**Type Conversion in Python**

Python defines type conversion functions to directly convert one data type to another which is useful in day-to-day and competitive programming. This article is aimed at providing information about certain conversion functions.

There are two types of Type Conversion in Python:

1. Implicit Type Conversion
2. Explicit Type Conversion

## **Implicit Type Conversion**

In Implicit type conversion of data types in Python, the Python interpreter automatically converts one data type to another without any user involvement. To get a more clear view of the topic see the below examples.

**Example:**

* Python3

x = 10

  print("x is of type:",type(x))

  y = 10.6

print("y is of type:",type(y))

  z = x + y

print(z)

print("z is of type:",type(z))

**Output:**

x is of type: <class 'int'>

y is of type: <class 'float'>

20.6

z is of type: <class 'float'>

## **Explicit Type Conversion**

In Explicit Type Conversion in Python, the data type is manually changed by the user as per their requirement. With explicit type conversion, there is a risk of data loss since we are forcing an expression to be changed in some specific data type.  Various forms of explicit type conversion are explained below:

**1. int(a,** **base)**: This function converts**any data type to integer**. ‘Base’ specifies the**base in which string is** if the data type is a string.  
**2. float()**: This function is used to convert **any data type to a**floating-point**number.**

|  |
| --- |
| # Python code to demonstrate Type conversion  # using int(), float()    # initializing string  s = "10010"    # printing string converting to int base 2  c = int(s,2)  print ("After converting to integer base 2 : ", end="")  print (c)    # printing string converting to float  e = float(s)  print ("After converting to float : ", end="")  print (e) |

**Output:**

After converting to integer base 2 : 18

After converting to float : 10010.0

**3. ord() :**This function is used to convert a **character to integer.**  
**4. hex() :**This function is to convert **integer to hexadecimal string**.  
**5. oct() :**This function is to convert **integer to octal string**.

* Python3

# Python code to demonstrate Type conversion

# using  ord(), hex(), oct()

# initializing integer

s = '4'

# printing character converting to integer

c = ord(s)

print ("After converting character to integer : ",end="")

print (c)

# printing integer converting to hexadecimal string

c = hex(56)

print ("After converting 56 to hexadecimal string : ",end="")

print (c)

# printing integer converting to octal string

c = oct(56)

print ("After converting 56 to octal string : ",end="")

print (c)

**Output:**

After converting character to integer : 52

After converting 56 to hexadecimal string : 0x38

After converting 56 to octal string : 0o70

**6. tuple() :**This function is used to **convert to a tuple**.  
**7. set() :**This function returns the **type after converting to set**.  
**8. list() :**This function is used to convert **any data type to a list type**.

* Python3

|  |
| --- |
| # Python code to demonstrate Type conversion  # using  tuple(), set(), list()    # initializing string  s = 'srtut.com'    # printing string converting to tuple  c = tuple(s)  print ("After converting string to tuple : ",end="")  print (c)    # printing string converting to set  c = set(s)  print ("After converting string to set : ",end="")  print (c)    # printing string converting to list  c = list(s)  print ("After converting string to list : ",end="")  print (c) |

**Output:**

After converting string to tuple : ('s', 'r', 't', 'u', 't')

After converting string to set : {'s', 'r', 't', 'u'}

After converting string to list : ['s', 'r', 't', 'u', 't']

**9. dict() :**This function is used to **convert a tuple of order (key,value) into a dictionary**.  
**10. str() :**Used to **convert integer into a string.**  
**11. complex(real,imag) :**This function**converts real numbers to complex(real,imag) number.**

* Python3

|  |
| --- |
| # Python code to demonstrate Type conversion  # using  dict(), complex(), str()    # initializing integers  a = 1  b = 2    # initializing tuple  tup = (('a', 1) ,('f', 2), ('g', 3))    # printing integer converting to complex number  c = complex(1,2)  print ("After converting integer to complex number : ",end="")  print (c)    # printing integer converting to string  c = str(a)  print ("After converting integer to string : ",end="")  print (c)    # printing tuple converting to expression dictionary  c = dict(tup)  print ("After converting tuple to dictionary : ",end="")  print (c) |

**Output:**

After converting integer to complex number : (1+2j)

After converting integer to string : 1

After converting tuple to dictionary : {'a': 1, 'f': 2, 'g': 3}

**12. chr(number):**This function**converts number to its corresponding ASCII character.**

* Python3

|  |
| --- |
| # Convert ASCII value to characters  a = chr(76)  b = chr(77)  print(a)  print(b) |

**Output:**

L

M