**Day-01**

**28-02-2025**

**================**

**Type Conversion:**

**============**

-> When we want to convert the data from one format to another format, then we can use "type conversion".

-> Type conversion is called as "Type casting".

-> There are two types:

1) Internal Type conversion

2) External Type conversion

-> Internal Type conversion is also called as "Automatic type conversion".

Based on the operation, the system can automatically convert the data into another format is called as "Automatic Type conversion".

-> Internal Type conversion also called as "Implicit Typecasting".

a = 13

b = 4

print(type(a)) # int

print(type(b)) # int

a = a / b

print(type(a))

print(type(b))

-> External type conversion is also called as "Explicit Type conversion".

-> Based on the requirement, the programmer can convert the data into required format is called as "Explicit Type conversion".

-> To perform the explicit type conversion, we have pre-defined functions in python:

1) int()

2) float()

3) complex()

4) bool()

5) str()

-> Function:

is a named block,

block consisting of group of statements

to perform the specific functionality/task.

Ex: type()

-> Functions can be two types:

1) Built-in functions/pre-defined functions

2) User-defined functions

1) int():

---------

-> when we need to convert any data into integer format, we can use "int()".

-> int() can always convert the data into decimal format only.

Syntax:

int(data/name-of-data)

**rules for int():**

**---------------**

-> int() can convert any integral value like binary, octal and hexadecimal into decimal.

-> float formatted value can convert into decimal.

-> The complex number not possible to convert into decimal.

-> Boolean formatted data can convert into integer.

-> String to integer conversion:

if a string with digits -> possible to convert into an integer.

but if a string with 'alphabets' or with 'other characters' -> not possible to convert into an integer.

a = 0b11001

b = 0o121

c = 0xaf

print(int(a))

print(int(b))

print(int(c))

print(a)

print(b)

print(c)

print(int(123.001932))

print(int(12345678901e-7)) # 12345678901 X 10^-7

# print(int(123-234j))

print(int(True))

print(int(False))

print(int('10223'))

# print(int('abc'))

2) float()

----------

-> when we need to convert any data format into integral format, we can use "float()".

Syntax:

float(data/name-of-data)

**Rules:**

**------**

-> Any integral formatted value can convert into float.

-> Complex numbers are not allowed for the conversion into float.

-> Boolean to float conversion is possible.

-> String to float:

i) if a string with 'decimal' number

ii) if a string with 'float'

are allowed to convert into the float.

iii) Other than the above, any values not allowed to convert into float.

print(float(10229))

print(float(0b10101)) # binary --> decimal ---> float

print(float(0o102)) # octal --> decimal --> float

print(float(0xaf)) # hex --> decimal --> float

# print(float(102-230j))

print(float(True)) # boolean --> decimal --> float

print(float(False))

print(float('102')) # string --> decimal --> float

# print(float('0b1010101'))

print(float('12.002'))

# print(float('True'))

**Day-02**

**03-03-2025**

**================**

3) complex()

---------------

-> When we want to convert any data/value into complex number, we can use "complex()".

Syntax-1:

complex(data/variable)

Rules:

------

complex() can consider any given value as real part of the complex. So, it can add '0j' as an imaginary to that given real number to form a complex number.

1) Integer (any base) can possible to convert into complex.

binary --> decimal --> complex

octal --> decimal --> complex

hexadecimal --> decimal --> complex

2) Float value can convert into the complex value.

3) Boolean value can convert into the complex value.

4) String to complex:

i) when a decimal as string, can convert into the complex (only)

(No other base values of integer can convert into complex)

ii) when a float as string, can convert into the complex

iii) when a complex number as a string, we can convert into complex number.

print(complex(102)) # by adding '0j' as imaginary part to the given decimal

print(complex(0b10101)) # binary -> decimal -> complex by adding '0j' as imaginary

# complex() considers the give value as "real" part of complex and it can add '0j' as imaginary

print(complex(1.002))

print(complex(True)) # True --> 1 --> complex (1 + 0j)

print(complex('1302')) # str --> decimal --> complex

# print(complex('0b1010101'))

print(complex('1.0023'))

print(complex('12e-5'))

print(complex('12-23j'))

# print(complex('a'))

Syntax-2:

complex(value1/variable1, value2/variable2)

Here:

value1/variable1 --> real part of the complex number

value2/variable2 --> imaginary part of the complex number

Note:

-----

1) real and imaginary part can allow to define using either the positive values or negative values.

2) complex() can understand any number formatted data (even boolean vales also allowed) but not text formatted data.

print(complex(11,22)) # 11 -> real and 22 -> imaginary

print(complex(-101,-202))

print(complex(3021,0b11001))

print(complex(0XAf12,0O7654))

# print(complex('102','1.023'))

# print(complex(1-2j,2-3j))

print(complex(True,10))

# print(complex('102',100))

4) bool()

----------

-> When we want to convert any value into boolean value, we can use "bool()".

Syntax:

bool(value/variable)

Rules:

------

1) Any zero value (0, 0.0, 0+0j, 0-0j, 0b0, 0o0, 0x0,'') can understand as "False".

2) Any non-zero value (+eve, -eve, string with something) can understand as "True".

print(bool(0))

print(bool(0b00))

print(bool(0o00))

print(bool(0X00))

print(bool(0.0))

print(bool(0+0j))

print(bool(0-0j))

print(bool(''))

print(bool(1))

print(bool(-1))

print(bool(0.0001))

print(bool(1-0j))

print(bool('a'))

5) str()

---------

-> When we want to convert any data into string data, we can use "str()".

Syntax:

str(value/variable)

Rules:

------

1) Any integer formatted data, can convert into the string

2) Float value can convert into the string.

3) Complex value can convert into the string.

4) Boolean value can convert into the string.

print(type(str(123)))

print(type(str(0b1001)))

print(type(str(1.00023)))

print(type(str(1-23j)))

print(type(str(True)))

**Run-time variable:**

**============**

-> we can use input()

Syntax:

variable = input("message")

Note:

-----

input() can read any value as string by default.

-> to take any value using input(), we should define the type conversion to the input().

# to take an integer

a = int(input("Enter an integer value:"))

# to take float value

b = float(input("Enter a float value:"))

# complex number as input

c = complex(input("Enter some complex number:"))

# take the boolean as an input

d = bool(input("Enter the boolean value:"))

e = input("Enter any string:")

print(type(a))

print(type(b))

print(type(c))

print(type(d))

print(type(e))

Note:

-----

* Run-time variable definition is called as "Dynamic variable" definition
* This can always use in the "dynamic programs".