

# BASIC PYTHON PROGRAMMING

```
In [2]: import sys  
sys.version
```

```
Out[2]: '3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 6  
4 bit (AMD64)]'
```

```
In [3]: #Number -- Integer (int)  
#text- string (str) ''',''',''' '''
```

## WORK WITH NUMBERS

```
In [5]: 4+5
```

```
Out[5]: 9
```

```
In [6]: 5-6
```

```
Out[6]: -1
```

```
In [7]: 2*4
```

```
Out[7]: 8
```

```
In [8]: 4/2 #output in float
```

```
Out[8]: 2.0
```

```
In [9]: 4//2 #output in int
```

```
Out[9]: 2
```

```
In [10]: 2**4
```

```
Out[10]: 16
```

```
In [11]: 3%6
```

```
Out[11]: 3
```

## WORK WITH STRINGS

```
In [13]: 'SKAFTAB', "SKAFTAB", '''SKAFTAB''' #Strings or text always ends with quotes.
```

```
Out[13]: ('SKAFTAB', 'SKAFTAB', 'SKAFTAB')
```

## VARIABLE = OBJECT

```
In [15]: v = 7 # v is variable $ 7 is value.  
v
```

```
Out[15]: 7
```

```
In [16]: type(v)
```

```
Out[16]: int
```

```
In [17]: v=7  
v1 #NameError V1 is not defiend
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[17], line 2  
      1 v=7  
----> 2 v1  
  
NameError: name 'v1' is not defined
```

```
In [115... v=19  
v
```

```
Out[115... 19
```

```
In [117... v1="SK"  
v1
```

```
Out[117... 'SK'
```

```
In [120... a=2.3  
type(a)
```

```
Out[120... float
```

```
In [122... import sys  
sys.version
```

```
Out[122... '3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 6  
4 bit (AMD64)]'
```

```
In [124... nit=15  
NIT#Name error Cause its not defiend
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[124], line 2  
      1 nit=15  
----> 2 NIT  
  
NameError: name 'NIT' is not defined
```

In [126... nit

Out[126... 15

In [128... 1a=67  
1a#variable come first then num

```
Cell In[128], line 1  
    1a=67  
      ^  
SyntaxError: invalid decimal literal
```

In [130... a1=4  
a1

Out[130... 4

In [132... nit\$=45  
nit\$#special key not allowed except \_

```
Cell In[132], line 1  
    nit$=45  
      ^  
SyntaxError: invalid syntax
```

In [134... x\_train, x\_test, y\_train, y\_test= 80,20,70,30

In [136... x\_train  
x\_test  
y\_train  
y\_test#only last element will come cause print was not there

Out[136... 30

In [138... print(x\_train)  
print(x\_test)  
print(y\_train)  
print(y\_test)

80  
20  
70  
30

In [140... import keyword  
keyword.kwlist

```
Out[140... ['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 [142... if=90
if#syntaxError Cause "if" is a keyword
```

```
Cell In[142], line 1
    if=90
    ^
SyntaxError: invalid syntax
```

```
In [144... a12 = 40
a8 = 84
```

```
In [146... print(a12)
print(a8)
```

```
40
84
```

```
In [148... for = 20#syntax Error because these are keys
```

^

```
In [150... dei 10
```

^

```
In [152... For = 20
For
```

```
In [154... a = True
a
```

```
In [156... b = "True"
b
```

```
In [158... pi = 3.17
pi
```

```
In [160... pi = 3.20
pi
```

In [162...]

```
q = np.zeros((n, n))  
for i in range(n):  
    q[i, i] = 1  
    q[i, i+1] = 0.5  
    q[i+1, i] = 0.5  
    q[i+1, i+1] = 0.9
```

= 90

In [164... a

In [ ]:

## INT FLOAT BOOLEAN COMPKEX STRING

# INTEGER

```
In [170... i = 23  
i
```

```
Out[170... 23
```

```
In [172... type(i)
```

```
Out[172... int
```

```
In [174... print(type(i))
```

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

# FLOAT

```
In [177... petrol = 105.95  
petrol
```

```
Out[177... 105.95
```

```
In [179... type(petrol)
```

```
Out[179... float
```

```
In [181... print(type(petrol))
```

```
<class 'float'>
```

# BOOLEAN

```
In [184... b = true  
b#first letter should be in capital for bool
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[184], line 1  
----> 1 b = true  
      2 b  
  
NameError: name 'true' is not defined
```

```
In [186... b = True  
b
```

```
Out[186... True
```

```
In [188... b1 = False
b1
```

```
Out[188... False
```

```
In [190... True + False#Accor to program or Coding True = 1 & False = 0
```

```
Out[190... 1
```

```
In [192... False - True
```

```
Out[192... -1
```

```
In [194... True + True
```

```
Out[194... 2
```

```
In [196... True / False#undefiend
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
Cell In[196], line 1
----> 1 True / False

ZeroDivisionError: division by zero
```

```
In [198... False/True
```

```
Out[198... 0.0
```

```
In [200... False//True
```

```
Out[200... 0
```

## COMPLEX

```
In [203... c = a+bj #a=real part ,b= imaginary part & j is the square root of -1
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[203], line 1
----> 1 c = a+bj

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

```
In [205... c1 = 10 + 20j
```

```
In [207... c1.real
```

```
Out[207... 10.0
```

```
In [209... c1.imag #IMAGINARY
```

```
Out[209... 20.0
```

```
In [211... c1
```

```
Out[211... (10+20j)
```

```
In [213... c2 = 20 + 30j
c2
```

```
Out[213... (20+30j)
```

```
In [215... print(c1)
print(c2)
```

```
(10+20j)
```

```
(20+30j)
```

```
In [217... c1+c2
```

```
Out[217... (30+50j)
```

```
In [219... print(c1+c2)
```

```
(30+50j)
```

```
In [221... c3 = 20 + 60i#only j letter is allowed for complex
```

```
Cell In[221], line 1
```

```
c3 = 20 + 60i#only j letter is allowed for complex
```

```
SyntaxError: invalid decimal literal
```

```
In [223... c1-c2
```

```
Out[223... (-10-10j)
```

```
In [225... c1/c2
```

```
Out[225... (0.6153846153846154+0.0769230769230769j)
```

```
In [227... c1//c2
```

```
-----
TypeError
```

```
Traceback (most recent call last)
```

```
Cell In[227], line 1
```

```
----> 1 c1//c2
```

```
TypeError: unsupported operand type(s) for //: 'complex' and 'complex'
```

```
In [229... c1*c2
```

```
Out[229... (-400+700j)
```



# STRING

```
In [232... s = 'SK AFTAB'  
s
```

```
Out[232... 'SK AFTAB'
```

```
In [234... s1 = "SKAFTAB"  
s1
```

```
Out[234... 'SKAFTAB'
```

```
In [236... s2 = '''SK  
      AFTAB'''# only exceptable under the triple quoets  
s2 #\n is use for next line
```

```
Out[236... 'SK\n      AFTAB'
```

## STRING SLICING [:]

## INDEXING

```
In [240... s = "SK AFTAB"
```

```
In [242... s
```

```
Out[242... 'SK AFTAB'
```

```
In [244... s[:]#slice notaion or full or identity slic to access the entire sequence s[0:len(s
```

```
Out[244... 'SK AFTAB'
```

```
In [246... s[5] #FORWARD indexing start from 0.
```

```
Out[246... 'T'
```

```
In [248... s
```

```
Out[248... 'SK AFTAB'
```

```
In [250... s[-3] #BACKWARD Indexing Start from -1
```

```
Out[250... 'T'
```

```
In [252... b = True  
b
```

Out[252...] True

In [254...] `int(True)`

Out[254...] 1

In [256...] `int(False)`

Out[256...] 0

In [258...] `float(True)`

Out[258...] 1.0

In [260...] `float(False)`

Out[260...] 0.0

In [262...] `True`

Out[262...] True

In [264...] `False`

Out[264...] False

In [266...] `int`

Out[266...] int

In [268...] `complex`

Out[268...] complex

In [270...] `bool`

Out[270...] bool

In [272...] `str`

Out[272...] str

## SLICING INDEXING

In [275...] `s`

Out[275...] 'SK AFTAB'

In [277...] `s[1:3]`  *#(n-1) last index*

Out[277... 'K '

In [279... `s[1:5]`

Out[279... 'K AF'

In [281... `s[7]`

Out[281... 'B'

In [283... `s = "sk aftar"`

In [285... `len(s)` *#len is the total length of value including space.*

Out[285... 8

In [287... `s`

Out[287... 'sk aftar'

## TYPE CASTING

In [291... `int(2.3)` *#Cast or Convert From Float TO Int.*

Out[291... 2

In [293... `int(2.3,3.0)` *#only one parameter or argument is allowed.*

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[293], line 1  
----> 1 int(2.3,3.0)  
  
TypeError: 'float' object cannot be interpreted as an integer
```

In [295... `int(True)` *#cast from Bool to Int.*

Out[295... 1

In [297... `int(False)`

Out[297... 0

In [299... `True`

Out[299... True

In [301... `True + True`

Out[301... 2

```
In [303... int(1=2j) #Cannot cast complex to int because of its combine of num $ alph
```

```
Cell In[303], line 1
```

```
int(1=2j) #Cannot cast complex to int because of its combine of num $ alph
```

```
^
```

```
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

```
In [305... int("10")
```

```
Out[305... 10
```

```
In [307... int(''ten'')
```

```
-----  
ValueError
```

```
Traceback (most recent call last)
```

```
Cell In[307], line 1
```

```
----> 1 int(''ten'')
```

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

```
In [309... float(10)
```

```
Out[309... 10.0
```

```
In [311... float(10,29)#only one parameter is allow
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
Cell In[311], line 1
```

```
----> 1 float(10,29)
```

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

```
In [313... float(True)
```

```
Out[313... 1.0
```

```
In [315... float(False)
```

```
Out[315... 0.0
```

```
In [317... float(1+2j)#cannot cast complex
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
Cell In[317], line 1
```

```
----> 1 float(1+2j)
```

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

```
In [319... float("10")
```

```
Out[319... 10.0
```

In [321... `float("ten")`

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[321], line 1  
----> 1 float("ten")  
  
ValueError: could not convert string to float: 'ten'
```

In [323... `True + True`

Out[323... 2

In [325... `complex(10)`

Out[325... (10+0j)

In [327... `complex(10,20)`

Out[327... (10+20j)

In [329... `complex(1,2,3)` *#only 2 parameter is allowed*

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[329], line 1  
----> 1 complex(1,2,3)  
  
TypeError: complex() takes at most 2 arguments (3 given)
```

In [331... `complex(2.3)`

Out[331... (2.3+0j)

In [332... `complex(2,1.2)`

Out[332... (2+1.2j)

In [335... `complex(True)`

Out[335... (1+0j)

In [337... `complex(False)`

Out[337... 0j

In [339... `complex(1,True)`

Out[339... (1+1j)

In [341... `complex(2,False)`

Out[341... (2+0j)

```
In [343... complex(2.1,True)
```

```
Out[343... (2.1+1j)
```

```
In [345... complex(2.3,False)
```

```
Out[345... (2.3+0j)
```

```
In [347... complex(True,2)
```

```
Out[347... (1+2j)
```

```
In [349... complex(False,2.2)
```

```
Out[349... 2.2j
```

```
In [351... complex(True,3.4)
```

```
Out[351... (1+3.4j)
```

```
In [353... complex(True,"sk") #second arugument can't be true
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[353], line 1  
----> 1 complex(True,"sk")  
  
TypeError: complex() second arg can't be a string
```

```
In [355... complex("sk")
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[355], line 1  
----> 1 complex("sk")  
  
ValueError: complex() arg is a malformed string
```

```
In [357... complex("10")
```

```
Out[357... (10+0j)
```

```
In [359... complex(True,True)
```

```
Out[359... (1+1j)
```

```
In [362... complex("10","20")
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[362], line 1  
----> 1 complex("10", "20")  
  
TypeError: complex() can't take second arg if first is a string
```

In [364... `complex(10, "10")`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[364], line 1  
----> 1 complex(10, "10")  
  
TypeError: complex() second arg can't be a string
```

In [366... `complex("10", 100)`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[366], line 1  
----> 1 complex("10", 100)  
  
TypeError: complex() can't take second arg if first is a string
```

In [368... `bool(2)`

Out[368... `True`

In [370... `bool(0)`

Out[370... `False`

In [372... `bool(-1)`

Out[372... `True`

In [374... `bool(2,3)` *#Only one argument is allowed*

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[374], line 1  
----> 1 bool(2,3)  
  
TypeError: bool expected at most 1 argument, got 2
```

In [376... `bool("10")`

Out[376... `True`

In [378... `bool(2.3)`

Out[378... `True`

```
In [380...] bool(1+2j)
```

```
Out[380...] True
```

```
In [382...] bool(0+0j)
```

```
Out[382...] False
```

```
In [384...] bool(1+1j)
```

```
Out[384...] True
```

```
In [386...] bool(1+0j)
```

```
Out[386...] True
```

```
In [388...] bool(0+1j)
```

```
Out[388...] True
```

```
In [390...] bool("SK")
```

```
Out[390...] True
```

```
In [392...] bool()
```

```
Out[392...] False
```

```
In [394...] str(7)
```

```
Out[394...] '7'
```

```
In [396...] str(2.3)
```

```
Out[396...] '2.3'
```

```
In [398...] str("10")
```

```
Out[398...] '10'
```

```
In [400...] str("ten")
```

```
Out[400...] 'ten'
```

```
In [402...] complex("ten")
```

```
-----  
ValueError                                Traceback (most recent call last)  
Cell In[402], line 1  
----> 1 complex("ten")  
  
ValueError: complex() arg is a malformed string
```



```
In [404... str(True)
```

```
Out[404... 'True'
```

```
In [406... str(False)
```

```
Out[406... 'False'
```

```
In [408... str(1+2j)
```

```
Out[408... '(1+2j)'
```

```
In [410... str(-1)
```

```
Out[410... '-1'
```