Python

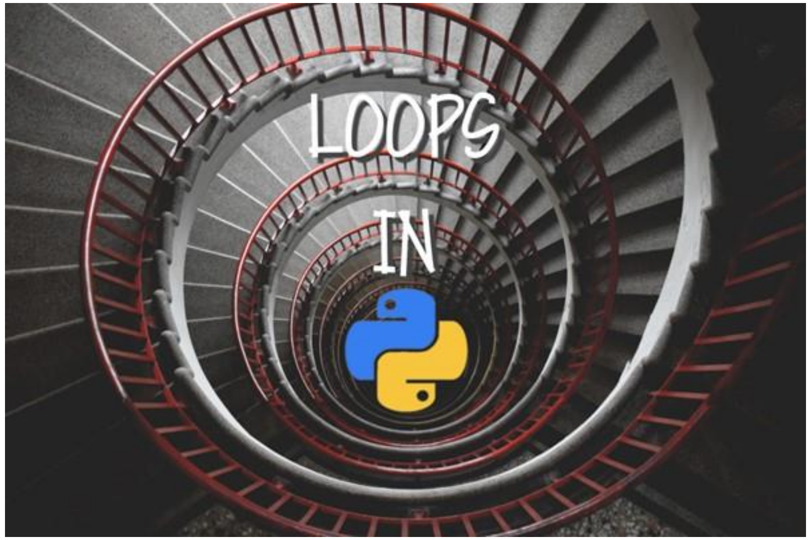
Full stack Skills Bootcamp

Introducing Python Loops

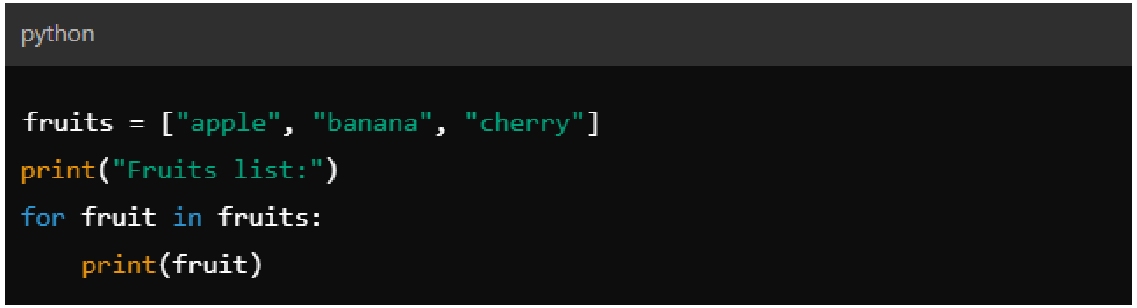
What are Loops?

- Loops are constructs that allow you to repeat a block of code multiple times based on a condition or the elements of a sequence.

- Types of Loops:
  For Loop: Iterates over a sequence (like a list or tuple).
  While Loop: Repeats if a condition is true.



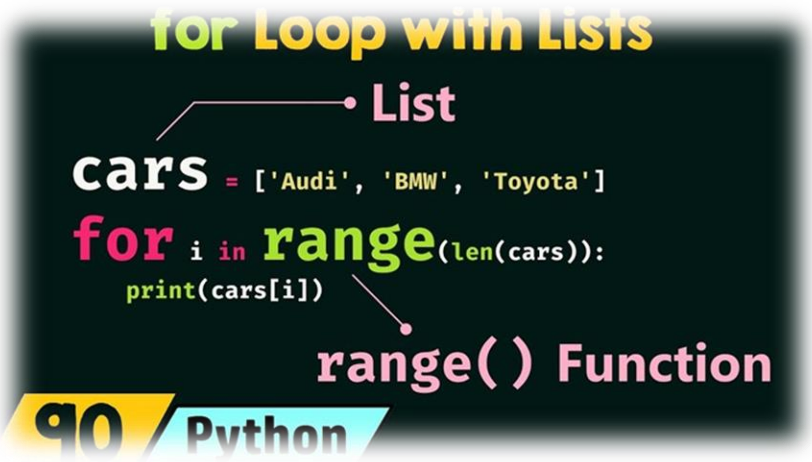
For Loop with Lists



python

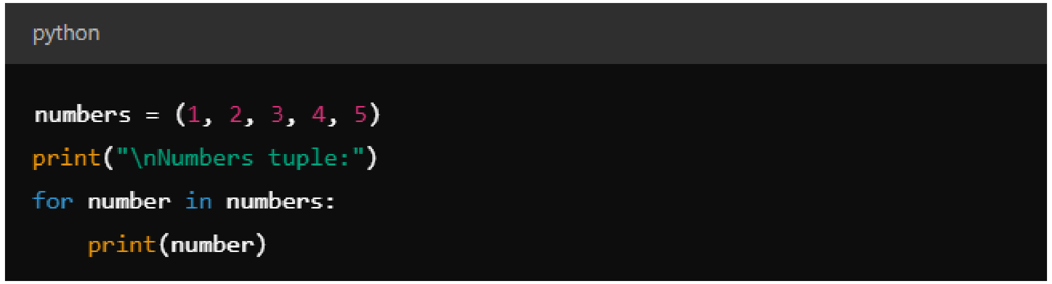
```python
fruits = ["apple", "banana", "cherry"]
print("Fruits list:")
for fruit in fruits:
    print(fruit)
```

- This loop iterates over a list of fruits, printing each fruit in the list.



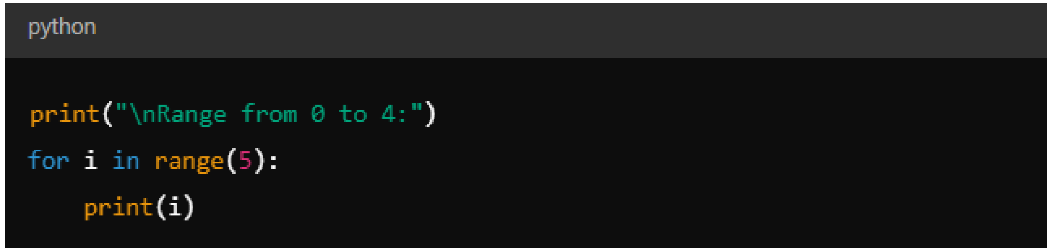
```python
cars = ['Audi', 'BMW', 'Toyota']
for i in range(len(cars)):
    print(cars[i])
```

For Loop with Tuples & Ranges

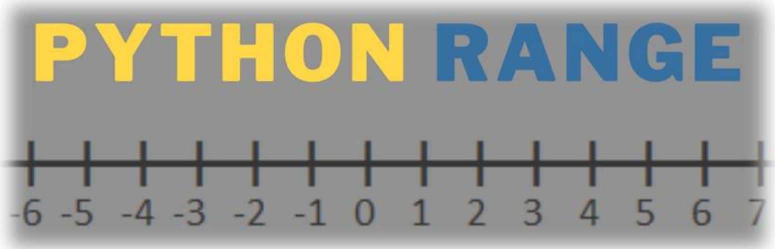


```python
numbers = (1, 2, 3, 4, 5)
print("\nNumbers tuple:")
for number in numbers:
    print(number)
```

- Like lists, this loop iterates over a tuple of numbers, displaying each number.

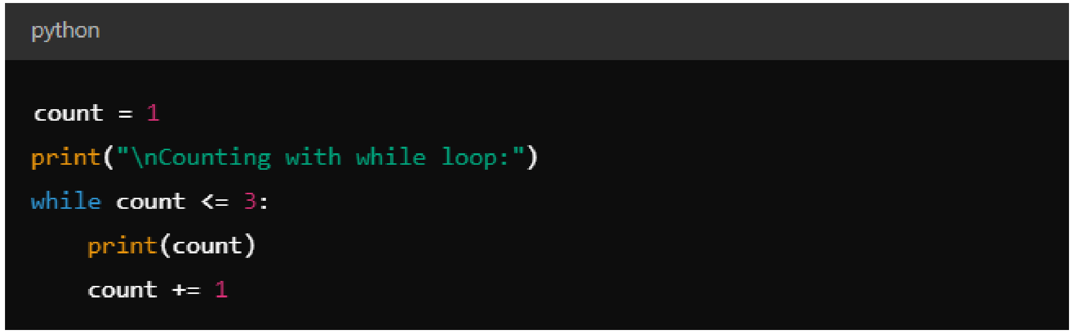


```python
print("\nRange from 0 to 4:")
for i in range(5):
    print(i)
```



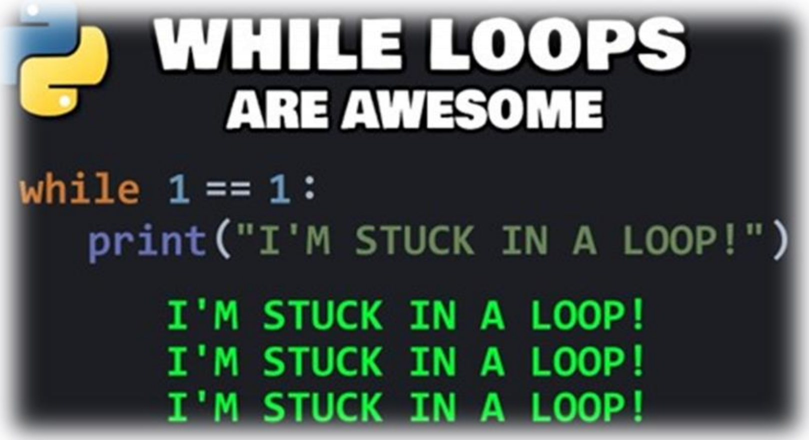
- This loop uses the range() function to iterate over a sequence of numbers from 0 to 4.

While Loop



```python
count = 1
print("\nCounting with while loop:")
while count <= 3:
    print(count)
    count += 1
```

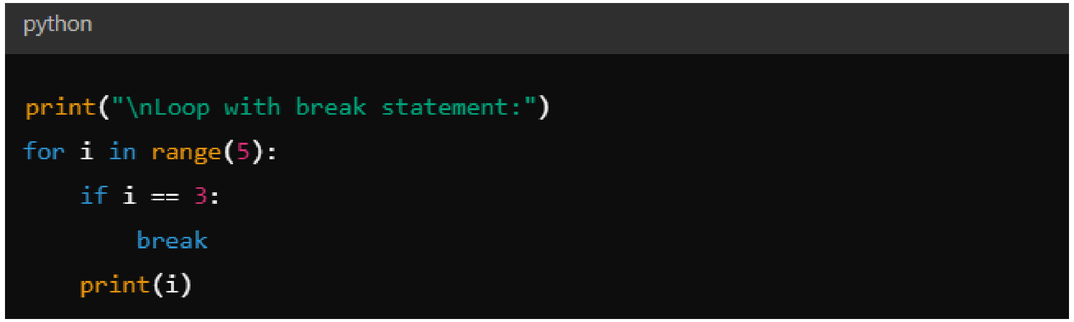
- This loop continues to execute as long as the count is less than or equal to 3, incrementing count each iteration



```python
while 1 == 1:
    print("I'M STUCK IN A LOOP!")
```

```
I'M STUCK IN A LOOP!
I'M STUCK IN A LOOP!
I'M STUCK IN A LOOP!
```

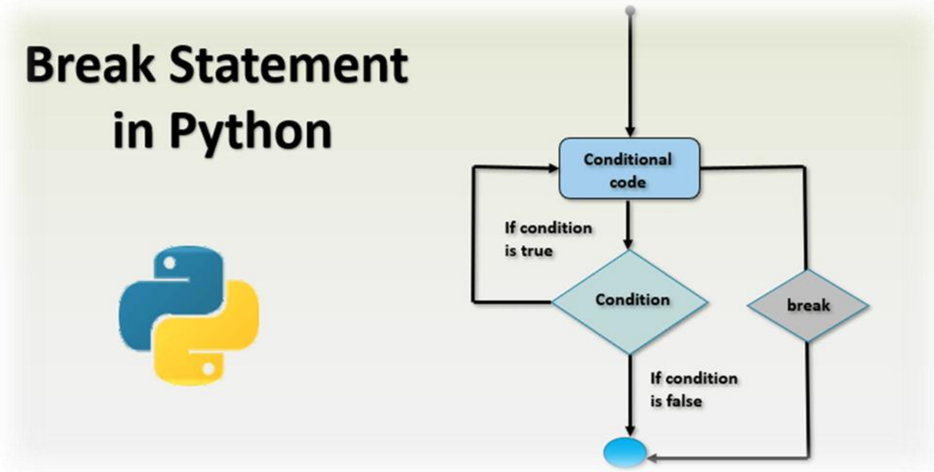
Break Statement



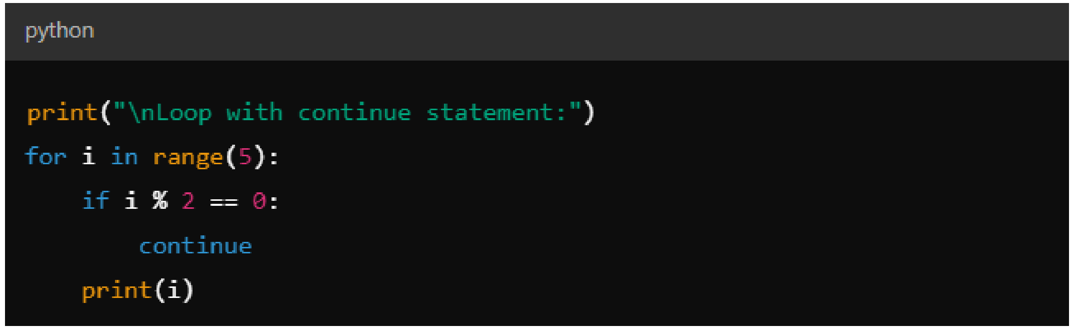
python

```python
print("\nLoop with break statement:")
for i in range(5):
    if i == 3:
        break
    print(i)
```

- The loop exits early when i reaches 3 due to the break statement.

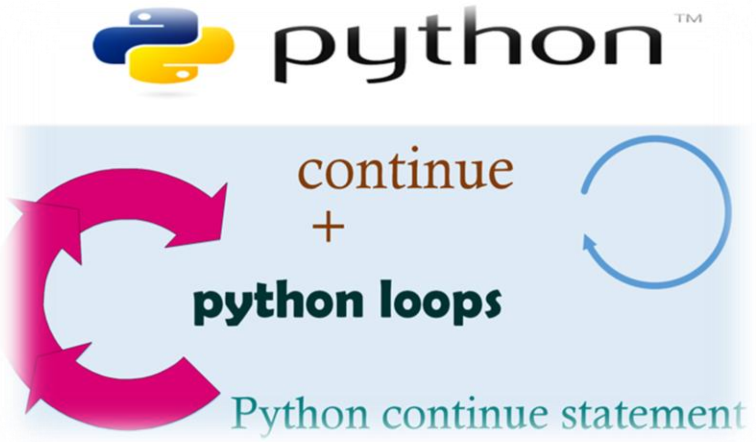


Continue Statement



```python
print("\nLoop with continue statement:")
for i in range(5):
    if i % 2 == 0:
        continue
    print(i)
```

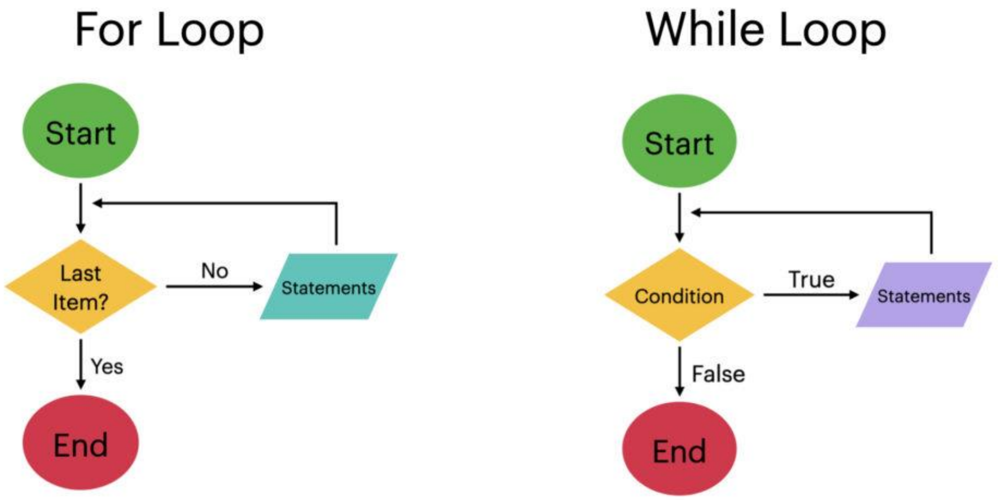
- This loop skips printing even numbers, only displaying the odd ones due to the continue statement.



Conclusion

Key Points

- Loops allow repetitive execution of code.
- For Loop: Best for iterating over sequences.
- While Loop: Useful when the number of iterations is not known.
- Control flow statements (break and continue) enhance loop functionality.


For Loop
Start
Last Item?
No
Statements
Yes
End
While Loop
Start
Condition
True
Statements
False
End