Welcome back everyone

Agenda :-

1. Compound Assignment operator
2. Identity operator
3. Membership operator
4. Precedence of operator
5. Decision Flow

Compound Assignment Operators

a++ ⇒ error

Python works on expression a lot like maths

pre increment & post increment is not valid / works in python

10 ⟶ 5 + 10

||

15 + (+10)

15 + (+(-10))

-(+15) + (+10)

5 ⟶ 5 + + (error)

5 + 10 ⇒ 5 + (+10)

Identity Operator

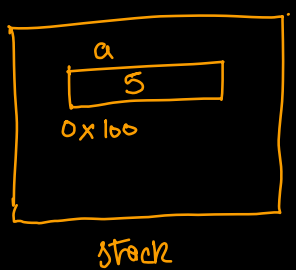
1] is
2] is not

2 purposes :-

1. It is used to check if value is of certain type

```
a=5          # int
type(a) is int
```

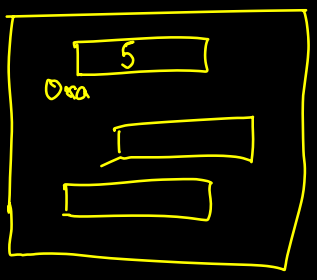
2. To verify if two references have same memory location



a = 5
↓
binary

0x ⟶

when you define a var
python assigns an address
to that var & stores the value



a = 5

b = 5

a	0x01
b	0x01

Store the above variables address as 1.

id (__) ⇒ address

var

Membership Operator

1] in
2] not in

teams = [___ , ___ , ___ , ___]

Chennai in team : True

Pune in team : False

Pune not in team : True

Collections = str
list
set
tuple
range
dict

To check if something is part of your collection, this is used.

Precedance of operator

Brochet of Division Multiplication Addition Subtraction

45 + 50/5 ⇒ 55

"Biryani" and "Dosa" in Orders

Operators	Meaning
`()`	Parentheses
`**`	Exponent
`+x`, `-x`, `~x`	Unary plus, Unary minus, Bitwise NOT
`*`, `/`, `//`, `%`	Multiplication, Division, Floor division, Modulus
`+`, `-`	Addition, Subtraction
`<<`, `>>`	Bitwise shift operators
`&`	Bitwise AND
`^`	Bitwise XOR
`|`	Bitwise OR
`==`, `!=`, `>`, `>=`, `<`, `<=`, `is`, `is not`, `in`, `not in`	Comparisons, Identity, Membership operators
`not`	Logical NOT
`and`	Logical AND
`or`	Logical OR

Associativity in Python

When 2 or more operators follow same precedence.

Python follows associativity almost all operators follow → left to right

5 and 6 and 7

ℓ ↓ r

↓

final answer

Operators	Associativity
() Highest precedence	Left - Right
**	Right - Left
+x , -x, ~x	Left - Right
*, /, //, %	Left - Right
+, -	Left - Right
<<, >>	Left - Right
&	Left - Right
^	Left - Right
|	Left - Right
Is, is not, in, not in, <, <=, >, >=, ==, !=	Left - Right
Not x	Left - Right
And	Left - Right
Or	Left - Right
If else	Left - Right
Lambda	Left - Right
=, +=, -=, *=, /= Lowest Precedence	Right - Left

Print statement

Format specifier

%d ⇒ integer
%s ⇒ string
%i ⇒ integer
%f = float

1) no. of format specifier and variable should match
2) %s can be used for non-string value
3) %d cannot be used for strings.

.format

new way of printing

Gets rid of %d, %s,
and makes string formatting
easier & more regular.