for	while	break	continue	pass
used for iterating over a sequence (like a list, tuple, string, or range) or any other iterable object.	used to execute a block of code as long as a condition is true	used to exit or break out of a loop prematurely	skip the current iteration and proceed to the next iteration of the loop	Used to write empty loops, null statement
Syntax : **for element in iterable:** # code to be executed for each element	**Syntax :** **while condition:** # code to be executed as long as the condition is true	**Syntax :** **break**	**Syntax :** **continue**	**Syntax :** **pass**

Example :
 	
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
print(fruit)
	Example :
	
	count = 0
	while count < 5:
	print(count)
	count += 1
		Example :
		
		for i in range(1, 11):
		if i == 6:
		break
		else:
		print(i, end=" ")
			Example:
			
			for i in range(1, 11):
			if i == 6:
			continue
			else:
			print(i, end=" ")
				Example:
				
				a = 10
				b = 20
				
				if(a<b):
				pass
				else:
				print("b<a")

With Different structures :
 	
1. Lists and Sequences:
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
print(fruit)
 	
2. Range:
for i in range(5):
print(i)
 	
3. String:
text = "Hello, World!"
for char in text:
print(char)
 	
4. Dictionary:
Iterating over keys:
person = {"name": "John", "age": 30}
for key in person:
print(key, person[key])
 	
Iterating over values:
person = {"name": "John", "age":
	With Different structures :
	
	1. Lists:
	numbers = [1, 2, 3, 4, 5]
	index = 0
	while index < len(numbers):
	print(numbers[index])
	index += 1
	
	2. Dictionaries:
	Iterating through keys:
	person = {"name": "Alice", "age": 30, "city": "New York"}
	keys = list(person.keys())
	index = 0
	while index < len(keys):
	key = keys[index]
	print(key, person[key])
	index += 1
	
	Iterating through values:
	values = list(person.values())
	index = 0
	while index < len(values):
	print(values[index])
	index += 1
		With Different structures :
		
		1. Terminating a Loop Early:
		numbers = [1, 2, 3, 4, 5]
		for number in numbers:
		if number == 3:
		print("Found 3")
		break
		
		2. User-Defined Exit Condition:
		while True:
		user_input = input("Enter 'quit' to exit: ")
		if user_input == 'quit':
		break
		
		3. Preventing Infinite Loops :
		count = 0
		while count < 10:
		print(count)
		if count == 5:
		break # Exit the loop to avoid an infinite loop
		count += 1
			With Different structures :
			
			1. Skipping Specific Items:
			numbers = [1, 2, 3, 4, 5]
			for number in numbers:
			if number == 3:
			continue # Skip processing of the number 3
			print(number)
			
			2. Skipping Iterations with User Input:
			for i in range(5):
			user_input = input(f"Skip iteration {i + 1}? (y/n): ")
			if user_input.lower() == 'y':
			continue # Skip the current iteration based on user input
			print(f"Processing iteration {i + 1}")
			
			3. Complex Conditions:
			for i in range(10):
			if i % 2 == 0:
			continue # Skip even numbers
				With Different structures :
				
				1. Creating a Placeholder Function:
				def my_function():
				pass # To be implemented later
				
				2. Creating an Empty Class:
				class MyClass:
				pass # To be expanded later
				
				3. Leaving a Conditional Block Empty:
				if some_condition:
				pass # Placeholder for future code
				else:
				# Do something else

30}
for key, value in person.items():
    print(key, value)

Iterating over key, value pair:
person = {"name": "John", "age": 30}
for key, value in person.items():
    print(key, value)

5. Nested Loops:
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for row in matrix:
    for item in row:
        print(item)

6. Enumerating:
fruits = ["apple", "banana", "cherry"]
for index, fruit in enumerate(fruits):
    print(f"Index {index}: {fruit}")

3. Strings:
text = "Hello, World!"
index = 0
while index < len(text):
    print(text[index])
    index += 1

4. Iterating with Conditions:
numbers = [10, 20, 30, 40, 50]
index = 0
while index < len(numbers):
    if numbers[index] == 30:
        print("Found 30!")
        break
    print(numbers[index])
    index += 1

4. Completing a Task Early:
for i in range(10):
    if i == 5:
        print("Task completed, exiting loop early")
        break
    print(i)

    if i == 7:
        continue # Skip the number 7
    print(i)

4. Conditional Filtering :
fruits = ["apple", "banana", "cherry", "date"]
for fruit in fruits:
    if len(fruit) < 6:
        continue # Skip processing of fruits with less than 6 letters
    print(fruit)