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 [2]:

```
false=10
```

In [3]:

```
false
```

Out[3]:

```
10
```

identifier

name given to entity like class,functions,variables etc

In [4]:

```
1var=20
```

```
  Input In [4]
    1var=20
    ^
SyntaxError: invalid decimal literal
```

In [ ]:

```
var1=20
```

In [ ]:

```
var1
```

In [ ]:

```
var@=30
```

In [ ]:

```
var_=30
```

In [ ]:

```
var_
```

In [ ]:

```
finally=40
```

In [ ]:

```
Finally=40
```

In [ ]:

```
Finally
```

In [ ]:

```
from=50
```

In [ ]:

```
From=50
```

In [ ]:

```
From
```

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

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

statement

In [ ]:

```
val5=10
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

indentation

In [ ]:

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

docstrings

In [ ]:

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

In [ ]:

```
square(2)
```

In [ ]:

```
square.__doc__
```

variables

In [ ]:

```
a=5
```

In [ ]:

```
a
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

data type

int float bool/boolean complex

In [ ]:

```
a=1254789631422
```

In [ ]:

```
type(a)
```

In [ ]:

```
b=10.2
b
```

In [ ]:

```
type(b)
```

In [ ]:

```
bool1=True
```

In [ ]:

```
bool1
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
type(x)
```

Strings

In [ ]:

```
str1="hello"
```

In [ ]:

```
type(str1)
```

In [ ]:

```
len(str1)
```

In [ ]:

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

In [ ]:

```
str2[0]
```

In [ ]:
```
len(str2)
```

indexing

In [ ]:
```
str3='welcome'
```

in python we always start indexing from 0

In [ ]:
```
str3[0]
```

In [ ]:
```
str3[-7]
```

Slicing

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

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

In [ ]:
```
str3[0:]
```

In [ ]:
```
str3[3:]
```

In [ ]:
```
str4="bangalore"
```

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

In [ ]:
```
str4[3:]
```

In [ ]:

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

In [ ]:

```
str5[5:]
```

In [ ]:

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

In [ ]:

```
str6='heloo'
```

In [ ]:

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

In [ ]:

```
del str6
```

In [ ]:

```
str6
```

string concatenation

In [ ]:

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

In [ ]:

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

In [ ]:

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

String Membership

In [ ]:

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

In [ ]:

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

In [ ]:
```
print('python' in mystr)
```

In [ ]:
```
print(' eve' in mystr)
```

In [ ]:
```
print('-eve' in mystr)
```

string partitioning

In [ ]:
```
mystr1='natural language with python and R and java'
mystr1
```

In [ ]:
```
l=mystr1.partition('with')
l
```

In [ ]:
```
mystr1.capitalize()
```

In [ ]:
```
mystr1.upper()
```

In [ ]:
```
mystr1.lower()
```

In [ ]:
```
mystr1.count('a')
```

In [ ]:
```
mystr2="   hello   "
mystr2
```

In [ ]:
```
mystr2.strip()
```

In [ ]:
```
mystr2.rstrip()
```

In [ ]:

```
mystr2.lstrip()
```

21 july 2023

In [16]:

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

```
2225333912944
```

In [17]:

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

```
2225313591472
```

In [12]:

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

In [18]:

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

Out[18]:

```
'beautiful morning'
```

In [19]:

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

In [20]:

```
str8
```

Out[20]:

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

In [21]:

```
str8.split()
```

Out[21]:

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

variable assignment

In [22]:

```
x=10
```

In [23]:

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

In [24]:

```
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 [26]:

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

```
12
8
20
```

In [27]:

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

```
7.5
7
1
```

In [28]:

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

```
4
```

Assignment

In [35]:

```
x=5
```

In [37]:

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

In [38]:

```
print(y)
```

```
12
```

In [39]:

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

In [40]:

```
print(z)
```

```
5
```

comparision /relational

== Equal

!= not equal

#> greater than

< less than

<= less than equal to

#>= greater than equal to

In [42]:

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

```
False
True
False
```

Logical operators

and

or

not

In [47]:

```
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 [49]:

```
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_13680\436760226.py:1: SyntaxW
arning: "is" with a literal. Did you mean "=="?
  print((10) is (10.0))
```

In [52]:

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

```
hello
welcome

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [52], 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 [55]:

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

In [54]:

```
type(a)
```

Out[54]:

```
str
```

In [56]:

```
type(b)
```

Out[56]:

```
int
```

Auto type casting

Forced Type casting

In [57]:

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

Out[57]:

```
7.23
```

In [58]:

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

Out[58]:

```
6.3
```

In [59]:

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

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [59], in <cell line: 1>()
----> 1 2+3+'data'+4.5

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

In [60]:

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

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Input In [60], in <cell line: 1>()
----> 1 12+15+'10'+2

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

In [61]:

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

```
1
```

In [64]:

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

```
1
```

In [65]:

```
a='10'
```

In [66]:

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

```
10
```

In [71]:

```
b='data123'
```

In [72]:

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

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Input In [72], in <cell line: 1>()
----> 1 print(int(b))

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

In [69]:

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

```
3
```

In [70]:

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

```
3.0
```

In [73]:

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

```
0.0
```

In [ ]:

```
```