

## Loops

### for loops

#### format:

```
for ( initialization; test; update)
{
    loop body
}
```

#### How does it work?

Initialize a variable.

Test the value of the variable.

If the test is **true**, execute the loop body. Then update the variable. Test again.

If the test is **false**, exit the loop and continue the program.

#### How do you update the variable?

You will either increment (add) or decrement (subtract) the variable by 1.

increment the variable by 1: `var_name++`

decrement the variable by 1: `var_name--`

#### Example

```
{
    int index, total = 0;

    for ( index = 1; index <= 5; index++ )
    {
        total = total + index;
    }

    print total;
}
```

#### memory

|  |              |   |   |   |   |    |    |
|--|--------------|---|---|---|---|----|----|
|  | <b>index</b> | 1 | 2 | 3 | 4 | 5  | 6  |
|  | <b>total</b> | 0 | 1 | 3 | 6 | 10 | 15 |

#### Steps

1. Initialize *index* to 1.
2. Test the condition. Is *index* <= 5? Yes. Execute the loop body.
3. Update the variable. *index* is now 2.
4. Test the condition. Is *index* <= 5? Yes. Execute the loop body.
5. Update the variable. *index* is now 3.
6. Test the condition. Is *index* <= 5? Yes. Execute the loop body.
7. Update the variable. *index* is now 4.
8. Test the condition. Is *index* <= 5? Yes. Execute the loop body.
9. Update the variable. *index* is now 5.
10. Test the condition. Is *index* <= 5? Yes. Execute the loop body.

11. Update the variable. *index* is now 6.
12. Test the condition. Is *index* <= 5? No. End the loop.
13. Continue with the rest of the pseudocode.
14. Print the value of *total*.

### while loops

#### **format:**

```
while ( condition ) do
{
    loop body
}
```

#### **How does it work?**

Test the condition.

If the condition is **true**, execute the loop body.

If the condition is **false**, exit the loop. Continue with the rest of the program.

#### Example

```
{
    int result = 1;
    int count = 1;

    while ( count <= 10 ) do
    {
        result = result * count;
        count = count + 2;
    }

    print total;
    print count;
}
```

#### **memory**

|               |                  |
|---------------|------------------|
| <b>result</b> | 1 1 3 15 105 945 |
| <b>count</b>  | 1 3 5 7 9 11     |

#### Steps

1. Test the condition. Is *count* <= 10? Yes. Execute the loop body.
2. Test the condition. Is *count* <= 10? Yes. Execute the loop body.
3. Test the condition. Is *count* <= 10? Yes. Execute the loop body.
4. Test the condition. Is *count* <= 10? Yes. Execute the loop body.
5. Test the condition. Is *count* <= 10? Yes. Execute the loop body.
6. Test the condition. Is *count* <= 10? No. End the loop.
7. Continue with the rest of the pseudocode.
8. Print the value of *total* and *count*.