Welcome to this CoGrammar Task Walkthrough: Task 4

The session will start shortly...

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

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Task Walkthrough: Iteration

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*.

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

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)

CoGrammar

For Loops

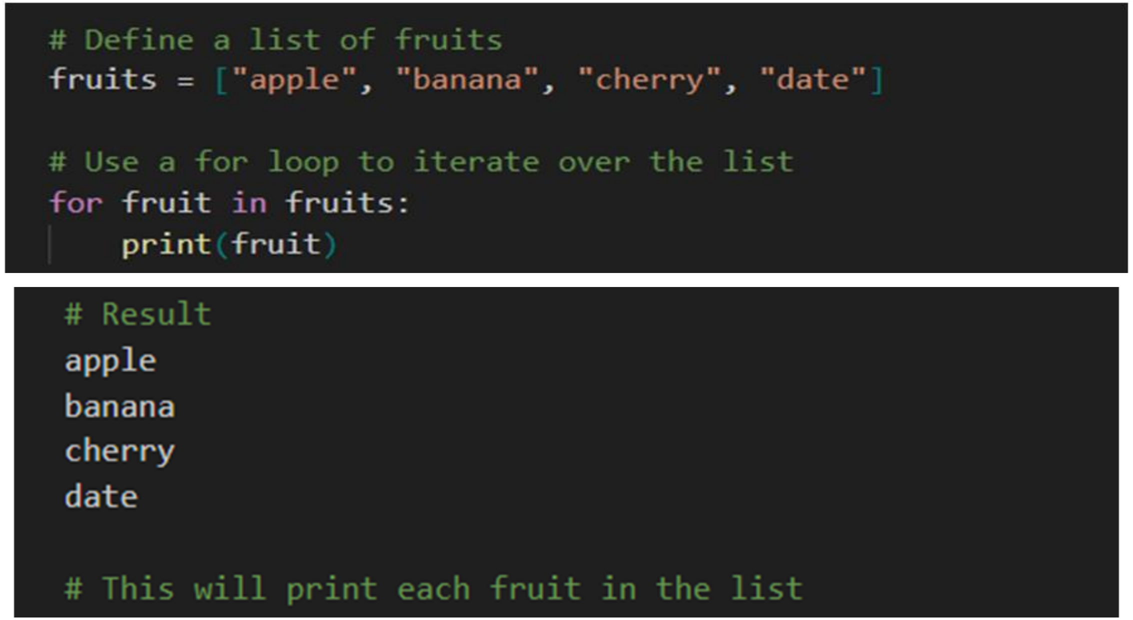
CoGrammar

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
```

For Loop Example



```python
# Define a list of fruits
fruits = ["apple", "banana", "cherry", "date"]

# Use a for loop to iterate over the list
for fruit in fruits:
    print(fruit)
```

```
# Result
apple
banana
cherry
date

# This will print each fruit in the list
```

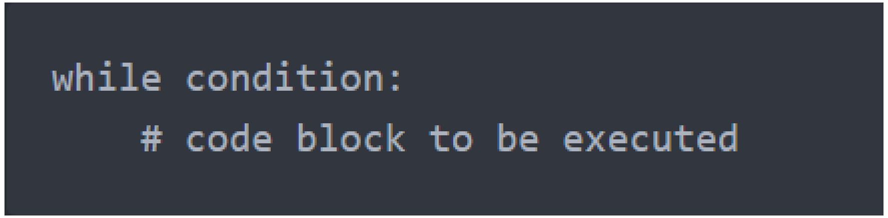
CoGrammar

While Loops

CoGrammar

While Loops

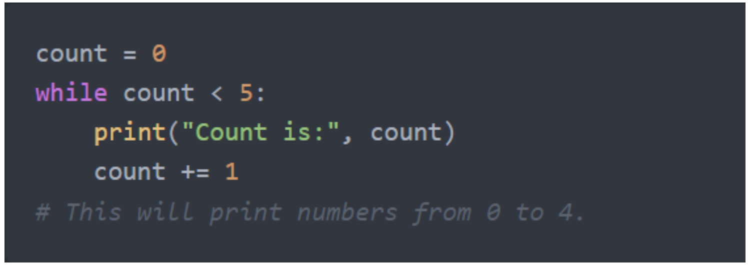
- While loops are control flow structures that repeatedly execute a block of code as long as a specified condition is true.
- These are used when you want to execute a block of code repeatedly as long as a specified condition is true. They continue iterating until the condition becomes false.



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

CoGrammar

While Loop Example

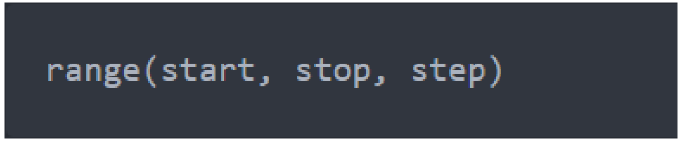


```python
count = 0
while count < 5:
    print("Count is:", count)
    count += 1
# This will print numbers from 0 to 4.
```

- while count < 5: Start of a while loop. It checks if the value of count is less than 5. If this condition is true, the code block inside the loop will execute. If the condition is false, the loop will terminate.
- print("Count is:", count): This line prints the current value of count along with the text "Count is:". Since count is initially 0, it will print "Count is: 0".
- count += 1: This line increments the value of count by 1 in each iteration of the loop. So, after the first iteration, count becomes 1, then 2, etc.

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)
```

CoGrammar

For Loops – Range Function

- Range() takes three arguments: start, stop, and step.
- start: The starting value of the sequence (inclusive). If not provided, it defaults to 0.
- stop: The ending value of the sequence (exclusive). This is a required argument.
- step: The increment between each value in the sequence. If not provided, it defaults to 1.

CoGrammar

Range Function Example

```
for i in range(start, stop, step):
    # code block to be executed
```

```
for i in range(1, 6):  # This will iterate from 1 to 5
    print(i)
```

- This loop will print numbers 1 through 5. Remember, the stop value is exclusive, so the loop stops before reaching 6.

Part 1 Walkthrough

CoGrammar

Auto-graded task 1

Follow these steps:

- Create a file called **while.py**.
- Write a program that continually asks the user to enter a number.
- When the user enters "`-1`", the program should stop requesting the user to enter a number. Please be aware that `0` is not a valid input.
    - Hint: think about how you might **exit** the loop if `-1` is entered.
- The program must then calculate the average of the **valid** numbers entered, excluding the `-1` and `0`.
- Use a while loop to achieve the continuous prompting and number collection.

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

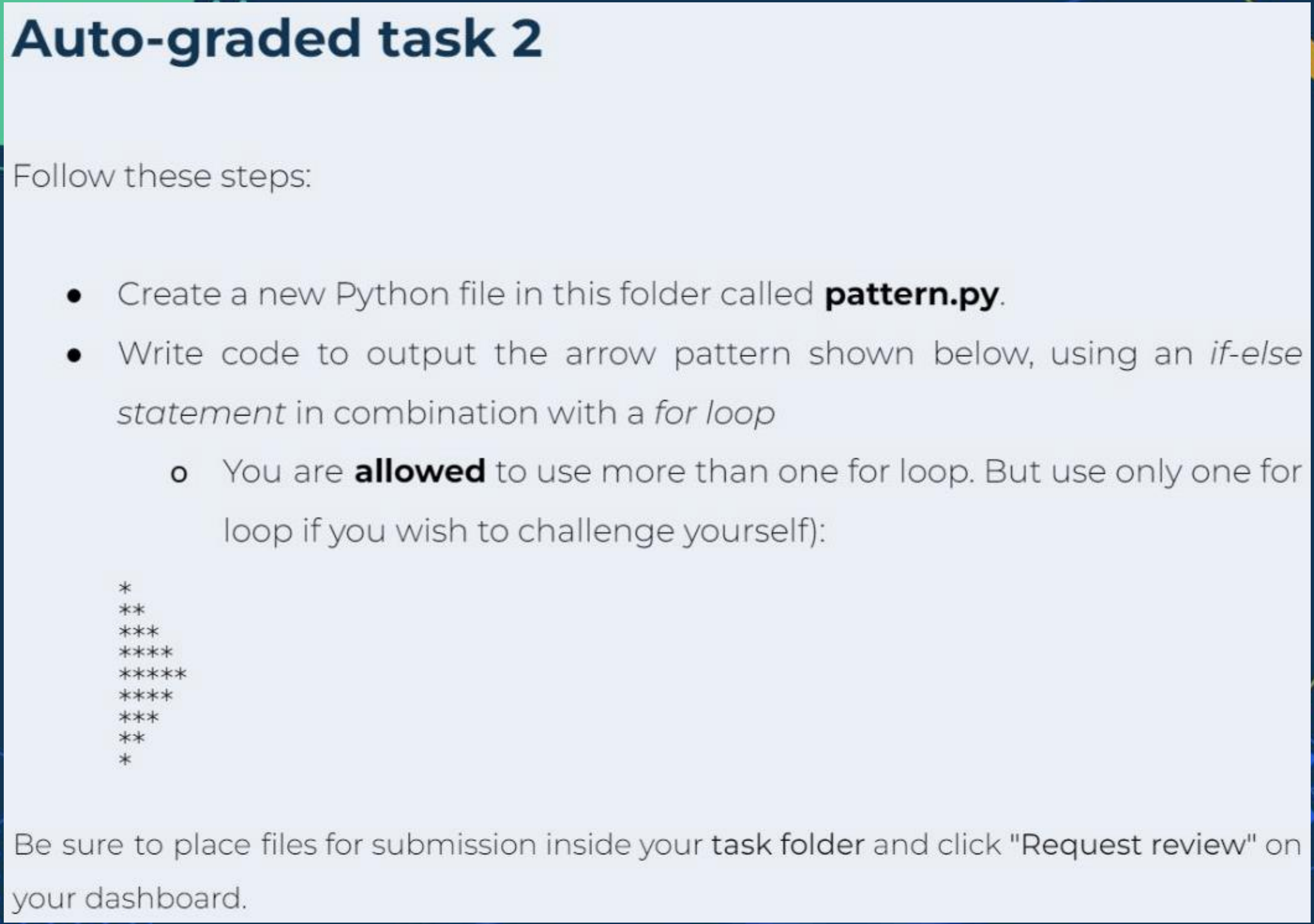
while.py

Task Objective

- The objective of this task is to demonstrate your understanding of while loops and their application in repetitive tasks by working with user input.
- You'll use the while loop repetition structure to repeatedly perform actions based on user input, such as:
  - Continuously prompting the user to enter numbers.
  - Stopping the loop when a specific condition is met (input of -1).
  - Calculating the average of the numbers entered, excluding the termination value (-1).
- This task is designed to enhance your problem-solving skills and deepen your understanding of loops and conditional logic.

Part 2 Walkthrough

CoGrammar



Auto-graded task 2

Follow these steps:

- Create a new Python file in this folder called **pattern.py**.
- Write code to output the arrow pattern shown below, using an *if-else statement* in combination with a *for loop*
  - You are **allowed** to use more than one for loop. But use only one for loop if you wish to challenge yourself):

```
*
**
***
****
*****
****
***
**
*
```

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

pattern.py

Task objective

- The objective of this task is to demonstrate your knowledge of for loops and conditional logic by generating a specific pattern output.
- You'll use an if-else statement in combination with a single for loop to:
  - Dynamically create and output lines of varying patterns (e.g., **, *, *******, etc.).
  - Apply logic within the loop to control the pattern based on conditions.
- This task is designed to enhance your problem-solving skills and deepen your understanding of loop structures and decision-making in Python.

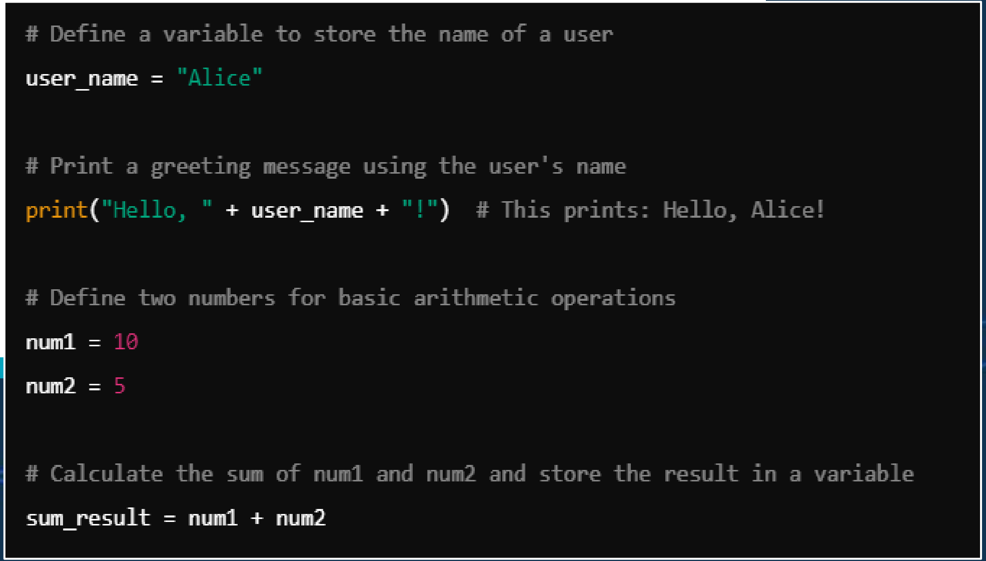
CoGrammar

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



```python
# 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
```

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar