Loop Control Statements

Statement	Description
`break`	Exits the loop immediately
`continue`	Skips the current iteration and moves to the next

Example: Using `break`

```
for i in {1..10}; do
    if [ $i -eq 5 ]; then
        echo "Breaking at $i"
        break
    fi
    echo "Iteration: $i"
done
```

Example: Using `continue`

```
for i in {1..5}; do
    if [ $i -eq 3 ]; then
        echo "Skipping $i"
        continue
    fi
    echo "Iteration: $i"
done
```

Looping Through Command Output

```
for file in $(ls *.txt); do
    echo "Processing file: $file"
done
```

Summary

- **`for` loop**: Iterates over a range or list.
- **`while` loop**: Runs while a condition is true.
- **`until` loop**: Runs until a condition becomes true.
- **Use `break`** to exit a loop early.
- **Use `continue`** to skip an iteration.