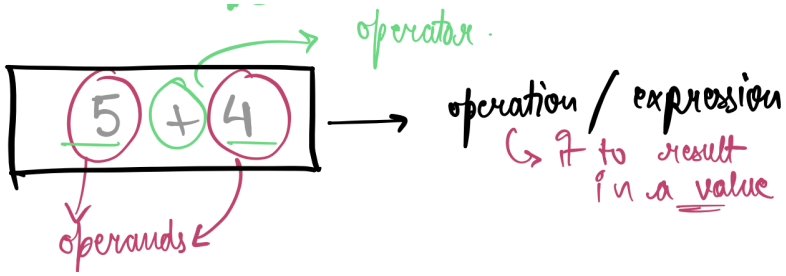
Operators

Content

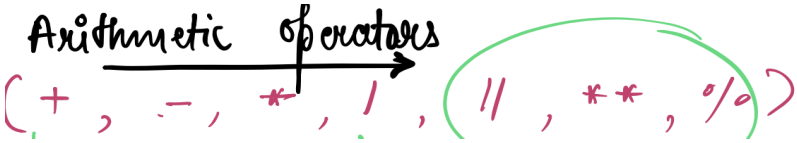
Operators

- **Operators** are symbols of operation.
- **Operands** are values on which operation is happening
- Operation/Expression are combination of operands and operators
- Operation/Expression gives a result after execution.



Arithmetic operators

- operators such as `+`, `-`, `*`, `/`, `//`, `**`, `%`




```
# Addition
10 + 3
```

```
    13
```

```
# Subtraction
10 - 3
```

```
    7
```

```
# Multiplication
10 * 3
```

```
    30
```

- The return type of the division operator `/` is always a floating point object.

```
# Division
10 / 3
```

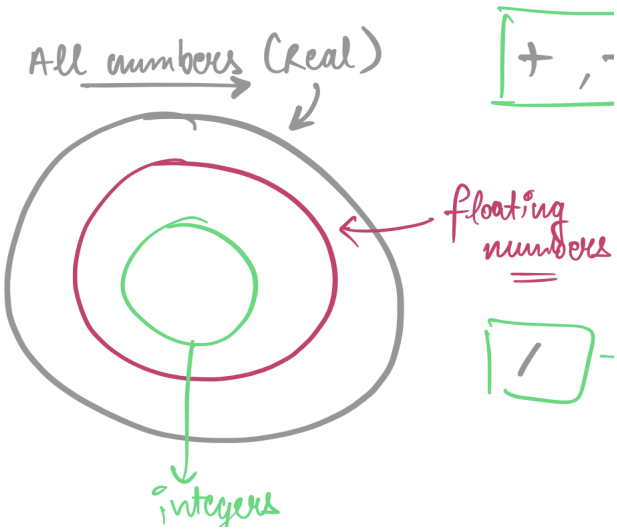
```
3.3333333333333335
```

```
# float + float -> float
5.0 +5.0  #10.0
```

```
10.0
```

Subset diagram of number system -

integers ⊂ floats ⊂ real numbers



- Python retains as much information as it can.
  - for +, -, * if one of the operand is float, then result is float.
  - for / operator, the result is always float.

Quiz-1

Question: What would be the output of the following?

```
x = 10
y = 2.5
print(x / y)
```

A. `4`

B. `4.0`

C. `4.5`

Ans: B. `4.0`

Quiz-2

Question: What would be the output of the following?

```
1.0+2
```

A. `3.0`

B. `"3.0"`

c. `3`

Ans: A. `3.0`

Some more operators -

Exponentiation operator

`x**y` = x^y

```
5 ** 2
```

```
25
```

```
5 ** -1
```

```
0.2
```

```
5 ** 0.5 # square root
```

```
2.23606797749979
```

```
5.0 ** 2
```

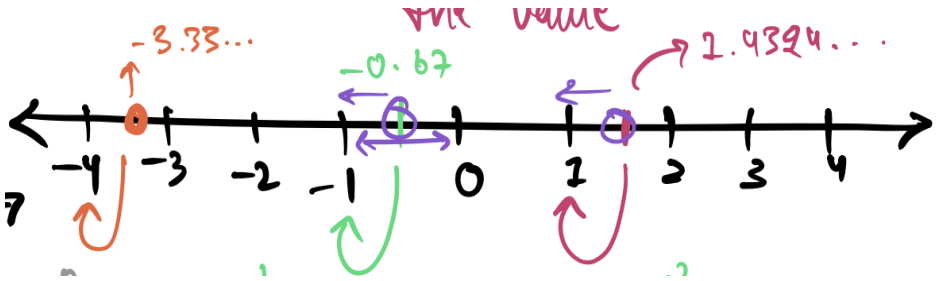
```
25.0
```

Floor Division operator (`//`)

`x//y` = floor(x/y)

Floor function -

- The floor() function gives the biggest integer less than the value.
- On a number line, it gives the closest integer to the left of the value.
- Example:
  - floor(-0.67) gives -1
  - floor(2.3) gives 2




```
10//3
```

```
3
```

```
-10//3
```

```
-4
```

Modulus operator (`%`)

`x % y` -> remainder of `x / y`

- If `x` is **'+ve'** -> remainder of `x / y`
- If `x` is **'-ve'** -> `y - (x % y)`

```
10 % 3
```

```
1
```

```
-10 % 3
```

```
2
```

Quiz-3

Question: Predict the output.

```
print(10**-1)
```

A. `10`

B. `1/10`

C. `0.1`

Ans: C. `0.1`

Quiz-4

Question: What would be the output of the following?

```
15//2
```

A. `7.5`

B. `7`

C. `1`

Ans: B. `7`

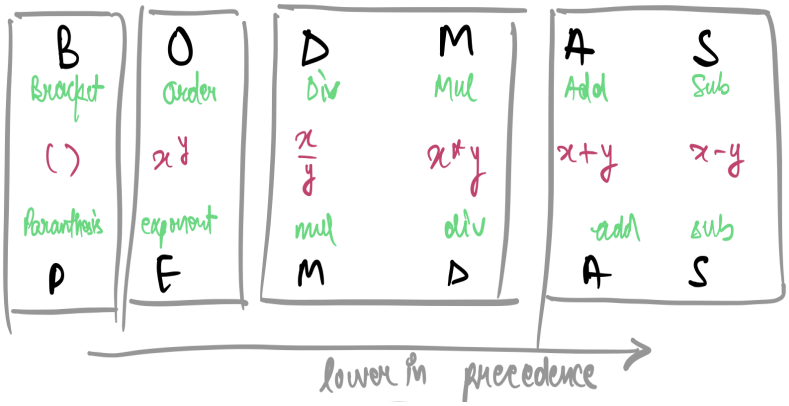
Precedence of operators

Question: Predict the ouptut

```
10-4*2
```

Ans: 2

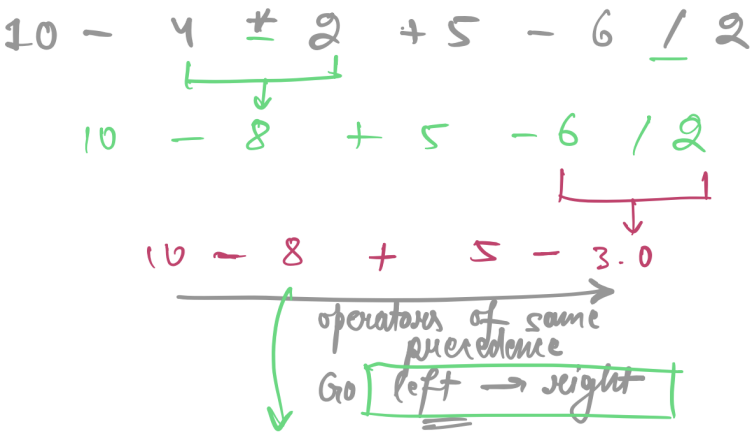
- Division and Multiplication have the same order of precedence.
- Addition and Subtraction have the same order of precedence.
- In case of encountering operators with same precedence, go from **left to right**.

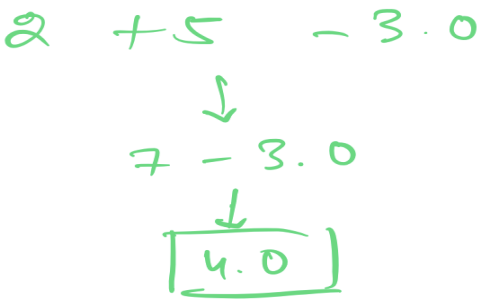


Question: Predict the output.

```
10-4*2+5-6/2
```

Ans: 4.0





Quiz-5

Question: Predict the output.

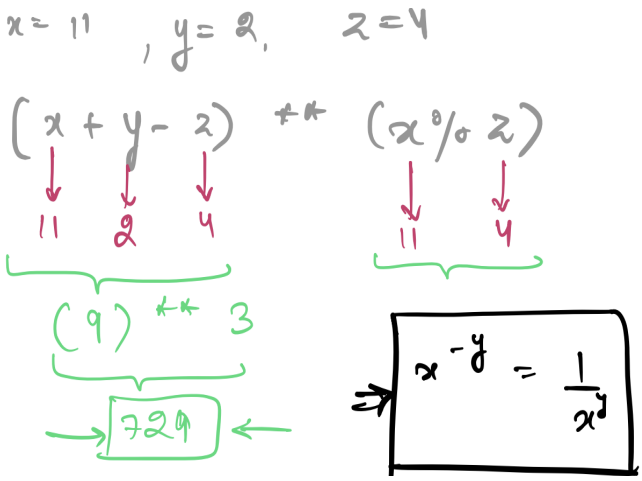
```
x = 11
y = 2
z = 4
res = (x + y - z) ** (x % z)
print(res)
```

A. 729

B. 81

C. 9

Ans: A. 729



The `bool()` function

In Python,

- Every number except 0 is `True`.
- Every non-empty strings is `True`.

```
bool(1)
```

```
    True
```

```
bool(0)
```

```
    False
```

```
bool(-123)
```

```
    True
```

```
bool("Hello")
```

```
    True
```

```
bool("")
```

```
    False
```

```
bool(" ")
```

```
    True
```

Question: Predict the output

```
print(bool('false'))
```

A. `True`

B. `False`

Ans: A. `True`

Question: Predict the output

```
print(int(True))
```

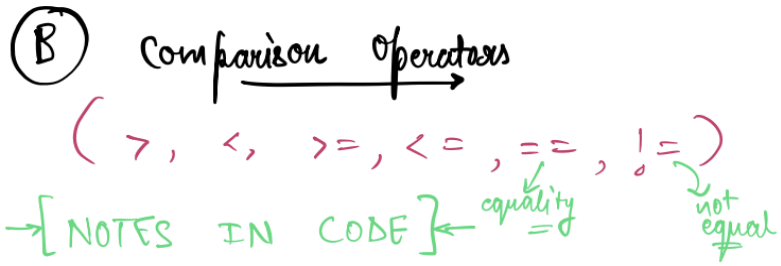
A. 1

B. 0

Ans: A. 1

Comparison operators

- operators such as `>, <, >=, <=, ==, !=`




```
my_marks = 50
my_cousin_marks = 90
```

```
my_marks > my_cousin_marks
```

```
    False
```

```
my_marks == my_cousin_marks
```

```
    False
```

```
my_marks = 90
```

```
my_marks == my_cousin_marks
```

```
    True
```

```
my_marks!=my_cousin_marks
```

```
    False
```

Assignment operators

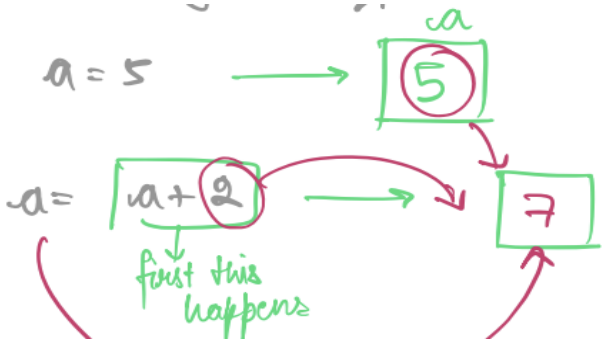
- `=` is an assignment operator.
- It assigns the RHS operand value to the LHS operand.

```
a = 4 * 4
a
```

```
    16
```

```
a = 5
a = a + 2
a
```

```
    7
```




```
# Question
# What will be the output?

marks = 50
correction = 0.5
marks = marks + correction

marks = marks + correction
marks
```

```
51.0
```

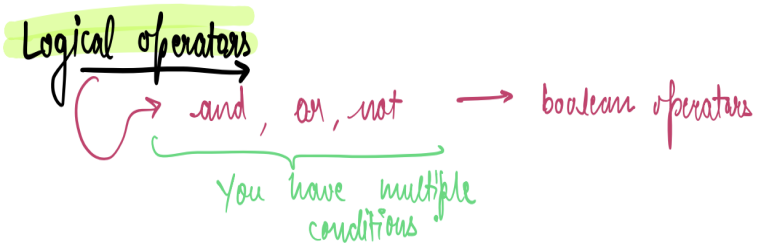
```
# Question
# What will be the output?

a = 5
a += 2
print(a)
a -= 2
print(a)
a *= 2
print(a)
```

```
7
5
10
```

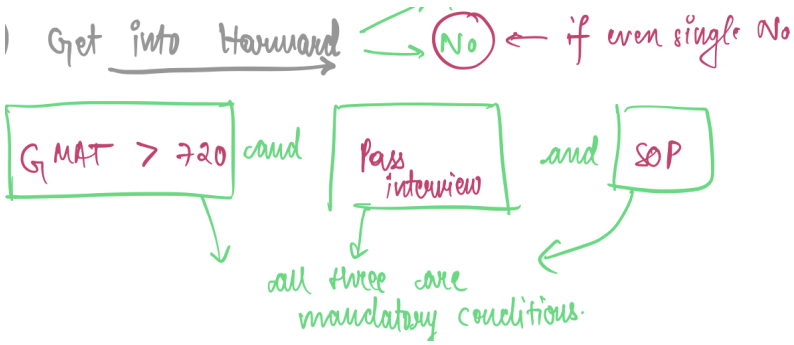
Logical operators

- `and`, `or`, `not`
- used when there are multiple conditions

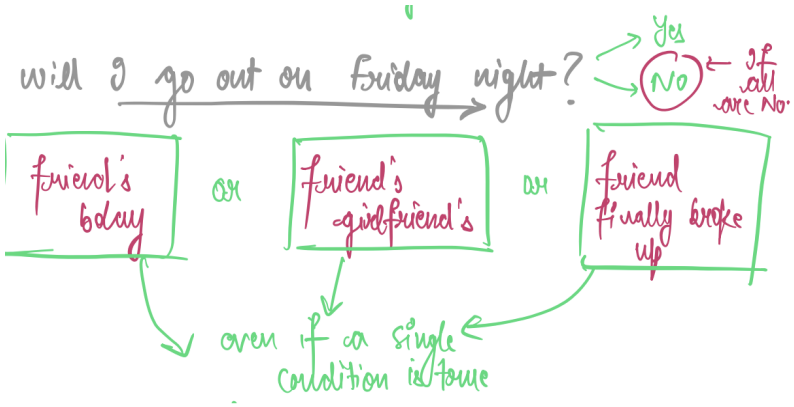


`and` operator

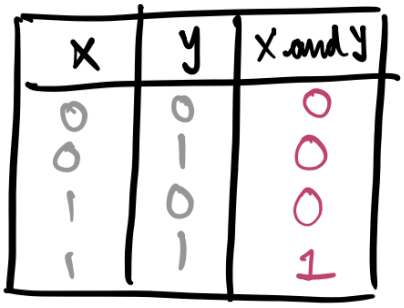
- `True` only if all the conditions are satisfied.



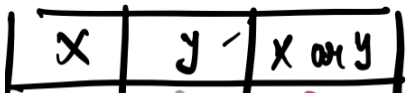
- `or` operator
  - `True` if any one of the conditions are satisfied.



- **Truth Tables**
  - **x** and **y** are conditions
  - 1 is `True`, 0 is `False`
- Truth table for `and`



- Truth table for `or`



Question: Predict the output of the following pieces of code.

```
a=10
```

```
a>5 or a>20 or a<2 #True


a>5 and a==20 and a<2 #False
```

Check if a number is a multiple -

```
amount = 1000
amount % 500 == 0
```

```
True
```

ATM Dispatch

```
amount= int(input("Please enter the amount to withdraw: "))
amount % 500 == 0 or amount % 200 == 0
```

```
Please enter the amount to withdraw -2039
False
```

`not` operator

- Works only with **boolean** operands.

```
print(not True)

print(not False)

print(not 24 < 56)
```

```
False
```