

## The For Loop

The for loop is the most common example of an unconditional loop, and is found in a similar form in many different programming languages. It is a very convenient type of loop to use when a known or measurable number of repetitions must be done.

### Structure of a For Loop

---

```
for( i=0; i<10; i++)           //Loop to display #'s from 0 to 9
{
    printf("The value of i is %i\n",i);
}
```

---

### Important Features of the For Loop

- You do NOT put a semicolon on the end of the for statement.
- The for loop has a **loop control variable**, which in this case is the integer **i**.
- The loop control variable is given:
  - à an initial value (0 in this example)
  - à an end value (9 is the final value in the example)
  - à an way to change its value (the i++ in the example)
- You may put as many operations as wanted between the {}
- Each of the expressions in the round brackets is separated with a semicolon
- Any start, end and change values may be used that are either integer or character values.

### Exercises:

Do each of the following in a simple C program to experiment with the **for** loop.

1. Write a for loop that :
  - Displays all of the multiples of 10 from 10 to 100 in the console.
  - Displays a 'countdown' from 10 to 1 in the console. (use i--)
  - Displays all multiples of 3 from 33 down to 3
  - Calculates the sum of the integers from 1 to 1000 and displays the sum only.
  - Displays the alphabet (use a char for the loop control variable)
2. Write a C program which allows the user to enter an integer value (positive only) for which the program will then display all of the factors of the number.
3. Write a C program which allows a user to enter an integer value (positive only) and the program will identify whether or not the integer is a prime number.