25-oct

Typecasting

- changing one datatype to another datatype

- int to float
- int to str
- int to boolean

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

Out[1]: int

In [2]:
```
# int to float

float_num = float(number)

float_num, type(float_num)
```

Out[2]: (100.0, float)

In [3]:
```
# int to string

str_num = str(number)
str_num, type(str_num)
```

Out[3]: ('100', str)

In [4]:
```
bool(number)
```

Out[4]: True

In [7]:
```
print(float(100))
print(str(100))
print(bool(100))
```

```
100.0
100
True
```

In [8]:
```
print(float(-100))
print(str(-100))
print(bool(-100))
```

```
-100.0
-100
True
```

In [9]:
```
print(float(0))
print(str(0))
```

```
print(bool(0))
```

```
0.0
0
False
```

- **Note**
- Boolean conversation of any integer value other than zero becomes True
- Boolean Convertion of zero is False

Float to other data types

In [10]:
```
print(int(100.25))  # 100
print(str(100.25))  # '100.25'
print(bool(100.25))  # True
```

```
100
100.25
True
```

In [12]:
```
print(int(-100.25))  # -100
print(str(-100.25))  # '-100.25'
print(bool(-100.25))  # True
```

```
-100
-100.25
True
```

In [13]:
```
print(int(0.0))  # 0
print(str(0.0))  # '0.0'
print(bool(0.0))  # False
```

```
0
0.0
False
```

Note

- Boolean conversation of any float value other than zero becomes **True**
- Boolean Convertion of zero is **False**

String to other datatypes

In [17]:
```
#int("Apple")  #Error
#float("apple") # Error
bool("Apple") #True
```

Out[17]: True

In [18]:
```
bool('0') # this is also string that's why its comes true
```

Out[18]: True

In [20]:
```
bool('') # meaning of empty string is off or False
```

Out[20]: False

In [21]:
```
int('10')
```

Out[21]: 10

In [22]:
```
float('10')
```

Out[22]: 10.0

In [23]:
```
bool('10')
```

Out[23]: True

Note

- Always remember
  - True means **on**
  - False means **Off** === 0 === ' ' (empty)
  - 0 for int and float == **false**
  - ' ' empty in string == **false**

In [26]:
```
int('10.5')  # fails or error
int('10')    # works
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[26], line 1
----> 1 int('10.5')
      2 int('10')

ValueError: invalid literal for int() with base 10: '10.5'
```

In [28]:
```
float('10') # Works
float('10.5') # works
```

Out[28]: 10.5

- **Always remember float is a boss**

In [1]:
```
int(10.5) # It works converting float to int is possible
int('10.2') # but fails converting string float value to int is not possible
```

Out[1]: 10

Note

- It works converting (float) to (int) is **possible**
- but fails converting (string float value) to (int) is **not possible**

In [ ]:
```

```