

AP $a_1 \rightarrow$ $a_n \rightarrow$ $\{ \text{Source} \}$ $\xrightarrow{\text{Slicing} \rightarrow \text{set}}$ $\xrightarrow{\text{String}} \xrightarrow{\text{List}} \xrightarrow{\text{tuple}}$

Python Input
Step
+ve $\rightarrow 0$
-ve $\rightarrow -1$

$a : b : c$
 $\downarrow \quad \downarrow \quad \rightarrow \text{Step size}$
 Starting Index Step size end index
 +ve $\rightarrow b-1 \leq$ $a < b$
 -ve $\rightarrow a > b$
 $-ve \rightarrow b+1 \leq$

$$-16 : -3 : -2$$

① Stepsize sign -ve $a > b \rightarrow \underline{\text{True}}$
 $-16 > -3 \rightarrow \underline{\text{False}}$

No Expression
 $\overbrace{\quad \quad \quad}^{\text{ }}$

$$3 : 19 : 3$$

Step 1 \rightarrow +ve $a < b \rightarrow \underline{\text{True}}$

$$3 < 19 \rightarrow \text{True}$$

Step 2
last value
 +ve
 $\rightarrow b-1$
 $\rightarrow 19-1$
 $= 18$

$a_1, a_1+d, a_1+2d, a_1+3d, \dots, a_1+nd$

$\boxed{3, 6, 9, 12, 15, 18} \times$

$$-3 \quad -27 \quad -3$$

① $\stackrel{-ve}{=}$

$$-3 > -27 = \underline{\text{True}}$$

② $\begin{array}{c} -ve \\ b+1 \\ \hline -27+1 \end{array}$

$$\boxed{-3, -6, -9, -12, -15, -18, -21, -24} \times$$

$$= -26 \quad \checkmark$$

$$a = 'N^{\circ 1 2 3 4 5 6} I S H A N T'$$

$$b = a[3 : \underbrace{(len(a) : 2)}_{3 : 7 : 2}]$$

$$\boxed{3, 5} \times$$

$$b = 'H N'$$

$$x = x[::16::2]$$

+ve
step
size
 \Rightarrow
 $a = 0$
 $b = 16$
 $c = 2$

$$x[::::3]$$

$a = 0$
 $b = \text{len}(x)$

$$c = 3$$

$$x[::] \text{ or } x[:, :]$$

$a = 0$
 $b = \text{len}(x)$
 $c = 1$

-ve step size

$$x :: -1$$

$$x \rightarrow -1$$

$$y \rightarrow -(len(x) + 1)$$

* Concatenation seq →
 → str
 → list
 → tuple

Multiplying A seq
With Integer
Numbers.

$$\Rightarrow [1, 2, 3]$$

$$111111$$

$$(1, 2) * 2$$

$$\Rightarrow \underline{(1, 2, 1, 2, 1, 2, 1, 2)}$$

left + right
Same Data type

Str + Str → Str

List + List → List

tuple + tuple → tuple

String

! Nishat123 !
" Nishat 123 "
\n → newline
\t → tab
\ → escape char

✓ # " " " " = / / / / }
✓ # " " = " "



Python Keywords

Keywords are special reserved words which convey a special meaning to the compiler/interpreter. Each keyword have a special meaning and a specific operation. List of Keywords used in Python are:

True	False	None	and	as
asset	def	class	continue	break
else	finally	elif	del	except
global	for	if	from	import
raise	try	or	return	pass
nonlocal	in	not	is	lambda



Python Identifiers

File, dirname

These can be variables, class, object, functions, lists, dictionaries etc.

There are certain rules defined for naming i.e., Identifiers.

- ✓ I. An identifier is a long sequence of characters and numbers.
- ✓ II. No special character except underscore (_) can be used as an identifier.
I module Name
- ✓ III. Keyword should not be used as an identifier name.
- ✓ IV. Python is case sensitive. So using case is significant.
- ✓ V. First character of an identifier can be character, underscore (_) but not digit.

✓ [A-Z a-z 0-9 _]

Var-1 = ✓
Var = ✗

-Var ✓



Python Operators

$$5/2 = 2.5$$

$$5//2 = \underline{\underline{2}}$$

Python supports the following operators:

- Arithmetic Operators. **Arithmetic Operators:**
- Relational Operators

Operators	Description
//	Perform Floor division(gives integer value after division)
+	To perform addition
-	To perform subtraction
*	To perform multiplication
/	To perform division
%	To return remainder after division(Modulus)
**	Perform exponent(raise to power)

$$\begin{array}{l} 2)5(2 \\ \underline{-4} \\ \hline 1 \\ \hline 5)2(1 \\ \underline{-2} \\ \hline 0 \end{array}$$

$$2^2 \rightarrow 4$$

$$2+2 \rightarrow 4$$



Python Operators

Relational Operators:

True / False

Conditions

Operators	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to



Python Operators

Assignment Operators:

$$\begin{aligned} a &= 5 & \rightarrow 5 \\ \text{print}(a) & & \\ a &+= 5 & \rightarrow [a = a + 5] \\ \text{print}(a) & \rightarrow 20 \end{aligned}$$

Operators	Description
=	Assignment
/=	Divide and Assign
+=	Add and assign
-=	Subtract and Assign
*=	Multiply and assign
%=	Modulus and assign
**=	Exponent and assign
//=	Floor division and assign

Logical Operators:

Operators	Description
and	Logical AND(When both conditions are true output will be true)
or	Logical OR (If any one condition is true output will be true)
not	Logical NOT(Compliment the condition i.e., reverse)

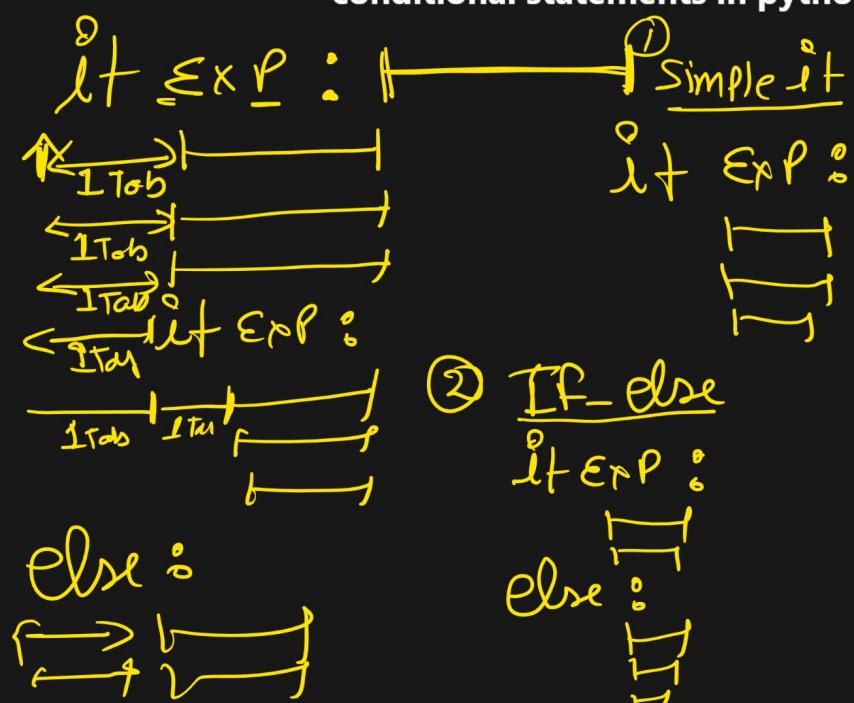
$$\begin{array}{cc} T & \xrightarrow{\text{not}} F \\ F & \xrightarrow{\text{not}} T \end{array} \quad \boxed{\text{not } q}$$

a	b	$\stackrel{\text{and}}{=}$	$a \cdot b$
F	F		F
T	F		F
F	T		F
T	T		T

OR

a	b	$a+b$
F	F	F
T	F	T
F	T	T
T	T	P

conditional statements in python



③ Chained + Elif

if Exp1 :

[]

elif Exp2 :

[]

elif Exp3 :

[]

else :

[]

① Simple If

```
if <conditional>:  
    <Statement to be executed if the conditional evaluates to be True>
```

② if-else

```
if <conditional>:  
    <Statement 1 to be executed if the conditional evaluates to be True>  
else:  
    <Statement 2 to be executed if the conditional evaluates to be False>
```

③ if-elif-else

```
if <conditional 1>:  
    <Statement 1 to be executed if the conditional 1 evaluates to be True>  
elif <conditional 2>:  
    <Statement 2 to be executed if the conditional 2 evaluates to be True and conditional 1 evaluates to be false>  
else:  
    <Statement 3 to be executed if the conditional evaluates to be conditional 1 and 2 evaluates to be false>
```

Prob 1. W.A.P To Take Input 0-100 and create Conditions

Note: Also Handle the odd Inputs.

→ $10 < \text{var} < 60$

↳ $\text{var} > 43$

The value $\{\text{var}\}$ is large!

$35 < \text{var} < = 43$

The value $\{\text{var}\}$ is medium.

The value $\{\text{var}\}$ is small.

→ The value $\{\text{var}\}$ not in range $10 < \text{var} < 60$

Var

- Input
- ① 43.5
 - ② Absurd (23)
 - ③ (1, 2, 3)
 - ④ 188 → The value $\overset{\circ}{\text{is}}$ out of range (0-100).
 - ⑤ 90 - ✓
 - 46 - ✓
- Enter the valid Integer
Number r