

## **EXPERIMENT NO: 02**

**AIM:** To test the functionality of various data types and type conversion.

Theory:

### 1] Data types:

Every value in python has a data type. Since everything is an object in python programming, data types are actually classes and variables are instance(object) of these classes. In programming, data types is an important concept. Python has the following data types built-in by

default, in these categories:-

Text type: str.

Numeric types: int, float, complex.

Sequence types: list, tuple, range.

Mapping type: dict.

Set types: set, frozenset

Boolean type: bool.

Binary types: bytes, bytearray, memoryview

We can get the data type of any object by using the type() function.

### 2] Type conversion: -

There are two types of type conversion in python.

- Implicit type conversion.
- Explicit type conversion.

→ Implicit type conversion:

In Implicit type conversion of data types in python, the python interpreters automatically converts one data type to another without any other involvement.

→ Explicit type conversion:

In Explicit type conversion in python, the data type is manually changed by the used as per their requirement.

1) int(a, base): This function converts any data type to integer. Base specifies the base in which string is if the data types is a string.

2) float(): This function is used to convert any datatype to a float-point number.

3) ord(): This function is used to convert any character to integer.

4) hex(): This function is used to convert integer to hexadecimal string.

- 5) oct(): This function is used to convert integer to octal string.
- 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.
- 9) dict (): This function is used to convert a tuple of order ( key, values) into a dictionary.
- 10) str(): Used to convert an integer to a string.
- 11) complex(real, img): This function converts real numbers to complex (real, img) numbers.
- 12) chr (number): This function converts numbers to its corresponding ASCII characters.

Q1) Write a program to define data types in Python

```
x= "Hello World"
```

```
y=20
```

```
z=30.5
```

```
b=True
```

```
print(type (x))
```

```
print(type (y))
```

```
print(type (z))
```

```
print(type (b))
```

```
# Dict
```

```
c= { "name" : "John", " age" : 36}
```

```
print(type (c))
```

```
# Set
```

```
d={"apple", "banana", "cherry" }
```

```
print(type (d))
```

**OUTPUT 1:**

```
<class 'str'>
```

```
<class 'int'>
<class 'float'>
<class 'dict'>
<class 'bool'>
```

## Q2) Write a program to implement implicit type conversion .

```
# Converting integer to float.
num_int=123
num_flo= 1.23
num_new = num_int + num_flo
print("Value of num_new:", num_new)
print("Datatype of num_new: ", type(num_new))
```

### OUTPUT 2:

```
Value of num_new: 124.23
Datatype of num_new: <class 'float'>
```

## Q3) Write a program to implement explicit type conversion-

```
1 num_int = 123
2 num_str = "456"
3 print("Data type of num_int:",type(num_int))
4 print("Data type of num_str before Type Casting:",type(num_str))
5 num_str = int(num_str)
6 print("Data type of num_str after Type Casting:",type(num_str))
7 num_sum = num_int + num_str
8 print("Sum of num_int and num_str:",num_sum)
9 print("Data type of the sum:",type(num_sum))
```

### OUTPUT:

```
D:\SBMP\SUBJECTS\Python\Progs>python Explicit.py
Data type of num_int: <class 'int'>
Data type of num_str before Type Casting: <class 'str'>
Data type of num_str after Type Casting: <class 'int'>
Sum of num_int and num_str: 579
Data type of the sum: <class 'int'>
```

.....

### **Convert two given lists into a Dictionary.**

```
keys = ['a', 'b', 'c', 'd']
values = [1, 2, 3, 4]
# Convert lists to dictionary
result_dict = dict(zip(keys, values))
print("Resulting Dictionary:", result_dict)
```

---

### **List to Set – Remove Duplicates**

**Real Time:** Removing duplicate entries

```
users = ["A", "B", "A", "C"]
unique_users = set(users)
print(unique_users)
```

### **Bank Transaction System**

**Real Time:** Balance stored as string

```
balance = "10000"
withdraw = 2500

new_balance = int(balance) - withdraw
print ("Remaining Balance:", new_balance)
```

**CONCLUSION:** Hereby, we performed the functionality, of various data types and type conversions.

---