```
In [1]: 9

Out[1]: 9

In [2]: 10+5

Out[2]: 15

In [3]: (10+5)*3

Out[3]: 45

In [4]: 10+5*3

Out[4]: 25

In [5]: 10/5 #float div

Out[5]: 2.0

In [6]: 10//5 # int div

Out[6]: 2

In [7]: "nareshit"

Out[7]: 'nareshit'

In [8]: 'nareshit'

Out[8]: 'nareshit'

In [9]: #errors
        #compile time error:
        #logical error
        #run time error

In [10]: '''hgicjugjmcoicopj
         hufgvhcorgh nvmojgivj
         jfhvgiojveoifjvourgbfihg
         ihoiehoifjvmcoigojoifjcj'''

Out[10]: 'hgicjugjmcoicopj\nhufgvhcorgh nvmojgivj\njfhvgiojveoifjvourgbfihg\nihoieh
         oifjvmcoigojoifjcj'

In [11]: s = '''hgicjugjmcoicopj
         hufgvhcorgh nvmojgivj
         jfhvgiojveoifjvourgbfihg
         ihoiehoifjvmcoigojoifjcj'''
```

```
In [12]: s

Out[12]: 'hgicjugjmcoicopj\nhufgvhcorgh nvmojgivj\njfhvgiojveoifjvourgbfihg\nihoieh
         oifjvmcoigojoifjcj'
```

```
In [13]: import sys
         sys.version

Out[13]: '3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC
         v.1916 64 bit (AMD64)]'
```

```
In [14]: import keyword
         keyword.kwlist

Out[14]: ['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 [15]: len(keyword.kwlist)

Out[15]: 35
```

```
In [16]: #variable
         a=5
         a

Out[16]: 5

In [17]: type(a)

Out[17]: int

In [18]: id(a)

Out[18]: 140708695544744

In [19]: b=5.5
         c="alice"
         b

Out[19]: 5.5

In [20]: type(b)
         print(type(b))

         <class 'float'>

In [21]: print(a)

         5

In [22]: type(c)

Out[22]: str

In [23]: print(c)

         alice

In [24]: f1 = 3e0
         f1

Out[24]: 3.0

In [25]: f2 = 3e1
         f2

Out[25]: 30.0

In [26]: f3 = 3e2
         f3

Out[26]: 300.0
```

```
In [27]: f4 = 2.3e4
         f4

Out[27]: 23000.0

In [28]: True

Out[28]: True

In [29]: False

Out[29]: False

In [30]: None

In [31]: True + False

Out[31]: 1

In [32]: True+True

Out[32]: 2

In [33]: True + False *True - False

Out[33]: 1

In [34]: int(True)

Out[34]: 1

In [35]: print(True*2)

         2

In [36]: n = 10
         m= 20
         add =n+m
         print("the add of the value",add ,"this is the answer")

         the add of the value 30 this is the answer

In [37]: c=10+20j
         c

Out[37]: (10+20j)

In [38]: type(c)

Out[38]: complex
```

```
In [39]: d=30+20j
         d

Out[39]: (30+20j)

In [40]: c+d

Out[40]: (40+40j)

In [41]: c.real

Out[41]: 10.0

In [42]: c.imag

Out[42]: 20.0

In [43]: s = 'abc'
         s

Out[43]: 'abc'

In [44]: type(s)

Out[44]: str

In [45]: s1 ="abcd"
         s1

Out[45]: 'abcd'

In [46]: s2 = '''abcde'''
         s2

Out[46]: 'abcde'

In [47]: s3 = s+s1+s2
         s3

Out[47]: 'abcabcdabcde'
```

```
In [48]: s4 = s*s1*s2
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[48], line 1
----> 1 s4 = s*s1*s2

TypeError: can't multiply sequence by non-int of type 'str'
```

```
In [49]: s5 = """Hello
World"""
s5
```

```
Out[49]: 'Hello\nWorld'
```

```
In [50]: s2[2]
```

```
Out[50]: 'c'
```

```
In [51]: s2[10]
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[51], line 1
----> 1 s2[10]

IndexError: string index out of range
```

```
In [52]: s2[-3]
```

```
Out[52]: 'c'
```

```
In [53]: for i in s3:
    print(i)
```

```
a
b
c
a
b
c
d
a
b
c
d
e
```

```
In [54]: s3

Out[54]: 'abcabcdabcde'

In [55]: s3[1:6]

Out[55]: 'bcabc'

In [56]: s3[-5:-2]

Out[56]: 'abc'

In [57]: s3[-5:4]

Out[57]: ''

In [58]: s3

Out[58]: 'abcabcdabcde'

In [59]: s3[0:10:2]

Out[59]: 'acbdb'

In [60]: #Type casting
         int(2.3)

Out[60]: 2

In [61]: int(True)

Out[61]: 1

In [62]: int(1+2j) # complex to int is not possiable

         ---------------------------------------------------------------------------
         -
         TypeError                                 Traceback (most recent call las
         t)
         Cell In[62], line 1
         ----> 1 int(1+2j)

         TypeError: int() argument must be a string, a bytes-like object or a real
         number, not 'complex'

In [63]: int("10")

Out[63]: 10
```

```
In [64]: int("hello")

         ---------------------------------------------------------------------------
         -
         ValueError                                Traceback (most recent call las
         t)
         Cell In[64], line 1
         ----> 1 int("hello")

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

```
In [65]: float(1)

Out[65]: 1.0
```

```
In [66]: float(False)

Out[66]: 0.0
```

```
In [67]: float(1+2j) # complex to float is not possiable

         ---------------------------------------------------------------------------
         -
         TypeError                                 Traceback (most recent call las
         t)
         Cell In[67], line 1
         ----> 1 float(1+2j)

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

```
In [68]: float("10")

Out[68]: 10.0
```

```
In [69]: float("ten")

         ---------------------------------------------------------------------------
         -
         ValueError                                Traceback (most recent call las
         t)
         Cell In[69], line 1
         ----> 1 float("ten")

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

```
In [70]: int(10.2,12.3) # only one argument is possiable in int
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[70], line 1
----> 1 int(10.2,12.3)

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

```
In [72]: float(10,12) # only one argument is possiable in float
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[72], line 1
----> 1 float(10,12)

TypeError: float expected at most 1 argument, got 2
```

```
In [75]: complex(1)

Out[75]: (1+0j)
```

```
In [76]: complex(1,2)

Out[76]: (1+2j)
```

```
In [77]: complex(1,2,3)
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[77], line 1
----> 1 complex(1,2,3)

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

```
In [80]: complex(10.2,23.8)

Out[80]: (10.2+23.8j)
```

```
In [81]: complex(True,False)

Out[81]: (1+0j)
```

```
In [82]: complex(False,True)

Out[82]: 1j
```

```
In [83]: complex("10")

Out[83]: (10+0j)
```

```
In [84]: complex("ten")

---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[84], line 1
----> 1 complex("ten")

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

```
In [85]: complex('10','20')

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[85], line 1
----> 1 complex('10','20')

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

```
In [86]: complex('10',20)

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[86], line 1
----> 1 complex('10',20)

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

```
In [87]: complex(10,'20')

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[87], line 1
----> 1 complex(10,'20')

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

```
In [88]: str(10)

Out[88]: '10'

In [89]: str(10,12)

         ---------------------------------------------------------------------------
         -
         TypeError                                 Traceback (most recent call las
         t)
         Cell In[89], line 1
         ----> 1 str(10,12)

         TypeError: str() argument 'encoding' must be str, not int

In [90]: str(10.222)

Out[90]: '10.222'

In [91]: str(True)

Out[91]: 'True'

In [92]: str(False)

Out[92]: 'False'

In [93]: str(1+2j)

Out[93]: '(1+2j)'

In [94]: bool(10)

Out[94]: True

In [95]: bool('0')

Out[95]: True

In [96]: bool(11)

Out[96]: True

In [97]: bool(1+2j)

Out[97]: True

In [98]: bool(0)

Out[98]: False
```

```
In [99]: bool()

Out[99]: False

In [100]: bool( )

Out[100]: False

In [101]: bool(2.3)

Out[101]: True

In [102]: bool(0+0j)

Out[102]: False

In [103]: bool("ten")

Out[103]: True

In [ ]:

In [ ]:
```