CoGrammar

Welcome to this session:
Task Walkthrough - Data Types and Conditionals

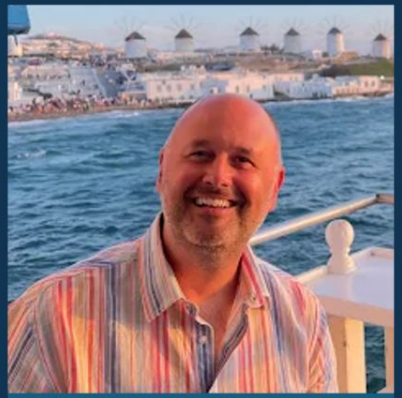
The session will start shortly...

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

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

Skills Bootcamp Data Science

- 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** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- 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**

Skills Bootcamp Data Science

- 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***
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- Manipulate strings using Python string methods to analyze and transform text.
- Perform basic mathematical operations and implement user inputs for computational tasks.
- Apply conditional logic to make decisions and evaluate scenarios programmatically.
- Develop modular programs with reusable functions for solving real-world problems.
- Extend their understanding by combining string and numerical operations in data-driven tasks.

Lecture Overview

- ➔ Presentation of the Task
- ➔ Variables
- ➔ Data Types
- ➔ Conditionals
- ➔ Iteration
- ➔ Task Walkthrough

Python Task

Imagine you're a data scientist analyzing survey responses and performance metrics. You'll create a Python application to process user inputs, manipulate data, and evaluate results dynamically. This task extends what you've learned by integrating string manipulation, numerical operations, and decision-making into a single project.

- **Text Analysis Module**
- **Numeric Insights Module**
- **Performance Evaluation Module**

Python Task

Text Analysis Module

Input: Ask the user to input a short survey response (e.g., “I love data science!”).

Tasks:

- Calculate the length of the response.
- Replace the most common letter in the response with #.
- Display the response with every word reversed but in the same order.
- Extract and display the first three words.

Python Task

Performance Evaluation Module

Input: Ask the user to input three times (in minutes) for an employee's performance in reading, planning and typing a final report.

Tasks:

- Calculate the total time.
- Evaluate the employee's performance using these criteria:
  - Gold Medal: Total time ≤ 90 minutes
  - Silver Medal: Total time ≤ 120 minutes
  - Bronze Medal: Total time ≤ 150 minutes
- No Medal: Total time > 150 minutes

What does the len() function in Python return?

A. The first character of a string
B. The number of items in an object (e.g., characters in a string)
C. The total memory size of a variable
D. The last character of a string

Which of the following is a valid conditional statement in Python?

A. if x > y:
B. if x > y then
C. if x > y: followed by an indented block of code
D. Both A and C

Variables and String Manipulation

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

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

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	!=

CoGrammar

Iteration

- Iteration refers to the process of executing a set of instructions repeatedly.
- For loops and while loops are commonly used to handle repetitive tasks in Python.
- Condition-based iteration allows the loop to continue or stop based on a condition (e.g., user input or reaching a specific value)

For Loops

- For loops are control flow structures used to iterate over a sequence (such as a list, tuple, string, etc.) and execute a block of code for each element in the sequence.
- For loops are used when you know the number of times you want to execute a block of code.

```
for item in sequence:
    # code block to be executed
```

While Loops

- While loops are control flow structures that repeatedly execute a block of code as long as a specified condition is true.
- These are used when you want to execute a block of code repeatedly as long as a specified condition is true. They continue iterating until the condition becomes false.

```
while condition:
    # code block to be executed
```

For Loops – Range Function

- Range is a built-in Python function used to generate a sequence of numbers. It is commonly used with for loops.
- Ranges in for loops are a way to specify a sequence of numbers that you want to iterate over. The range() function generates this sequence of numbers based on the arguments you provide.

```
range(start, stop, step)
```

Python Task

Imagine you're a data scientist analyzing survey responses and performance metrics. You'll create a Python application to process user inputs, manipulate data, and evaluate results dynamically. This task extends what you've learned by integrating string manipulation, numerical operations, and decision-making into a single project.

- **Text Analysis Module**
- **Numeric Insights Module**
- **Performance Evaluation Module**

Python Task

Text Analysis Module

Input: Ask the user to input a short survey response (e.g., “I love data science!”).

Tasks:

- Calculate the length of the response.
- Replace the most common letter in the response with #.
- Display the response with every word reversed but in the same order.
- Extract and display the first three words.

Python Task

Numeric Insights Module

Input: Ask the user to input three numbers representing survey scores (e.g., satisfaction, engagement, and ease of use).

Tasks:

- Calculate the sum, average, and product of the scores.
- Identify the highest and lowest scores.
- Compare the scores to a benchmark of 7 and display a summary.

Python Task

Performance Evaluation Module

Input: Ask the user to input three times (in minutes) for an employee's performance in reading, planning and typing a final report.

Tasks:

- Calculate the total time.
- Evaluate the employee's performance using these criteria:
  - Gold Medal: Total time ≤ 90 minutes
  - Silver Medal: Total time ≤ 120 minutes
  - Bronze Medal: Total time ≤ 150 minutes
- No Medal: Total time > 150 minutes

Which method is used to reverse a string in Python?

A. str.reverse()
B. String slicing (e.g., str[::-1])
C. reversed(str)
D. str[::-1] and reversed(str)

What is the output of the following operation: 10 + 20 // 3?

A. 10.0
B. 16
C. 16.666...
D. 20

Summary

- ★ **String Manipulation:**
  Length calculation, character replacement, and word reversal.
- ★ **Numerical Operations:**
  Basic arithmetic, comparisons, and summary statistics.
- ★ **Conditional Logic:**
  Using if-elif-else statements for decision-making.
- ★ **Modular Programming:**
  Breaking down a larger task into smaller, manageable modules.
- ★ **Integrating Multiple Skills:**
  Combining string and numerical operations for comprehensive problem-solving.

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education