Welcome to this CoGrammar session:

Iteration

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

Stay Safe Series:

Mastering Online Safety One Week or Step at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalisation, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* is designed to guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Beware of the Ask: Handling Requests for Personal Information

Sharing personal information without careful consideration can expose you to identity theft, financial loss, or privacy breaches. It's crucial to be vigilant and recognise when requests for such information are legitimate or potentially harmful. Always verify before you comply—your personal information is your responsibility to safeguard.

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 (C11) or 22 December 2024 (C12):

- **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 (C11) or 02 February 2025 (C12):

- **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 (C11) or 30 March 2025 (C12).

✓ 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 (C11) or 04 May 2025 (C12).
  - **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

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Iteration

Polls

CoGrammar

Poll

1. **Which of the following statements about iteration in Python is true?**
   A. A for loop is used to iterate over items of a sequence.
   B. A while loop executes only once regardless of the condition.
   C. Iteration is only possible with numeric data types.
   D. A loop cannot be used to iterate over a string.

Poll

2. **What is the main advantage of using a while loop over a for loop?**

   A. It’s always faster.

   B. It’s easier to read.

   C. It can run until a condition is no longer met, regardless of the number of iterations.

   D. None of the above.

CoGrammar

Learning Outcomes

- Understand the syntax and structure of loops in Python.
- Implement loops to solve repetitive tasks.
- Use loop control statements like break and continue to control the flow of loops.
- Implement nested for loops for more complex iterations.
- Understand the purpose and usage of the range() function in Python.
- Apply range() with different parameters to control the start, stop, and step values.

CoGrammar

Iterations

CoGrammar

Iterations

Can you think of situations in real life where you do something repeatedly?

CoGrammar

Real World Scenario: Sorting Laundry

- **Example:** Sorting Laundry
  - For Each item in the Basket:
    - Pick up an item
    - Check what type of clothing it is (shirt, pants, colour, etc.)
    - Place it into the correct pile
  - Repeat the process until there are no items left in the basket.

Iterations

- Iterations refer to the process of repeatedly executing a set of instructions or a block of code.
- In programming, iterations are commonly associated with loops, where the same code block is executed multiple times, either for a specified number of times or until a certain condition is met.

CoGrammar

Iterations in Coding

- Each execution of the code block within a loop is called an iteration. Iterations are essential for automating repetitive tasks and processing collections of data efficiently.
- *" Loops are like magic tricks in programming that help us avoid doing the same thing over and over again. Instead of writing the same code again, we use loops to make the computer do it for us. This saves time and make our programs neat and tidy. "*
- In coding, there are two kinds of loops: **while loops**, which we use when we want to keep doing something until a certain condition is true, and **for loops**, which we use when we can determine how many times we want to repeat something.

CoGrammar

While Loops

CoGrammar

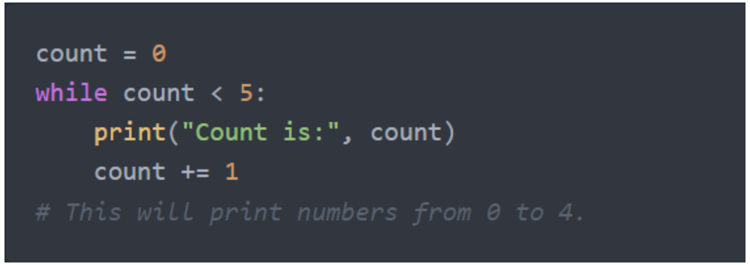
While Loops

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

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

CoGrammar

While Loop Example



```
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

CoGrammar

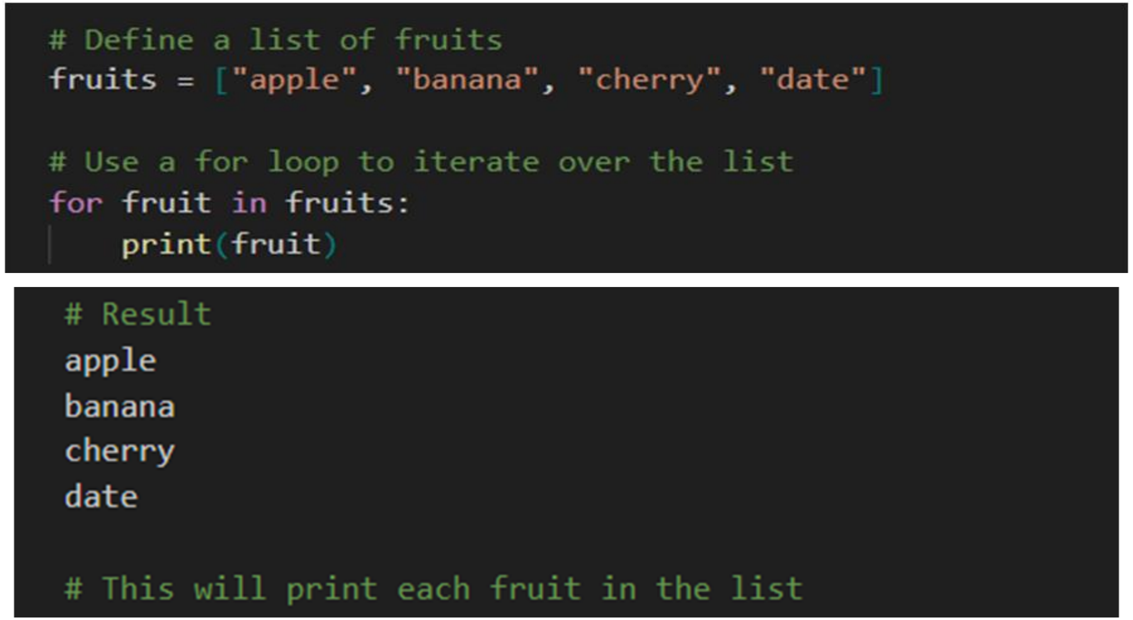
For Loops

- 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 can determine the number of times you want to execute a block of code.

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

CoGrammar

For Loop Example



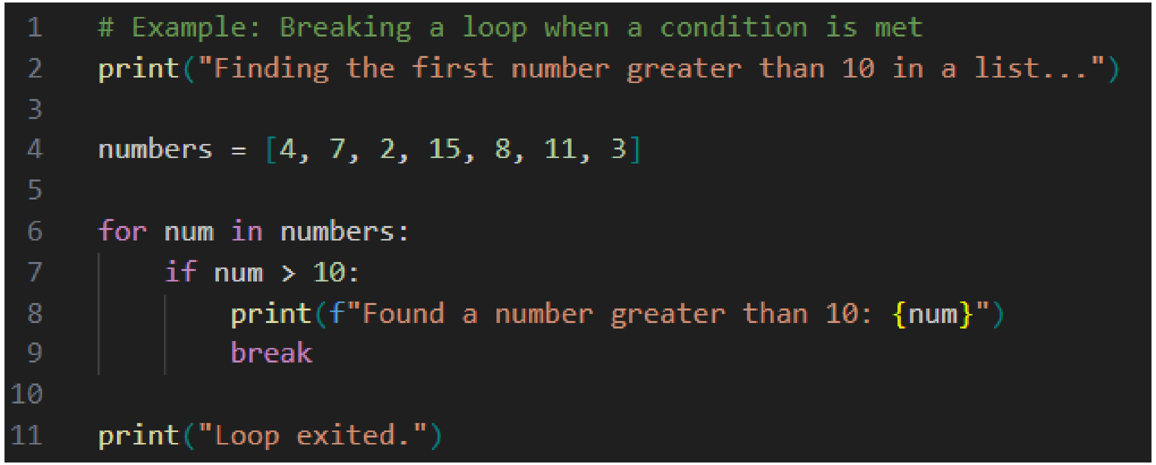
```python
# Define a list of fruits
fruits = ["apple", "banana", "cherry", "date"]

# 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

Loop Control Statement: break

- The break statement exits the loop early, stopping further iterations.

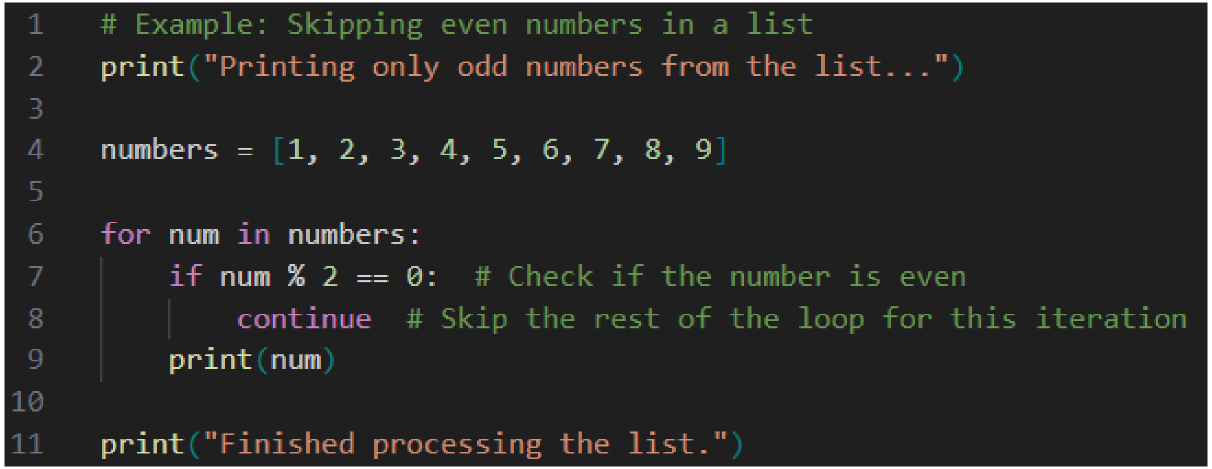


```python
# Example: Breaking a loop when a condition is met
print("Finding the first number greater than 10 in a list...")

numbers = [4, 7, 2, 15, 8, 11, 3]

for num in numbers:
    if num > 10:
        print(f"Found a number greater than 10: {num}")
        break

print("Loop exited.")
```

CoGrammar

Loop Control Statement: continue

- The continue statement skips the current iteration and moves to the next one.

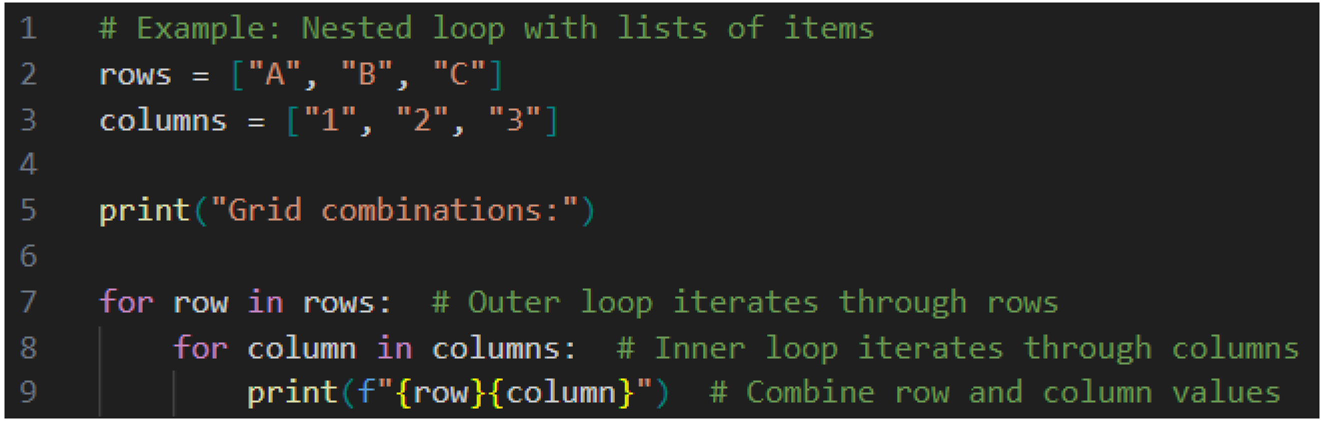


```python
# Example: Skipping even numbers in a list
print("Printing only odd numbers from the list...")

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

for num in numbers:
    if num % 2 == 0:  # Check if the number is even
        continue  # Skip the rest of the loop for this iteration
    print(num)

print("Finished processing the list.")
```

CoGrammar

Nested Loops

- Definition: A loop inside another loop
- Use case: For working with multi-dimensional data (e.g. matrices)



```python
# Example: Nested loop with lists of items
rows = ["A", "B", "C"]
columns = ["1", "2", "3"]

print("Grid combinations:")

for row in rows:  # Outer loop iterates through rows
    for column in columns:  # Inner loop iterates through columns
        print(f"{row}{column}")  # Combine row and column values
```

For Loops – Range Function

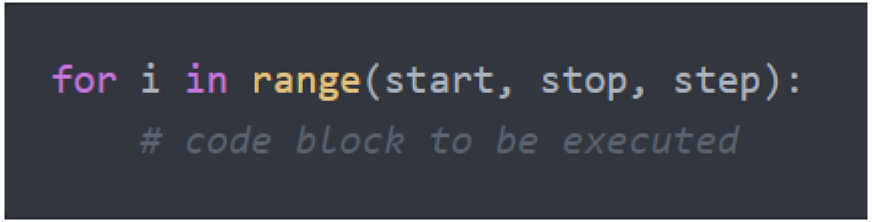
- Range is a built-in Python function used to generate a sequence of numbers. It is commonly used with for loops.
- Ranges in for loops are a way to specify a sequence of numbers that you want to iterate over.
- The range() function generates this sequence of numbers based on the arguments you provide.

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

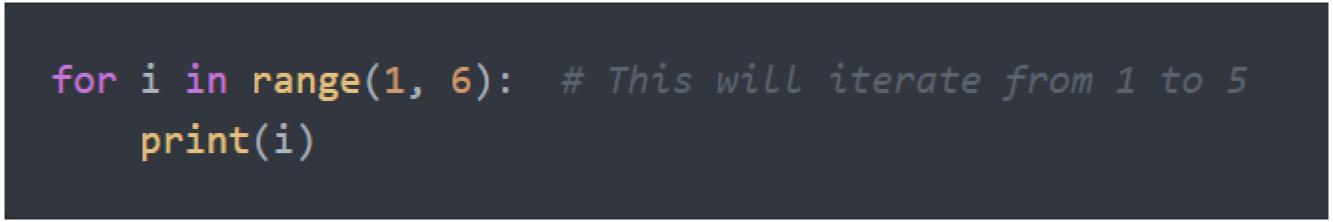
For Loops – Range Function

- 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.

Range Function Example



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



```python
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.

CoGrammar

Let’s take a short break

CoGrammar

Let’s get coding!

CoGrammar

Polls

CoGrammar

Poll

1. **Which of the following is a valid way to iterate over a list in python?**
   A. while element in my_list:
   B. for element in my_list:
   C. repeat element in my_list:

CoGrammar

Poll

2. **What is the purpose of the continue statement in a loop?**

   A. Exit the loop

   B. Skip the current iteration

   C. Reset the loop

CoGrammar

Conclusion and Recap

- for Loop: Iterates over a sequence (like lists, strings, or ranges).
- while Loop: Repeats as long as a specified condition is true.
- break stops loop execution early, while continue skips to the next iteration without completing the current one.
- Nested Loops: Loops within loops to handle multi-dimensional data or complex problems.
- range() Function: Generates a sequence of numbers for loop iterations. Parameters: start, stop, and step to control iteration.

CoGrammar

Learner Challenge

- Write a program to find and print the sum of numbers from 1 to 100 that are divisible by 3.

CoGrammar

Additional Resources

- Iterations

  https://docs.python.org/3/library/stdtypes.html#iterator-types

- Python For Loops

  https://www.w3schools.com/python/python_for_loops.asp

  https://realpython.com/python-for-loop/

- Python While Loops

  https://www.geeksforgeeks.org/python-while-loop/

CoGrammar

Questions and Answers

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar