input()
input() is a predefined function in python.
Input() function read a value from keyboard(stdin)
After reading value from keyboard, it returns that value as string type.
Syntax: input([prompt])

Prompt is message/string which is displayed on console before reading value. Defining prompt is optional.

Example:
```
a=input("Enter the value of a:")
b=input("Enter the value of b:")
uname=input("UserName:")
password=input("Password:")
```
Output:
```
Enter the value of a:10
Enter the value of b:20
UserName:nit
Password:naresh
>>>
```

Example:
```
n1=input("Enter first number")
n2=input("Enter second number")
n3=n1+n2
print(n1,n2,n3)
print(type(n1),type(n2),type(n3))
```

Output:
```
Enter first number10
Enter second number20
10 20 1020
<class 'str'> <class 'str'> <class 'str'>
>>>
```

Type conversion or type casting
type conversion is process of converting an object of one type to another type.
This type conversion is used using type conversion functions.

Int()
Int() is predefined function in python.
This function is used to convert int->int,float->int,string->int,bool->int

Syntax-1: int(value)
Syntax-2: int(value,base=10)

Syntax-1 is used to convert int→int, float→int, bool->int

```
>>> x=int(65)
>>> print(x,type(x))
65 <class 'int'>
>>> y=int(1.456)
>>> print(y,type(y))
1 <class 'int'>
>>> z=int(True)
>>> print(z,type(z))
1 <class 'int'>
>>> p=int(False)
>>> print(p,type(p))
0 <class 'int'>
>>>
```

Syntax-2 is used to convert string->int

```
>>> a=int("45")
>>> print(a,type(a))
45 <class 'int'>
>>> b=int("0b1010")
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    b=int("0b1010")
ValueError: invalid literal for int() with base 10: '0b1010'
>>> b=int("0b1010",base=2)
>>> c=int("0xab",base=16)
>>> d=int("0o34",base=8)
>>> e=int("65")
>>> print(a,b,c,d,e)
45 10 171 28 65
```

```
>>>
>>> f=int("abc")
Traceback (most recent call last):
  File "<pyshell#17>", line 1, in <module>
    f=int("abc")
ValueError: invalid literal for int() with base 10: 'abc'
>>> p=int("1.5")
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    p=int("1.5")
ValueError: invalid literal for int() with base 10: '1.5'
>>>
```

Example:
 # write a program to add two integers

```
n1=int(input("Enter first number"))
n2=int(input("Enter second number"))
n3=n1+n2
print(n1,n2,n3)
```

Output:

```
Enter first number45
Enter second number65
45 65 110
>>>
=== RESTART: C:/Users/user/Desktop/python6pm/py6.py ===
Enter first number0xa
Traceback (most recent call last):
  File "C:/Users/user/Desktop/python6pm/py6.py", line 3, in <module>
    n1=int(input("Enter first number"))
ValueError: invalid literal for int() with base 10: '0xa'
>>>
```

Example:
write a program to add two hexadecimal integers

```
n1=int(input("Enter first number"),base=16)
n2=int(input("Enter second number"),base=16)
n3=n1+n2
print(hex(n1),hex(n2),hex(n3))
```

```
print(n1,n2,n3)
```

Output:

```
Enter first number65
Enter second number76
0x65 0x76 0xdb
101 118 219
>>>
=== RESTART: C:/Users/user/Desktop/python6pm/py7.py ===
Enter first number0x65
Enter second number0x76
0x65 0x76 0xdb
101 118 219
>>>
```

Note: giving prefix to the value given inside string is optional

float()

It is a predefined function in python.
This function is used to convert int-float,float-float,string-float

Syntax: float(value)

Value can be int,float,string (float→fixed,exponent,int),bool

```
>>> f1=float()
>>> print(f1)
0.0
>>> f2=float(45)
>>> print(f2,type(f2))
45.0 <class 'float'>
>>> f3=float(1.5)
>>> print(f3,type(f3))
1.5 <class 'float'>
>>> f4=float("1.5")
>>> print(f4,type(f4))
1.5 <class 'float'>
>>> f5=float("15e-1")
>>> print(f5,type(f5))
1.5 <class 'float'>
```

```
>>> f6=float("15")
>>> print(f6,type(f6))
15.0 <class 'float'>
>>> f7=float(True)
>>> print(f7,type(f7))
1.0 <class 'float'>
>>>
```