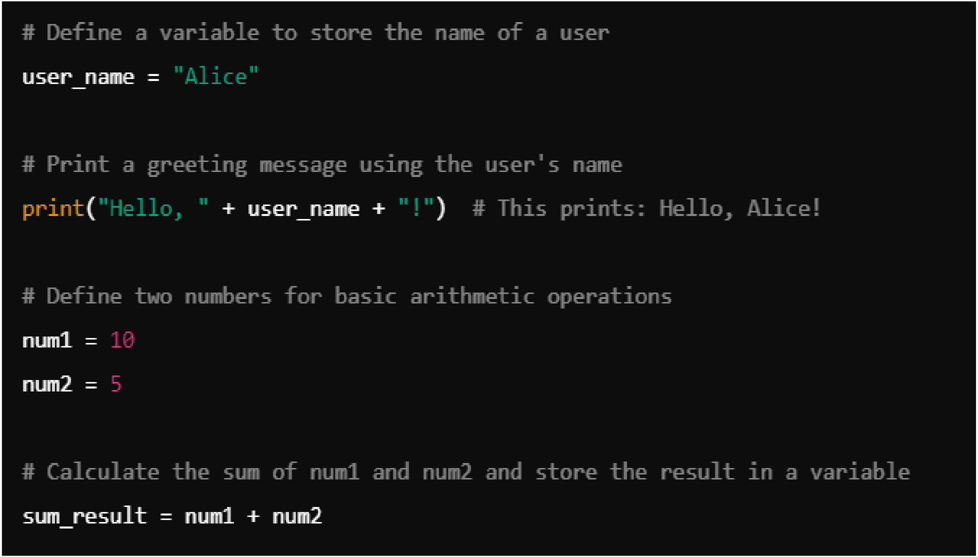
CoGrammar

Documentation and Style

- Add comments to your code. Explain your approach, and/or how your code works.
- Consult the Python PEP8 guidelines: https://peps.python.org/pep-0008/

Pay close attention to:

- Variable names
- Spacing around operators
- Separating logical sections
- Indentation



```
# Define a variable to store the name of a user
user_name = "Alice"

# Print a greeting message using the user's name
print("Hello, " + user_name + "!")  # This prints: Hello, Alice!

# Define two numbers for basic arithmetic operations
num1 = 10
num2 = 5

# Calculate the sum of num1 and num2 and store the result in a variable
sum_result = num1 + num2
```

Learner Challenge

For those who are looking for an additional challenge, expand the functionality of your café program with the following features:

- Dynamic Menu Management:
  - Allow the user to add, remove, or update items in the menu, stock, and price dictionaries dynamically.
  - Implement input validation to ensure that prices are non-negative numbers and stock values are integers.
- Low Stock Alert:
  - Automatically identify items with low stock (e.g., less than 5 units) and print a restocking alert for these items.

Learner Challenge

For those who are looking for an additional challenge, expand the functionality of your café program with the following features:

- Order Processing System:
  - Prompt the user to place an order by entering item names and quantities.
  - Check if the items are available in stock and calculate the total cost of the order.
  - Deduct the ordered quantity from the stock and notify the user if an item is out of stock.

Learner Challenge

For those who are looking for an additional challenge, expand the functionality of your café program with the following features:

- Order Processing System:
  - Prompt the user to place an order by entering item names and quantities.
  - Check if the items are available in stock and calculate the total cost of the order.
  - Deduct the ordered quantity from the stock and notify the user if an item is out of stock.

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar