

Type Conversion in Python

<https://www.geeksforgeeks.org/type-conversion-in-python/>

В Python есть два типа преобразования типов:

1. Неявное преобразование типов
2. Явное преобразование типов

- **Неявное преобразование типов**

При неявном преобразовании типов данных в Python интерпретатор Python автоматически преобразует один тип данных в другой без участия пользователя.

```
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'>
```

Как мы видим, тип данных «z» автоматически изменился на тип «float», в то время как одна переменная x имеет целочисленный тип, а другая переменная y — тип float.

Причина того, что значение с плавающей запятой не преобразуется в целое число, связана с продвижением типа, которое позволяет выполнять операции путем преобразования данных в тип данных большего размера без потери информации. Это простой случай неявного преобразования типов в python.

- **Явное преобразование типов**

При явном преобразовании типов в Python тип данных изменяется пользователем вручную в соответствии с его требованиями. При явном преобразовании типов существует риск потери данных, поскольку мы заставляем выражение измениться в каком-то конкретном типе данных. Различные формы явного преобразования типов объясняются ниже:

1. int(a, base) : эта функция преобразует **любой тип данных в целое число** . 'Base' указывает базу, в которой находится строка, если тип данных является строкой.

2. float() : эта функция используется для преобразования **любого типа данных в число с плавающей запятой**.

```
# 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(): эта функция используется для преобразования **символа в целое число**.

4. hex(): эта функция предназначена для преобразования **целого числа в шестнадцатеричную строку** .

5. oct(): эта функция предназначена для преобразования **целого числа в восьмеричную строку** .

```
# 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**.

```
# Python code to demonstrate Type conversion
# using tuple(), set(), list()

# initializing string
s = 'geeks'

# 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 : ('g', 'e', 'e', 'k', 's')

After converting string to set : {'k', 'e', 's', 'g'}

After converting string to list : ['g', 'e', 'e', 'k', 's']

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): Эта функция преобразует действительные числа в комплексные(real,imag) числа.

```
# 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(число): эта функция преобразует число в соответствующий символ ASCII.

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

Output:

```
L
M
```