Type Conversion in Python

↳ Process of converting one datatype to another

- Implicit (automatically)
- Explicit (manually)

① Implicit type Conversion

```
x = 10    #int
y = 5.5    # float
z = x+y    # x is converted to float
print(z)
print(type(z))
```

② Explicit type Conversion

- int()
- float()
- str()
- list()
- tuple()
- set()

① String to int

```
age = "28"
new_age = int(age)
print(new_age)
print(type(new_age))
```

② Integer to String

```
number = 30
new_number = str(number)
print(new_number)
```

③ list to set

```
numbers = [1, 2, 3, 2, 1]
new_numbers = set(numbers)
print( — )
```

④ String to float

```
pi = "3.14"
new_pi = float(pi)
```

print( )

⑤ type to list

```
x = (10, 20)
new_x = list(x)
print(new_x)
    ↳ type(-)
```

Operators

↳ Symbols used to perform operations on variables & values (operands)

```
a + b
↑
```

① Arithmetic operators

			o/p
①	+	10 + 2	12
②	-	10 - 2	8
③	*	10 * 3	30
④	/	10/3	3.33

⑤ % = 10%3 1

⑥ // (floor division) = 10//3 3

⑦ Exponential (**) = 2**3 8

② Comparision / Relational operators

① == Equal to

② != Not equal to

③ >

④ <

⑤ >=

⑥ <=

10==5 F

10!=5 T

10>5 T

10<5 F

10>=10 T

10<=5 F

③ Assignment operators

① =

② +=

③ -=

④ *=

⑤ /=

⑥ %=

⑦ //=

x = 10

x+=5

x-=5

x*=5

x = x+5

x = x-5

x = x*5

x%=5 x = x%5

(3) **=

x = 10 % 3

x %=

10 % 3 ⇒ 1

x = 1

(4) Logical operators

(1) and
(2) or
(3) not

A and B

T T ⇒ T
T F ⇒ F
F T ⇒ F
F F ⇒ F

or

T T ⇒ T
T F ⇒ T
F T ⇒ T
F F ⇒ F

```
x = 10
print( x > 5 and x < 20)
print ( not (x > 5)) => false
```

(5) Bitwise operators ⇒ 10 → 1010

AND 5 & 3 = 1
OR 5 ' 3
NOT
XOR
Left shift

Right shift

5 ⇒ 101

3 ⇒ & 011

001 ⇒ 1

2

101

(OR) 011

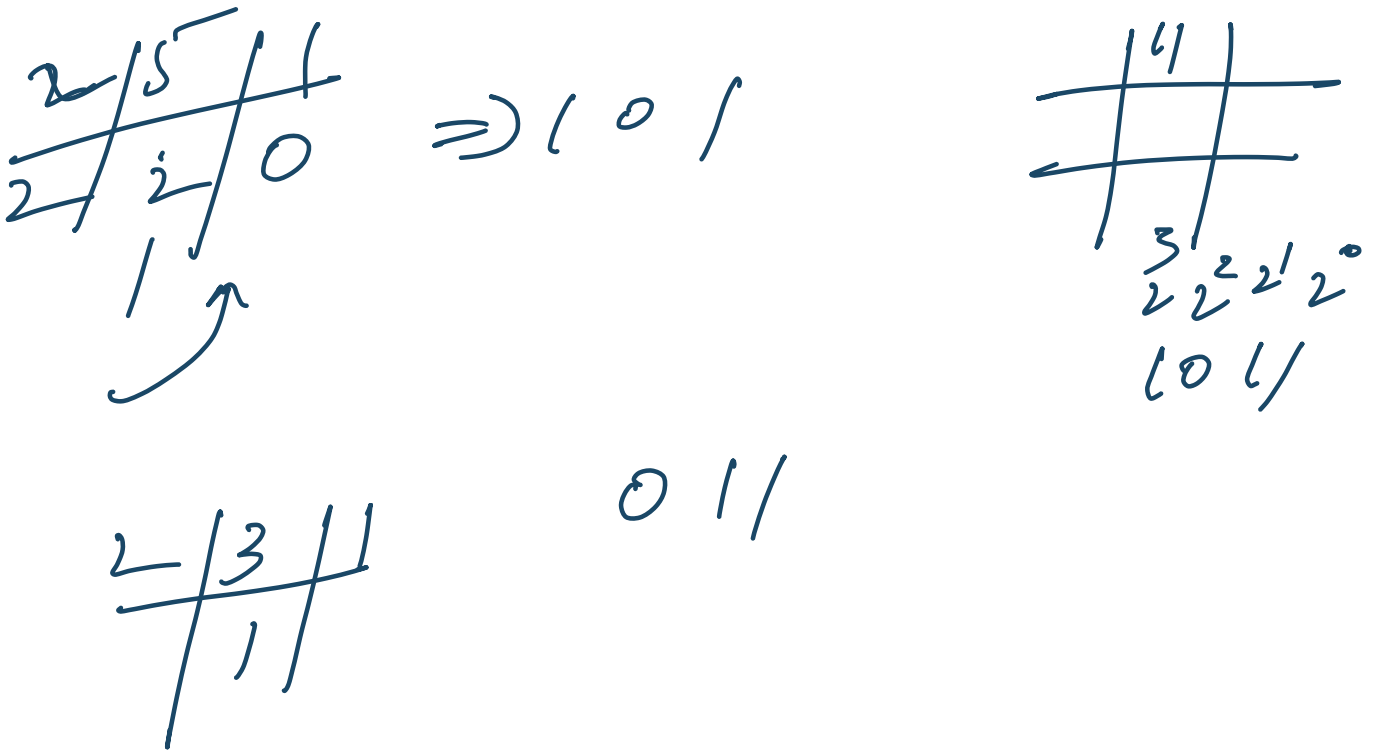
111

8 ⇒ [1, 1, 2, 3, 4, 2, 5]

⇒ 1 2 3 4 5

XOR

A	B	A ^ B
1	1	0
0	1	1
1	0	1
0	0	0



6. membership operators

in

not in

name = ["Arun", "Sourav", "Sumit"]

print("Arun" in name) True

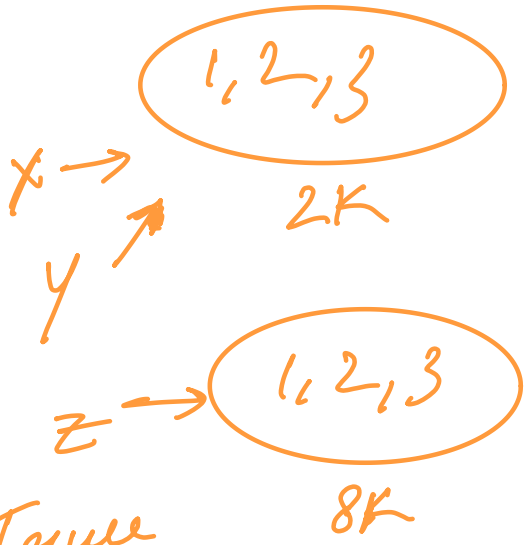
```
name = ["Arun", "Sourav", "Sumit"]
print("Arun" in name)    # True
print("Arun" not in name)     #False

print("A" in "Arun")
print("B" not in "Arun")
```

7. Identity operators

- is
- is not

↳ it compares the memory location of the object

```
x = [1, 2, 3]
⇒ y = x
z = [1, 2, 3]
print(x is y) // True
print(x is z) //
```



Basic Rules in Python

1. Case-sensitive (name / Name)
2. uses indentation to define blocks of code
3. No need of semicolon
4. Comment ⇒
   - Single line Comment ⇒ #
   - multi line Comment ⇒ 

↳ use triple quotes

⇒ Indentation

Input & output

↳ print()

↳ input()

↳ By default it takes string as input

age = input("Enter your Age") // 25

↳ "25"

```
name = input("Enter Your Name ")
print("Name:", name)

age = int(input("Enter Your age "))
print("Age:", age)
print(type(age))
```

Taking multiple inputs

↳ split()

```
x, y  = input("Enter 2 numbers separated by space ").split()
print("first number: ", x)
print("second number: ", y)
```

Printing output

```
x = 10
y = 20
sum = x+y
print("The sum of", x, "and", y, "is", sum)
print(f"The sum of {x} and {y} is {sum}")
```

```
print("Hello")
print("World!")


print("====================")


print("Hello", end = " ")
print("World!")
```