break and continue

In Python, `break` and `continue` are control flow statements used within loops to alter their behavior.

1. `break` Statement:

The `break` statement is used to exit a loop prematurely. When a `break` statement is encountered inside a loop, the loop is immediately terminated, and the program continues with the next statement after the loop.

Example: Finding the First Occurrence

```python
# Search for a specific element in a list and break when found
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

target = 5
found = False

for num in numbers:
    if num == target:
        found = True
        break  # exit the loop when the target is found

if found:
    print(f"{target} found in the list.")
else:
    print(f"{target} not found in the list.")
```

2. `continue` Statement:

The `continue` statement is used to skip the rest of the code inside a loop for the current iteration and move on to the next iteration.

Example: Skip Odd Numbers

```python
# Print only even numbers from a list
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for num in numbers:
    if num % 2 != 0:
        continue  # skip to the next iteration if the number is odd
    print(num)
```

Example Combining `break` and `continue`:

```python
# Find and print the first even number greater than 5 in a list
numbers = [1, 3, 5, 6, 8, 10]

for num in numbers:
    if num <= 5:
        continue  # skip to the next iteration if the number is less than or equal to 5
    elif num % 2 == 0:
        print(f"The first even number greater than 5 is: {num}")
        break  # exit the loop once the first even number greater than 5 is found
```

In this example, the `continue` statement is used to skip numbers less than or equal to 5, and the `break` statement is used to exit the loop once the first even number greater than 5 is found.

These control flow statements provide flexibility in loop constructs, allowing you to tailor the loop's behavior based on specific conditions. `break` is useful for prematurely exiting a loop, while `continue` is helpful for skipping specific iterations.