

Loops

Print "Hello" 100 Times

The Ugly Option

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello\n");
5      printf("Hello\n");
6      printf("Hello\n");
7      printf("Hello\n");
8      printf("Hello\n");
9  }
```

A Neat Option

```
1  #include <stdio.h>
2  int main()
3  {
4      for(int i=1; i<=100; ++i)
5      {
6          printf("Hello\n");
7      }
8  }
```

++i is same as writing i=i+1

Terminology

```
1  #include <stdio.h>
2  int main()
3  {
4      for(int i=1; i<=100; ++i)
5      {
6          printf("Hello\n");
7      }
8  }
```

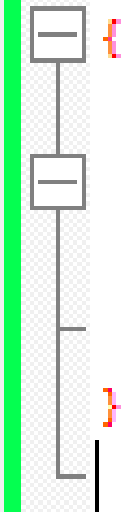
Int i=1; → Loop initialization

i <=100; → Loop Termination
Condition

++i → Loop Increment

Scope of a Variable

```
1  #include <stdio.h>
2  int main()
3  {
4      for(int i=1; i<=100; ++i)
5      {
6          printf("Hello\n");
7      }
8  }
9
```



Since **int i** is inside **for**, it cannot be used outside.

We say that “The **scope** of **i** is within the for { } block”.

Another Way to Declare Loop Initialization Variable

```
1  #include <stdio.h>
2  int main()
3  {
4      int i = 1;
5      for(; i<=100; ++i)
6      {
7          printf("Hello\n");
8      }
9  }
```

This also works! Now **i** can be used anywhere in the **main** function even outside the **for** loop.

Quiz

How many times will this loop run?

```
1  #include <stdio.h>
2  int main()
3  {
4      for(int i = 1; i <=10; i)
5      {
6          printf("Hello\n");
7      }
8  }
```

It never terminates!

While loop

```
1  #include <stdio.h>
2  int main()
3  {
4      int i = 1;
5      while(i <=10)
6      {
7          i++;
8          printf("Hello\n");
9      }
10 }
11
```

Another way of looping!

i++ is known as **Post Increment** whereas **++i** is called **Pre Increment**.

Difference between ++i and i++

```
1  #include <stdio.h>
2  int main()
3  {
4      int i = 1;
5      int j = 1;
6      j = ++i;
7      printf("%d", j);
8  }
```

Outputs 2

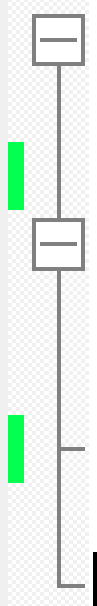
```
1  #include <stdio.h>
2  int main()
3  {
4      int i = 1;
5      int j = 1;
6      j = i++;
7      printf("%d", j);
8  }
```

Outputs 1

Similarly, you may decrement too. i-- will do that.

Yet Another Loop

```
1  #include <stdio.h>
2  int main()
3  {
4      int i = 1;
5      do
6      {
7          i++;
8          printf("Hello\n");
9      } while(i <=10);
10 }
11
```

A diagram illustrating the execution flow of a do-while loop. It consists of a vertical line with two rectangular boxes, each containing a horizontal dash. The first box is positioned between lines 3 and 4 of the code, and the second box is between lines 6 and 7. A vertical line connects the two boxes. A horizontal line extends from the left side of the first box to the left margin. Another horizontal line extends from the right side of the second box to the right margin, where it meets the closing curly brace of the loop on line 9. There are two green vertical bars: one on the left margin between lines 5 and 6, and another on the left margin between lines 9 and 10.

Break Out of Loop

```
1  #include <stdio.h>
2  int main() {
3      char key;
4
5      printf("I will not end till you press X:\n");
6      while(1) {
7          scanf("%c", &key);
8          if (key == 'X')
9              break;
10     }
11     printf("Goodbye!\n");
12 }
13
```

while(1) marks endless loop.

In C, any non-zero value is considered as true.

This Works Too! But, Very Ugly!!

```
1  #include <stdio.h>
2  int main() {
3      char key;
4
5      printf("I will not end till you press X:\n");
6      while(-5) {
7          scanf("%c", &key);
8          if (key == 'X')
9              break;
10     }
11     printf("Goodbye!\n");
12 }
```

Continue: Skip Remaining Lines in a Loop

```
1  #include <stdio.h>
2  int main() {
3      int i = 1;
4      printf("Print 1 to 10. But not 5. \n");
5      while(1) {
6          if (i == 5)
7              continue;
8          printf("%d \n", i);
9          i++;
10     }
11     printf("Goodbye!\n");
12 }
```

So, what did we discuss?

- for loop
- while loop
- post and pre increment
- do while loop

Questions?

Computational Thinking

Match Stick Game

- In this Puzzle there are 21 Match Sticks.
- You and Computer will pick up the sticks one by one.
- Sticks can be picked from 1 to 4.
- **The one who picked up the last stick, is the loser.**

<https://atozmath.com/Games/21MatchStick.aspx>