PYTHON

UNIT -1 : Introduction to python

Lec-2

Today's Target

- Type conversion
- Data Type
- AKTU PYQs

By Pragya rajvanshi

B.tech,M.tech

Gateway classes

Type conversion

The act of changing an **object's data type** is known as type conversion.

Implicit conversion(Automatic conversion)

- Python interpreter **automatically** converts **one data type to another** without any **user involvement.**
- There is **no loss of data.**

```
1  x=10
2  y=20.5
3  z=x+y
4  print(z)
```

```
30.5
```

Explicit conversion(type casting)

- The data type is **manually changed by** the user as per their requirement
- There is a **risk of data loss** since we are forcing an expression to be changed in some specific data type.

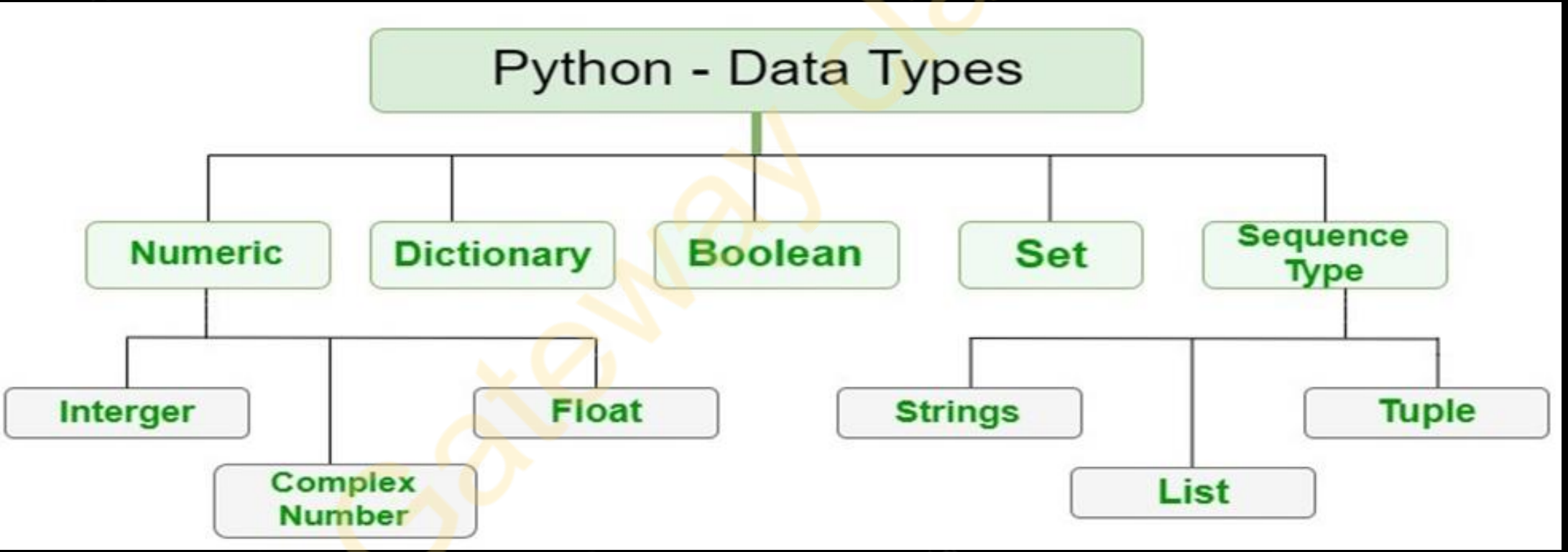
```
1  s="10010"
2  z=int(s,2)
3  print(z)
4  w=float(z)
5  print(w)
```

```
18
18.0
```

Gateway classes

DATA TYPE

- Python Data types are the classification or categorization of data items.
- A data type defines the kind of values a variable can hold and determines what operations can be performed on that data.



GATEWAY CLASSES
Gateway Classes

NUMERIC

- Python, numeric data type is used to hold **numeric values.**
- **Int -** holds signed integers of non-limited length.
- **float -** holds floating decimal points and it's accurate up to 15 decimal places.
- **complex -** holds complex numbers.

```
1  a = 5
2  print("Type of a: ", type(a))
3  b = 5.0
4  print("Type of b: ", type(b))
5  c = 2 + 4j
6  print("Type of c: ", type(c))
```

```
Type of a:  <class 'int'>
Type of b:  <class 'float'>
Type of c:  <class 'complex'>
```

GATEWAY CLASSES
Gateway Classes

SEQUENCING TYPE

String

- **The sequence of characters** in the quotation marks can be used to describe the **string.**

```
1 str1="hello"
2 str2='hi'
3 print(str1,"is of type",type(str1))
4 print(str2,"is of type",type(str2))
```

```
hello is of type <class 'str'>
hi is of type <class 'str'>
```

```
1 str1="hellostudents"
2 str2="python"
3 print(str1[0:2])
4 print(str2[2:4])
5 print(str1[3])
6 print(str2[3])
7 print(str1*2)
8 print(str2*3)
9 print(str1+str2)
```

```
he
th
l
h
hellostudentshellostudents
pythonpythonpython
hellostudentspython
```

GATEWAY CLASSES
Gateway classes

LIST

- List is an **ordered collection** of **similar or different types of items** separated by **commas** and enclosed within brackets [ ].

```
[1, 3, 4, 5]
<class 'list'>
3
5
[1, 3, 4]
[1, 3, 4, 5]
[1, 3, 4, 5, 2, 3, 4, 'hello']
[1, 3, 4, 5, 1, 3, 4, 5]
```

```
l1=[1,3,4,5]
l2=[2,3,4,"hello"]
print(l1)
print(type(l1))
#indexing
print(l1[1])
print(l1[-1])
#slicing
print(l1[0:3])
print(l1[0:])
#concatenation
print(l1+l2)
#repetition
print(l1*2)
```

Gateway Classes

Tuple

- Tuple is an **ordered sequence** of items same as a list. The only difference is that tuples are **immutable**. Tuples once **created cannot be modified.**
- we use the **parentheses ( )** to store items of a tuple.

```
tup1=(1,2,3,4)
print(type(tup1))
tup2=("hello",2.3,4,3,4)
print(tup1[1])
print(tup2[-1])
print(tup1[1:3])
print(tup1+tup2)
print(tup1*2)
tup1[1]=6
```

```
<class 'tuple'>
2
4
(2, 3)
(1, 2, 3, 4, 'hello', 2.3, 4, 3, 4)
(1, 2, 3, 4, 1, 2, 3, 4)
Traceback (most recent call last):
  File "/home/main.py", line 9, in <module>
    tup1[1]=6
TypeError: 'tuple' object does not support item assignment
```

Gateway Classes

Dictionary data type

- Dictionary is an **ordered collection of items**. It stores elements in **key/value pairs.**
- Here, **keys are unique identifiers** that are associated with each value .

```
1  S1={"sectionA":"good","sectionB":"excellent","sectionC":"good"}
2  print(S1)
3  print(type(S1))
4  print(S1.keys())
5  print(S1.values())
6  print(S1["sectionA"])
```

```
{'sectionA': 'good', 'sectionB': 'excellent', 'sectionC': 'good'}
<class 'dict'>
dict_keys(['sectionA', 'sectionB', 'sectionC'])
dict_values(['good', 'excellent', 'good'])
good
```

Gateway classes

Boolean

- The Boolean value can be of two types only i.e. either True or False.

  The output indicates the variable is a Boolean data.

```
a=True
print(a)
print(type(a))
b=False
print(b)
print(type(b))
```

```
True
<class 'bool'>
False
<class 'bool'>
```

Sets

- Set items are unordered, unchangeable, and do not allow duplicate values.

GATEWAY CLASSES
Gateway classes

GATEWAY CLASSES

```
s={"banana","grapes","papita","mango","mango"}
print(s)
print(type(s))
s2={"1","hello",2.5}
print(s2)
s[1]="bye"
print(s)
```

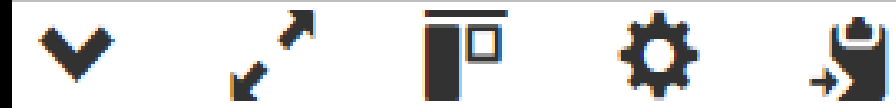
input

```
{'mango', 'grapes', 'banana', 'papita'}
<class 'set'>
{2.5, 'hello', '1'}
Traceback (most recent call last):
  File "/home/main.py", line 6, in <module>
    s[1]="bye"
TypeError: 'set' object does not support item assignment
```

Gateway classes

GATEWAY CLASSES

```
1  s1={"hi","hello","jjj","hi" }
2  print(s1)
```



```
{'hello', 'hi', 'jjj'}
```

Gateway classes



Gateway Classes