input( )

Used to take an input from the user and it will return in the form of <class 'str'>
input(prompt=None) -> str

#App-1

```
#input(prompt=None) -> str
'''
print("Enter u r name : ")
n=input( )   #shashi
print("U r Name is Mr|Mrs. ",n)  '''
```

#App-2

```
c=input("Enter u r city : " )
print("U r city is : ",c)
```

#Ex

```
#input(prompt=None) -> str
a=input("Enter u r age : ")
print("U r age : ",a)
print("Type of variable a is : ",type(a))
```

Eg:

```
#Sum of Two numbers
x=input("Enter a number : ")
y=input("Enter b number : ")
s=x+y
print("Sum of two is : ",s)

# str + str -> str [concatenation ] joining
#  Input: "sai"+"baba" -> "saibaba"
# "10"+"20" -> "1020"
```

Typecasting:

- It is the process of converting the value from one type to another
- In order to convert the value from one to another type in python for every data there is equality function provided
  - int -> int( ) | float -> float( ) | str -> str( )
  - list -> list( ) | tuple -> tuple( ) | dict -> dict( ) bool -> bool( ) …

- int typecasting

#int type casting -> int( )

float -> int valid

```
x=12.1212
y=int(x)
print(type(x),type(y),sep='---')
print("x : ",x,"y: ",y)
```

complex -> int

```
print("complex - > int ")
x=(10+20j)
'''
y=int(x)
Traceback (most recent call last):
  File "E:/Python_Online9/TYPECASTING/IntTypeCastingEx.py", line 12, in
<module>
    y=int(x)
TypeError: can't convert complex to int  '''
```

str -> int

```
x="10"
y=int(x)
print(type(x),type(y),sep='---')
print("x : ",x,"y: ",y)

'''
y=int("10.20")
```

```
Traceback (most recent call last):
  File "E:/Python_Online9/TYPECASTING/IntTypeCastingEx.py", line 21, in
<module>
    y=int(x)
ValueError: invalid literal for int() with base 10: '10.12' '"
```

bool -> int

```
x=True      #True -> 1 False ->0
y=int(x)
print(type(x),type(y),sep='---')
print("val x : ",x,"val y: ",y)
```

Float Type Casting :

```
# float typecasting -> float( )
```

#int -> float

```
x=12
y=float(x)
print(type(x),type(y),sep='   --->   ')
print("val of x : ",x,"val of y : ",y)
```

#complex -> float

```
x=(10+20j)
'''
y=float(x)
Traceback (most recent call last):
  File "E:/Python_Online9/TYPECASTING/FloatTypeCasting.py", line 11, in
<module>
    y=float(x)
TypeError: can't convert complex to float  '"
```

#str -> float

```
x="10.2334"  # x="10"  valid
y=float(x)
```

```
print(type(x),type(y),sep='   --->   ')
print("val of x : ",x,"val of y : ",y)

#bool -> float
x=False                #True -> 1(int) -> 1.0 [float]
y=float(x)           #False -> 0(int) -> 0.0 [float]
print(type(x),type(y),sep='   --->   ')
print("val of x : ",x,"val of y : ",y)
```

String Typecasting using str( ) :

```
#string typecasting using str( )
```

#int-> str

```
x=10
y=str(x)
print(type(x),type(y),sep='  ---> ')
print(x,y,sep=' ---> ')
```

#float -> str

```
x=12.12
y=str(x)
print(type(x),type(y),sep='  ---> ')
print(x,y,sep=' ---> ')
```

#complex -> str

```
x=(10+20j)
y=str(x)
print(type(x),type(y),sep='  ---> ')
print(x,y,sep=' ---> ')
```

#bool ->str

```
x=True
y=str(x)
print(type(x),type(y),sep='  ---> ')
print(x,y,sep=' ---> ')
```

Bool typecasting Using bool( ):

#bool typecating using bool( )

#int -> bool

```
x=-200   #only 0 is False
y=bool(x)
print(type(x),type(y),sep='  --> ')
print("val of x : ",x,"val of y : ",y)
```

#float -> bool

```
x=0.1    #0.0 only false
y=bool(x)
print(type(x),type(y),sep='  --> ')
print("val of x : ",x,"val of y : ",y)
```

#complex-> bool

```
x=(0+0j)
y=bool(x)       # complex -> bool --> False real and img is 0
print(type(x),type(y),sep='  --> ')
print("val of x : ",x,"val of y : ",y)
```

#str-> bool

```
x="123"  #True
x="0" #True
x="srinivas"
x=" "
x=""
y=bool(x)
print(type(x),type(y),sep='  --> ')
print("val of x : ",x,"val of y : ",y)
```

Ex :
```
x=""  # ""-> bool [False]
x=[]   # [ ]-> bool [False]
x=()   # () -> bool [False]
x=set() # set( ) -> bool [False]
x={}  #{} -> bool [False]
x=None # None -> bool [False]
y=bool(x)
print("Result is : ",y)   #Filter()
```

Complex typecasting

#Complex typecasting using complex(x)

```
# here x rep -- > real
```

#int -> complex

```
x=10
y=complex(x)  # (10+0j)
print("Result is : ",y)
```

#float -> complex

```
x=12.12
y=complex(x)  #(12.12+0j)
print("Result is : ",y)
```

#str -> complex

```
x="10"
y=complex(x)
print("Result is : ",y)
```

#bool -> complex

```
x=False
y=complex(x)  # (0+0j) --> 0j
print("Result is : ",y)
```

Complex(x)

Complex(x,y) x-real, y-imag

Note: If first input string, it won't allow the second argument

```
#int,int -> valid
#float,float -> valid
#int,float -> valid
#float,int -> valie

#complex(x,y)
x=10
y=20
z=complex(x,y)
print("Result is : ",z)
```

#x-int,y-float

```
x=10
y=12.12
z=complex(x,y)
print("Result is : ",z)
```

#str,int

```
x="10"
y=12
'''
z=complex(x,y)
print("Result ",z)
Traceback (most recent call last):
  File "E:/Python_Online9/TYPECASTING/ComplexTypeCastingDemo2.py", line 21,
in <module>
    z=complex(x,y)
TypeError: complex() can't take second arg if first is a string"'

'''
z=complex(y,x)
TypeError: complex() second arg can't be a string"'
```