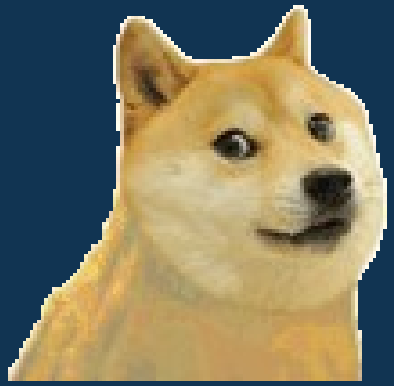
Welcome to the Loops in Python Lecture



The session will start shortly...

CoGrammar

Johannesburg Team Housekeeping

- Please be mindful and respectful to everyone in this supportive learning environment. Mutual respect and tolerance are fundamental values we uphold.
- There are no bad or silly questions—feel free to ask anything! You can ask Sashlin or myself questions at any time, regardless of the situation. Even if you find yourself in a dire situation—like stuck in quicksand—you're still welcome to ask us a question (though we recommend calling or shouting first!).
- A few additional reminders for onsite behavior:
  - Keep shared spaces tidy—clean up after yourself in the break areas.
  - Please mute your devices during sessions to minimize distractions.
  - Avoid making personal phone calls in common areas—use designated quiet zones if you need to step away.
- Additionally, please remember to put any dishes in the sink before 2 p.m., as Lizbeth will have already finished for the day. If you’re feeling unwell, kindly inform Ingrid or myself via email.

Learning Objectives

- Understand the purpose, syntax, and differences between for loops and while loops.
- Write Python programs using both types of loops.
- Explain how these loops control the flow of a program.

What Are Loops ?

- Loops allow us to repeat tasks in programming.
- Instead of writing repetitive code, loops help us automate tasks.

For Loops

- A **for** loop iterates over a sequence (such as a list, string, or range) and executes a block of code for each element in that sequence.

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

- For loops are great when you know the number of iterations beforehand

Syntax Breakdown

```
numbers = [1, 2, 3, 4]
for num in numbers:
    print(num)
```

While Loops

- A **while** loop continues to execute as long as a specified condition remains **true**.

```
while condition:
    # Code to execute
```

- While loops are useful when the number of iterations isn't predetermined and depends on a condition being met.
- **NB:** while loops can lead to infinite loops if the condition never becomes false.

Syntax Breakdown

```
count = 1
while count <= 5:
    print(count)
    count += 1
```

For Loop vs While Loop

For Loop	While Loop
Iterates over a known sequence	Repeats as long as a condition is true
Used when you know the number of iterations.	Used when the number of iterations is uncertain.
Example: Iterating through a list	Example: Keep asking for a password

Pitfalls To Avoid

- **For Loops:**
  - Iterating Over the Wrong Sequence
  - Changing the Sequence Inside the Loop
  - Off-by-One Error
- **While Loops**:
  - Infinite Loops
  - Incorrect Condition
  - Forgetting to Break the Loop

Q & A SECTION

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

Recap

- **For Loops:** Best for iterating over sequences.
- **While Loops:** Best for condition-based loops.
- **Loops** help reduce repetitive code and errors.

Thank you for attending