In [2]:
```
true=True

# true is variable
# True is a Boolean value we are storing in true
```

In [3]:
```
true
```

Out[3]:
```
True
```

In [ ]:
```
name 'true' is not defined
name  'hyd' is not defined
name   'India' is not defined
```

In [5]:
```
n1=100
```

In [6]:
```
n1
```

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

In [7]:
```
# 15 days our main goal
# to avoid the syntax error
type true
```

```
  Cell In[7], line 3
    type true
         ^
SyntaxError: invalid syntax
```

In [8]:
```
type(true)
```

Out[8]:
```
bool
```

In [ ]:
```
100
```

In [ ]:
```
1)mistakes are very very common
2) avoid the syntax error
3) You will think about logic
4) 1min=== 1hour
1st === 10th === 25th ==== 100th
```

- intger data type int
- float data type float
- string data type str
- Boolean data type bool

Type casting

- Convert one data type to another data type is called as type casting
- which means we wants to convert

- intger type to all other(float,str,bool) data types
- float type to all other(int,str,bool) data types
- str type to all other(int,float,bool) data types
- bool type to all other(int,float,str) data types

Integer to other data types

$int - float$

In [9]:
```
num=100
type(num)
```

Out[9]:
```
int
```

In [12]:
```
float_num=float(num)
type(float_num)
```

Out[12]:
```
float
```

In [13]:
```
str_num=str(num)
str_num
```

Out[13]:
```
'100'
```

In [14]:
```
type(str_num)
```

Out[14]:
```
str
```

In [15]:
```
bool(num)
```

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

In [16]:
```
float(200),str(200),bool(200)
```

Out[16]:
```
(200.0, '200', True)
```

In [17]:
```
float(-200),str(-200),bool(-200)
```

Out[17]:
```
(-200.0, '-200', True)
```

In [18]:
```
# when integer of boolean converstion become False
bool(0)
```

Out[18]:
```
False
```

Note

- Intger =0 then only boolean conversion becomes False
- Otherwise for any postive number or any negative number it becomes True

Float to another data type

In [ ]:
```
int(200.5)  # 200
str(200.5)  # '200.5'
bool(200.5) # True
```

In [20]:
```
int(200.5),str(200.5),bool(200.5)
```

Out[20]:
```
(200, '200.5', True)
```

In [21]:
```
int(200.5),str(200.5),bool(0.0)
```

Out[21]:
```
(200, '200.5', False)
```

String to another data types

In [ ]:
```
# case-1
str='apple'
# case-2
str1='10'
# case3:
str2='10.5'
```

In [ ]:
```
int('apple') # error
float('apple') # error
bool('apple')  # True
```

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

In [ ]:
```
int('10.5') # error
float('10.5') # 10.5
bool('10.5')  # True
```

In [ ]:
```
int('10')  # pass
int('10.5') # Fail
float('10')# pass
float('10.5')# pass
```

- Float is the boss
- Float of any value either it is integer in strings format
- Or float in strings format will works
- But Integer will works only for integer in strings format
- Integer type casting will fail for float in strings format

In [24]:
```
bool('')
```

Out[24]:
```
False
```

In [23]:
```
bool(0)
```

Out[23]: False

- Empty string of boolean give Flase
- False means nothing either 0 or empty string

In [ ]:
```
# boolean to other type

True ==== int  float  str
```

In [25]:
```
int(True)
```

Out[25]: 1

In [26]:
```
float(True)
```

Out[26]: 1.0

In [27]:
```
str(True)
```

Out[27]: 'True'

In [28]:
```
int(False)
float(False)
str(False)
```

Out[28]: 'False'

In [ ]:
```
############ Int to other##################
float(100)
str(100)
bool(100)

############# Float to other ##############
int(100.5)
str(100.5)
bool(100.5)

bool(0)

################ Str to other ##################
int('apple') # error
float('apple') # error
bool('apple')  # True

int('10') # 10
float('10') # 10.0
bool('10')  # True

int('10.5') # error
float('10.5') # 10.5
bool('10.5')  # True

bool('')

#######################bool to other#####################################
```

```
int(False)
float(False)
str(False)

int(True)
float(True)
str(True)
```