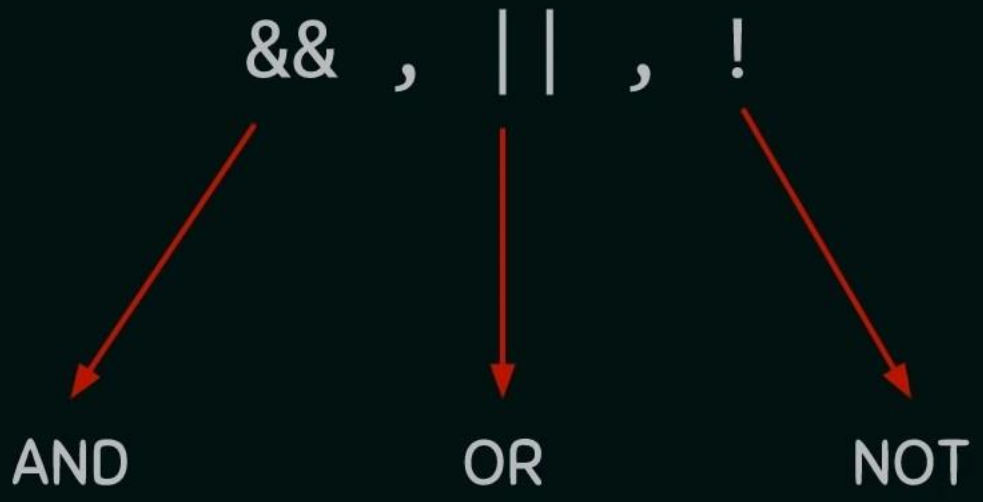
Logical operators



Logical operators

- && and || are used to combine two conditions.

  && – returns TRUE when *all* the conditions under consideration are true and returns FALSE when any one or more than one condition is false.

  For example:

```
int a = 5;

if (a == 5 && a != 6 && a <= 56 && a > 4)
{
    printf("Welcome to this beautiful world of operators");
}
```

Logical operators

- && and || are used to combine two conditions.

  || – returns TRUE when *one or more than one* condition under consideration is true and returns FALSE when all conditions are false.

  For example:

```
int a = 5;

if (a != 5 || a == 6 || a >= 56 || a > 4)
{
    printf("Welcome to this beautiful world of operators");
}
```

Logical operators

- ! operator is used to complement the condition under consideration.

  !- returns TRUE when condition is FALSE and returns FALSE when condition is TRUE.

  For example:

```
int a = 5;

if (!(a == 6))
{
    printf("Welcome to this beautiful world of operators");
}
```

Concept of short circuit in Logical operators

Short circuit in case of &&: simply means if there is a condition anywhere in the expression that returns false, then the rest of the conditions after that will not be evaluated.

```c
#include <stdio.h>

int main() {
    int a = 5, b = 3;
    int incr;

    incr = (a < b) && (b++);
    printf("%d\n", incr);
    printf("%d", b);
    return 0;
}
```

False = 0
True = 1

Concept of short circuit in Logical operators

Short circuit in case of | |: simply means if there is a condition anywhere in the expression that returns True, then the rest of the conditions after that will not be evaluated.

```c
#include <stdio.h>

int main() {
    int a = 5, b = 3;
    int incr;

    incr = (a > b) || (b++);
    printf("%d\n", incr);
    printf("%d", b);
    return 0;
}
```