Type conversion functions or type casting
Converting one type of value to another type is called type casting or type conversion.

1. int()
2. float()
3. complex()
4. bool()
5. str()

int() function
This function is used to perform the following conversions

1. int to int
2. str to int
3. float to int
4. bool to int

Syntax-1: int(value) This syntax is used to convert int to int, float to int,bool to int

Example:

```
>>> x=50
>>> y=int(x)
>>> print(x,y,sep="\n")
50
50
>>> p=1.45
>>> q=int(p)
>>> print(p,q,sep="\n")
1.45
1
>>> b1=True
>>> r=int(b1)
>>> print(b1,r)
True 1
>>> b2=False
>>> r1=int(b2)
>>> print(b2,r1)
False 0
```

Syntax-2: int(value,base=10) This syntax is used to convert string representation of integer to integer

```
>>> s1="65"
>>> i1=int(s1)
>>> print(s1,i1,sep="\n")
65
65
>>> print(type(s1),type(i1))
<class 'str'> <class 'int'>
>>> s2="abc"
>>> i2=int(s2)
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    i2=int(s2)
ValueError: invalid literal for int() with base 10: 'abc'
>>> s3="1.5"
>>> i3=int(s3)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    i3=int(s3)
ValueError: invalid literal for int() with base 10: '1.5'
>>> s4="0xabc"
>>> i4=int(s4)
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    i4=int(s4)
ValueError: invalid literal for int() with base 10: '0xabc'
>>> i4=int(s4,base=16)
print(s4,i4)
0xabc 2748
print(s4,hex(i4))
0xabc 0xabc
>>> s5="0b1010"
>>> i5=int(s5,base=2)
>>> print(s5,i5)
0b1010 10
>>> print(s5,bin(i5))
0b1010 0b1010
>>> s6="0o45"
```

```
>>> i6=int(s6,base=8)
>>> print(s6,i6)
0o45 37
>>> print(s6,oct(i6))
0o45 0o45
>>> s7="abc"
>>> i7=int(s7,base=16)
>>> print(s7,i7)
abc 2748
>>> print(s7,hex(i7))
abc 0xabc
>>> s8="101"
>>> i8=int(s8)
>>> print(i8)
101
>>> i9=int(s8,base=2)
>>> print(i9)
5
```

Example:

```
# write a program to add two integers
# input two integers from keyboard

n1=int(input("Enter the value of n1"))
n2=int(input("Enter the value of n2"))
n3=n1+n2
print(n1,n2,n3)
```

Output:

```
Enter the value of n110
Enter the value of n220
10 20 30
```

Example:

```
# write a program to input two hexadecimal integers and add

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 numbera
enter second numberb
0xa 0xb 0x15
10 11 21

Example:
write a program to input two binary integers and add

```
n1=int(input("Enter first binary integer"),base=2)
n2=int(input("Enter second binarr integer"),base=2)
n3=n1+n2
print(bin(n1),bin(n2),bin(n3))
print(n1,n2,n3)
```

Output:
Enter first binary integer101
Enter second binarr integer1010
0b101 0b1010 0b1111
5 10 15

Note: base conversion function return string value

```
>>> n1=45
>>> type(n1)
<class 'int'>
>>> n2=bin(n1)
>>> type(n2)
<class 'str'>
>>> print(n2)
0b101101
```

float function
This function is used to perform the following conversions

1. float to float
2. int to float
3. string to float

Syntax: float(value) → this function convert input value into float value and return

```
>>> i1=15
>>> f1=float(i1)
>>> print(i1,f1)
15 15.0
>>> f2=1.5
>>> f3=float(f2)
>>> print(f1,f2)
15.0 1.5
>>> print(f2,f3)
1.5 1.5
>>> s1="1.5"
>>> f4=float(s1)
>>> print(s1,f4,sep="\n")
1.5
1.5
>>> print(type(s1),type(f4))
<class 'str'> <class 'float'>
>>> s2="15e-1"
>>> f5=float(s2)
>>> print(s2,f5)
15e-1 1.5
>>> print(type(s2),type(f5))
<class 'str'> <class 'float'>
>>> f6=float("65")
>>> print(f6)
65.0
```

Example:

```
# write a program to find area of triangle
# input : base,height
# area=0.5*base*height
# output: area

base=float(input("Enter Base"))
height=float(input("Enter Height"))
area=0.5*base*height
print("Area of triangle is ",area)
```

Output:

```
Enter Base1.2
```

Enter Height1.3
Area of triangle is 0.78

1. Write a program to input rs and convert into dollar
2. Write a program to input dollar and convert to rs
3. Write a program to find area of rectangle
4. Write a program to find area of circle