Python type casting or type conversion

In [1]:
```
int(2.3)
```

Out[1]: 2

In [2]:
```
int(2.3,4.5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 int(2.3,4.5)

TypeError: 'float' object cannot be interpreted as an integer
```

In [3]:
```
int(True)
```

Out[3]: 1

In [4]:
```
int(False)
```

Out[4]: 0

In [5]:
```
int(True,False)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 int(True,False)

TypeError: int() can't convert non-string with explicit base
```

In [6]:
```
int(true)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[6], line 1
----> 1 int(true)

NameError: name 'true' is not defined
```

In [9]:
```
print(int(2.3)) # float to int
print(int(True)) #Boolean to int
print(int('1')) # String to int
```

```
2
1
1
```

In [8]:
```
print(int(1+2j)) # complex to int
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 print(int(1+2j))

TypeError: int() argument must be a string, a bytes-like object or a real number, no
t 'complex'
```

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

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

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

In [13]:
```
print(float(2))
print(float(False))
```

```
2.0
0.0
```

In [14]:
```
print(float(2+3j))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 print(float(2+3j))

TypeError: float() argument must be a string or a real number, not 'complex'
```

In [15]:
```
print(float('two'))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[15], line 1
----> 1 print(float('two'))

ValueError: could not convert string to float: 'two'
```

In [16]:
```
print(float('2'))
```

```
2.0
```

In [17]:
```
complex(10)
```

Out[17]: (10+0j)

In [18]:
```
complex(10,20)
```

Out[18]: (10+20j)

In [19]:
```
complex(10,20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 complex(10,20,30)

TypeError: complex() takes at most 2 arguments (3 given)
```

In [20]:
```
complex(2.3,5.6)
```

Out[20]: (2.3+5.6j)

In [21]:
```
complex(2.3,5)
```

Out[21]: (2.3+5j)

In [23]:
```
complex(5,2.3)
```

Out[23]: (5+2.3j)

In [24]:
```
complex(False,False)
```

Out[24]: 0j

In [25]:
```
complex('1',23)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 complex('1',23)

TypeError: complex() can't take second arg if first is a string
```

In [26]:
```
complex(1,'23')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 complex(1,'23')

TypeError: complex() second arg can't be a string
```

In [27]:
```
complex('1','23')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 complex('1','23')

TypeError: complex() can't take second arg if first is a string
```

In [28]:
```
complex('1')
```

Out[28]: (1+0j)

In [29]:
```
print(bool(1))
print(bool(1.1))
print(bool(1+2j))
print(bool('nit'))
```

```
True
True
True
True
```

In [30]:
```
bool( )
```

Out[30]: False

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

Out[31]: False

In [32]:
```
str(2)
```

Out[32]: '2'

In [33]:
```
str(2.2)
```

Out[33]: '2.2'

In [34]:
```
str(True)
```

Out[34]: 'True'

In [35]:
```
str(1+2j)
```

Out[35]: '(1+2j)'

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: