**Base Conversions**

**04-03-2025**

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

**Q: How we can define the variable with binary value, octal value and hexadecimal values in the run-time:**

Syntax for taking binary value in run-time is:

variable-name = int(input("Enter the binary value:"),2)

Here:

for the int(): we are providing two parameters

1) input()

2) base value of the binary ==> 2

Syntax for taking an octal value in run-time is:

variable-name = int(input("Enter the octal value:"),8)

Syntax for taking an Hexadecimal value in run-time is:

variable-name = int(input("Enter the Hexadecimal-value:"),16)

a = int(input("Enter the binary value:"),2)

b = int(input("Enter an Octal value:"),8)

c = int(input("Enter an Hexadecimal value:"),16)

print(type(a))

print(type(b))

print(type(c))

**Q: IS IT POSSIBLE TO PRINT BINARY AS BINARY, OCTAL AS OCTAL AND HEXADECIMAL AS HEXADECIMAL?**

Yes, we can do this using base conversions.

-> Base conversions ==> converting the value of one base to another base

-> For base conversions, we have three pre-defined functions:

1) bin()

2) oct()

3) hex()

bin():

------

-> bin() can be used to convert any base value into base-2 (binary)

Syntax:

bin(any-base-value)

-> the decimal value by dividing with '2' continuously until the value reached to '0' and while the division we need to capture the remainders and those remainders can arrange from bottom to top after the completion of the division.

for octal:

0 --> 000

1 --> 001

2 --> 010

3 --> 011

4 --> 100

5 --> 101

6 --> 110

7 --> 111

-> the given octal value with octal letters, each octal letter can expand with 3-bit binary to get final binary of the given octal.

-> For hexadecimal:

0 --> 0000

1 --> 0001

2 --> 0010

3 --> 0011

4 --> 0100

5 --> 0101

6 --> 0110

7 --> 0111

8 --> 1000

9 --> 1001

a --> 1010

b --> 1011

c --> 1100

d --> 1101

e --> 1110

f --> 1111

-> so, each hexadecimal digit of hexadecimal number need to expand with 4-bit binary and join with all to get the final binary for the given hexadecimal.

Note:

-----

1) Result type of bin() is "str"

we cannot use float, complex and strings for the base conversions.

a = 102 # decimal

b = 0o102 # octal

c = 0xaf12 # hexadecimal

d = bin(a)

print(bin(a)) # decimal to binary

print(bin(b)) # octal to binary

print(bin(c)) # hexadecimal to binary

print(type(d))

# print(bin(1.2)) # value error

# print(bin(1-2j))

# print(bin('101'))

print(bin(True))

print(bin(False))

oct():

------

-> when we need to convert any base value into octal (base-8), we can use "oct()".

Syntax:

oct(any-base-value)

-> In the given binary, from right to left the total binary can split into pairs, each pair has a length of 3. Then each 3-bit pair convert into octal.

-> The given decimal can divide with '8' till the value reached to '0' and while the division arrange all the remainders from bottom to top.

-> Hexadecimal value --> binary --> octal

a = 0b101010101

b = 102

c = 0x1021

print(oct(a))

print(oct(b))

print(oct(c))

print(oct(True))

hex()

-----

-> when we need to convert any base value into hexadecimal (base-16), we can use "hex()".

Syntax:

hex(any-base-value)

-> the given binary value can be split into multiple pairs. Each pair has the length of '4'. And finally each pair need to convert into the hexadecimal character.

-> Divide the decimal with '16' and arrange all the remainders from bottom to top after the division.

-> Octal --> binary --> hexadecimal

a = 0b1010101

b = 102

c = 0o102

print(hex(a))

print(hex(b))

print(hex(c))

Final Output:

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

a = 0b1010101

b = 0o102

c = 0x12af

print("Binary value = ",bin(a))

print("Octal value = ",oct(b))

print("Hexadecimal value = ",hex(c))