Type casting

Data type converstion

converting one data type to another data type

int

float

str

bool

complex

int to other data types

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

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

In [3]:
```
# convert int  to   float
float_number=float(number)
float_number

# 100 ======== 100.0
# int ======== float
```

Out[3]:
```
100.0
```

In [4]:
```
type(float_number)
```

Out[4]:
```
float
```

In [6]:
```
# convert into str type
string_number=str(number)
string_number
```

Out[6]:
```
'100'
```

In [7]:
```
type(string_number)

# 100 ============= '100'
# int =============  str
```

Out[7]:
```
str
```

In [8]:
```
# convert into boolean
bool(number)
```

Out[8]:
```
True
```

In [ ]:
```
bool(200)
bool(300)
```

In [11]:
```
bool(200)
```

Out[11]:
```
True
```

In [14]:
```
n=300
float(n)
```

Out[14]:
```
300.0
```

In [15]:
```
bool(-200)
```

Out[15]:
```
True
```

In [16]:
```
bool(0)
```

Out[16]:
```
False
```

Other than zero everything is True under the bool operations

In [19]:
```
# int to complex
a=200
complex(a)  # 200+0j
```

Out[19]:
```
(200+0j)
```

In [20]:
```
a=200
b=300
complex(a,b) # 200+300j
```

Out[20]:
```
(200+300j)
```

In [23]:
```
complex(0,300)
```

Out[23]:
```
300j
```

In [ ]:
```
# int to float
# int to str
# int to bool
# int to complex
```

float to other data types

In [24]:
```
number1=10.5
number1
```

Out[24]: 10.5

In [25]:
```
type(number1)
```

Out[25]: float

In [27]:
```
# float to integer
int_number=int(number1)

# 10.5 =========== 10
# float ========= int
int_number
```

Out[27]: 10

In [28]:
```
type(int_number)
```

Out[28]: int

In [30]:
```
#float to string
str(number1)
```

Out[30]: '10.5'

In [31]:
```
#float to bool
bool(number1)
```

Out[31]: True

In [32]:
```
# float to complex
complex(number1)
```

Out[32]: (10.5+0j)

In [ ]:
```
number2=100.2
int(number2)   # 100
str(number2)   # '100.2'
bool(number2)  # True
complex(number2)# 100.2+0j
```

String to other data types

In [ ]:
```
string1='python'

int(string1)     # error   python   6
float(string1)
bool(string1)
complex(string1)
```

In [33]:
```
string1='python'

int(string1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[33], line 3
      1 string1='python'
----> 3 int(string1)

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

In [34]:
```
string1='python'
float(string1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[34], line 2
      1 string1='python'
----> 2 float(string1)

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

In [35]:
```
string1='python'
bool(string1)
```

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

In [37]:
```
string2=""   # empty string means nothing mentioned in quotes
bool(string2)
```

Out[37]:
```
False
```

Empty string will provide FALSE under bool type

In [38]:
```
string1='python'
complex(string1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[38], line 2
      1 string1='python'
----> 2 complex(string1)

ValueError: complex() arg is a malformed string
```

In [ ]:
```
# Name error
# Value error
```

In [ ]:
```

```

In [ ]:
```
# when it is zero  it is False
# when it is empty string it is False
```

In [ ]:
```
# when it False
```

In [ ]:
```
# english letters can not convert into maths
```

In [ ]:
```
what is "with base 10"

give me 1000rs

0b1000000
```

In [ ]:
```
string3='10'
int(string3)    # 10
float(string3)  # 10.0
bool(string3)   #
complex(string3)
```

In [42]:
```
string3='10'
complex(string3)
```

Out[42]:
```
(10+0j)
```

In [ ]:
```
string4='10.5'
int(string4)    # 10
float(string4)  # 10.0
bool(string4)   #
complex(string4)
```

In [44]:
```
string4='10.5'  # internally what type float
int(string4)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[44], line 2
      1 string4='10.5'  # internally what type float
----> 2 int(string4)

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

In [45]:
```
int('10')   # This will works  10 is generally int=== quotes
```

Out[45]:
```
10
```

In [46]:
```
int('10.5') # 10.5 ==== float   # it will fail
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[46], line 1
----> 1 int('10.5')

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

In [ ]:
```
float('10')   # 10.0
```

In [47]:
```
float('10.5')
```

Out[47]:
```
10.5
```

In [ ]:
```
# float is the boss 10  10.5  it will give the answer
# int will works only on integer value in quotes
# int will not  works on float value in quotes
```

In [ ]:
```
string1='python'
string2='10'
string3='10.5'
string4='10+20j'
```

In [ ]:
```
string4='10+20j'
int(string4)          # error
float(string4)        # error
bool(string4)         # True
complex(string4)      #
```

In [51]:
```
string4='10+20j'
complex(string4)
```

Out[51]:
```
(10+20j)
```

In [ ]:
```
maths ====== maths
english === english
maths+english ==== english
```

In [ ]:
```
int('10')
# check-1:  '10'  10 is what?  = integer it is in quotes
# int('10')   both are same group it will works

int('10.5')
# check-1 :  '10.5' ========= 10.5 = float
# check-2:  int with float   , this will fail


# floa is the boss
# float('10') and float('10.5') both will works
```

In [ ]:
```
Complex to other data types
```

In [ ]:
```
complex_number= 10+20j
int(10+20j)
float(10+20j)
bool(10+20j)
str(10+20j)
```

python

esc+m

do shift+enter

Note:

- Other than zero every numerical either it is int or float it is True under boolean type
- If it is empty string , it is False under boolean type
- int('10.5') provides error
- complex('10+20j') will convert into complex values only
- above two examples we need to observe two points
  - what is the type cast
  - what is the base data type in quotes
  - if both are same it will works
  - int is integer type , but '10.5' is float in quotes both are different so error
  - complex type , inside btackets '10+20j' is complex in quotes so No error

In [ ]:

In [ ]: