# Python - Module 5: Operations and Expressions

## Python

# Module 5: Operations and Expressions

## Video: Arithmetic Operators

Hello, welcome back to the Python session.

In this session we are going to discuss about

automatic operators, different operator.

First we see different operators in Python.

So basically these operators are different types like uh,

automatic operators,

relational operators,

large scale operators,

and next category thematic assignment operators

and one more bitwise operators.

In this session, mainly we ate on automatic operators.

I'm ing notes for you letter and download this notes.

Let's come into the automatic operators.

So plus this is for addition.

So minus depends Here the symbol is hyen.

This is for subtraction.

There is a hashtag symbol. This is for multiplication

slash this is for division

And double slash, this is for floor division.

What exactly each one I'll explain

there is a percentage symbol.

This is a modular operator.

This is the remainder calculation

and still we have other operators

double star.

That is double symbol. This is to find power values.
Simply, you can say it as a power operator.
So these are the automatic operators using Python.
So for each one, example
plus for addition,
suppose your statement is your expression is 10 plus 20.
It returns. But
suppose you assigned value 10 to the A and 22 B.
If your statement is C equal to A plus B.
If you said print something like sum of
A and B
is C, what is the expected output
for this sum off?
10 and 20.
Is
that okay?
Similarly, theen symbol which is minus,
this is for software action,
10 minus 20.
We get minus 10, a hundred minus 70,
we get 30 programmatically if we want to say equal
to 70 something B equal to 20.
Now C equal to A minus B.
Now if you print it C,
what is 70 minus 20 here?
That 50, the value of C is 50.
Similarly, you want to multi, you want
to perform multiplication operation.
Then opera operators has take symbol
simply you can say it as a star.
The star mark.
This is for multiplication.
For example, 10 has 3 23.
It returns 200.

I'm taking your salary by example.

Your salary is one line.

I want to compute tax value.

Suppose 10% of tax. I want to apply salary.

Two point 10%.

Point 10 by a hundred means that point.

Now what is the tax value here as

for this 10,000?

Okay, similarly, house intelligence, HRA want to compute

salary into point to something 20%.

I'm applying so that the value

of HRA will be here 20,000.

Now I want to compute by using all expressions.

I want to compute net salary where it is salary

plus HRA minus tax.

What is the value here for the salary?

10,000 plus 2000 of HRA minus th 10,000

of 20,000 HRA 10,000 tax.

The total will be one lack

11,000, one lack 10,000.

Okay? This is the reason

that way you can use automatic operations

to perform different competitions.

Still more operators, we have step

by step wheel

slash This is where

division, some

of I said 10 slash five.

That will stand by five. What is the result? Two.

Okay, suppose five by two,

you get two five.

Similarly, double slashes.

This is called a flow division

10 double slash two by seven.

What is the meaning of it?

Divide 10 with the two by two.

Two fives tab. What is the remainder values? Zero.

Now this is low division value.

So what it exactly,

for example, if I say

15 by six, what is the result we get?

6, 2 12 Still please. There.

Point six.

In software using the slash, I'm going to use

double slash six.

What we get? 15 six.

Six two. Remainder value three.

This two is loaded simply against say quotient.

Okay, this

is got it.

So 15 slash two is two point something

where it returns only two.

That is a difference between slash

and w slash

in the next expressions spot we see how

to use them in into the programming

and one more operator is spending,

other two operators are spending.

We need to discuss person.

Then this is where, what is this? Operator's score.

This operator is score mod below operator.

This is to find remainder value.

This same previous example, I'm taking

16 slash three.

What are the result? Five something.

If I say 16 double slash three, it gives you

five 16.

That model operator. What are the remainder?

I divide 16 with

by

3, 3, 5, 15 for the remainder.

This is the remainder

what I got here, not this expression.

So this is the difference between slash double slash

and this model operator, the passengers.

The last operator here,

double star.

This is to find power values

for example two.

The of five I one. What is the meaning of two?

The power of five

two.

So how to get this two double star.

That's it.

Okay, so what is the value?

Two. Two. Two is four. Four. Eight. 2 2 8 18.

Two 16. 16 eight. Two.

That it

that it not 30.

That it? Q

eight again.

Two. Two for 16. 16 to do 30. Correct.

So similarly, one more thing you want to find out.

Square, by using the same technique,

for example square root five, how

to give the python expression for this

without using any predefined functions.

Of course we have predefined functions in pythons letter.

We discuss about those functions actually fight

to five square root of five minutes.

What is the meaning? It's a fight

to the in python.

How do to get the expression for this five double star
one by two
or you can say file double star in the place
of one by two.
I'm giving point. This is the meaning of square off.
So this is about simply
automatic operators used in the python.
In the next session, we are going
to discuss about a relational operators.
Thank you very much. See you in the next session.

## Video: Relational Operators

Welcome back to the Python session.
In this session we are going
to discuss about relational operators.
So generally these relational operators are used
to compare two variables or two values.
These are used to
compare two variables
or values.
So for
comparison, we have different operators here.
These operators are called relational operators
and also we can call them as conditional operators.
And one more name, but these operators,
comparison operators,
comparison operators.
So below the list of different operators,
one is w equal to
not equals to
greater than.
Greater than or equals to less than

less than our equals.
So these are the six operators used
for the comparing two variables.
So the first one W equal to this is
for equality.
Check
if the, if the values are equal, it reads true,
otherwise false,
it rids billion value, which is true.
For example, I'm saying 10 equal to
20, the condition is false.
Here it for example,
10 W equals to 20, sorry, 20 is done.
Let's
next operator not equals.
For example, 10 not equals to 20.
This time what will happen? The 10 is the not equals to 20.
That's why it for example,
10 not equals to 10.
Now in this case it
and whether value are bigger than the current value
or not, you want to check
What they see.
How greater than,
so suppose age, I'm taking a variable.
Age is 25, condition is greater.
20. What is the value
of H 25?
25 greater than 20, but this is true.
It returns
similarly is greater than 30.
Amp.
A is greater than 30. It returns false
because value is 25.

25 greater than 30. It is false.
This is about not equals to any greater than
and one more thing
greater than or equals indicates
that greater or equals
greater than or equals.
Example, I'm taking your salary,
I'm assigning value one like to the cell.
The tradition is like this salary greater than
Oracles two 50,000.
True or false is one like actually one like is greater than
Oracles two 50, which is true.
One more example on the same salary
greater than Oracles two,
one like Actually here.
S is that is greater
than other equal to one likeness.
Greater or equals anything is okay.
But yeah, Sally is not greater than one
like but equals to one.
Like still it is
the loss condition.
Sally, greater than our equals two likes.
Now here the value one, like one leg greater than another
to two likes, condition, fault returns, false.
Excuse me. The next operator here,
less than,
let's talk about it every age.
Again, the age is 25.
My given conditions like this is less than 30.
Just substitute here. The 25 is less than 30.
The condition is, and one more
statement is less
than 25.

There is value 25, 25, less than 25.
That is false because both are equal.
Similarly, less than are equal to operators,
less than are equals.
Same age. I'm saying less than Oracle equals to 25.
Now in this case it,
okay, so the value can be less be equals.
Anything is okay. That is what less than are request.
So these are all the Alaska operators.
Generally user conditions,
whenever we want to check, sign any condition logical that,
that the relational operators are used.
Okay, so another name
for these relational operators in three different names you
can use the other name relational operators.
What is another name?
Comparison operators,
another name conditional operators.
So use it to check conditions.
But in all these examples, we have used only one condition.
Whenever we want to use more than one condition, we have
to go for large scale operators.
So basically large scale operators are used to
check multiple conditions.
In our next session completely, we discuss about
large scale operators.
Thank you very much. Let's meet in the next session.

## Video: Logical Operators

Welcome to Python session.
In this session we are going
to discuss about logical operators.

Logical operators, basically three

and or not.

So what exactly These things we'll discuss now

and if all

conditions are true, final result is

this is correct

and final result is true

in case of our, if at least one condition is true,

that means any one condition,

if at least one condition is true.

So final result is

not simply isn't a negation of the,

that means not of true is false.

Similarly opposite, not a false is true.

Let's say

let's discuss more on this topic

for easy understanding.

I'm writing on end table.

Let's say condition two conditions I'm taking

for easy understanding

condition one,

condition two and final result.

So here I'm using symbols for true capital.

Ty for false capital, capital at array of five minutes

for example, false condition is true.

Second condition also true.

Then final result is look good to the second rule.

True and false. But what is the rule for that?

All conditions must be true,

but here only one condition is two.

Second one is false. So the result is here. False.

Similarly, false and false.

False and false. Definitely the false

look at tro all the given conditions are true.

Then only that big end but this is adopted.
I'm taking about or same kind of table.
I will consult for our operator.
So it is condition one,
condition two, and final result.
So let's look into this. True, true, true.
Guess how far true false one true enough total thing will
become to false.
True, false.
False. This is false.
Based on this you can understand one thing.
If all conditions are false then only final result is false.
This is the case for our operator. Okay,
next one is not, not a simply negation
the condition if any condition is true,
but that if you apply the negation like not the threatens
false simply negation.
For example, salary value 10,000
are one lack my condition.
Is salary greater than one lack?
Actually the value is one lack.
One lack greater than one lack, definitely that is false.
But the same thing, if you apply the negation not
of salary greater than one lack,
now it returns true.
Okay, so not of false is true not
of tool is false like that.
These are the three logical operator operators.
So generally these logical operators are used to apply
for more than one condition.
For example, condition one
and or all condition.
This is a fault after this one more hopper,
one more function I have to use means one more condition.

You want to apply again end or all then condition.
So here of the release example,
there are three conditions between each condition.
One operator again go
between condition two and condition three.
Another last operator. If I have young conditions,
we have to give young minus one
to last kilometers
either end or on.
So well all conditions must be true.
This is whatever expectation. Then go for end.
Anyone condition is true.
Can you go for or right
on That way you can decide these things
look into different examples based on this.
So suppose I want
only males data
then what is the condition for this?
Think that there is a color variable called gender.
The condition is gender equals to
let's say the gender values.
Gender values are like BF, like that gender equals
to um, that is a condition.
So here no need to use any logical operator
because single condition,
it's a single condition.
No need to use logical operators.
One more example, just look into this.
So from the given data, what I want is I want males from
from high travel
for example, this value variable is gender
and city value is location.
Now condition for the gender based gender male screens the
condition is gender equals to

double equals to because you are comparing.
Yeah. And one more condition for the location,
location equals pattern.
Now condition one condition I'm
keeping And operator here.
Okay, so this is condition one
and this is conditioned both should be true.
That's why. And operator for example,
the details are like this.
His name is Ravi. He is a male from Pune.
What happened First condition through
second condition,
location equal toba actually location value Pune Pune equal
toba is false.
True and false. What will happen? False.
So that this record will be that resected
like this.
One more test we do for the sale.
Previously we said gender is equal to male
and location equal to
the operator.
I'm giving you same previous condition I'm take
check for under person.
Suppose her name is Lata, female
from Hydrovac whatever
first condition, true second condition to
final, that means
accepted similarly.
Gary, one more record. I'm testing mail.
Let's say Gary J this letter,
this in this case both conditions.
First condition falls and secondary condition is also false.
Finally, what false So that it ize
in this whenever we are giving multiple conditions,

then only how to use last scale operator

of course in this example and is are right.

Similarly, one example I'm taking for our operator,

I was after this example.

I want, okay,

I want from,

I want all records.

I mean I want all the employees from

and Dell.

So remaining city people I don't want

now observe here how to give the condition,

tell me the condition for hydro.

The condition will be like this location equal to

hydro

Location equal to Pune.

That is second condition

location equal to de

this is now here it is condition one,

condition two condition.

Again more than one condition between them operator circuit,

but so here right operators, okay?

Because cities work for example, if you are from,

definitely are not from the remaining cities.

So I'm from Delhi, definitely not from first.

How is the chance here? Any one

condition can be true at a time.

All conditions at a time cannot be true over here.

That's why R is the right operator.

Look at this, Mr.

From whatever the condition, first condition,

true or second.

False or false.

One true enough, final result is true.

For example, she is from Delhi.

First one is false or second is false
or thought condition is true.
Final letters and one more Mr.
John
from chairman.
What happens here? First is false second, false
third also false.
So the final is that false
because no one, no condition is true here.
So this is the example for the our operator.
So I tell you one situation for not operator
for example if the requirement is like this,
I want all employees
except from
Aaba tell you
that they have employees from hundred different cities.
Accept these three remaining you want then
apply the negation condition.
Three ways you can do two ways.
Wave one already have a condition.
Suppose only hba want, only
people want already have planned a condition.
Location equals to aaba
or location equals to
Pune
or location equals to.
So now you'll get only the people from Pune and Delhi.
I'm going to apply. I don't want these people then they have
to say not okay, this is
so finally tell this
Ravi from Puni, what about this?
This condition false, this condition true,
this condition false between them
what is operator final result.

True. So here we are applying, we are applying not not
of truth is false from Pune.
Actually don't want Pune because except from Pune.
Except from Pune. Delay one remaining people.
So that we are in this case Ravi, that is not
of true is false, not a false is true.
Okay? So in another style we can apply this
it the not symbol
look into the second way for the same question.
So look into the question again, the task.
I want all employees except from Aaba daily. Now, right?
Independent condition, I don't want from Aaba.
What is the condition? Location not equals to
aaba condition.
I don't want two. Then condition, location
not equals to
location not equals to.
So here I don't want this,
I don't want this, I don't want this.
At time all the three conditions should be trooper.
That's why here operator?
Yeah, yeah,
this see wait, got
Uh, retest that from Pune.
What about the first condition?
Pune articles through une, articles to Pune false
location articles to delete through who And false
and true in case of and all things should be true, right?
But here this is false. So that final result is
false because of false.
This record is rejected. Got it.
For example, your name is running,
we are from China for this what happens,
this condition, true Shana article, true

articles to delete true, true and true and true.

Finally, final result is true.

So that ran from Chennai is accepted like

that you can apply the different things.

Now we have seen separate examples per end,

separate example per r and also separate examples per not.

I want to use the combination of end, end or operators.

Look into this

we can use the and or combinations.

The task here,

I want male employees

from Hyderabad, Delhi, Pune, Hyderabad,

Delhi, same thing.

Females from

Pune, Mumba

and all always which includes both males and female.

All from,

so this is my requirement, think

that there is a location column

or C uh, there is other columns like uh,

variables like a city and gender.

Now I want to apply this.

Let we go to the next screen for this.

Look back to the tasks.

Males from Del Postal condition only for the males

gender equal to male

and you want, you want males from haba and deities.

So that condition city equals to Haba.

That is one condition.

City equals two in Delhi.

This is second condition again here.

Condition one, look at only the city

condition one, condition two.

Your city cannot be two, your city can be either or Delhi or not of anyone.

The chances, anyone. That's why here operate

risk or operate.

Now you see the center thing is condition one, the center

part is conditioned again between these two things.

Alaska operator here I'm keeping the operator and

because here gender should be male at the same time.

City can be either higher above or so

and or commerce.

This is condition number one. What about the females?

Look into the task. I want females from Mumbai.

Then condition should be gender is equal to female

and city

equals to une

or city equals to double equals to.

Whenever you are checking equality,

that should be double equal to operate.

Okay, double equal

number

Now this is condition one conditioned

and one more thing you need to concentrate for from China.

I want all people. That means from Sinai,

I want males, females, both.

That's why no need to give you want,

you are expecting all genders.

That's why no need to give any gender specific

condition directly.

The thought condition is city equals

in this case.

Condition one, condition two. This is condition three.

Again, multiple conditions between these conditions.

Again, laws per, for example,

you are a male from definitely will not be in into

next two categories.

For example, you are from haii,
but definitely you are not into foster two categories,
so your chances you can be at any one place.
That's why here one more operator or one more operator
or, so the center thing is condition one,
this is condition two.
This is condition between these three things.
Again, large scale. Now observe many conditions are normally
the two we have used to mix of and, and our
and our operators complex.
Okay? This is simply about logical operators.
In the next session, we learn about how
to use all these operators in the form
of expressions, practical use.
Okay, thank you very much. Let's meet in the next session.
Our topic is writing expressions with Python practically.
Okay, let's do the lab exercise. Lab exercise on the back.
Thank you very much.

## Video: Arithmetic Expressions Lab

Sessions we have discussed about the different types
of operators, like automatic operators,
relational operators, and logical operators.
Today in this, the practical part of uh, how
to give the expressions we see, okay,
so we have different types of expressions.
So these expressions can be classified as
I read the same thing on the notes.
Then we see the practical part of it.
What type of expressions are automatic expressions?
The second category, relational expressions,
thought logical expressions,

the both category, membership expressions
and fifth category bitwise
identity Before that, identity expressions
and six five BITWISE expressions.
So let's see all these operations step by step.
Now I'm presenting this lab demo to the Google collab,
okay, so which is organized by the Google.
Google. So it provides a runtime computing mission
to run your code even
for machine learning, such kind of operations.
Also it provides G missions
and TP missions so that without having to any infrastructure
and without having to install anything, okay,
you can execute all the python port.
So already we discussed the different
automatic operators faster.
Let me list off those operators. One is plus minus.
The iPhone symbol is by the subtract.
Next asterisk symbol simply says star.
This is for multiplication slash for division.
What is this? Score module, low operator
and a double slash module L is for to find the remain.
A double slash is for floor.
The patient takes quotient.
When you divide some number with the other number,
the ENT part will take it.
And to find out the power value you have double stars.
So that is to find out the power values.
So now let's see the expressions here.
For example, I'm assigning some value to the variable
B, equal to something A three.
Now A plus B. Let's say C equal to A plus B.
I got here the expression is here, A plus B.
Now C, you're assigning the result

of the A plus B into the variable C.

Now I'm going to print something somehow.

Yeah, and B

is six.

Now what is the expected output here?

Value 10, B, value three

and C value 10 plus three depends 30.

We'll be getting that

just what it is.

Connecting with the runtime. Just after what is happening.

So now execution is completed. Just looking into this.

Some of 10 and three is starting similarly, let's play

with all of the remaining automatic operators.

I'm saying C, D equal A minus,

no a value 10, B, value three.

So that what will get the D, which is seven

independently, we'll check each automatic operator first.

Let's say equal to gain to three Y to B,

eight 10, B three, 10 to three.

It should get 30, 20 E, 10 to three 30.

This is for multiplication. Same thing.

I want remainder value when I divide

that return with the three.

Okay, I want to find out remainder value

before we are going for the remainder.

Let's say division value.

Let's say your fig, yay by B, that means a slash bill.

So it returns actually yeah, value is integer.

B value integer

and divided by three will get,

definitely will get a float value.

Just look into this. The value is float

3.333 something.

Okay. Similarly, I want to know like

what is the remainder value when I divide that?

Uh, 10 with three.

So a percentage,

that means this operator is called a modello operator, A

person symbol with the B.

So definitely the remainder B three three is

nine 10 minus nine one.

The remain is one here.

And let's check with the double slash.

What is that double slash shape? The is part floor division.

The is quotient valley.

So what exactly the ENT already I explain you.

So here for example, if I divide 10

with three, what is the question?

Three threes nine, I valley one

ion it's three.

There's three is floor division,

technically this is called floor division.

The operator to find out this one double slash

just look into the output.

10 double slash three. I got three here.

Got it the same time. I want to find out some power values.

Yeah, double stop.

For example, let's look into with directly with values.

So 10 is the value I want 10 square.

10 square means 10 double star.

So 10 double star. Two means 10 to the two. That is just 10.

Square means certain I got 10.

So even uh, like for example, if it is simple value

and titan, it can easily compute it.

Suppose there is a big value like a hold to the power

of something I want, or two double star eight.

That is do power of eight, which is 256.

So just let's look into very biggest value, like two

to the four of 32.
So in this way, similarly want to find out square
for example, to find out the square root.
Actually here we have something in import math.
Math sq opt like 25.
What is the meaning of it? Square root off 25.
So this is what your expertise square root off 25 is
how much that is Five.
But without using this predefined function, also
by using this double star operator,
that means power operator.
You can find out this square root.
So what is the meaning of uh uh, square root?
Old did you pour off one bite?
Suppose 25, the double star. I'm saying one by two.
What is expected result? Expected result is what?
Expected result is five, but you don't get five.
Just check what happens. 12.5, I got it.
So this is a mistake like uh, faster double star is computed
after that, that result is divided by the two.
To overcome this, what I have to say, keep this one
by two in the brackets.
Now one by 2, 2 1 by two is separat allocated.
Now 25 to the four. That one by two means 0.5.
That will be upright. Now square root, 5.0. Exactly. Got it.
Other easy way, you can say the same expression 25.
That will start off 0.5.
One by two means in the place of one by two.
Here I have apps 0.5.
So here I'm getting square. Similarly Y through.
Do you want to find out my like uh, fourth route
or QIC route or fifth route?
You want to find out, for example, you want to know
fourth route of hundred.

Okay, so simply say a hundred double star.

Fourth route means that is three four of one by four.

Now we get both in this way.

We have different operators in different states.

Let's say some real time example.

For example, your salary is one lack.

I want to find out tax.

Suppose your tax is 10% on the salary.

Our salary into 10%. 10% means standby by a hundred.

Instead of saying this has 10 by a hundred, you can say 10

by a hundred is value is 0.1.

You get 0.1.

Similarly, HRA HR is equal to salary into point.

That means that 20% you are sending us.

Similarly net salary won become

for the net salary inputs are salary tax, hr, I'm using

that word salary minus tax minus plus hr.

You'll get net salary.

Now let's print all the things.

Print salary,

the salary value hand printing here.

Similarly, the tax value I'm printing

HRA

net

chase net.

Now look into this one.

So here to find out the tax.

HR, a net for all these things,

automatic expressions are used.

Got it. Similarly, want to find out any number,

whether it is a even number or odd number.

We want to find out, for example,

let's go talk about hundred is 1 27

even number or number.

I want two find. Okay
then what is the even number?
If I divide, I divide any number with the two
and the remainder should be it zero.
That's why a person K mod operator, the person operator with
two equals two is zero.
Now if it is even number, it returns true.
If it is not even number, it returns false.
Just look into this, this conditions false means the 1 27 is
not even number.
Let's say a equal to 1 28,
which is even number now a personel.
Two double equals two zero.
Now it returns has 1 28 is
a even number.
So conditionally you can check.
Later we discuss about the if conditional statements.
Just simply I'm writing on condition.
If yay or two equal to zero,
then print.
I'm printing A is given number.
Yes, the condition falls miss, definitely
that is odd number.
Really? Yay is odd number.
Understand. So
here something comm, Mr.
Dr. Just check this execution.
So it's all the practical part
for the operators.
In the next session we see some expressions based
on operators.
Okay? That means we are going to work
with the relational expressions in the next session.
Okay, thank you very much. Let's meet in the next session.

# Video: Relational Expression Lab

In the previous session we have discussed about
automatic expressions.
Okay? Now we see relational expressions.
Before we proceed into the practical part, I just try
to recall what all the operators available in the,
as part of the relational.
So there is another name for the relational operators,
comparison operators.
And one more time you can use conditional operators.
All these relational operators are used
to check the conditions, okay?
So like you are comparing always two
values to variable values.
That's why these operators also called us operators.
So for equal check here we have double equals to
the double equals to is for equal check.
You can say equals to greater than
and the symbol is for less than
and one more greater than or equals.
Two
can be greater or even it can be equal also.
Then it returns true
similarly, less than a equals.
And I suppose if I want to say not equals true, the negation
this exponential and equals.
This is for inequality Check.
Inequality check simply can say not
equals,
let's say the practical part of it.
So I'm coming to Google color to execute all these things.
Relational expressions,

the support, these examples,
I'm taking your age for example.
Your age is 25. The value 25 year
as assigning into variable page
and age greater than 18 M.
Look at this as value actually here, 25.
But you are comparing with 80 but it is greater than 18
Or not. We are checking.
So definitely the 25 is greater than 18.
That's why condition. Let me give one more condition which
can be the false for the same thing
is greater than I'm giving.
30 now is well 25
but I'm checking with whether it is greater than 30.
Now it returns false for example, to join my course.
The as limit is 80.
If it is greater than 18, then only he's eligible.
Otherwise he is not eligible.
Think that this is the credit here conditionally.
I can write a statement like if he is greater than 18
and print eligible
for example, condition falls.
Then I'm giving yes print
the student or the the proper candidate is not
letter about the yip statement statement.
What all the rules for everything like we'll discuss later
already in the coding rules we explain like how
to write these yip statements.
But more detail is specifically about if statement I create
a few more sessions per you.
Now check this actually as value is what? 25.
25 greater than 80.
Now P sales for example, I'm changing the value
one more candidate raise.

Now what about this ship? 15 greater than 18 condition.

False Because of false. What? It'll say what?

It'll print not eligible. Got it.

For example, if the value 18 look into this

18 greater than 18 again

for example minimum age is 18 at the time you can say

greater than r equals to 80.

So this is another upgrade of a greater than R equals

it can be greater equals little

bit in that is also that will be,

got it.

So this is greater than R equals for example, uh, the two,

the years limit, 40 years to become my student.

Think that. So then you can check the condition

if yes, greater than R equals two.

Now I'm checking with the max size limit

less than R equals two.

Let's say 50. Think that 50 is the maxis limit.

Okay? Now then print the candidate cells.

Otherwise not else.

Every is otherwise in python

and programmatically you have to

Say yes and then give your statement, right?

Not e. Suppose

what is value you have kept here previously?

19 Is there, let me keep some age

is uh, 55.

Think that this is age of one student

is applying for the course.

Now what did is say 55 55

less than article equal to 50 condition falls.

That's why it is saying not eligible. Same thing.

Test with 35.

35 is eligible. That's say exactly equal to 50.

So 50 also I'm keeping 51

not least understood.

That can be less or it can be equal to that 50.

And both the cases, the condition is true

then it returns true.

If it is true, it is printing. Got it.

This is less than less than our equals still not equals.

If you want to check, suppose equal check you want to,

you want to see for example, name, I'm taking my name bar

and his gender, but it's male.

Mm. If the person is male

or not, you want to check then the condition

for equality gender equals too.

Yeah. Now in this

case that is true.

Got it. That gender value is Yeah.

You are comparing with yeah for example gender value.

I'm keeping something like this gender value upper case.

Yeah. Now check the same thing gender equals

to Yeah.

Now here upper case letter is different

and a lawyer case letter step

that it is ing false.

Got it. But in both the cases I am expecting that is

to be true whether user has a type a small LA

or upper cm, anything to be true here then we have

to use a lawyer function, gender, talk to lawyer.

Even though upper case value is there

that is converted into lawyer, the coed value are checking,

comparing with smaller.

Now for small and or capital

in both the cases you'll get it wrong.

But suppose general values like this, yeah.

Now generally is equal to

you say definitely the condition is false.
Definitely we will get false.
Now based on this conditionally
We to operate this, then you need
to write the statement like this.
If gender lawyer
equals to yes, if yes print
that's applicant is female.
If condition falls, then
print applicant.
Yes, of course I have only two options that are male
or female conditions true.
I'm printing as female. Condition response are pretty
as male here.
Same thing. Inequality. Uh, inequality. Exactly.
Wanted for example gender
not equals to Yes,
actually currently the gender is what
female look at the value of the gender.
So current value is female, right?
I'm checking gender not equals to. Yeah.
Gender value of
and are not equals to comparing
with the you have not equals to F condition.
False actually f equal of threat.
So here you will get false here
for example, gender value is the,
I'm comparing gender not equals to.
Yeah. So what is the value of gender?
Yeah, you are not equal inequality check
you're doing with yf.
Yum not equals to yf condition is true like this.
Okay, so here we have seen the execution of equal to
for quality check not equals to

for inequality check rather than rather than
or equals to less than, less than are equals to
all examples she has.
So this is about the relational expressions, okay?
All these relational operators use in expressions
independently cannot do anything.
Just why using rather than what you can do.
I'm saying only greater than I'm running it.
It is giving you all these operators to be placed
in expressions only.
So how to give the expressions.
This is, but in all these expressions
you are checking, you are giving only single expressions
left side of the operator, one variable or value
or right side of the operator.
One variable or valid.
What if, if I have multiple conditions to be checked
at the time we had to use logical operators
in the logical expressions in our next session
we completely did the practical session, practical part
of the logical expressions.
Okay, thank you very much. Let's meet in the next session.

## Video: Logical Expression Lab

Relational expressions.
In this session we are going with the logical expressions
probably we discussed what are the
different logical operators.
So I just recall those things.
The logical operators
and are not.
So it means all conditions must be true.

If all conditions are true, then on the final condition,
that means the final result is true.
So in case of far at least one condition,
at least one condition will be true.
Suppose I have high conditions, four conditions falls,
but one condition is true but still that result is true.
If all conditions falls, then on the final result is false.
So not means you know negation.
So in the previous sessions we discussed about reus,
now we see the practical part of it.
I'm starting notebook. Let's go with the fresh notebook.
Look at this. I'm taking your salary
for example, a salary 70,000
and name that salary.
So I'm just validating whether salary
greater than 50 and less than one mark.
One like one like
or mark salary greater that
50 K and less than
or equals to one black.
Let's say here also greater than oracles.
What this, how to write the condition.
So look at the condition,
the variable itself greater than Oracle two, two 50,000.
That is first condition
and salary less than Oracle equal two one like
so here, two conditions are,
so I'm expecting both conditions to be true.
That's why that means whether salary is between in between
50 and one like you are checking other here.
What is the salary value you have again? 70,000.
Now what about first condition?
First condition is true and what about second condition?
70,000 is less than four to one. That is also true.

So that what is the final result?
Okay, so this is a condition check for if salary is
between in between 50,009
one, not check this.
So here what is the final result? Check it.
So finally it is true. Same thing.
One more salary.
Not between 50 K
and a hundred K.
A hundred K means smaller this time
what I want less than 50,000
and greater than one lack I So
however, the condition salary less than 50,000
and salary greater than one lack here.
What is operator and or, or the right operator is
or now check it here.
Actually we a salary. 70,000. What about first condition?
First condition, false.
70,000 is not less than 50,000
or what about the second condition?
70,000 greater than one life. This is also false.
What is the final result? False. Got it.
For example, same thing. Do check for the 40,000.
Suppose your salary is 40,000.
What about first condition here? 40,000, less than 50.
That is true or salary Greater
than one lack.
This is false. The 40,000 greater than one lack is false.
Not true or false In case
of far one true enough final condition will be
that means 40,000 is eligible for this.
Got it. Similarly, one more value.
I'm taking one lack 10,000.
What about first condition?

One lack 10,000 less than 50,000. This is false.
Second condition one lack 10,000
is greater than one lack.
One lack. 10,000 is greater than one lack. That is true.
False are true. This is true
because one proof is okay.
Now let's validate and check it here.
My condition is this one.
What is the value? Salary value here? 70,000. Hmm.
Let me keep one second. Let's test all the phases.
In the first phase I'm taking salary as 30,000
condition two or false.
Your condition is true because
of the first one is true but second one is false.
One true enough. So final result is,
okay, let's check with another here.
I'm taking salary one lack 20,000.
So fast condition falls here. One lack 20 less than 50.
That is false but one lack 20 greater than one lack.
That is true. So false are true is true because of why.
Okay and check in between value, we'll take it.
Suppose your sally now is 90,000.
So first condition 90,000 less than 50, false
90,000 greater than one lack false.
Both are false. So
that if all conditions false final result in all false
and this got it.
So here a quick summary about two, about two operations.
So suppose if the task is like this.
If you want to check Sally between 50,000
and onelan, that means a hundred K.
Then the condition should be Sally greater than our equal
to 50,000 and salary
less than article equal two one.

Same thing. Not between salary, not between
50 K and on that K in this case in case of positive,
I got the end operator definitely,
definitely needs negative.
I will get all operator. This is positive statement,
this is negative statement in the positive statement end up
here means in its negative statement or
or will come into the picture so
that what is the condition here?
Salary less than 50,000
or salary greater than one black like this.
This is uh our pre previous example
what we have executed similarly.
One more task we just observed
to get more ly have it.
City, city column. Suppose your name again.
One second I'm taking your name.
Your name is Ravi
suppose city value.
Hyderabad. Okay, so the task is like this.
I want to check for whether the person
is from Hyderabad, Delhi Uni.
Okay these three cities are valid for me, okay?
I want to check whether he is from one city of Hyderabad,
Delhi Uni, okay,
so like the condition for listing
deliver.
Now what is the condition? If the column name is city,
the variable name is city.
City equals to that is one condition
city equals to this is under the condition
city equals to this is next condition.
Now I have more than one condition here.
Definitely lost operator required.

So between each condition you need one operator
Here I'm keeping or
or suppose if I give you end, first of all,
if I give you end, suppose if you are from Hydrovac,
definitely you are not from remaining cities,
definitely a condition will become false.
For example, another person is from Pune.
Definitely first two conditions will be false in case
of an all must be true, right?
That's why here, how is the chance anyone condition
to be true If you're from Hyderabad,
definitely you're not from Delhi and Pune.
If you are from Pune definitely are not
from Hyderabad and Delhi.
That's why what is a operator?
R is operator because your situation is any one condition
can be two here, not all at a end.
That's why R is right also.
Okay, just check it whether is from
the wanted list stop cities or not to check it.
So this time it returns true. Got it.
For example, let's take another profile here.
Let say Ronnie from Shana.
So here I got false
because post condition falls that condition falls
cloud condition is also false.
All our false means in case of our final result is once.
Got it. So here Ronnie running profile
is not eligible for this.
One more last profile I'm testing.
Mr. Wayne is from uni.
So Pune is uh, one of the one city are not here.
Whatever first condition, false second condition,
false third condition, true one condition

to total result will be
right in this multiple conditions.
So suppose if the task is like this,
what I want is
I want accept
aaba remaining city people are eligible,
accept and aaba.
Now that means here city should not be,
city should not be Pune.
What is the condition? Now city equals to city
not equals to negotiation equal to means double, equal to
not equals to that explanatory mark.
Mark symbol and then equal not equals to not equals
to one more condition.
City not equals to P.
So how many conditions are here? There are two conditions.
Two conditions means here fastest should not be so.
Uh, like conditions it should not be, it should not be Pune.
Both should not be understand.
So that that means both conditions to be true.
That's why other here
whenever I'm asking Pune daily, people
here are here definitely needs negative.
We end up here. There is a small, okay, just validate this.
For example here the city valley something Pune,
what about this post condition une bad condition.
True. What about second condition? Pune not equals to.
Pune is false. Pune equals to Pune is true
but Pune not equals to Pune is false.
Okay? One condition true here
but second condition is false.
That's why total result is false.
For example, I'm moving one more value here.
City equals to Delhi.

What about false condition here?
City Delhi nautical also had a bad roof
and Delhi nauticals to une.
That is also true. Both are true so that he is a true.
So here final observation here if you want positive here,
if r appeared, you know in positive case R appeared
definitely need negative case and will be upgrade.
Okay? Now let's take this example
our condition une.
Why is false? Totally false because false condition true
but second condition is false.
One false enough total thing will become false yet because
and is expecting all conditions to be true.
Okay? I'm replacing this
with the Delhi first condition.
True. Second also true.
That's why for Delhi the given condition is
true like this.
But in all these examples may maybe in the salary examples
or maybe in the city examples there are or
and anyone operator may have used, I want
to use the combination of both R
and n For example, if the task is like this,
if my task is what I want is
iaba daily people and their salary should be
greater than or equals
to one.
Like this is what expectation.
So person taba Delhi
at the same time the salary should be greater than Oracle
equals two one.
Like now tell me the condition for the city only for city.
City should be hba Delhi.
Anyone is okay what the condition for hba

city equals two aaba.
I'm typing the city name of short names for the city
and city equals to Delhi.
So between them Sal operator are so the center thing treated
as one condition and one more condition is required.
That is for the salary. What
is the condition for salary here?
The salary greater than Oracle equal to one lack.
So this entire left part condition one,
this is second condition.
Now between them again operator city to be true salary
to be true here strictly that's why.
And observe in city within the city
or operator between city and salary
and operator for example, your name ra,
the name of the profile is Ra is Sally is Lac, 20,000
and he's from
now whatever.
Just validate this first.
Internally validate this first condition,
two second condition.
False here,
greater the article two one like
then to our false, what will happen to our false
true and
what about S condition?
True again. True and true. Final result. True.
Understood. That's why Mr.
Zel will kind it for this criteria. Got it.
Let's check it past
current, but I'm changing this
check in the next example.
That's a Chris in the
name solid and thousand.

What about city? City condition is true

but s conditions now.

So Mr. Chris is not eligible for this

like Sally.

So city condition is true finally because he is from,

but Sally is not met one false enough in the case of end,

total thing will become false.

That's why he's not Kris' not eligible.

Let's check with one more thing Sally.

One like 20 but city is Chennai.

Now what happened here? City condition got fails

because he's not from Hyderabad at the same time he is not

from de, he is from Chennai.

That city condition got fail but Sally is still there.

Is result is false because city criteria met Sally.

Sorry Sally criteria met city criteria not met.

So Karen is not eligible here. And one more drive.

I say Karen, my

from that

to Sally is 20,000.

Hmm for the current I here please check.

What about city criteria, false salary criteria? False.

Both are false. That's why Karen,

my is not also is also not

eligible in this way.

You can use combinations of your end or it.

Got it. And one more thing.

What are the thing that we have for this criteria?

I'm applying negation. Just check what happens

and keeping in one bracket I want to apply its negation.

For the negation, the not symbol that I have used.

What is not? Not of false.

Okay, but finally here it is not accepting like this.

So apply with the knot function.

Is there any solution with this? Not off.
And what knot is the, so you can use this symbol
for the knot when
whenever you are checking inequality, knot equals to like
that not for entire condition.
Got it. Actually here, what is the criteria?
Accept to deli that is a meaning.
At the same time, Sally should not be greater than Oracle.
Equal two. One lack.
Actually this is positive case per the positive.
If you apply the negation, which is not, finally it turns
to false.
Not of true is false. Not a false is.
Okay? So this is about logical expressions
in our real programming part we see more complications
with the deep statements and statements like that.
Okay, so in the next session we are going to work
with the topic called programming.
Using these expressions up to now like uh, okay,
before going for that one.
Also like we see some other type of, uh,
expressions after that.
Finally, we include all these things into programming.
Now independently after you have checked up
to now independently have checked up
to now automatic operations, relational operations,
relational expressions, and logical expressions.
Okay? In the next class there are special operators
which are called as a membership operators, like in
or not in such kind of operators.
So in the next session we'll be discussing about that. Okay?
Thank you very much. Let's meet in the next session.

# Video: Membership Expressions Lab

Operators and collects expressions

with some practical stuff.

Now let's enter into the topic of membership operators.

There are two membership operators here in

the number one yin and not in

in operator.

Second one is not in operator.

These two are not statements.

Simply can call them as operators

first.

Uh, let me explain some example letter functionality.

I will explain. I'm taking a string. Let's say hello.

I'm checking the word Y,

the character Y, yes or no.

This is a condition E in.

Yes, the letter E available

or not available if available.

It returns. And one more.

I'm keeping E, L, L in.

Yes. What happens?

L-E-L-L-L Available. Yes or not? Yes.

So it returns and one more condition I'm

giving you that says Z zin.

Yes. In the hello, the as well is hello.

Within the hello there is no character concept.

Now it returns box

On that way in operator you can apply on the strings.

Same thing. It's opposite.

For example, that's a fruit

equal apple.

I'm checking. P not

in fruit.

P not in, but the word P,

the character P available in apple.

Okay, but you said not in red, not in.

So it returns false.

Okay, one more example.

Z not in fruits

here you said he applied some

actually Z is not available on the apple.

So basically that is a false

but not in he applied not a falses.

You get true. So based on this

you can guess what is the functionality.

You can guess what is the functionality of the scene

and not in operator.

That will keep something into the notes.

Let me explain the functionality

in if given item

available in the collection

that says string is also acting as a collection.

As a collection of characters like given item

available in the collection.

Then returns

if not available returns.

False.

So this is functionality of again,

and if you come to the mountain,

it given item not available, it true

opposite if available.

Its false.

Okay? This is the functionality opinion not in operators.

Let's see the practical part of it.

I'm opening Google collab notebook,

come to the first notebook.

Let's take our beginning example, which is, hello

I'm checking for HRE in S.

The letter E available in the hello.

That's why it has to written.

That's why it is uh, connected. The run tap.

Now it disconnected. Same thing.

Bunch of characters you can give suppose LL

in that means in, yes.

So E, L, L, L.

Instead of characters available in

that given thing.

That's why it returns true. Let's say

Z, the character in yes,

but in the hello, there is no Z character.

That's why it returns false. Got it.

Same thing I'm applying with negation Z not in.

Yes, that basic J is not available in the yes.

Then it returns

true by example.

The letter ye in fact available in, yes

we are saying ye not in.

Yes, if you're available it returns false power

not actually E available.

That's why it returns false. So this is the functional.

After this there is a line

of comment is on Python.

Okay? Now the word class available in the line or not.

I want to check. I want

to check whether the class word available in the given line

class in

or line available.

That's why it returns true. Okay? Same thing.

Whether this line is related with python,

Python line or not.

That means Python related comment or not.

You want to check then check like this python in line.
Still it true, it doesn't matter
what is the position of that line.
Maybe Python can be at lost or in the middle
or in the beginning it doesn't matter.
It is checking word by word
if available returns true otherwise false.
Okay, same thing. I'm checking Java.
Java in line.
Java actually not available. That's why false.
Let's go with its negation.
The same Java not in line.
Actually Java not available. Then it returns true
because of noting
similarly, let's say Python.
Python noting line,
actually Python available in case
of noting if available it returns false.
Okay? This is in and knotted.
So this in can be applied not only on the strings,
you can apply on different types of collections.
What are the basic collections in Python list type
dictionary set.
Let's try all these things.
In all these both types of collections.
I'm taking one list subject already in data type spot we
have discussed about list double
dictionary, set all these things.
Suppose I'm giving value something like this.
10, 30, 50, 80 and 90.
Now I want to check whether the value 50 available
in the given list or not.
15, yes, 50 available.
That's why it has to return true check hundred

not in X, actually hundred not available in the list.

If not available then only returns true.

For example, I'm checking hundred yin X

in fact hundred not available.

So that falls, that is a difference

between in end not in in case of in if available,

true in case of not in if not available then true.

Okay. For example,

I want to check for the multiple values.

30, 50, 30, 50 is the list

in I'm saying x check.

What happens? Actually 30 50 is available,

but still it is showing error.

The principal because it is taking each element

is the first element is a list of 30 50,

not is the second element is a list,

not is the third element is a 30 50 list.

Not fourth knot, fifth knot.

That's why that has return. False.

Same thing apply with negation.

x clear supply.

Not now it returns

because not of true is false, not

of false is true previously got false.

That's why when you say nothing,

not of false will be applied.

That's why I got it. So in this way, listed, uh, that uh,

in not in these operators are called membership operators.

This can be applied on list.

Collectionals, same thing you can apply on Apple.

Let's say I'm taking a top, top off

the employee id 1 0, 1 name Ravi

is salary 90,000 and mail

and designation software engineer

and he is from the department.
Let's I depart. So think that here,
there is a structure here for the info.
What is the structure? Id name salary next bill.
The fourth one is gender. The next field is designation.
Next field is or department.
So department is it.
Now you want to check whether it department,
whether the word maybe in the first, it doesn't matter
what is the position of it, whether the word it available in
this info not you want to check it in info
check this output, it is available in the last position,
which is as a department.
So that's why this is true. Got it. Same thing.
The word marketing available in this or not.
We just confirmed it. I want to check for the department.
Marketing, marketing in
info that is not available.
So that false for the same thing I'm going to apply.
It's negation marketing Martin,
Martin info marketing.
Actually not available. If not available
then only it returns, right?
Got it. Similar style these membership operators seen
and not in can be applied on dictionary apps.
What is a dictionary? Dictionary is a collection
of key and value page.
I'm taking a sample dictionary. Let's say the name D info.
I'm taking your ID as key.
So is a key value pay always plus one is key.
Second one is value. Already, you know the key rule,
key rules key should be unique.
Value can be duplicated.
For example, 1 0 1 is uh,

details like 90,000.

Think that the employee ID

and salary our bank account number balance

and for the 1 0 2 90,000

because value can be duplicated

and 1 0 3 40,000 1 0 9 between, uh,

after 1 0 3 and uh, before 1 0 9.

There are no account numbers here.

The next account hold is 1 0 9 is balance amount 80.

Next one 20. Balance amount one.

Think that in this way there are some five

ways just come from this.

Come from this. Learn off the info.

There are five element I want to check.

Suppose treat that key. The first one is key.

That is account number, bank account number.

Second one is balance amount.

I want to check whether this is available,

whether a particular record number is available

in the dictionary or not.

I'm checking with 1 0 1 1 0 1 in DO.

Check this now 1 0 1 available.

That by for example,

one level triple uh, triple one.

In fact 111 is not available

in a given dictionary.

Now you'll get in case of in operator, you'll get false.

Got it. Let's supply its negation.

The P is taking only key part.

It is not checking the value part.

Triple one n

dean let's supply this time.

Not in opera, not in actually triple one not available.

So that if not available then only that is not in opera.

Once again, look at the din info.
There are five page, the account numbers are key
and balance amount as well.
But I want to check in the values.
But if you apply in operator
or not in operator on a dictionary, it is checking only
for the, it is checking only for the dictionary.
So that means key spot only.
It is checking, but I want to check within the values
then the technique, we need to separate it.
Let's look into this. The info do values only values
spot will be separated.
Once values got separated within the
on the value spot you can apply in, not in operators.
Now let's do, I'm going to check for 90,000.
The value 90,000 in DFO
dot values directly.
I dfo it is second only.
Keys only in the keys.
90,000 keys not available,
but I want to check within the values dot values.
Now it returns true because 90,000
value appeared in two types.
It doesn't matter how many times it appeared.
Least one time it's available, it returns similarly.
I want to check with the tool la I want to check
with the tool la ax in
din info dot values.
Now ax not available returns, false
apply the negation, same two LA not
the input values.
So actually two lax is not available if not available,
then only not in returns.
True. I got so up to now.

We applied this up

to now we have applied this in operator

or not in operator on strings, on list on top and on al.

Now one more collection spending for us. That is set. Okay?

Set is a collection of unique items. So you can apply this.

Uh, membership operators.

These membership operators are set

also, let me take a sample.

Your sql, I'm keeping it as a set.

Hundred, 200, 300. Let me take one more. 300, 300.

It doesn't matter how many three hundreds you are giving,

how many times the values is repeated.

Set is not allowing duplicated duplicate values.

It takes only unique values. Let's give 400 and 500.

So actually here I have five to five

and, uh, three totally eight values.

Okay? But in fact in the yes, how many values will be there?

Only three values, five values

because 304 times we have one,

but it has taken only one time.

Now, this, uh, this a what is the type of object?

It's a set, no type of s just say and confirm it.

This is a set. Now I want to, within this set,

whether a particular item available or not.

I want to check. Suppose value four 50 in Yes, I'm set

where CA set, but four 50 not available.

So that falls, lets check for the 400,

400 in years, but 400 is available so that it returns

one more four 50,

which value?

Four 50. Four 50.

I'm applying not, not in. Yes.

In fact, four 50 not available. So that it has to return.

If not available, then only it is written. Got it.

So this is the thing about membership operator
without membership operators also,
you can play all these things, but you have to use
multiple conditions between each condition for the, in
our operator have to use for the replacement of not in
the end operator have to use.
Okay? Instead of doing like a inin that the end or,
or such kind of things directly can apply
in, not in operators.
Got it. So that the exhibition will be faster.
At the same time, sometimes it will be confused whether in
this situation where are to be used
or had to be used in the beginning at the time
of a beginner level programmer.
Okay? So simply you can avoid the
confusions by using this one.
Okay? So it's a discussion for the, for this session.
In the next session, we're going to discuss on
something objects called
identity objects.
Okay? Up to now we have discussed in all our previous
sessions about the expressions, automatic expressions.
We have seen relational expressions,
and in the last class we have discussed about logical
expressions in this session, membership operators
and expressions, both theory and practical warehouse.
Okay? In the next class we work
with identity objects and identity.
I, I like checking the identity expressions.
Okay, let's meet in next session. Thank you very much.

## Video: Identity Expression Lab

In the last class we have seen about membership operators,
like now one more interesting operator
that is identity operator
to in this session we'll discuss about it.
So mainly here there are two operators.
Number one is the
negation for it not is
so where exactly this thing is useless.
If memory location of two variables is say
if memory location of two variables
is equal, then it returns true.
If not
returns false.
Whereas same thing like opposite the negation action
if memory not equals, if memory location not equals,
then it returns true
not equals then returns.
So if equals
it returns false.
Okay, that is similar to in
and noting here is not is so here what exactly the concept
of memory location, we try to understand it before that.
I copy this notes into the your document.
A small example I'm creating only stop it.
Let's say X is a list
keeping some values from 2034.
If I say Y equal to X.
Now let's print both the things print X
and print what for the both we get
the same output, which is done
20, 30, 40 here.
When you say Y equal to X, what happens internally is
that means in ram,
in the ram in some location one object is created.

The list object. The list object is here.
10, 20, 34.
For this, the variable name assign that is X.
When you say Y equal to x for the same memory location
one more alliya will be set.
Now in fact there are two variables,
but memory location is set.
Okay?
There are two variables
and that memory location is same here.
Okay? Now in this moment,
if you check a condition like this,
Yex is one.
Both memory locations same or not? Yes.
That's why it has to red done same thing.
Opposite action. XE, not
or not is what it returns opposite
false, okay,
so how to make sure that the memory,
memory location is equal or same or not.
Let's say the practical part of it.
Lemme open that Google collab for this execution.
Okay, let's create one list first, let me run the runtime.
Now it is connecting the run python runtime in the cloud.
I'm creating one list, same previous value I'm taking
10, 20, 30.
But after that, why the exam
before this some operation will play later.
We think about this allocation.
Just try to compare both the values.
Now print x print value,
same values in both.
That's why I'm changing some value in the X.
For example, the fastest cell, which is index number zero

X of zero, previous value was 10.

First value was done. I'm going to make it as hundred.

Now print both

X and Y.

Now here actually I did a change in X,

but this change reflected into Y.

In fact where I did change,

I did change at XI did not do anything for the Y.

But in Y also, same change happen.

Let's do some change in the Y for example, last element,

I want to make it as 400 Y off minus one

way off minus one equals two.

That's of 400. Now print both values.

Print x, print. What?

Now check the values. I made a change at Y,

but in fact the change reflected into x excels

why this is happening because the memory location of X

and Y in both are same.

How to make it true simply say X is Y.

That is true.

That means both memory, location is same.

That's why if you do something change in the X,

the same change will happen in one.

Okay, it seems like that,

but in fact that memory location is same.

You are indirectly you are updating that memory location,

values, understand,

look into that idea of x.

Run it. What is id Id for X?

This is a memory location id. Four x is this one ID four Y.

What is that?

Understood, same Id generated for both, right?

Got it. For example, X equal

to one more example.

1, 2, 3, 4, 1, 1, 2, 4, 5

Y equal to XIZ

and look into the jet Z is equal to

the comprehensive style.

I'm applying expression follow loop, left side expression,

right set follow loop in the place of where I'm being

for V in X.

How many values are there in X? There are four values.

So that that means loop transfer loopit rates

for four values at each iteration

the value is coming into V.

I'm going to take that V.

I'm not doing any modification for that value.

Now gender is also created.

Just check what are the types of each inhibitor.

Print X type of X, sorry,

and a print Y type of Y,

print Z, the type of check.

Now everything is a list come from this

by checking the

output server list.

But I'm saying that XC is Y, the condition is true

because memory, location of X

and Y not check

the Xs Z.

False because memory location is different.

That means if I do some changes in the X, the changes won't,

won't be reflected into J.

Same thing. If I operate on jet,

the changes won't be reflected in X,

but if I do some change in X,

the change will be reflected in Y.

If I do some change in Y, same thing.

Reflection will happen at X. Got it.

Uh, now opposite X is not one
actually XY memory location sale in the case of is not
if location sale, it returns false.
For example, X is not on set. Now it returns.
Got it. Now I'm doing some change in X. Let's say X of zero.
First element time. I'm dating previous value one.
Now I'm making it as hundred. Confirm this.
Print X, print y, print it.
What is your expectation in X and Y?
Same values, but inject the bold values
because it did not do any change in inject.
Now from this output, XY has same values
that is having different values
because the changes are not reflected into jet
because memory location is different.
Similar style. I do some objection in jet
it says J off minus one.
I'm going to modify the last value in the jet 500.
Now print all the things, print X,
print Y, and print.
I said check this.
Now actually I did change it J,
but we changes are not reflected in XY
because their memory locations are different.
Let's compare their memory locations, how
to get its memory location id idea of X
idea of Y
and ID of side for X.
Y. Same values you'll see,
but that is will see a different value.
Confirm this ideal ID. Same for X and Y, right?
Okay. And for ax also same id
but for J different, right?
Remember here E is different, double equal is different.

I'm saying that X equals two y, what happens?

Two X is y true,

but there is a difference between W equal two and X is Y.

So what is W equal two is

checking check C.

Quality

that means contains same values or not,

but it'll check it.

But in case of it is checking what even those same values,

even those same values, the condition may be true or false.

It is taking what? Memory, location checks.

Memory location is same or

that is a difference between list.

Got it. Same x not equals to YX is not Y on the,

this is about identity operators and their expressions.

Okay, so in the next session we are going

to discuss about Bitwise operators.

Thank you very much. Let's meet in the next session.

## Video: Bitwise Operators and Expressions

Discussed about identity operators and their expressions.

Okay, previous to previous class,

we discussed about membership operators, like in now ease,

not easy, the identity operators.

So in Python, one more special operators,

that is bwe operators.

In this session we discussed about that

Bwe operators.

So especially into electronic devices,

especially in the digital electronic space,

these bit device operators are most useful.

So generally in regular application programming you don't

use, but whenever you are trying
to embed your code into the chips, okay, so at that moment,
this bit five to increase the speediness within the chip
level, okay?
These bit fives operators will be helpful for,
of course there are many, but for a four, just
for understanding purpose here I listed about two,
like there are two symbols here, two operators here.
But is ampers percent singular amp percent.
This screen is end, that means bit twice. And operator,
similarly, there is a pipe, this is R
bit nor normal.
R bitwise operator.
This two operators how behave
and how are behaves in just second.
So when now we cannot A give a by
or any is any number into the by bits.
So that will to the boundary format.
Uh, for example, I want to convert a five.
What is a way just observe.
I want to convert this into a boundary array,
which is a four size.
Okay? Two twos. What is the remainder?
Two, twos four. Remain value. One, two ones.
Remainder value zero two zeros.
What is the remainder value one.
Now what is the binary format of this?
Five zero coming disorder. Coming disorder.
Okay. Zero one. Zero one.
Okay, this is a binary format of similarly.
Let's talk about three.
I want to convert this three into
binary two. One
Two, remain value.

One, two zeros. One.

What is the format? So come in this sort

0 1 1 of course here, uh,

I'm comparing with four size.

That's why I keep one more zero extra

because in the left side, zero there is no value.

Got it. So now remember this

double zero one, double zero, double one.

This is binary format of three

and 0, 1, 0 1.

This is binary format of five. Now let's play this bit.

Device operators on this one. What happens? Just check it.

Sorry,

I will keep it to the dock part.

For example, our expression is uh, three end of five.

How it works. You'll see my statement is

three and five.

This is expression. Hmm.

What is the value of three bin format of three?

Zero. Zero and one. One. What is the value of five?

The binary format of five, what we got previously.

0, 1, 0 1.

Other here if all our ones,

the end behavior is like this,

it compares each bit,

each corresponding bit.

If both are one

returns, one

other here, zero, zero.

Both are not one. So that 0 0 1, both are not ones.

0, 1, 0, 0.

Now here in the last, both are one

that means one.

Now the result is in binary

that into like a byte.

That means number numerical format. If you cannot, this

Other here, this is two deliver of zero.

Two, deliver of one because all values are zero. One, two.

This two of two, that means two square. This is two q.

Now format is like this.

Multiply this zero and a two Q zero into two square,

zero into two of one, one in two of zero.

What is the result of zero into two Q? Definitely zero.

Anything is zero. Zero into

two square, zero into three.

Four of one plus one in 2, 3, 4 of zero.

What do you get? Zero plus.

Zero plus zero plus one and two.

One. One and one is one. All left side are zero.

Final result is one.

That means when you say three

and five, your reply is your result is one.

Okay? So similarly for our operator, what happens

there is a symbol called pipe symbol.

Let's try to understand what what will happen.

In the case of our for same three

and five values, we'll see

three pipe symbol, a pipe symbol four or our operator

and five amp say what is the result of the three when format

for the 3 0 0 1 1.

Is it true? Confirm this.

0, 0 1 1 we have taken

for

3, 5, 0, 1, 0, 1, 0, 1, 0 1.

This is four, five. What are operator doing?

Pipes.

If anyone is one.

So it is also like comparing each bit,

compare each bit.

If anyone is one.

One,

okay, if both are zero, zero.

If both are zero, zero

based on this T tell me what I can see here.

Zero. Zero. What is expected?

If both are zero, but it has to return zero.

If anyone is one again. One zero.

Anyone is one. This is one.

At least one should be there one.

Now both are ones understood.

So in the boundary format, the result is zero,

triple 1, 1, 1, 1.

Now this is a binary format

that means bit format cannot into bytes.

That means numeric format. Hmm? What is your style?

Start from two deeper of zero. This is of one here.

Two. That means two square here. It doesn't matter.

Anything into zero is zero. Okay, lemme write it.

Two Q, command summit zero into,

zero into one in two

and one in two and one in.

Come on. Do the same thing for each. One.

0 2 2 Q plus

one in two, two square plus

one in two of one plus

one in two of zero.

Next zero to zero anything zero one in two, two square

four, one in two.

Two of one of zero is one,

one into one, one.

So final result is what? Seven.

Got it now three.

The five five, the result is seven.

Same thing when you applied three

and which was that with five?

I got one whether our calculation correct

or not, we'll test it

practically in this way.

The bitwise operators will be working

across notebook for this,

but we cannot with the run time disconnecting now

I'm directly applying 3, 2, 3 and five.

We had previously what is our expectation?

Three and five. How much

One is the result?

Now confirm. Well

that means our calculation is correct and the same thing.

Let's apply for the R

3 5 5.

What is our expected value? Three or five. How much?

Seven do it. That is seven.

That is our previous explanation of the,

the computational explanation is correct for the

bitwise operators.

So finally get clarity.

What is the functionality of end bitwise and

and uh, functionality of bitwise are in the case of end,

it compares each value.

Both are once then only rating one. That's why zero.

Zero, zero. So compare this zero.

Zero is 0, 0 1 is 0, 0 1 is zero zero.

Everything is zero because it is

expecting all should be ones.

But in the last of the right side, one,

one, then only it is written.

Understand. And next,

bitwise our operator here.

The functionality of this bitwise operators

0, 0, 0, 0.

One, one. 1, 0 1. That is what? One is.

One is okay. If any at least one. One is there. It one.

Understand. So zero. Zero no ones.

That's why 0 0 1, at least one. One is there? 1 0 1.

One one. Both are ones. At least one. It is one. One.

It is expected. Understand.

Okay, so this is about the between operators.

In the next session we are going

to discuss about try

to use all these expressions

programmatically in one program.

Okay? Thank you. Let's meet in the next session.

## Video: Arithmetic Assignment Operators

Hello, welcome to Python Sessions.

In the last session we discussed

and uh, did some practical work with BITWISE operators.

So today we are going to discuss about automatic

assignment operators.

Let's start it directly, practically

our initial examples.

I will insert, demonstrate with Google color.

Max will use ID like vs.

Code pie chart, such kind of ideas.

So I'm connecting with the runtime.

For example, there is a variable, yay.

The value is now for the

variable, I want to add 10.

And traditionally you can say equal to a plus 10.

Now already here the value a hundred of

that, you're adding 10th.

Now the latest value of a

one, so the same thing.

This is here equal to a assignment operator,

automatic assignment operator.

I want to, I want to show you,

let's say yay latest value.

Now one 10 yay plus equals to to the existing value.

I want to add 150. So what is our previous value of it?

Current value one 10 for the 1 10 50 is added.

So instead of the meaning

of the center expression is a equal to a plus 50.

So here a value one 10 and plus 51 60. Now one 60 stored.

Yeah, the equivalent expression is simply in short,

a plus equals to 50.

Now, latest value of a one 10 plus 50, which is one six.

You can apply the same kind of model, same kind

of assignment for decrements,

like a subtraction multiplication for everything.

Let's say yay minus equals to I'm keeping 40.

So the meaning of the statement year to,

yeah, minus 40, what the value now?

1 60, 1 60 minus 40, 1 20.

That will be, that is the latest value of that is one 20.

Same multiplication. Let's a C equal to five.

I want to make this valid.

Double C, that asterisk symbol, which is

for the multiplication equals to

what is the meaning of the statement.

C, equal to C and two, two.

Now C, value five, five into two.

Now latest value, the C is 10.

Now C latest is 10.

I'm trying, I'm applying same kind of operation
with the divided by the slash two.
The meaning of this statement C equal two C.
So C by two, which is the latest value of C is five.
So same thing with the, some realtime data will, like
for example, the person name is,
is 50,000 or one like I'm taking
now I want to give
20% increment.
What is the expression of salary plus equals to
Salary into point 20% means
20 by a hundred, 20 by a hundred means point.
So previous value one,
like one like into 0.2 20,000, that 20,000 is added
to salary.
Now what is the meaning of full meaning
of this entire expression?
Salary equal to salary plus
salary into point
instead of equal to, again, same variable plus.
And then this expression directly can say salary
plus equals to salary.
Two point. Now check the latest salary
of previously one lack.
Now it should be one lack 20,000.
Got it. Similarly, I want to apply the tax.
So this tax that is the independent tax expression we are
giving because tax variable previously not available,
it's about 10% tax.
I want to apply. I can say Sally into point.
Now tax is 10%, one lack, 10% means 10,000.
This is not automatic assignment operator.
This is simply assignment operator understand.
For example, he see previous one lack 20,000.

Now I want to give decrement of 30%.
I want to give decrement of 30%, then
give decrement
or 30% to salary.
No salary because of decrement.
I'm saying minus equal, equal salary into 0.3.
The full meaning of the center expression salary, equal
salary minus
salary into 0.3.
So what are the value? One like 20,000 on the one like
20,000, the 30% is calculated.
So that amount will be directed from this one.
Now the latest salary here
after decrement,
which is 84,000.
Got it. That means 36,000 is uh, up,
not directed from here.
Got it. So in this way you can use
automatic operator and incremental
as automatic operator.
Decremental automatic operator.
If it is incremental plus equals to your saying incremental
minus equals to.
Okay. So in the next examples we'll discuss about
automatic operators are operators on
all types of data types.
Maybe on the independent data types in the,
like a single value data types
and uh, collection data types,
everything on automatic operators we'll apply
on all data types.
We see that about, we see about
that part in the next session.
Thank you very much, but.

# Video: Arithmetic Assignment Datatypes

Hello, welcome to Python sessions.

So in the last session, what we discussed

in the last session, we discussed about automatic assignment

operators, right?

So in this session we are going to do same kind of automatic

operators on different data types,

how it works, just check it.

I'm starting a new notebook.

So connecting with python runda, I'm going to take a,

let's say equal hundred, our data type of cept

other here, VA plus

inte then is also anti in plus inte.

You get incept and one more thing

ya inte, but 10.25 is a float.

Now integer plus float always it returns float.

For example, basic equal

30.5, which is a float web

for the B I'm adding one integer,

another value will be floats.

Okay? All these are the ab,

what are the variables have taken?

And these variables are holding simulated.

That means single value. Now you can apply these operators

on string cells, but all operators are not allowed.

Depends on the data. Different operators are allowed.

For example, I'm taking string,

let's say a SQL to follow

data hypothesis string.

So for this, yes I'm adding a number

that will be invalid expression

because string plus in is invalid.

Same thing, yes.
Plus, let's say why now string buys a string.
Now here what happens is string
of concord nation will happen.
No, just check it. Yes. Says hello buy has.
So the next year adding bye string plus string
string Concord nation.
Suppose between them I need space. Yes.
Plus give a space plus and then one.
Now after this output including spaces
that was con coordinated.
Hello. After that space,
after that one, that means plus operators
allowed between the strings.
Similarly, another operator is allowed,
which is multiplication operator as strict symbol.
Okay? If it is applied,
suppose string, uh, I'm taking a string value,
which is iPhone.
I'm multiplying with the 10.
That means a 10 times the iPhone,
the string iPhone will be repeated.
Now check the output. Same iPhone is for 10 tap.
Okay, one not thing. Let's, let's say A, B, C, A, B.
I'm saying in two three, the same A, B, C is repeated for
three types like A, B, C, A, B, C, A, B, C.
So that means on stringing what are the,
what operators are allowed on string
plus multiplication operators are allowed.
So what I try with the yes minus three
inval operation, yes, by three inval
is only allowed operations on the string is plus 10 stop.
Okay? Similarly, let's take another different data type.
Let's take a list

here.

X equal, I'M writing list of some values.

10, 20, 30, okay,

yeah, x plus hundred.

So please tell me the data types here.

X list plus in, actually this is invalid application,

okay, this is invalidation.

So list two plus inte is invalid.

But a list, two plus list, try to do that.

The access list of this more list M adding.

Now access five elements, upper plus,

there are two elements in the list.

Now this is valid operation between list objects.

If you apply the plus list con coordination will have that.

I'll just check it. Five plus two total seven elements will

be con coordinated as a single string.

Check the output. Okay, list

to plus list is a valid application.

Same thing. Let's take Y equals stream

here I'm taking 102 hundred.

There are two values Y in two three

similar to what happened when I applied the multiplication.

Operat must a string is repeated for 10 times.

Okay, so now Y into three.

I said why has 102 hundred values a hundred,

200 are repeated for three times.

One more list will be created. That has total six values.

Just check it under 200, a hundred, 200,

a hundred, 200 likes.

So that means on list what automatic operators

one is plus second one

is multiplication.

Operators are so remaining operators,

not only these two are allowed.

Similarly, I'm testing with the tap.

I'm going to do the same trial on the tap.

So this is a symbol for the tap.

Let's say hundred, 200,

301 more thing.

I'm taking a new double

506 sample.

Now both are doubles here. Info plus new.

So info has three elements

and then new has two elements, totally five elements.

You should get it. So plus solves.

So plus is allowed between double solves.

Okay, look into this. X or x, X is a list.

Try to say info plus x. What up? And just check it.

Info is a apple here, X is a list.

Invalid top plus

list,

invalid other.

This info plus x is actually list.

I'm converting that into double now both are doubles.

Now check this is valid.

Optic info has three elements.

The X has five elements.

All the five elements have a con as a single double.

Okay, one more trial we do

with the multiplication operator.

Let's look into info again, I have three values

now info into, there are three values.

All the three values are repeated.

Again, two times that will be repeated. Now check it.

So a hundred, 200, 300, 1, time hundred, 200 per time.

Again second time.

So on top what operators are allowed

again plus multiplication operators are allowed.

Now I'm going to take the same operators trials on

the dictionary.

Object. D,

C, DA.

C means stands for dictionary is a

collection oft

and value based.

Let's take some the info for our bank account.

I'm taking your account number as key.

Your balance amount as well.

1 0 3 50,000 1 0 3.

That's a 90 plus thing that I have total.

Three account calls in my bank.

So account number has key that balance amount as well.

Okay, now 1 0 2 is depositing

10,000.

So 1 0 2 previous balances thousand.

Now if you are, if fees deposit ticket it 10,000 means

that the 10,000 should be added.

So for this bank of 1 0 2 plus equals

incremental assessment.

Okay? Plus equal equals to 10,000.

Now check latest values of it. Latest values of the 1 0 2.

So previously 50,000. Now his balance is 60,000.

Similarly, 1 0 3 is withdrawing 30,000

1 0 3 is with withdrawing.

30,000 previous balance 90.

If 30,000 is withdrawn, 90 minus 30, the balance,

latest balance should be 60,000.

Now bank of 1 0 3

here, decremental.

Assignment minus equal, equal minus equal, equal to 30.

Check the latest values of the bank.

Now this one,

so on dictionary, water operators are allowed,
all operators are allowed on
dictionary base.
All operations are cloud
like uh, plus minus,
multiplication, division,
Floor division, power values,
everything is a lot on the dictionary
but on the value part, okay, so previously you are adding,
you are adding some value to the 1 0 2.
You are directing some value from 1 0 3 like that.
But what if like uh, between the dictionary,
if I apply the operations, what happens,
let's say look into the bank.
I have three accounts. One more branch.
I'm taking, let's say bank two.
Think that this is of a different branch.
I'm taking 2 0 1 10,002 zero,
two 30,000.
Think that in the branch two bank is branch one,
bank two is branch two.
Between banks. Between these dictionaries, I'm going
to apply optimatic operation.
What happens? Just check it. It is not allowable.
Okay, so between dictionaries here, the comment
within a dictionary you can apply all these operators.
But between the dictionaries, across the dictionaries,
you cannot apply other medical operations.
Okay? So between dictionaries
we cannot apply
automatic operations.
This point you just remember
now I'm going to take one more collection, which is set.
What is set? Set is a collection of unique items.

I'm taking set one.

Let's say a hundred, 200, 300

and 400, 500 total of five values I have taken.

Even if I type, if I give duplicate values sheet one set,

I'm taking set to hundred

200, 700, 900, 300.

So here five values in the five values in the S.

Okay, other here, here the special operators.

If I say yes, one plus S two, what happens?

Invalid, same minus.

Okay. Yes. One minus S two here minuses alone. What happens?

You just check it. The values which are available only in

the S one but not available in the S two.

Okay. Yes. One minus S two.

What it is giving the values

which are available only in as well,

but not in

so here minuses.

So minus is not allowed of at least not allowed for couple.

Not uh, not allowed of between the dictionaries

but within the dictionary.

Minuses allowed because we not arrive with the draw.

The amount to be directed.

We have seen that, uh, in the previous example. Got it.

Similarly, I'll apply something. Just one.

The law operates just two. Yes.

One, just look at what happens.

Just by seeing this output, you just guess it.

I have given the pipe operator.

Actually it is a bit twice Operator. Okay, between the sets?

Yes. One is a set, yes. Two is a set between the sets.

I have given pipe operator. That means our operator.

What happens to the union? Office one and has three happen.

But duplicate simulator a hundred once union is done in the S one.

Five elements. Yes. Two, five elements total. 10 elements.

But a hundred is taken only one time.

200 taken one time 300 taken 100. Okay.

400, 500, 700, 900. Like this.

That means here what is happening? What functionality?

Union. Got it.

Similarly, end operator.

I'm bit that S one a percent,

yes, you'll get common.

So in both sets, a hundred, 200, 300 a day. Common elements.

Understand? So this is what like uh,

a simple, I'm taking my friends.

My friends are Ravi,

Gary Siri and man,

and these are my friends.

Think that these are my Facebook friends

and I'm taking your friends.

Your friends are Ravi.

Ravi are also my friend and also and sir,

so between you and me, what are the common friends?

I want common friends now.

So you can say in Facebook,

technology mutual friends, look at this.

The common friends equal to my friends the intersection

of the intersection and percent.

And your friends.

Now just look into this common.

What is your understanding? Common friends from both.

Who? Only my friends. Those friends should not be with you.

Who are those GI money When

that means that these are friend of friends to you.

I'm your friend and gi money.

When are my friends but not with you.

Understand those are friend of friends party.

Tomorrow when I will log into the Facebook,

these three will be recommendation to you.

You may know this person. Got it.

That means the recommendation to you.

Your recommendation,

the recommendations for you is my friends,

minus your friends.

Now look into the final output. Your recommendations,

check the output, give money.

Previously you don't know them.

Now these three are recommendation.

Who are these three friend of friends to you?

Similarly, I read my recommendations, my recommendation.

I'm just reversing the previous acquisition.

Your friends minus my friends.

Now just look into this panel output of it,

my recommendation.

Got it. Jun,

were not only exclusively your friends, I don't know them.

So, but you are already my friend

and these three are friend of friends to me.

That's why these three are recommendation to me tomorrow.

If I sign into Facebook, these three will be recommendation

to understand.

So I'm not way here.

The allowed operators

on set one is minus

five and percent.

And what is one more thing?

So only these three operators we have taken. Got it.

These are the allowed operations.

We can do this for minus only available

with you, but not with me.

So in the case of my friends, manage
or friends means only my friends, but not your friends.
Got it. I want common friends.
I want common friends. And percent intersection
and percent is for intersection.
Here. Understand, let's apply the pipe.
What happens? Your friend, my friends
pipe, your friends.
Just check this. I got union.
Understand. So that means totally
your friends and my friends.
Okay, this is union is happening.
Finally, what is your understanding?
The single pipe is
for union.
Not single amp percent is for intersection.
I mean common to say common friends minuses were
for what?
Only available with set one,
but not with set.
So all these sessions we discussed about
in all our previous sessions we discussed about different
operators, different types of operators,
and how to apply them, apply them, and different data types.
All these things we have seen in one single session.
In the next session onwards, we are going
with the Stan statements.
Okay, thank you very much. Let's meet in the next session.
Bye-bye.