welcome

welcome

shift+enter=run the cell

dd=delete the cell

Keywords

In [1]:

```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'brea
k', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finall
y', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonloc
al', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yiel
d']
```

In [6]:

```
false=10
```

In [7]:

```
false
```

Out[7]:

```
10
```

identifier

name given to entity like class,functions,variables etc

In [8]:

```
1var=20
```

```
  Input In [8]
    1var=20
    ^
SyntaxError: invalid decimal literal
```

In [9]:

```
var1=20
```

In [10]:

```
var1
```

Out[10]:

```
20
```

In [11]:

```
var@=30
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [11], in <cell line: 1>()
----> 1 var@=30

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

In [12]:

```
var_=30
```

In [13]:

```
var_
```

Out[13]:

```
30
```

In [14]:

```
finally=40
```

```
  Input In [14]
    finally=40
    ^
SyntaxError: invalid syntax
```

In [15]:

```
Finally=40
```

In [16]:

```
Finally
```

Out[16]:

```
40
```

In [17]:

```
from=50
```

```
  Input In [17]
    from=50
         ^
SyntaxError: invalid syntax
```

In [18]:

```
From=50
```

In [19]:

```
From
```

Out[19]:

```
50
```

comments in python :used to explain the code for more readablity

In [20]:

```
print('python')    #define python
```

```
python
```

In [21]:

```
#single line comment
#multi line comment
```

In [22]:

```
"this
is python
session""
```

```
  Input In [22]
    "this
    ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [23]:

```
'''welcome
to
the
india'''
```

Out[23]:

```
'welcome\nto\nthe\nindia'
```

statement

In [24]:

```
val5=10
```

In [25]:

```
p1=20+30
p1
```

Out[25]:

```
50
```

In [26]:

```
p2=20+30\
+40+50\
+70+80
p2
```

```
  Input In [26]
    p2=20+30\
              ^
SyntaxError: unexpected character after line continuation character
```

In [27]:

```
p2=20+30\
+40+50\
+70+80
p2
```

Out[27]:

```
290
```

indentation

In [28]:

```
x=10
if x==10:
    print('x is equal to 10')
```

```
x is equal to 10
```

docstrings

In [29]:

```
def square(num):
    """square function will return the square of a number"""
    return num**2
```

In [30]:

```
square(2)
```

Out[30]:

```
4
```

In [31]:

```
square.__doc__
```

Out[31]:

```
'square function will return the square of a number'
```

variables

In [32]:

```
a=5
```

In [33]:

```
a
```

Out[33]:

```
5
```

In [34]:

```
p=50
q=25
r=q
```

In [35]:

```
print(id(p))
```

```
2056507950864
```

In [36]:

```
print(id(q))
```

```
2056507950064
```

data type

int float bool/boolean complex

In [37]:

```
a=1254789631422
```

In [38]:

```
type(a)
```

Out[38]:

```
int
```

In [39]:

```
b=10.2
b
```

Out[39]:

```
10.2
```

In [40]:

```
type(b)
```

Out[40]:

```
float
```

In [41]:

```
bool1=True
```

In [42]:

```
bool1
```

Out[42]:

```
True
```

In [43]:

```
print(bool(-2))
```

```
True
```

In [44]:

```
x=2+3j
```

In [45]:

```
type(x)
```

Out[45]:

```
complex
```

Strings

In [46]:

```
str1="hello"
```

In [47]:

```
type(str1)
```

Out[47]:

```
str
```

In [48]:

```
len(str1)
```

Out[48]:

```
5
```

In [49]:

```
str2=" hello  python "
```

In [50]:

```
str2[0]
```

Out[50]:

```
' '
```

In [51]:

```
len(str2)
```

Out[51]:

```
15
```

indexing

In [52]:

```
str3='welcome'
```

in python we always start indexing from 0

In [53]:

```
str3[0]
```

Out[53]:

```
'w'
```

In [54]:

```
str3[-7]
```

Out[54]:

```
'w'
```

Slicing

In [55]:

```
str3[3:6]
```

Out[55]:

```
'com'
```

In [56]:

```
str3[3:7]
```

Out[56]:

```
'come'
```

In [57]:

```
str3[0:]
```

Out[57]:

```
'welcome'
```

In [58]:

```
str3[3:]
```

Out[58]:

```
'come'
```

In [59]:

```
str4="bangalore"
```

In [60]:

```
str4[2:5]
```

Out[60]:

```
'nga'
```

In [61]:

```
str4[3:]
```

Out[61]:

```
'galore'
```

In [62]:

```
str5='data science'
```

In [63]:

```
str5[5:]
```

Out[63]:

```
'science'
```

In [64]:

```
str5[2:8]
```

Out[64]:

```
'ta sci'
```

In [65]:

```
str6='heloo'
```

In [66]:

```
str[3]='l'
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [66], in <cell line: 1>()
----> 1 str[3]='l'

TypeError: 'type' object does not support item assignment
```

In [67]:

```
del str6
```

In [68]:

```
str6
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [68], in <cell line: 1>()
----> 1 str6

NameError: name 'str6' is not defined
```

string concatenation

In [69]:

```
s1='data'
s2='science'  #data science
```

In [70]:

```
print(s1,s2,"dvnksdvnvdkndvn")
```

```
data science dvnksdvnvdkndvn
```

In [71]:

```
print(s1+'-'+s2)
```

```
data-science
```

String Membership

In [72]:

```
mystr='hello everyone'    #in,not in membership operator
```

In [73]:

```
print('hello' in mystr)
```

```
True
```

In [74]:

```
print('python' in mystr)
```

```
False
```

In [75]:

```
print(' eve' in mystr)
```

```
True
```

In [76]:

```
print('-eve' in mystr)
```

```
False
```

string partitioning

In [77]:

```
mystr1='natural language with python and R and java'
mystr1
```

Out[77]:

```
'natural language with python and R and java'
```

In [78]:

```
l=mystr1.partition('with')
l
```

Out[78]:

```
('natural language ', 'with', ' python and R and java')
```

In [79]:

```
mystr1.capitalize()
```

Out[79]:

```
'Natural language with python and r and java'
```

In [80]:

```
mystr1.upper()
```

Out[80]:

```
'NATURAL LANGUAGE WITH PYTHON AND R AND JAVA'
```

In [81]:

```
mystr1.lower()
```

Out[81]:

```
'natural language with python and r and java'
```

In [82]:

```
mystr1.count('a')
```

Out[82]:

```
8
```

In [83]:

```
mystr2="   hello   "
mystr2
```

Out[83]:

```
'   hello   '
```

In [84]:

```
mystr2.strip()
```

Out[84]:

```
'hello'
```

In [85]:

```
mystr2.rstrip()
```

Out[85]:

```
'   hello'
```

In [86]:

```
mystr2.lstrip()
```

Out[86]:

```
'hello   '
```

21 july 2023

In [87]:

```
str6="heloo"
print(id(str6))
```

```
2056614426992
```

In [88]:

```
x=str6.replace('o','l')
print(id(x))
```

```
2056585744752
```

In [89]:

```
str7='good morning'
```

In [90]:

```
str7.replace("good", "beautiful")
```

Out[90]:

```
'beautiful morning'
```

In [91]:

```
str8='one two three four five six seven'
```

In [92]:

```
str8
```

Out[92]:

```
'one two three four five six seven'
```

In [93]:

```
str8.split()
```

Out[93]:

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

variable assignment

In [94]:

```
x=10
```

In [95]:

```
x,y,z=20,25.5,'hello'
```

In [96]:

```
print(x)
print(y)
print(z)
```

```
20
25.5
hello
```

Operators

Arthmetic

Assignment

comparition /relational

logical

membership

identity

bitwise

Arthmetic =====

Addition

Substraction

Multiplication

Division

Floor division

Modulos

Exponentional

In [97]:

```
print(10+2)
print(10-2)
print(10*2)
```

```
12
8
20
```

In [98]:

```
print(15/2)
print(15//2)
print(15%2)
```

```
7.5
7
1
```

In [99]:

```
print(2**2)
```

```
4
```

Assignment

In [100]:

```
x=5
```

In [101]:

```
y=10
y+=2    #y=y+2
```

In [102]:

```
print(y)
```

```
12
```

In [103]:

```
z=15
z-=10    #z=z-10
```

In [104]:

```
print(z)
```

```
5
```

comparision /relational

== Equal

!= not equal

#> greater than

< less than

<= less than equal to

#>= greater than equal to

In [105]:

```
x=10
y=20
print(x==y)
print(x<y)
print(x>y)
```

```
False
True
False
```

Logical operators

and

or

not

In [106]:

```
x=10
y=20
print((x<5) and (y<10))
print((x<5) or (y<10))
print((x==10) and (y==20))
print((x==10) or (y==20))
```

```
False
False
True
True
```

identity operators

In [107]:

```
print((10) is (10.0))
```

```
False

<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
C:\Users\Mukund\AppData\Local\Temp\ipykernel_2652\436760226.py:1: SyntaxWa
rning: "is" with a literal. Did you mean "=="?
  print((10) is (10.0))
```

In [108]:

```
print('hello')
x=10
if (x>5):
    print('welcome')
print(hdsvvhsvn)
print('python')
```

```
hello
welcome

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [108], in <cell line: 5>()
      3 if (x>5):
      4     print('welcome')
----> 5 print(hdsvvhsvn)
      6 print('python')

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

Type Casting

In [109]:

```
a='2'
b=2
```

In [110]:

```
type(a)
```

Out[110]:

```
str
```

In [111]:

```
type(b)
```

Out[111]:

```
int
```

Auto type casting

Forced Type casting

In [112]:

```
4+3.23+False    #4.00+3.23+0.00
```

Out[112]:

```
7.23
```

In [113]:

```
3+2.3+False+True   #3.00+2.3+0.00+1.00
```

Out[113]:

```
6.3
```

In [114]:

```
2+3+'data'+4.5
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [114], in <cell line: 1>()
----> 1 2+3+'data'+4.5

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [115]:

```
12+15+'10'+2
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [115], in <cell line: 1>()
----> 1 12+15+'10'+2

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [116]:

```
print(int(True))
```

```
1
```

In [117]:

```
print(int(bool("data")))
```

```
1
```

In [118]:

```
a='10'
```

In [119]:

```
print(int(a))
```

```
10
```

In [120]:

```
b='data123'
```

In [121]:

```
print(int(b))
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Input In [121], in <cell line: 1>()
----> 1 print(int(b))

ValueError: invalid literal for int() with base 10: 'data123'
```

In [122]:

```
z=3.2
print(int(z))
```

```
3
```

In [123]:

```
x=3
print(float(x))2
```

```
  Input In [123]
    print(float(x))2
                   ^
SyntaxError: invalid syntax
```

In [124]:

```
print(float(False))
```

```
0.0
```

Python KT session 24 July 23

List

In [125]:

```
list1=[10,20,30,40,50] #homogenous same kind of data
print(list1)
print('hello')
```

```
[10, 20, 30, 40, 50]
hello
```

In [126]:

```
type(list1)
```

Out[126]:

```
list
```

In [127]:

```
list2=[60,20.5,'happy']  #hetrogenous  multiple kind of data
list2
```

Out[127]:

```
[60, 20.5, 'happy']
```

In [128]:

```
type(list2)
```

Out[128]:

```
list
```

In [129]:

```
list3=[[1,2,3],[4,5,6]]   #nested list
```

In [130]:

```
type(list3)
```

Out[130]:

```
list
```

Properties of List

In [131]:

```
#list store element in sequential order
#list store hetrogenous element
#list allow duplicate value
#list are mutable or changeable
```

In [132]:

```
list4=[1,1,20,30,40]
id(list4)
```

Out[132]:

```
2056615048768
```

In [133]:

```
list4
```

Out[133]:

```
[1, 1, 20, 30, 40]
```

In [134]:

```
list4[1]=5
```

In [135]:

```
list4
id(list4)
```

Out[135]:

```
2056615048768
```

In [136]:

```
list5=[10,50,7,80,90,5,20,100]
```

In [137]:

```
list5[-4]
```

Out[137]:

```
90
```

In [138]:

```
list5[-7]
```

Out[138]:

```
50
```

In [139]:

```
list5[-5:-2]    #start index postion:
```

Out[139]:

```
[80, 90, 5]
```

In [140]:

```
list6=[[1,2,3],[4,5,6]]
```

In [141]:

```
list6[1][1]
```

Out[141]:

```
5
```

In [142]:

```
list7=[10,50,7,80,90,5,20,100]
```

In [143]:

```
list7.append(99)
```

In [144]:

```
list7
```

Out[144]:

```
[10, 50, 7, 80, 90, 5, 20, 100, 99]
```

In [145]:

```
list7.append(105)
```

In [146]:

```
list7
```

Out[146]:

```
[10, 50, 7, 80, 90, 5, 20, 100, 99, 105]
```

In [147]:

```
list7.pop()
```

Out[147]:

```
105
```

In [148]:

```
list7
```

Out[148]:

```
[10, 50, 7, 80, 90, 5, 20, 100, 99]
```

In [149]:

```
list7.pop()
```

Out[149]:

```
99
```

In [150]:

```
list7
```

Out[150]:

```
[10, 50, 7, 80, 90, 5, 20, 100]
```

In [151]:

```
list7.pop(2)
```

Out[151]:

```
7
```

In [152]:

```
list7
```

Out[152]:

```
[10, 50, 80, 90, 5, 20, 100]
```

In [153]:

```
list7.insert(0,5)
```

In [154]:

```
list7
```

Out[154]:

```
[5, 10, 50, 80, 90, 5, 20, 100]
```

In [155]:

```
list7.reverse()
```

In [156]:

```
list7
```

Out[156]:

```
[100, 20, 5, 90, 80, 50, 10, 5]
```

In [157]:

```
list7.sort()
```

In [158]:

```
list7
```

Out[158]:

```
[5, 5, 10, 20, 50, 80, 90, 100]
```

In [159]:

```
list7.sort(reverse=True)
```

In [160]:

```
list7
```

Out[160]:

```
[100, 90, 80, 50, 20, 10, 5, 5]
```

Tuple

In [161]:

```
tup1=(10,20,30,40)
tup1
```

Out[161]:

```
(10, 20, 30, 40)
```

In [162]:

```
type(tup1)
```

Out[162]:

```
tuple
```

In [163]:

```
tup2=50,60,40,70
tup2
```

Out[163]:

```
(50, 60, 40, 70)
```

In [164]:

```
type(tup2)
```

Out[164]:

```
tuple
```

Properties of tuple

In [165]:

```
#tuple store element in sequential order
#tuple allow duplicate values
#tuple is immutuble or unchangeable
```

In [166]:

```
tup3=(1,1,5,7,8)
```

In [167]:

```
tup3
```

Out[167]:

```
(1, 1, 5, 7, 8)
```

In [168]:

```
tup3[1]=9
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [168], in <cell line: 1>()
----> 1 tup3[1]=9

TypeError: 'tuple' object does not support item assignment
```

In [169]:

```
tup4=(10,20,30,50,70,80,90)
```

In [170]:

```
tup4[3:5]
```

Out[170]:

```
(50, 70)
```

In [171]:

```
tup4.index(80)
```

Out[171]:

```
5
```

In [172]:

```
tup4.count(50)
```

Out[172]:

```
1
```

Set

In [173]:

```
set1={10,20,60}
set1
```

Out[173]:

```
{10, 20, 60}
```

In [174]:

```
type(set1)
```

Out[174]:

```
set
```

In [175]:

```
set2={60,50,10,20,80,10,50}
len(set2)
```

Out[175]:

```
5
```

In [176]:

```
set2
```

Out[176]:

```
{10, 20, 50, 60, 80}
```

Properties of Set

In [177]:

```
#set store element in random order
#set didn't allow duplicate values
#set is immutuble or mutuble
#indexing and slicing is not possible in set
```

In [178]:

```
list10=[]  #empty list
type(list10)
```

Out[178]:

```
list
```

In [179]:

```
tup10=()  #empty tuple
type(tup10)
```

Out[179]:

```
tuple
```

In [180]:

```
set3=set()  #empty set
type(set3)
```

Out[180]:

```
set
```

In [181]:

```
set4={10,20,50,70,80}
```

In [182]:

```
set4[1:3]  #indexing /slicing is not possible
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [182], in <cell line: 1>()
----> 1 set4[1:3]

TypeError: 'set' object is not subscriptable
```

In [183]:

```
set4.remove(70)
```

In [184]:

```
set4
```

Out[184]:

```
{10, 20, 50, 80}
```

In [185]:

```
set4.discard(80)
```

In [186]:

```
set4
```

Out[186]:

```
{10, 20, 50}
```

In [187]:

```
set5={10,20,50,70,80,90,100}
```

In [188]:

```
set5.remove(105)
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
Input In [188], in <cell line: 1>()
----> 1 set5.remove(105)

KeyError: 105
```

In [189]:

```
set5.discard(109)
```

Dictionary

In [190]:

```
dict1={}   #empty dictionary
type(dict1)
```

Out[190]:

```
dict
```

In [191]:

```
dict2={'name':'rohit','age':25}
```

In [192]:

```
dict2
```

Out[192]:

```
{'name': 'rohit', 'age': 25}
```

In [193]:

```
dict2.keys()
```

Out[193]:

```
dict_keys(['name', 'age'])
```

In [194]:

```
dict2.values()
```

Out[194]:

```
dict_values(['rohit', 25])
```

In [195]:

```
dict2.items()
```

Out[195]:

```
dict_items([('name', 'rohit'), ('age', 25)])
```

In [196]:

```
dict2['location']='chennai'
```

In [197]:

```
dict2
```

Out[197]:

```
{'name': 'rohit', 'age': 25, 'location': 'chennai'}
```

In [198]:

```
dict2['location']='mumbai'
```

In [199]:

```
dict2
```

Out[199]:

```
{'name': 'rohit', 'age': 25, 'location': 'mumbai'}
```

In [200]:

```
dict3={1:'one',2:'two','A':['shankar','balaji','girish']}
```

In [201]:

```
dict3
```

Out[201]:

```
{1: 'one', 2: 'two', 'A': ['shankar', 'balaji', 'girish']}
```

In [202]:

```
dict3['A'][1][2]
```

Out[202]:

```
'l'
```

In [203]:

```
dict4={0:{'one':1,'two':2},1:{'city':'banglore','area':'whitefield'},'three':['Mumbai','P
```

In [204]:

```
dict4
```

Out[204]:

```
{0: {'one': 1, 'two': 2},
 1: {'city': 'banglore', 'area': 'whitefield'},
 'three': ['Mumbai', 'Pune']}
```

In [205]:

```
dict4[1]['area']
```

Out[205]:

```
'whitefield'
```

In [206]:

```
dict4[0]['two']
```

Out[206]:

```
2
```

In [207]:

```
dict4['three'][0]
```

Out[207]:

```
'Mumbai'
```

In [208]:

```
dict4['three'][1][2]
```

Out[208]:

```
'n'
```

In [209]:

```
dict4[1]['city']
```

Out[209]:

```
'banglore'
```

In [210]:

```
dict4.pop('three')
```

Out[210]:

```
['Mumbai', 'Pune']
```

Conditional Statement

In [211]:

```
#if
#if---else
#if--elif--elif....
#nested if
```

In [212]:

```
x=10
if x==10:              #true
    print('matched')
print('hello')
```

```
matched
hello
```

In [213]:

```
x=20
if x==10:               #false
    print('matched')
print('hello')
```

```
hello
```

In [214]:

```
y=15
if y>20:
    print('welcome')
print('python')
```

```
python
```

In [215]:

```
y=15
if y<20:
    print('welcome')
print('python')
```

```
welcome
python
```

In [216]:

```
#write down the code as a output "logged in" when password is 12345
```

In [217]:

```
y=12345
if y==12345:
    print('logged in')
```

```
logged in
```

In [218]:

```
x=10
y=20
print('hello')
if(x+y==30 and x<y):    #true
    print('yes')
else:
    print('no')
print('welcome')
```

```
hello
yes
welcome
```

In [219]:

```
x=13
if x%2==0:              #false
    print('x is even')
else:
    print('x is odd')
```

```
x is odd
```

In [220]:

```
y=12
if y%3==0:               #cond true
    print('divisible')
else:
    print('not divisible')
```

```
divisible
```

In [221]:

```
x=2
if x==1:                #false
    print('python')
elif x==2:              #true
    print('jupyter')
elif x==3:
    print('ML')
else:
    print("AI")
```

```
jupyter
```

In [222]:

```
x=10
if x==1:
    print('python')
elif x==2:
    print('jupyter')
elif x==3:
    print('ML')
else:
    print("AI")
```

```
AI
```

In [223]:

```
light_color='pink'
if light_color=='red':
    print('stop')
elif light_color=='green':
    print('go')
elif light_color=='orange':
    print('slow')
else:
    print('invalid color')
```

```
invalid color
```

In [224]:

```
light_color='green'
x=10
if light_color=='red':
    if x==10:
        print('hello')
    print('stop')
elif light_color=='green':
    print('go')
elif light_color=='orange':
    print('slow')
else:
    print('invalid color')
```

```
go
```

In [225]:

```
light_color='red'
x=11
if light_color=='red':
    if x==10:
        print('hello')
    print('stop')
elif light_color=='green':
    print('go')
elif light_color=='orange':
    print('slow')
else:
    print('invalid color')
```

```
stop
```

input method

In [*]:

```
light_color=input("Enter the color of the traffic light(red/green/orange): ")

if light_color=='red':
    print('stop')
elif light_color=='green':
    print('go')
elif light_color=='orange':
    print('slow')
else:
    print('invalid color')
```

In [ ]:

```

```