Literals and Data types

Literal is nothing value/data which never changed.
1. Numeric Literals
    a. Integer Literal
    b. Float Literal
    c. Complex Literal
2. Non Numeric Literals
    a. Boolean
    b. String
    c. None

Data types
Data types are used to allocate memory for data/literals.
Python data types are classified into two categories.
1. Scalar Data types
2. Collection Data types

Scalar data types are used to allocate memory for one value

1. Int
2. Float
3. Complex
4. Bool
5. NoneType

Collection data types are used to allocate memory for more than one value.

1. Sequences
    a. List
    b. Tuple
    c. String
    d. Range
    e. Bytes
    f. bytesarray
2. Sets
    a. Set
    b. frozenset
3. Mappings
    a. Dictionary

Python support 14 standard data types.

Int data type and integer literal/data/value

Integer data type is used to reserve memory for integer value/data.
An integer is numeric value without fractional part or decimal part.

What is variable?
A variable is an identifier, which is used to identify value.
Variable is named memory location.
Variable is container which contains value.

```
>>> a=10
>>> type(a)
<class 'int'>
>>> a
10
>>> a=20
>>> a
20
>>> a=a+5
>>> a
25
>>> pass=100
SyntaxError: invalid syntax
>>> int a=10
SyntaxError: invalid syntax
```

How to represent integer value/literals in python?
In python these integer values are represented in 4 formats
1. Decimal integer
2. Octal integer
3. Hexadecimal integer
4. Binary integer

Decimal, Octal, Hexadecimal and binary are called number system.
Number system defines set of rules and regulations for defining numbers in computer science.

Decimal Integer

An integer value with base 10 is called decimal integer. This integer is created using digits which range from 0-9. This integer is not prefix with 0. This integer is prefix with + or –

```
>>> b=123
>>> b
123
>>> c=056
SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers
```

Octal Integer
An integer value with base 8 is called octal integer.
This integer is created using digits which range from 0-7.
This integer is prefix with 0o or 0O.

```
>>> x=0o12
>>> x
10
>>> type(x)
<class 'int'>
>>> y=0o89
SyntaxError: invalid digit '8' in octal literal
```

Applications of octal integers
1. Representing an integer value which does not allows 8 and 9 digits.
2. In Assembly Language data is represented in octal format.

Decimal to Octal	Octal to Decimal

Decimal to Octal

$(15)_{10}$ ———— $(0o17)_8$

```
8 | 15
8 |  1 | 7
  |    | 1
```

$(10)_{10}$ ———— $(0o12)_8$

```
8 | 10
8 |  1 | 2
  |    | 1
```

Octal to Decimal

$(0o\ 17)_8$ ———— $(15)_{10}$

$8^0 x7 + 8^1 x1$

$7+8=15$

$(0o\ 12)_8$ ———— $(10)_{10}$

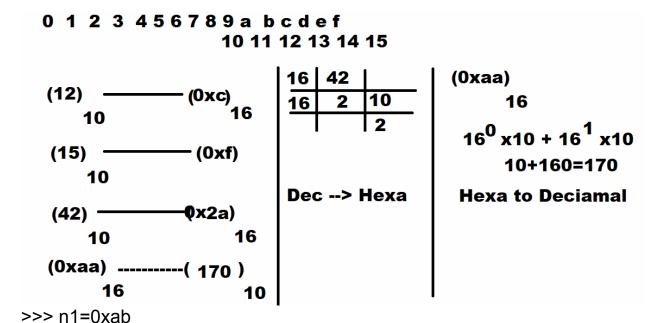$8^0 x2 + 8^1 x1$

$2+8=10$

Hexadecimal Integer
An integer value with base 16 is called hexadecimal integer.
This integer is prefix with 0x or 0X.
This integer is created using digits range from 0-9 and a-f/A-F
Larger integer values are represented in hexadecimal format.

Applications
1. Color values
2. Memory Address
3. Register Address

0 1 2 3 4 5 6 7 8 9 a b c d e f
                        10 11 12 13 14 15

$(12)_{10}$ ———— $(0xc)_{16}$

$(15)_{10}$ ———— $(0xf)$

$(42)_{10}$ ———— $(0x2a)_{16}$

$(0xaa)_{16}$ ----------$(170)_{10}$

```
16 | 42
16 |  2 | 10
   |    |  2
```

Dec --> Hexa

(0xaa)₁₆ →

$(0xaa)_{16}$

$16^0 x10 + 16^1 x10$

$10+160=170$

Hexa to Deciamal

>>> n1=0xab

```
>>> n1
171
>>> n2=0XABC
>>> n2
2748
>>> n3=0xbad
>>> n3
2989
>>> n4=0xabg
```

Binary integer

An integer value with base 2 is called binary integer
This integer is prefix with 0b or 0B
This integer is created using 0 and 1

Applications

1. Machine Language
2. Low Level Programming/Embedded Applications

```
>>> a=0b1010
>>> a
10
>>> b=0b101
>>> b
5
>>> c=0b102
SyntaxError: invalid digit '2' in binary literal
```

Decimal to Binary	Binary to Decimal

$(12)_{10} \longrightarrow 0b1\,1\,0\,0)_2$

$(0b1100) \longrightarrow (\ 12\ )_{10}$

2	12	
2	6	0
2	3	0
2	1	1
		1

$2^0 x0 + 2^1 x0 + 2^2 x1 + 2^3 x1$

$0+0+4+8$

```
>>> n1=9999999999999999999999999999999999999999
```

```
>>> n1
99999999999999999999999999999999999999
>>>
n2=99999999999999999999999999999999999999999999999999
>>> n2
99999999999999999999999999999999999999999999999999
```

The size of int data type is dynamic, the size is base on value.

codewithsatishgupta