# Day 2: Loops & Comprehensions (24/07/2025)

Today’s focus was on two powerful core Python concepts:  
1. Loops – for repeating tasks  
2. List Comprehensions – a cleaner way to build lists

## 🔁 Loops

Loops are used to execute a block of code repeatedly.

1. For Loop – Iterate over lists, tuples, sets, dictionaries, and strings:

fruits = ["Banana", "Orange", "Apple", "Guava", "Grapes"]  
for fruit in fruits:  
 print(fruit)

2. While Loop – Runs while a condition is true:

i = 1  
while i <= 6:  
 print(i)  
 i += 1

3. Break and Continue – Control the flow of loops:

# Break Example  
nums = [1, 2, 3, 4, 4, 5, 9, 8, 10]  
for n in nums:  
 if n == 4:  
 break  
 print(n)  
  
# Continue Example  
for n in nums:  
 if n == 4:  
 continue  
 print(n)

## 🧠 List Comprehensions

A cleaner and more efficient way to create lists in a single line.

1. Create a list of squares using for loop and list comprehension:

# Using loop  
squares\_loop = []  
for i in range(1, 6):  
 squares\_loop.append(i \* i)  
  
# Using comprehension  
squares\_comp = [i \* i for i in range(1, 6)]  
print(squares\_loop)  
print(squares\_comp)

2. Create a list of even numbers from 1 to 10:

evens = [x for x in range(1, 11) if x % 2 == 0]  
print(evens)

3. Convert list of strings into uppercase:

fruits = ["Banana", "Orange", "Apple", "Guava", "Grapes"]  
upper\_words = [w.upper() for w in fruits]  
print(upper\_words)

⚠️ Note: If you miss the parentheses in `w.upper()`, you'll collect function references instead of calling them.

## 🔧 Mini Task

1. Get numbers from 1 to 20 using a while loop:

i = 1  
nums = []  
while i <= 20:  
 nums.append(i)  
 i += 1  
print(nums)

2. Filter multiples of 3 using list comprehension:

multiples = [x for x in range(0, 21) if x % 3 == 0]  
print(multiples)

3. Use a loop to print Even/Odd for each number from 1 to 20:

for i in range(1, 21):  
 if i % 2 == 0:  
 print(f"{i} is Even")  
 else:  
 print(f"{i} is Odd")