Type Casting notes2

07 June 2025 17:04

5.Conversion of string to other data types:

Examples:-

```
s='sunday fun day'
int(s)
Output: Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    int(s)
ValueError: invalid literal for int() with base 10: 'sunday fun day'
float(s)
Output: Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    float(s)
ValueError: could not convert string to float: 'sunday fun day'
complex(s)
Output : Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    complex(s)
ValueError: complex() arg is a malformed string
bool(s)
True
list(s)
['s', 'u', 'n', 'd', 'a', 'y', ' ', 'f', 'u', 'n', ' ', 'd', 'a', 'y']
tuple(s)
('s', 'u', 'n', 'd', 'a', 'y', ' ', 'f', 'u', 'n', ' ', 'd', 'a', 'y')
set(s)
{'a', ' ', 'd', 'n', 's', 'y', 'u', 'f'}
dict(s)
Output: Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    dict(s)
ValueError: dictionary update sequence element #0 has length 1; 2 is required
```

Source Type	Destination Type
String	Int(only if the string having data in the form of integer it will convert) Float(only if the string having data in the form of float it will convert) Complex(only if the string having data in the form of complex it will convert) Boolean List Tuple Set(data loss, it eliminates duplicates) Dictionary--->Error

Note: only if the string having data in the form of integer then it is possible to convert string to integer,float,complex

Example:

```
e='67'
int(e)
67
float(e)
67.0
complex(e)
(67+0j)
bool(e)
True
list(e)
['6', '7']
tuple(e)
('6', '7')
set(e)
{'7', '6'}
dict(e)
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    dict(e)
ValueError: dictionary update sequence element #0 has length 1; 2 is required
```

Example:

```
d='77.8'
int(d)
```

```
Output:-Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    int(d)
ValueError: invalid literal for int() with base 10: '77.8'
float(d)
77.8
complex(d)
(77.8+0j)
bool(d)
True
list(d)
['7', '7', '.', '8']
tuple(d)
('7', '7', '.', '8')
set(d)
{'7', '8', '.'}
```

Example:-

```
f='20+6j'
f
Output:'20+6j'
int(f)
Output:Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    int(f)
ValueError: invalid literal for int() with base 10: '20+6j'
float(f)
Output:Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    float(f)
ValueError: could not convert string to float: '20+6j'
complex(f)
Output:(20+6j)
bool(f)
Output: True
list(f)
Output:['2', '0', '+', '6', 'j']
tuple(f)
Output:('2', '0', '+', '6', 'j')
set(f)
Output:{'+', '6', '0', 'j', '2'}
dict(f)
Output:Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    dict(f)
ValueError: dictionary update sequence element #0 has length 1; 2 is required
```

6.Conversion of list to Other data types:

Examples:

```
l=[10,20,'hai',10+7j,True]
int(l)
Output: Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    int(l)
TypeError: int() argument must be a string, a bytes-like object or a real number, not 'list'
float(l)
Output: Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    float(l)
TypeError: float() argument must be a string or a real number, not 'list'
complex(l)
```

```
Output: Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    complex(l)
TypeError: complex() first argument must be a string or a number, not 'list'
bool(l)
Output:True
tuple(l)
Output:(10, 20, 'hai', (10+7j), True)
set(l)
{True, 10, 20, 'hai', (10+7j)}
dict(l)
Output:Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    dict(l)
TypeError: cannot convert dictionary update sequence element #0 to a sequence
```

Example: l=[(12,34),'ab',[8,9],{6,3}]
dict(l)
Output:{12: 34, 'a': 'b', 8: 9, 3: 6}

Source Type	Destination Type
List	Bool String Tuple Set(1.only if list consist of immutable values. 2. it Will eliminate duplicate values) Dictionary(1.Each Collection should Only collection values. 2.Each collection should have length as 2)

7.Conversion of tuple to other data types:

Example: t=(1,2,3,3.4,'py',[1,2])

```
int(t)
Output: Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    int(t)
Type Error: int() argument must be a string, a bytes-like object or a real number, not 'tuple'
float(t)
Output: Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    float(t)
Type Error: float() argument must be a string or a real number, not 'tuple'
complex(t)
Output: Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    complex(t)
Type Error: complex() first argument must be a string or a number, not 'tuple'
bool(t)
Output: True
str(t)
Output:"(1, 2, 3, 3.4, 'py', [1, 2])"
list(t)
```

Output:[1, 2, 3, 3.4, 'py', [1, 2]]
set(t)
Output: Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    set(t)
TypeError: unhashable type: 'list'
dict(t)
Output: Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    dict(t)
Type Error: cannot convert dictionary update sequence element #0 to a sequence

Example: k=((1,2),(6,7))
dict(k)
Output:{1:2,6:7}

Example: r=(1,2,3,4,5)
set(r)
Output: {1,2,3,4,5}

Source Type	Destination Type
Tuple	Bool String list Tuple Set(1.only if list consist of immutable values. 2. it Will eliminate duplicate values) Dictionary(1.Each Collection should Only collection values. 2.Each collection should have length as 2)

8.Conversion of set to other data types:

Example :s={1,2,3,4,5,8,23,56}
int(s)
Output: Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    int(s)
TypeError: int() argument must be a string, a bytes-like object or a real number, not 'set'
float(s)
Output: Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    float(s)
 TypeError: float() argument must be a string or a real number, not 'set'
complex(s)
Output:Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    complex(s)
TypeError: complex() first argument must be a string or a number, not 'set'
bool(s)
Output: True
str(s)
Output:'{1, 2, 3, 4, 5, 8, 23, 56}'
list(s)
Output:[1, 2, 3, 4, 5, 8, 23, 56]
tuple(s)
Output:(1, 2, 3, 4, 5, 8, 23, 56)
dict(s)
Output:Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    dict(s)
TypeError: cannot convert dictionary update sequence element #0 to a sequence

Example: o={(8,9),'bp',(5,6)}
dict(o)

Output:{(8:9),'b':'p',5:6}

Source Type	Destination Type
Set	Bool String list Tuple Dictionary(1.Each Collection should Only collection values. 2.Each collection should have length as 2)

9.conversion of dictionary to other data types:

Example:

```
m={'a':10,'b':20,'c':30,'e':40}
int(m)
Output:Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    int(m)
TypeError: int() argument must be a string, a bytes-like object or a real number, not 'dict'
float(m)
Output:Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    float(m)
TypeError: float() argument must be a string or a real number, not 'dict'
complex(m)
Output:Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    complex(m)
TypeError: complex() first argument must be a string or a number, not 'dict'
bool(m)
Output:True
str(m)
Output:"{'a': 10, 'b': 20, 'c': 30, 'e': 40}"
list(m)
Output:['a', 'b', 'c', 'e']
tuple(m)
Output:('a', 'b', 'c', 'e')
set(m)
Output:{'b', 'e', 'c', 'a'}
```

Note: list,tuple,set will return only keys .because values don't contain address.

When ever we want access values we will make use of "values()"

Syntax: variable name. values()

Example: j={'a':20.'b':30}

```
list(j.values())
[10, 20]
tuple(j.values())
(10, 20)
set(j.values())
{10, 20}
```

When ever we want access values and keys we will make use of "items()"

Syntax: variable name. Items()

Example: j={'a':20.'b':30}

```
list(j.items())
[('a', 10), ('b', 20)]
set(j.items())
{('a', 10), ('b', 20)}
tuple(j.items())
(('a', 10), ('b', 20))
```

Source Type	Destination Type
Dictionary	Bool String list Tuple → it will returns only keys set