Welcome to the CoGrammar

Skills Bootcamp: Iteration

The session will start shortly...

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

Cyber Security 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** 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**

Cyber Security Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**

- We would love your **feedback** on lectures: **Feedback on Lectures**

- Find all the lecture **content** in you **Lecture Backpack** on GitHub.

Safeguarding & Welfare

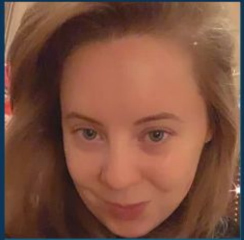
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

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

Rafiq Manan



Charlotte Witcher

Nurhaan Snyman

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Learning Objectives & Outcomes

- Define and differentiate between for loops and while loops in Python.
- Explain the purpose of loops
- Identify scenarios where loops are useful for iterating over data.
- Write code that uses for and while loops to iterate over sequences.
- Use loop control statements like break and continue to control the flow of loops.
- Design programs that employ loops to solve practical problems.

CoGrammar

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

CoGrammar

Iteration

October 2024

Iteration

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

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 socks, e.t.c)
        - Place it into the correct pile
    - Repeat the process until there are no items left in the basket.

Polls

Please have a look at the poll notification and select an option.

What do you think a loop does?

A. Repeats a set of instructions for each element in a sequence or until a condition is met.
B. Calls a block of code based on specific user input
C. Executes code randomly based on the program's state
D. Stores data temporarily in memory for quick access

Polls

Please have a look at the poll notification and select an option.

Have you used loops in any programming language before?

A. Yes
B. Unsure
C. No

CoGrammar

Iteration: Definition

- **Iteration** is the process of repeating a set of instructions in code until a condition is met or for a specific number of times.

CoGrammar

Iteration: Importance

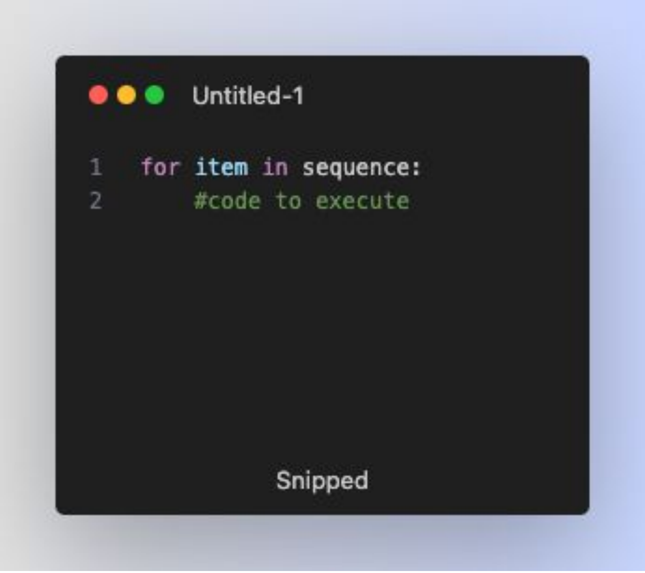
- Loops save time and effort by automating repetitive tasks, making code more efficient and easier to maintain.

Types of loops in python: For Loop

- **For loop:**
  - Ideal for iterating over a sequence (like a list, string, or range).
  - Repeats for each element in the sequence.

- **Use Cases:**
  - Iterating through a list of elements.
  - Repeating an operation a set number of times.
  - Accessing elements in a collection.

Types of loops in python: For Loop



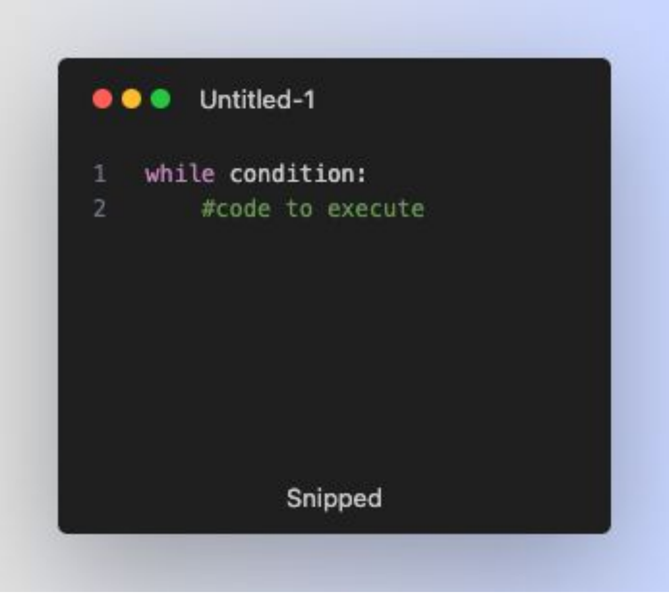
```
for item in sequence:
    #code to execute
```

CoGrammar

Types of Loops: While Loop

- **While loop:**
  - Repeats as long as a given condition is True.
  - Used when the number of iterations isn’t known beforehand.
- **Use cases:**
  - Continuously checking for a condition to be met (e.g., waiting for user input).
  - Creating loops where the number of iterations is not predetermined.

Types of Loops: While Loop



```
while condition:
    #code to execute
```

CoGrammar

Let’s take a break

CoGrammar

Stay Safe Series:

Mastering Online Safety One Week 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 radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

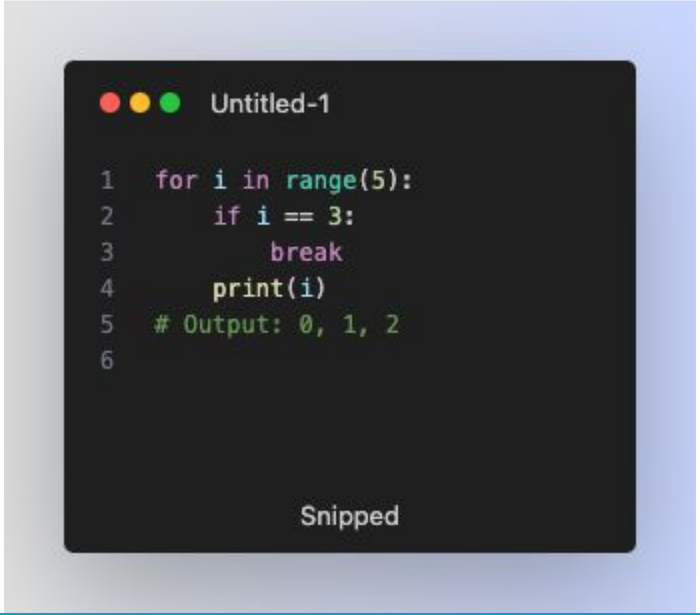
As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

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 recognize when requests for such information are legitimate or potentially harmful. Always verify before you comply—your personal information is your responsibility to safeguard.

Loop Control Statement: break

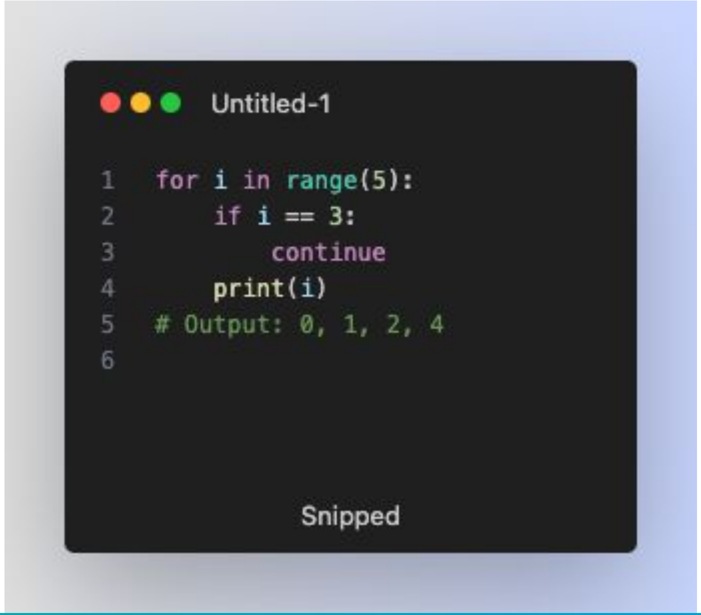
- Break Statement:
  - Exits the loop early, stopping further iterations.



```python
for i in range(5):
    if i == 3:
        break
    print(i)
# Output: 0, 1, 2
```

Loop Control Statement: continue

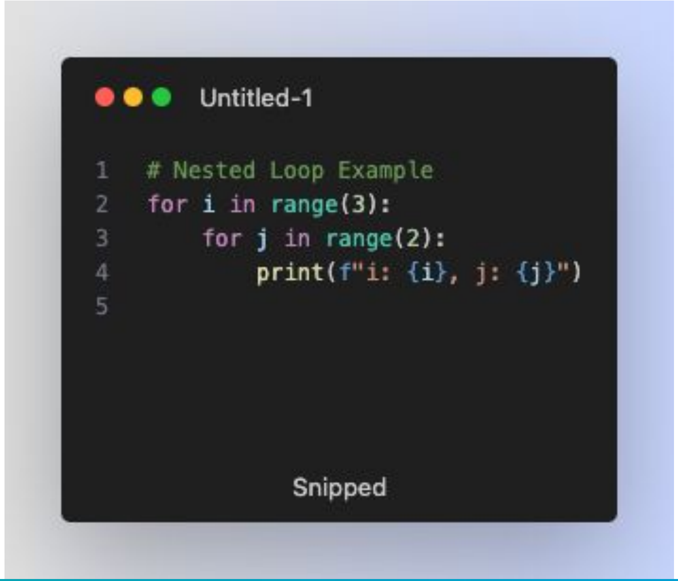
- Continue statement
  - Skips the current iteration and moves to the next one.



```python
for i in range(5):
    if i == 3:
        continue
    print(i)
# Output: 0, 1, 2, 4
```

Nested Loops

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



```python
# Nested Loop Example
for i in range(3):
    for j in range(2):
        print(f"i: {i}, j: {j}")

```

Challenge:

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

Polls

Please have a look at the poll notification and select an option.

Which of the following is a valid way to iterate over a list in python

A. for element in myList:

B. while element in myList:

C. repeat element in myList:

CoGrammar

Polls

Please have a look at the poll notification and select an option.

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

Summary

- **Loops** automate repetitive tasks and are essential for processing collections or executing code multiple times.
- **For Loops** iterate over a sequence (like lists or ranges) and are ideal when the number of iterations is known.
- **While Loops** continue executing as long as a given condition is true, useful for indefinite repetition.
- **Break** stops loop execution early, while **Continue** skips to the next iteration without completing the current one.
- Effective use of loops simplifies code, reduces redundancy, and enhances readability.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar