**Conditional Statement**

In a 'C' program are executed sequentially. This happens when there is no condition around the statements. If you put some condition for a block of statements the flow of execution might change based on the result evaluated by the condition. This process is referred to as decision making in 'C.' The decision-making statements are also called as control statements.

In 'C' programming conditional statements are possible with the help of the following two constructs:

1. if statement

2. if-else statement

It is also called as branching as a program decides which statement to execute based on the result of the evaluated condition.

**if statement**

It is one of the powerful conditional statement. If statement is responsible for modifying the flow of execution of a program. If statement is always used with a condition. The condition is evaluated first before executing any statement inside the body of If. The syntax for if statement is as follows:

if (condition)

{

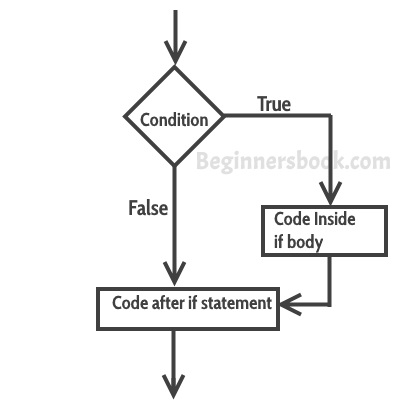
//Block of C statements here

//These statements will only execute if the condition is true

}

The condition evaluates to either true or false. True is always a non-zero value, and false is a value that contains zero. Instructions can be a single instruction or a code block enclosed by curly braces { }.

### Flow Diagram of if statement



### Example of if statement

#include <stdio.h>

int main()

{

int x = 30;

int y = 43;

if (x<y)

{

printf("Variable x is less than y");

}

return 0;

}

**Output:**

Variable x is less than y

**Explanation:** The condition (x<y) specified in the “if” returns true for the value of x and y, so the statement inside the body of if is executed.

## if else statement

If condition returns true then the statements inside the body of “if” are executed and the statements inside body of “else” are skipped.  
If condition returns false then the statements inside the body of “if” are skipped and the statements in “else” are executed.

if(condition)

{

// Statements inside body of if

}

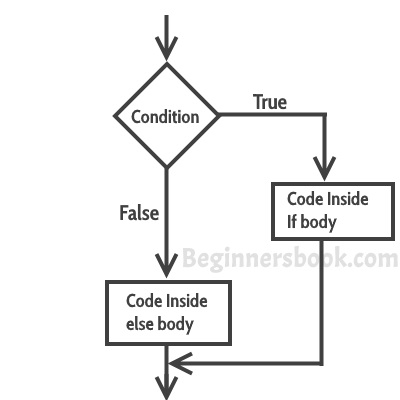
else

{

//Statements inside body of else

}

### Flow diagram of if else statement



Following programs illustrate the use of the if-else construct:

We will initialize a variable with some value and write a program to determine if the value is less than ten or greater than ten.

Let's start.

#include<stdio.h>

int main()

{

int num=19;

if(num<10)

{

printf("The value is less than 10");

}

else

{

printf("The value is greater than 10");

}

return 0;

}

Output:

The value is greater than 10

## Nested if-else Statements:

When a series of decision is required, nested if-else is used. Nesting means using one if-else construct within another one.

Let's write a program to illustrate the use of nested if-else.

#include<stdio.h>

int main()

{

int num=1;

if(num<10)

{

if(num==1)

{

printf("The value is:%d\n",num);

}

else

{

printf("The value is greater than 1");

}

}

else

{

printf("The value is greater than 10");

}

return 0;

}

Output:

The value is:1

#include<stdio.h>

int main()

{

int A=100,B=20,C=999;

if(A>B)

{

if(A>C)

{

printf("A is largest");

}

else

{

printf("C is largest");

}

}

else

{

if(B>C)

{

printf("B is largest");

}

else

{

printf("C is largest");

}

}

return 0;

}

#include<stdio.h>

int main()

{

char T;

printf("enter one character");

scanf("%c",&T);

if(T=='a'||T=='A'||T=='e'||T=='E'||T=='i'||T=='I'||T=='o'||T=='O'||T=='u'||T=='U')

{

printf("entered character is Vowel");

}

else

{

printf("entered character is Consonant");

}

return 0;

}

#include<stdio.h>

int main()

{

int N;

printf("enter number");

scanf("%d",&N);

if(N%2==0 && N%5==0)

{

printf("entered number is divisble by 2 and 5");

}

else

{

printf("entered number is NOT divisble by 2 or 5");

}

return 0;

}

#include<stdio.h>

int main()

{

char T;

printf("enter one capital alphabet");

scanf("%c",&T);

T=T+32;

printf("%c",T);

return 0;

}