

## Type casting

Convert one data type to another data type

- Integer
- float
- string
- boolean

### Integer to all other data types

- Integer to float
- Integer to string
- Integer to boolean

```
In [1]: number=100  
type(number)
```

```
Out[1]: int
```

```
In [3]: number_float=float(number) # float converstion
```

```
In [4]: number,number_float
```

```
Out[4]: (100, 100.0)
```

```
In [5]: type(number),type(number_float)
```

```
Out[5]: (int, float)
```

*int-str*

```
In [7]: number_string=str(number)
```

```
In [8]: number,number_float,number_string
```

```
Out[8]: (100, 100.0, '100')
```

```
In [9]: type(number),type(number_float),type(number_string)
```

```
Out[9]: (int, float, str)
```

*int-bool*

```
In [12]: number_bool=bool(number)  
number_bool
```

```
Out[12]: True
```

```
In [13]: number,number_float,number_string,number_bool
```

```
Out[13]: (100, 100.0, '100', True)
```

```
In [14]: number1=200 # int
number1_float=float(number1) # float
number1_string=str(number1) # string
number1_bool=bool(number1) # bool

number1,number1_float,number1_string,number1_bool
```

```
Out[14]: (200, 200.0, '200', True)
```

```
In [15]: bool(100),bool(-100),bool(0)
```

```
Out[15]: (True, True, False)
```

### Note

- Boolean conversion of 0 gives False
- Boolean conversion of +ve and -ve integer numbers gives True

```
In [19]: #I am getting results for last code only
a=10
b=20
a,b
```

```
Out[19]: (10, 20)
```

### Float to other data types

- Float to integer
- Float to string
- Float to boolean

```
In [23]: number2=100.25
int(number2),str(number2),bool(number2),bool(0.0)
```

```
Out[23]: (100, '100.25', True, False)
```

```
In [ ]: int(100.25) # 100
str(100.25) # '100.25'
bool(100.25) # True
bool(0.0) # False
```

```
In [24]: print('python')
```

python

### String to all other data types

- String to integer

- String to float
- String to boolean

```
In [26]: type('100')
```

```
Out[26]: str
```

```
In [27]: type('python')
```

```
Out[27]: str
```

```
In [28]: type('100.25')
```

```
Out[28]: str
```

### Case – 1

```
In [ ]: # string to all other
        int('100') # 100
        float('100') # 100.0
        bool('100') # True
```

```
In [29]: int('100')
```

```
Out[29]: 100
```

```
In [30]: float('100')
```

```
Out[30]: 100.0
```

```
In [31]: bool('100')
```

```
Out[31]: True
```

```
In [ ]: int('100.25') # faol
        float('100.24') # 100.24
        bool('100.25') # True
```

```
In [32]: int('100.25')
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[32], line 1
----> 1 int('100.25')

ValueError: invalid literal for int() with base 10: '100.25'
```

```
In [34]: float('100'), float('100.25')
```

```
Out[34]: (100.0, 100.25)
```

```
In [ ]: int('100') # works,
        int('100.25') # Fail
```

### Note

- float is boss
- integer conversion of float value in string format will fail

```
In [ ]: int(100.25) # 100
        int('100.25') # Fail
        int(100)      # 100
        int('100')   # 100
```

```
In [ ]: float(100.25) # 100.25
        float(100)    # 100.0
        float('100') # 100.0
        float('100.25') # 100.25
```

```
In [ ]: #integer will accept the float value    ex: int(100.25) works

        #integer will not accept the float value in strings format ex: int('100.25') fa
```

### Case – 3

```
In [ ]: int('python') # Fail
        float('python') # Fail
        bool('python') # True
```

### Boolean to all other data types

```
In [ ]: bool('100') # True
        bool('100.25') # True
        bool('0') # True
        bool('0.0') # True
        bool('python') # True
        bool(100) # T
        bool(-100) # T
        bool(100.25) # T
        bool(0) # F (Off)
        bool(0.0) # F (off)
        bool('') # F (off)
```

```
In [35]: bool('100')
```

```
Out[35]: True
```

```
In [ ]: TypeError: 'float' object is not callable
```

```
In [36]: float(100)
```

```
Out[36]: 100.0
```

```
In [ ]: int(100.25) # 100
        int('100.25') # error
        int(True) # 1
        int(False) # 0
```

```
In [ ]:
```

```
In [ ]: float(100) # 100.0
float('100.25') # 100.25
float(True) # 1.0
float(False) # 0.0
```

```
In [ ]: str(100) # '100'
str(100.25) # '100.25'
str(True) # 'True'
str(False) # 'False'
```

```
In [37]: int('100')
```

Out[37]: 100

```
In [38]: int('100.25')
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[38], line 1
----> 1 int('100.25')

ValueError: invalid literal for int() with base 10: '100.25'
```

```
In [39]: bool(-190)
```

Out[39]: True

```
In [ ]:
```