

* Bitwise operator:

The bitwise operator act on bits and perform bit-by-bit operations. They operate on binary numbers.

A bitwise operators ($\&$, $|$, \sim , \wedge , \gg , \ll)

$$Ex = 2^n$$

2^4	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	
16	8	4	2	1		8	4	2	1	(binary convert)

$$0000 = 0$$

$$0001 = 1$$

$$0010 = 2$$

$$0011 = 3$$

$$1000 = 4$$

$$1001 = 5$$

$$1100 = 12$$

$$1111 = 15$$

$$1100 = 12$$

$$1010 = 10$$

Ex: do bitwise operation for $15 \& 12$, $15 \vee 12$

$$\Rightarrow \begin{matrix} 1 & 1 & 1 & 1 & = 15 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 0 & 0 & = 12 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 0 & 0 & = 12 & (\text{and}) \end{matrix}$$

$$\begin{matrix} 0 & 0 & 1 & 1 & = 3 & (\text{or}) \end{matrix}$$

$$Ex: 10 \& 4$$

$$\Rightarrow \text{Binary value } 1010 \& 0100$$

$$1010 = 10$$

$$0100 = 4$$

$$\underline{0000} = 0 \text{ (and operation)}$$

Ex: $13 \gg 2$

$$\begin{array}{r} 1101 = 13 \\ \swarrow \searrow \searrow \\ 0010 = 2 \\ \cdot 01 \end{array}$$

Ex: $13 \gg 2$ ($\gg = \searrow$)

$$\begin{array}{r} \text{TE H T U} \\ 1101 = 13 \\ \swarrow \searrow \searrow \searrow \\ 00101 = 3 \\ \text{ignore} \end{array}$$

Ex: $6 \ll 2$ ($\ll = \swarrow$)

$$\begin{array}{r} 0110 = 6 \\ \text{Lack TE TH H T U} \\ 011000 = 24 \end{array}$$

* Operator procedure:

Ex:

$$2 + 7 \times 3$$

and not

$$\rightarrow 2 + 21$$

$$\rightarrow 9 \times 3$$

$$\rightarrow 23 \text{ (result)}$$

$$\rightarrow 27 \text{ (wrong)}$$

* This is bcoz (*) has more procedure than (+). if we want to execute (+) first then we should use parenthesis.

Ex:

$$(2 + 7) \times 3$$

$$\rightarrow 9 \times 3$$

$$\rightarrow 27 \text{ (result)}$$

* Procedure Table:

(), **, +x, -x, nx, x, /, //, %, +, -, <<, >>, &, ~, ', ==, !=, >, <, <=, is, is not, in, not in, not, and, or

Conditional Statements :

In programming, there are situations where we need to make some decisions, whether to execute blocks of code or not.

- * It allows us to make decisions in code.
- * They check conditions (Expression that result in True or false and execute different blocks of code accordingly).

* Types of Conditional Statement in python:

(i) The IF - STATEMENT:

A block of statement would be executed if the condition given is True, if false the block is skipped.

Syntax:

if (condition):

Statement (#only executed if True, otherwise skip)

Ex: a = int(input("Enter a num:"))

if (a > 5):

print("you entered num is > 5")

O/P:

Enter a num = 10

you entered num is 10

Alt O/P:

Enter a num = 3

Skipped

(ii) IF ELSE STATEMENT :-

The Else block is an alternative to if block which executes only when the condition of if block becomes false.

* The Else block doesn't have any condition.

Syntax

if (condition):

Statement

Else:

Statement

Ex: age = int(input("Enter a age:"))

if (age > 18)

print("Eligible to drive")

print("Apply for license")

Else:

print("Not Eligible to drive")

print("Try after turning 18")

print("Thank you")

O/P

Alt O/P

Enter a age = 16

Not Eligible to drive

Try after turning 18

Thank you

Enter a age = 22

Eligible to drive

Apply for license

Thank you

The IF - ELIF - ELSE LADDER:

- * If there are more than 2 scenarios or conditions then if-elif-else block is used.
- * The ladder starts with an if block, then we write as many elif blocks needed as per requirement, finally the else block is written.

Syntax:

if (condition):

Statement

elif (condition):

Statement

elif (condition)

Statement

else:

Statement

Ex:

```
pe = float(input("Enter your pe:"))
```

```
if (pe > 90):
```

```
    print("Grade A")
```

```
elif (pe > 60):
```

```
    print("Grade B")
```

```
elif (pe > 40):
```

```
    print("Grade C")
```

```
else:
```

```
    print("fail")
```

O/p:

Enter your pe = 95

Grade A

Alt O/p

Enter your pe = 46

Grade C

(iv) The NESTED If : (using one if inside another)

An if statement can be nested, i.e. An if statement can contain another if statement,

Syntax :

if (condition) : # Outer if

if (condition 2) : # Inner if
(Statement)

Else :

Statement

Else :

Statement

Ex: write a program to check the given input is +ve
-ve or zero

=> n = int(input("Enter a num:"))

if (n >= 0) :

if (n > 0) :

print("+ve")

Else :

print("Zero")

Else :

print("-ve")

O/P :

Enter a num = 5

+ve

Ex : NESTED :

11.1 Grading System :
write a python program that takes marks as input.
if marks $\geq 90 \rightarrow$ print "Grade A"
Else if marks $\geq 75 \rightarrow$ print "Grade B"
Else if marks $\geq 50 \rightarrow$ print "Grade C"
Else \rightarrow print "fail"

\Rightarrow

```
marks = int(input("Enter your marks:"))  
if (marks >= 50):  
    if (marks >= 90):  
        print("Grade A")  
    elif (marks >= 75):  
        print("Grade B")  
    else :  
        print("Grade C")  
else :  
    print("fail")
```

O/P :
Enter your marks = 92
Grade A