

```

#identifiers
'''
    any name in python is called identifiers.
    the name may be variable name, class name or function name.
'''

#keywords
'''
    there are some words in python to represent some functionality such type of words
    are
    called reserved words

    there are 35 keywords available in python

    ['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']

'''

import keyword
print(keyword.kwlist)

#comments in python

# single line comment (#!/ =control+/)
#multiline comments
'''
df,djsc
csjaikfjq
ukyk bxurwq
'''

#some in built function
#1)print() -o print something
#2)id()-to check address of something
#3)type()- to check type of variable

#variable
'''
    tag given to a memory location
'''

```

```

x=2    #int
y=2    #float
print(id(x))
print(id(y))
name='rahul'
name1="rahul"
print(id(name))
print(id(name1))

print(type(name))
print(type(x))

#data types- datatype represent type of data inside variable.
#1)integer- used to store whole value/ integer value
x=2
print(x)
print(type(x))

#2)float=used to store floating value / floating point number
y=2.4
print(y)
print(type(y))

#3)complex-to store complex value i.e is in the form of real part and imaginary
part

z=2+3j
print(z)
print(type(z))
print(z.imag)
print(z.real)

#4)boolean value=true or false
#note:- in python true is considered as 1 and false is considered as 0
a=True
b=False
print(a)
print(b)
print(type(a))
print(type(b))
print(a+b)
print(True+True)
print(True+False)

```

#5)string (str)-collection of char inside single quote or double quote are called string

```
name='lalit'
name1="lalit"
print(name)
print(name1)
print(type(name))
print(type(name1))
```

#string indexing= positive as well as negative indexing possible

```
print('#####')
```

```
s='today is good day to learn python'
```

```
print(s[0])
```

```
print(s[1])
```

```
print(s[2])
```

```
print(s[3])
```

```
print(s[4])
```

```
print(s[5])
```

```
print(s[6])
```

```
print(s[7])
```

```
print(s[-1])
```

#string slicing=piasece of sometyhing

```
'''
```

```
str[startindex:stop index:steps]
```

```
'''
```

```
s='today is good day to learn python'
```

```
print(s[0:4:1])
```

```
print(s[-7::1])
```

#reverse stringh

```
print(s[::-1])
```

```
#####
```

#all fundcamental data types are immutable

```
'''
```

```
int
```

```
float
```

```
complex
```

```
str
```

```
bool
```

```
'''
```

#immutable=where we can not make any changes if changes make in the immutable

```

#object then whole object changes

#mutable=the data type whwere we can perform changes .

#####
#6)list
'''
order
sequence of charecter inside square brackect qumma sqperated.
mutable
hetrogeneouse/homogeneous data store
duplicate allowed
indexing/slicing possible
item ASSIGNMENT POSSIBLE
'''
l=[1,2,23,3,4,'rahul',2]
print(id(l))
l[0]=5
print(l)
print(id(l))

#indexing /slicing possible
l=[1,2,23,3,4,'rahul',2]
print(l[0])
print(l[0:4:])

#item assignment/assign value to index possible
l=[1,2,23,3,4,'rahul',2]
l[5]='sagar' #item assignemnt
print(l)

#####
#7)tuple=read only version of list
'''
order
immutable
item assignment not possible
indexing/ slicing possible
duplicates are allowed
home/hetro possible
'''
t=(1,2,23,3,4,'amol',2)
print(type(t))
print(t)
#item assignemnt not possible

```

```

#t[0]=56    #tuple' object does not support item assignment

#indexing and slicing possible
t=(1,2,3,4,4,5)
print(t[0])
print(t[1])
print(t[0:4])

#####
#8)set
'''
mutable
unordered collection of character inside curly braces
indexing/ slicing not possible
duplicates not allowed
homo/hetro possible
item assignment not possible
'''
s={1,2,3,4,4,'amol'}
print(s)
#print(s[0])  #'set' object is not subscriptable
#####
#9)dictionary/mapping/key-value pair
'''
mutable
key of dict is immutable and value can be any thing i.e mutable or immutable
#key can not be repeated or duplicate
access value using key
'''
d={(1,2,3,43):'rahul',2:'amol',3:'sagar'}
print(d[2])

#####
#####
#frozenset
#byte
#bytearray
#####

```