Loops

Print “Hello” 100 Times

The Ugly Option

```c
#include <stdio.h>
int main()
{
    printf("Hello\n");
    printf("Hello\n");
    printf("Hello\n");
    printf("Hello\n");
    printf("Hello\n");

```

A Neat Option

```c
#include <stdio.h>
int main()
{
    for(int i=1; i<=100; ++i)
    {
        printf("Hello\n");
    }
}
```

++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  }
9
```

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++

```c
#include <stdio.h>
int main()
{
    int i = 1;
    int j = 1;
    j = ++i;
    printf("%d", j);
}
```

Outputs 2

```c
#include <stdio.h>
int main()
{
    int i = 1;
    int j = 1;
    j = i++;
    printf("%d", j);
}
```

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
```

Break Out of Loop

```c
#include <stdio.h>
int main() {
    char key;

    printf("I will not end till you press X:\n");
    while(1) {
        scanf("%c", &key);
        if (key == 'X')
            break;
    }
    printf("Goodbye!\n");
}
```

while(1) marks endless loop.

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

This Works Too! But, Very Ugly!!

```c
#include <stdio.h>
int main() {
    char key;

    printf("I will not end till you press X:\n");
    while(-5) {
        scanf("%c", &key);
        if (key == 'X')
            break;
    }
    printf("Goodbye!\n");
}
```

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