Operators:

- Operators are special symbols or keywords used to perform operation on one or more values (operands).

○ Arithmetic assignment operators:

Operators	Example	Same as
= (Assignment)	x = 5	x = 5
+= (Add and assign)	x += 3	x = x + 3
-= (Subtract and assign)	x -= 3	x = x – 3
*= (Multiply and assign)	x *= 3	x = x * 3
/= (Divide and assign)	x /= 3	x = x / 3
%= (Modulus and assign)	x %= 3	x = x % 3
**= (Exponentiation and assign)	x **= 3	x = x ** 3
//= (Floor divide and assign)	x //= 3	x = x // 3

Comparison Operators:

Used to compare two values, returning a Boolean (True or False) results.

Let x = 4 and y = 5

Operators	Example	Result
==	x == y	False
!=	x != y	True
>	x > y	False
<	x < y	True
>=	x >= y	False
<=	x <= y	True

- Logical Operators:

Used to conditional statements

Operator	Example	Description
and (Logical AND)	(condition1) and (condition2)	TRUE → If both conditions satisfies FALSE → if any of the condition(s) is/are False
or (Logical OR)	(condition1) or (condition2)	TRUE → of any or all conditions are satisfied FALSE → if all conditions are not satisfied
not (Logical NOT)	not (condition)	Invert the Boolean result

- Membership Operators:

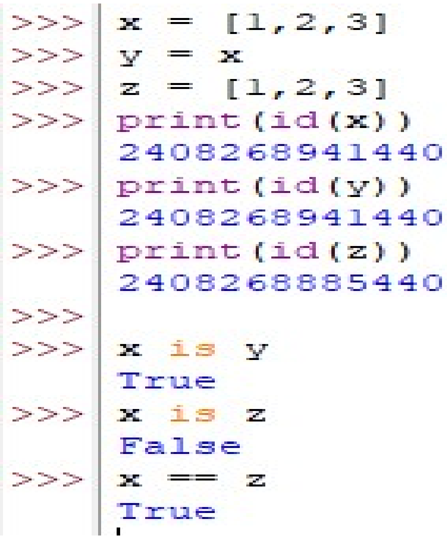
Used to test if a sequence contains a specific value for e.g. strings, lists

Operator	Example	Description
in (Is present in)	x in y	Returns TRUE if a sequence with the specified value is present in the object, otherwise FALSE
not in (Is not present in)	x not in y	Returns TRUE if a sequence with the specified value is not present in the object, otherwise FALSE

○Identity Operators:

Used to check of two variables refers to the same object in memory

Operator	Example	Description
is (is the same object)	x is y	Returns TRUE if both variables are the same object, otherwise FALSE
is not (is not the same object)	x is not y	Returns TRUE if both variables are not the same object, otherwise FALSE



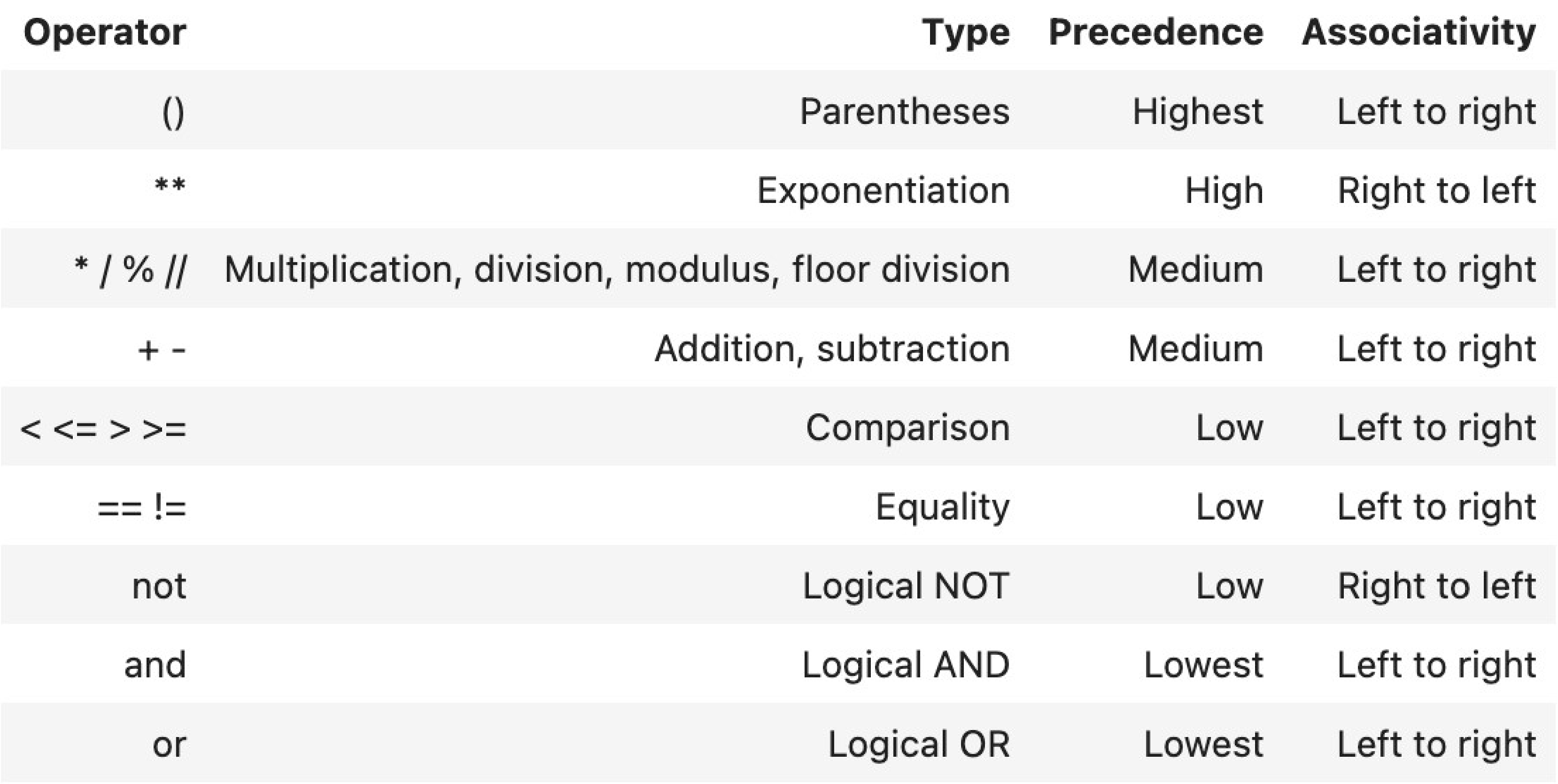
```
>>> x = [1,2,3]
>>> y = x
>>> z = [1,2,3]
>>> print(id(x))
2408268941440
>>> print(id(y))
2408268941440
>>> print(id(z))
2408268885440
>>>
>>> x is y
True
>>> x is z
False
>>> x == z
True
```

○ Bitwise Operators:

Bitwise operators are used to perform operation directly on the individual bits of integers. They treat integers as sequence of 1’s and 0’s and manipulate them on low-level.

Operators	Example	Description
& (Bitwise AND)	12 & 13	1100 & 1101 = 1100 (decimal 12)
| (Bitwise OR)	12 | 13	1100 | 1101 = 1101 (decimal 13)
~ (Bitwise NOT)	~12	Equivalent to –(x+1)
^ (Bitwise XOR)	12 ^ 13	1100 ^ 1101 = 0001 (decimal 1)
<< (left shift)	5 << 2	0101 << 2 = 10100 (decimal 20)
>> (Right shift)	20 >> 2	10100 >> 2 = 00101 (decimal 5)

Operators Precedence and Associativity:



Operator	Type	Precedence	Associativity
()	Parentheses	Highest	Left to right
**	Exponentiation	High	Right to left
* / % //	Multiplication, division, modulus, floor division	Medium	Left to right
+ -	Addition, subtraction	Medium	Left to right
< <= > >=	Comparison	Low	Left to right
== !=	Equality	Low	Left to right
not	Logical NOT	Low	Right to left
and	Logical AND	Lowest	Left to right
or	Logical OR	Lowest	Left to right

Operators Precedence and Associativity:

Sample Expression

Expression 1

=> 10 + 3 * 2 **3 – (6 // 2 + 1) * 3 % 4

=> 10 + 3 * 2 **3 – (3 +1) * 3 % 4

=> 10 + 3 * 2**3 – 4 * 3 % 4

=> 10 + 3 * 8 – 4 * 3 %4

=> 10 + 24 – 12 % 4

=> 10 + 24 – 0

=> 34 – 0

=> 34

Expression 2

⇒10 / 2 + 3 * 2 ** 3 – (6//2 + 1) * 3 % 4

⇒10/2 + 3 * 2 ** 3 – (3 + 1) * 3 % 4

⇒10/2 + 3 * 2 ** 3 – 4 * 3 % 4

⇒10/2 + 3 * 8 -4 * 3 % 4

⇒5 + 3 * 8 -4 * 3 %4

⇒5 + 24 – 12 % 4

⇒5 + 24 – 0

⇒29 – 0

⇒29

Expression 3

=> 10 * 2 + 3 * 2 ** 3 - (6//2 +1)/3 % 4

=> 10 * 2 +3* 2**3 -(3+1)/3 % 4

=> 10 * 2 +3* 2**3 -4/3 % 4

=> 10 * 2 +3* 8 -4/3 % 4

=> 10 * 2 +3* 8 -4/3 % 4

=> 20 + 3 *8 - 4/3 % 4

=> 20 + 24 - 4/3 % 4

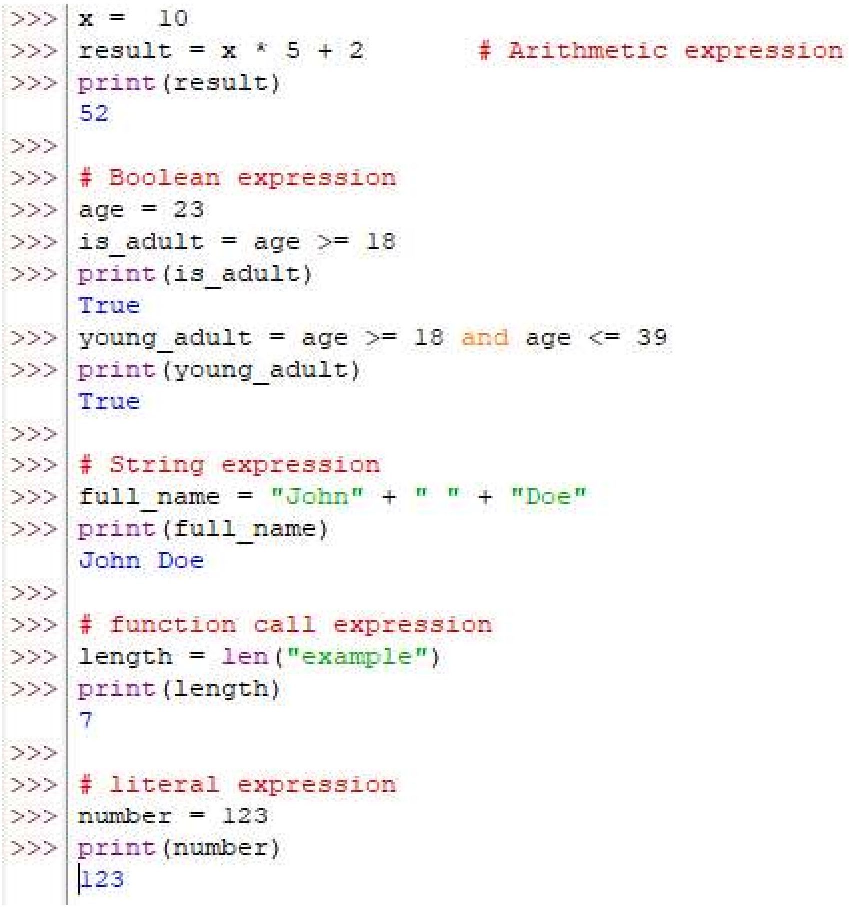
=> 20 + 24 - 1.3333 % 4

=> 20 + 24 - 1.3333

⇒ 44 - 1.3333

⇒ 42.66

Expressions:



```
>>> x =  10
>>> result = x * 5 + 2       # Arithmetic expression
>>> print(result)
    52
>>> 
>>> # Boolean expression
>>> age = 23
>>> is_adult = age >= 18
>>> print(is_adult)
    True
>>> young_adult = age >= 18 and age <= 39
>>> print(young_adult)
    True
>>> 
>>> # String expression
>>> full_name = "John" + " " + "Doe"
>>> print(full_name)
    John Doe
>>> 
>>> # function call expression
>>> length = len("example")
>>> print(length)
    7
>>> 
>>> # literal expression
>>> number = 123
>>> print(number)
    123
```

Basic Python Tasks:

1. Write Python program to take radius of circle as input and computes the area of circle ($A = \pi r^2$)
2. Write Python program to compute area of triangle ($\frac{1}{2} \times b \times h$)
3. Write Python program to solve (x + y) x (x + y). Let x =4 and y = 3
4. Write Python program to find distance between two points (x1, y1) and (x2, y2). Points coordinates are input by user.

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$