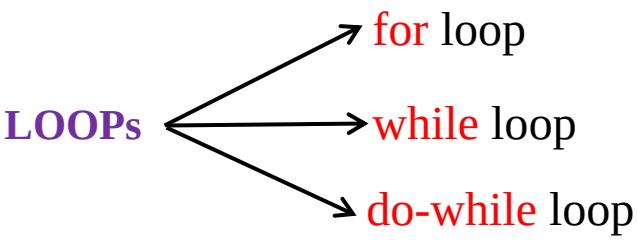
# for Loop in C

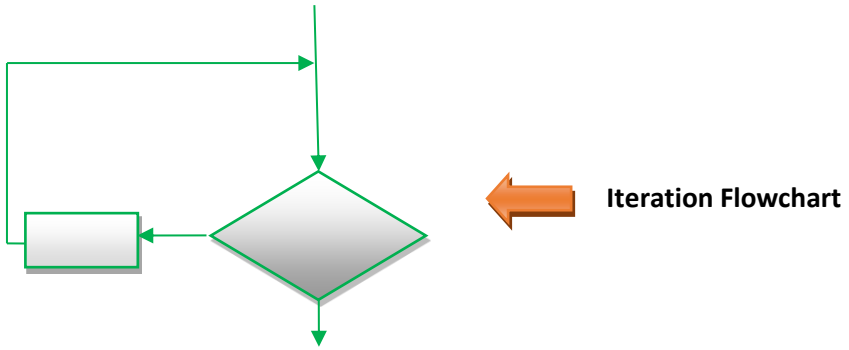
Loops programming construct is used to execute one or more instructions repeatedly until some condition is satisfied.

3 types of loops exist:




## for LOOP

for loop also known as **Iteration.** Iteration logic is used when one or more instructions may be executed several times depending on some condition.




### ITERATION

- Iteration comes from the word "reiterate", which means to repeat
- Iteration is a looping construct
- Iteration is a combination of decision and sequence and can repeat steps
- Iteration can be thought of as "while something is true, do this, otherwise stop"

## for loop syntax

```
for (initialization; condition; increment or decrement)
{
 ..........
 ..........
 ..........
 .                    <-------- program statements
 .
 .
 .
 ..........
}
```

The statements inside for loop gets executed until the condition become FALSE.

For loop also known as **iterative** statement. To iterate means to repeat. If we want to repeat execution of some action or statements several times, we use LOOPs.

## Practice Programs

(i)
```
#include<stdio.h>
void main()
{
        int x;

        for(x = 0; x<5; x++)
        printf("Stephen Hawking\n");
}
```

(ii)
```
#include<stdio.h>
void main()
{
        system("color fc");

        int x;
        for(x = 0; x<5; x++)
        printf("Stephen Hawking\n");
}
```

(iii)
```
#include<stdio.h>
void main()
{
        int x, y;

        for(x = 1; y = 5; x++)
        printf("Stephen Hawking\n");
}
```

(iv)
```
#include<stdio.h>
void main()
{
        int x, y = 10;

        for(x = 2; y == 10; x++)
        printf("Stephen Hawking\n");
}
```

(v)
```
#include<stdio.h>
void main()
{
        int i, x, y;
        x = 2;
        y = 3;

        for(i = 2; x+y; i++)
        printf("Stephen Hawking\n");
}
```

(vi)
```
#include<stdio.h>
void main()
{
        int i, x, y;
        x = 2;
        y = 3;

        for(i = 2; (x+y)>0; i++)
        printf("Stephen Hawking\n");
}
```

(vii)

```
#include<stdio.h>
void main()
{
        int i, x, y;
        x = 2;
        y = 3;

        for(i = 2; (x+y)<0; i++)
        printf("Stephen Hawking\n");
}
```

(viii)

```
#include<stdio.h>
void main()
{
        int i;

    for(i = 2; i<=5; i++)
     {
        printf("Stephen Hawking\n");
        printf("Galaxy\n");
      }
}
```

(ix)

```
#include<stdio.h>
void main()
{
        int i, x, y;

        for(i = 2; i<=5; i++)

          printf("Stephen Hawking\n");
          printf("Cosmologist\n");
          printf("Galaxy\n");
}
```

(x)

```
#include<stdio.h>
void main()
{
        int i, x, y;

        for(i = 2; i<=5; i++)
        {
          printf("Stephen Hawking\n");
          printf("Cosmologist\n");
          printf("Galaxy\n");
        }
}
```

**Program explanation**

Program 1

```
#include<stdio.h>

int main()
{
    int x;

    for(x = 0; x<5; x++)
    printf("Stephen Hawking\n");

    return 0;
}
```

```
for(x = 0; x<5; x++)
        cout<<"Stephen Hawking"<<endl;
```

| Value of x | Condition x<5 | Output |
|---|---|---|
| 0 | 0<5 True | Stephen Hawking |
| 1 | 1<5 True | Stephen Hawking |
| 2 | 2<5 True | Stephen Hawking |
| 3 | 3<5 True | Stephen Hawking |
| 4 | 4<5 True | Stephen Hawking |
| 5 | 5<5 False | • Loop stops here as conditions becomes false<br>• For loop come out of loop when condition becomes false<br>• As long as condition is true, the statements inside for loop gets executed |

Program 2

```
#include<stdio.h>

int main()
{
    system("color fc");

    int x;

    for(x = 0; x<5; x++)
    printf("Stephen Hawking\n");

    return 0;
}
```

Used to change output window background color and text color

system ("color xx");
xx can any of following 2 letters

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | Black | 4 | Red | 8 | Gray | C | Light Red |
| 1 | Blue | 5 | Purple | 9 | Light Blue | D | Light Purple |
| 2 | Green | 6 | Yellow | A | Light Green | E | Light Yellow |
| 3 | Aqua | 7 | White | B | Light Aqua | F | Bright White |

Program 3

```
#include<stdio.h>

int main()
{
    int x, y;

    for(x = 1; y = 5; x++)
    printf("Stephen Hawking\n");

    return 0;
}
```

for(x=1; y=5; x++)
initial value of x = 1
y = 5 is condition
5 is put into y and 5 is non-zero. So, the condition is always true.
So gives infinite loop

Program 4

```
#include<stdio.h>

int main()
{
    int x, y = 10;

    for(x = 2; y == 10; x++)
    printf("Stephen Hawking\n");

    return 0;
}
```

int x, y=10;
for(x=2; y = = 10; x++)

10 = = 10 ⟶ after comparing LHS & RHS, this statement returns TRUE

So, the statement following for loop gets executed infinitely ( ∞ )

Program 5

```c
#include<stdio.h>

int main()
{
    int i, x, y;
    x = 2;
    y = 3;

    for(i = 2; x+y; i++)
    printf("Stephen Hawking\n");

    return 0;
}
```

x = 2;
y = 3;
for(i=2; x+y; i++)

condition x+y = 2+3 =5 is always true
So, this is infinite loop

Program 6

```c
#include<stdio.h>

int main()
{
    int i, x, y;
    x = 2;
    y = 3;

    for(i = 2; (x+y)>0; i++)
    printf("Stephen Hawking\n");

    return 0;
}
```

for(i=2; (x+y)>0; i++)

Condition (x+y) > 0
(2+3) > 0 ⟹ 5 > 0 is always true.
So, this is also example of infinite ( ∞ ) loop

Program 7

```c
#include<stdio.h>

int main()
{
    int i, x, y;
    x = 2;
    y = 3;

    for(i = 2; (x+y)<0; i++)
    printf("Stephen Hawking\n");

    return 0;
}
```

for(i=2; (x+y)<0; i++)

Condition (x+y) > 0
(2+3) < 0 ⟹ 5 < 0 is FALSE.
So, there is no output

Program 8

```c
#include<stdio.h>

int main()
{
    int i;

    for(i = 2; i<=5; i++)
    {
        printf("Stephen Hawking\n")
        printf("Galaxy\n");
    }
    return 0;
}
```

```
for(i = 2; i<=5; i++)
    {
      cout<<"Stephen Hawking"<<endl;
      cout<<"Galaxy"<<endl;
    }
```

| Value of i | Condition i<=5 | Output |
|---|---|---|
| 2 | 2<=5 True | Stephen Hawking<br>Galaxy |
| 3 | 3<=5 True | Stephen Hawking<br>Galaxy |
| 4 | 4<=5 True | Stephen Hawking<br>Galaxy |
| 5 | 5<=5 True | Stephen Hawking<br>Galaxy |
| 6 | 6<=5 False | • Loop stops here as conditions becomes false<br>• for comes out of loop when condition becomes false<br>• As long as condition is true, the statements inside for loop gets executed |

Program 9

```c
#include<stdio.h>

int main()
{
    int i, x, y;

    for(i = 2; i<=5; i++)

        printf("Stephen Hawking\n");
        printf("Cosmologist\n");
        printf("Galaxy\n");

    return 0;
}
```

This example explains absence of braces {}
The first statement below for loop runs 4 times and the loop stops. Then for goes out of the loop and prints next 2 statements

Program 10

```c
#include<stdio.h>

int main()
{
    int i, x, y;

    for(i = 2; i<=5; i++)
    {
        printf("Stephen Hawking\n");
        printf("Cosmologist\n");
        printf("Galaxy\n");
    }

    return 0;
}
```

if we put braces, all 3 statements get executed 4 times