In [1]:

```
a=5.5
type(a)
```

Out[1]:

```
float
```

In [3]:

```
import sys
sys.version
```

Out[3]:

```
'3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 6
4 bit (AMD64)]'
```

In [9]:

```
nit=15 # python is case sensitive
NIT
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 2
      1 nit=15 # python is case sensitive
----> 2 NIT

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

In [7]:

```
nit
```

Out[7]:

```
15
```

In [11]:

```
1a=27 # python variable not starts with numeric value
1a
```

```
  Cell In[11], line 1
    1a=27
    ^
SyntaxError: invalid decimal literal
```

In [15]:

```
nit1=29 # python variable can end with numeric value
nit1
```

Out[15]:

```
29
```

In [21]:

```
x_train, x_test, y_train, y_test = 80, 20, 70, 30
```

In [23]:

```
x_train
x_test
y_train
y_test
```

Out[23]:

```
30
```

In [25]:

```
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

```
80
20
70
30
```

In [27]:

```
import keyword
keyword.kwlist
```

Out[27]:

```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [29]:

```
if = 90
if
```

```
  Cell In[29], line 1
    if = 90
       ^
SyntaxError: invalid syntax
```

In [31]:

```
a10 = 78
a9 = 89
```

In [33]:

```
print(a10)
print(a9)
```

```
78
89
```

In [39]:

```
for=10 # these are reserved keyword in python
for
```

```
  Cell In[39], line 1
    for=10 # these are reserved keyword in python
       ^
SyntaxError: invalid syntax
```

In [37]:

```
For=10
For
```

Out[37]:

```
10
```

In [41]:

```
a='True'
a
```

Out[41]:

```
'True'
```

In [43]:

```
a=True
a
```

Out[43]:

```
True
```

In [47]:

```
# Python DATA TYPES
```

In [49]:

```
i=25
i
```

Out[49]:

```
25
```

In [51]:

```
type(i)
```

Out[51]:

```
int
```

In [53]:

```
print(type(i))
```

```
<class 'int'>
```

In [55]:

```
petrol = 109.50 #value with decimal
petrol
```

Out[55]:

```
109.5
```

In [57]:

```
type(petrol)
```

Out[57]:

```
float
```

In [59]:
```
b=true
b
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[59], line 1
----> 1 b=true
      2 b

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

In [61]:
```
b=True
b
```

Out[61]:
```
True
```

In [63]:
```
True+True # True indicates 1 and False indicates 0 in python
```

Out[63]:
```
2
```

In [65]:
```
True+False
```

Out[65]:
```
1
```

In [67]:
```
False+True
```

Out[67]:
```
1
```

In [69]:
```
True-True
```

Out[69]:
```
0
```

In [71]:
```
False-True
```

Out[71]:
```
-1
```

In [73]:
```
c1 = 10 + 20j
c1
```

Out[73]:
```
(10+20j)
```

In [75]:
```
type(c1)
```

Out[75]:
```
complex
```

In [77]:
```
c1.real
```

Out[77]:
```
10.0
```

In [79]:
```
c1.imag
```

Out[79]:
```
20.0
```

In [83]:
```
c1
```
Out[83]:
```
(10+20j)
```

In [85]:
```
c2 = 20 + 30j
```

In [87]:
```
print(c1)
print(c2)
```
```
(10+20j)
(20+30j)
```

In [89]:
```
c1 + c2
```
Out[89]:
```
(30+50j)
```

In [91]:
```
c1 - c2
```
Out[91]:
```
(-10-10j)
```

In [93]:
```
c2-c1
```
Out[93]:
```
(10+10j)
```

In [95]:
```
c3 = 20+ 15i # only j symbol is alowed in imag part
```
```
  Cell In[95], line 1
    c3 = 20+ 15i
               ^
SyntaxError: invalid decimal literal
```

In [97]:
```
c3 = 20+ 15j
c3
```
Out[97]:
```
(20+15j)
```

In [99]:
```
c1 * c2
```
Out[99]:
```
(-400+700j)
```

In [101…
```
s = 'nareshit'
s
```
Out[101…
```
'nareshit'
```

In [103…
```
s1 = "naresh it"
s1
```
Out[103…
```
'naresh it'
```

In [105…
```
s2 = '''naresh
      it'''
s2
```

```
Out[105…  'naresh\n      it'

In [107…  # String Slicing in Python

In [109…  s

Out[109…  'nareshit'

In [111…  s[1]

Out[111…  'a'

In [113…  s[:]

Out[113…  'nareshit'

In [117…  s[3] # forward indexing

Out[117…  'e'

In [119…  s[-3] # Backward indexing

Out[119…  'h'

In [121…  s[1:7]

Out[121…  'areshi'

In [123…  s[1:13] # it wont throw error instead print till last character

Out[123…  'areshit'

In [125…  s[10]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[125], line 1
----> 1 s[10]

IndexError: string index out of range
```

```
In [127…  len(s)

Out[127…  8

In [129…  # Python Type casting

In [133…  int(2.3) # casting from float to int

Out[133…  2

In [135…  int(2.3,3.0)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[135], line 1
----> 1 int(2.3,3.0)

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

In [137…

```
int(True) # cast from bool to int
```

Out[137…

```
1
```

In [139…

```
int(False)
```

Out[139…

```
0
```

In [141…

```
int(2+3j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[141], line 1
----> 1 int(2+3j)

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

In [143…

```
int('10')
```

Out[143…

```
10
```

In [145…

```
int(10)
```

Out[145…

```
10
```

In [147…

```
int('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[147], line 1
----> 1 int('ten')

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

In [149…

```
int('five')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[149], line 1
----> 1 int('five')

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

In [151…

```
int('5')
```

Out[151…

```
5
```

```
In [153…  int(5)

Out[153…  5

In [155…  # Float type casting

In [157…  float(10)

Out[157…  10.0

In [161…  float(False)

Out[161…  0.0

In [163…  float(True)

Out[163…  1.0

In [165…  float('5')

Out[165…  5.0

In [167…  float('five')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[167], line 1
----> 1 float('five')

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

```
In [171…  float(20+3j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[171], line 1
----> 1 float(20+3j)

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

```
In [173…  # complex data type casting

In [175…  complex(10)

Out[175…  (10+0j)

In [177…  complex(10,20)

Out[177…  (10+20j)

In [179…  complex(20,30,40,50)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[179], line 1
----> 1 complex(20,30,40,50)

TypeError: complex() takes at most 2 arguments (4 given)
```

```
In [181…  complex(2.3)

Out[181…  (2.3+0j)
```

```
In [183…  complex('5')

Out[183…  (5+0j)
```

```
In [185…  complex(True)

Out[185…  (1+0j)
```

```
In [187…  complex(True,False)

Out[187…  (1+0j)
```

```
In [189…  complex(False)

Out[189…  0j
```

```
In [195…  complex('10')

Out[195…  (10+0j)
```

```
In [193…  complex('10','20')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[193], line 1
----> 1 complex('10','20')

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

```
In [ ]:
```