Welcome to this CoGrammar Task Walkthrough: Task 3

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: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: <u>Feedback on Lectures</u>

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:

- 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

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.

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.



Part 1 Walkthrough





Variables and String Manipulation

- Variables are containers that hold information.
- A string is simply a way to represent text in programming.
- Python identifies strings with the presence of quotation marks
 (" ")
- Strings can be joined, cut up, and measured.
- Built-in methods to manipulate strings



manipulation.py

Task objective

- The objective of this task is to demonstrate your knowledge of string manipulation by working with user input.
- You'll use different string methods to transform and manipulate the user input in specific ways, such as
 - calculating the string's length,
 - replacing characters,
 - slicing, and reversing portions of the string.
- This task is designed to enhance your problem-solving skills and deepen your understanding of how to work with strings.



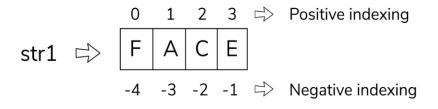
Auto-graded task 1

Follow these steps:

- Create a new Python file for this task and call it manipulation.py.
- Ask the user to enter a sentence using the input() function. Save the user's
 response in a variable called str_manip.
- Explore the string methods in here to help you solve the problem below:
- · Using this string value, write the code to do the following:
 - o Calculate and display the length of str_manip.
 - Find the last letter in str_manip sentence. Replace every occurrence of this letter in str_manip with '@'.
 - e.g. if str_manip = "This is a bunch of words", the output would be: "Thi@ i@ a bunch of word@"
 - o Print the last three characters in str_manip backwards.
 - e.g. if str_manip = "This is a bunch of words", the output would be: "sdr".
 - Create a five-letter word that is made up of the first three characters and the last two characters in str_manip.
 - e.g. if str_manip = "This is a bunch of words", the output would be: "Thids".

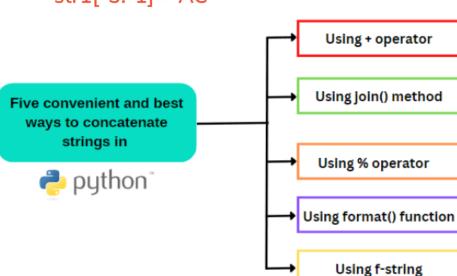
Be sure to place files for submission inside your task folder and click "Request review" on your dashboard.

String Slicing



$$str1[1:3] = AC$$

$$str1[-3:-1] = AC$$



Part 2 Walkthrough



Data Types Recap

- Data types in programming define the type of data a variable can hold and how that data can be used.
- Data types: Integers, Floats, Strings, and Booleans.
- Data types can be converted from one type to another Within reason!
- Arithmetic operations in Python: +, -, /, *, %, **
- Arithmetic built-in functions



numbers.py

Task objective

- The objective of this task is to build confidence and understanding basic arithmetic operations and user interaction.
- You'll perform calculations:
 - add up all the numbers,
 - subtract the second number from the first,
 - multiply the third number by the first,
 - and finally, divide the sum of all three numbers by the third.
- This exercise is designed to help you practice working with numbers and user inputs, giving you an understanding of how to handle basic operations.



Auto-graded task 2

Follow these steps:

- Create a new Python file called numbers.py.
- · Ask the user to enter three different integers.
- Then print out:
 - o The sum of all the numbers
 - o The first number minus the second number
 - The third number multiplied by the first number
 - The sum of all three numbers divided by the third number

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

Part 3 Walkthrough





Conditional Statements Recap

- Conditional statements are like the decision-makers in programming.
 They allow your code to choose different paths based on specific conditions.
- Conditional statements: if, elif, and else
- Comparison operators
- Logical operators: and, or & not

greater than	>
less than	<
equal to	==
not	!
greater than or equal to	>=
less than or equal to	<=
not equal to	1=



award.py

Task objective

- The objective for this task is to enhance your skills by focusing on operators, arithmetic operations, and conditional logic.
- In this task, you will create a program to calculate a triathlon participant's total time from their swimming, cycling, and running events. The program should then determine and display the appropriate award based on the total time.



Auto-graded task 3

Follow these steps:

- Create a new Python file in this folder called award.py.
- Design a program that determines the award a person competing in a triathlon will receive.
- Your program should read in user input for the times (in minutes) for all three events of a triathlon, namely swimming, cycling, and running, and then calculate and display the total time taken to complete the triathlon.
- The award a participant receives is based on the total time taken to complete the triathlon. The qualifying time for awards is any completion time between 0-100 minutes. Display the award that the participant will receive based on the following criteria:

Qualifying criteria	Time range	Award
Within the qualifying time.	0–100 minutes	Honorary colours
5 minutes off from the qualifying time.	101–105 minutes	Honorary half colours
10 minutes off from the qualifying time.	106–110 minutes	Honorary scroll
More than 10 minutes off from the qualifying time.	111+ minutes	No award

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

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
```



Questions and Answers





Thank you for attending







