

Control Flow – Overview

Control flow decides **how a program behaves**:

- Conditional statements
- Logical operators
- Loops (`for`, `while`)
- Loop control (`break`, `continue`, `pass`)

This module teaches how Python makes decisions and repeats actions.

Conditional Statements

Basic structure:

```
if condition:  
    block
```

A condition must evaluate to **True** or **False**.

Example:

```
age = 20  
if age >= 18:  
    print("Adult")
```

————— [finished] ———

```
Adult
```

if / else

```
if condition:  
    action_if_true  
else:  
    action_if_false
```

Example:

```
temperature = 10  
  
if temperature > 15:  
    print("Warm")  
else:  
    print("Cold")
```

_____ [finished] - _____

```
Cold
```


Logical Operators in Conditions

```
and  # both True  
or   # one True  
not  # invert
```

Examples:

```
age = 20  
has_id = True  
  
age >= 18 and has_id  
age < 18 or has_id  
not has_id
```

Nested Conditions

```
age = 25
citizen = True

if age >= 18:
    if citizen:
        print("Eligible")
    else:
        print("Not eligible")
```

----- [finished] -----

```
Eligible
```

Avoid deep nesting when possible.

for Loop – Basics

Iterates over a sequence:

```
for x in [1, 2, 3]:  
    print(x)
```

_____ [finished] _____

```
1  
2  
3
```

Strings:

```
for char in "Python":  
    print(char)
```

_____ [finished] _____

```
P  
y  
t  
h  
o  
n
```

for Loop with range()

```
for i in range(5):  
    print(i)
```

————— [finished] ———

```
0  
1  
2  
3  
4
```

Custom range:

```
for i in range(2, 10, 2):  
    print(i)
```

————— [finished] ———

```
2  
4  
6  
8
```

while Loop

Repeats while a condition is **True**.

```
count = 0
while count < 5:
    print(count)
    count += 1
```

_____ [finished] - _____

```
0
1
2
3
4
```

Loop Control – break

Stops the loop immediately.

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

_____ [finished] -

```
0
1
2
3
4
```

Loop Control – continue

Skips current iteration.

```
for i in range(5):
    if i % 2 == 0:
        continue
    print(i)
```

_____ [finished] -

```
1
3
```

Loop Control – pass

`pass` does nothing – placeholder.

```
for i in range(3):  
    pass
```

———— [finished] ————

```
for i in range(3):
```

———— [finished with error] ————

```
File "/tmp/nix-shell-25740-0/.presenterIOtbHV/snippet.py", line 1  
    for i in range(3):  
IndentationError: expected an indented block after 'for' statement on  
line 1
```

Example

```
1 total = 0
2
3 for i in range(1, 6):
4     if i % 2 == 0:
5         total += i
6     else:
7         continue
8
9 print("Sum of even numbers:", total)
```

————— [finished] ———

```
Sum of even numbers: 6
```

Mini Task

Write a script that:

1. Asks the user for a number
2. Prints all numbers from 1 to that number
3. Prints only **even numbers**
4. Skips odd numbers using `continue`
5. Stops entirely if number exceeds 100 using
`break`



$x += 1$

$x++$

$x -= -1$