Python

Python is an object-oriented, high-level, interpreted, general purpose and dynamic programming language.

It was developed by Guido van Rossum in 1991.

The name 'Python' came from an old BBC television comedy sketch series "Monty's Python Flying Circus".

In [1]:
```
#Simple python program
print("Hello World")
```

```
Hello World
```

In [2]:
```
print("Debashree Sahoo")
```

```
Debashree Sahoo
```

Keywords : Reserved Words (pre-defined words)

In [3]:
```
#to get python keywords
help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

In [4]:
```
help('if')
```

```
The "if" statement
******************

The "if" statement is used for conditional execution:

   if_stmt ::= "if" assignment_expression ":" suite
               ("elif" assignment_expression ":" suite)*
               ["else" ":" suite]

It selects exactly one of the suites by evaluating the expressions one
by one until one is found to be true (see section Boolean operations
for the definition of true and false); then that suite is executed
(and no other part of the "if" statement is executed or evaluated).
If all expressions are false, the suite of the "else" clause, if
present, is executed.

Related help topics: TRUTHVALUE
```

Variable : variables are the names given to the memory location to store a value or it is a container to store a value

We can use upper case letters(A-Z), lower case letters (a-z), digits(0-9)

We can't use special symbols except underscore(_)

We can't use keywords

We can't start with a digit

We can't use space in between the variable name instead we can use underscore.

Python is case sensitive

In [5]:
```
#we cant't use keyword as variable name
for = 10
```

```
  Cell In[5], line 2
    for = 10
        ^
SyntaxError: invalid syntax
```

In [6]:
```
#we can't start with a digit
123abc = 20
```

```
  Cell In[6], line 2
    123abc = 20
      ^
SyntaxError: invalid decimal literal
```

In [7]:
```
#we can't use space in between the variable name instead we can use underscore.
var abc = 30
```

```
  Cell In[7], line 2
    var abc = 30
        ^
SyntaxError: invalid syntax
```

In [8]:
```
#we can't use special symbols except underscore(_)
@!%^ = 40
```

```
  Cell In[8], line 2
    @!%^ = 40
     ^
SyntaxError: invalid syntax
```

In [10]:
```
#we can use upper case letters(A-Z), lower case letters (a-z), digits(0-9)
VAR = 50
print(VAR)
```

```
50
```

In [11]:
```
var = 60
print(var)
```

```
60
```

In [12]:
```
Var_123 = 100
print(Var_123)
```

```
100
```

In [ ]:
```

```

Datatypes

Primary

- **Numeric :**
  - Integer - int
  - Float - float
  - Complex - complex
- **Boolean(bool) :**
  - True
  - False

Sequential

- String - str
- List - list
- Tuple - tuple
- Set - set
- Dictionary - dict

In [13]:
```
#int
a = 10
print(a)
```

```
10
```

In [14]:
```
type(a)
```

Out[14]: int

In [15]:
```
#float
f = 2.5
print(f)
```

```
2.5
```

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

Out[16]: float

In [17]:
```
#complex
c = 2+5i
print(c)
```

```
  Cell In[17], line 2
    c = 2+5i
          ^
SyntaxError: invalid decimal literal
```

In [18]:
```
c = 2+5j
print(c)
```

```
(2+5j)
```

In [19]:
```
type(c)
```

Out[19]: complex

In [20]:
```
#Boolean
a = True
print(a)
```

```
True
```

In [21]:
```
type(a)
```

Out[21]: bool

In [ ]:

String (str) : sequence of characters stored inside quotation marks

In [22]:
```
a = 'Hello Everyone'
print(a)
```

```
Hello Everyone
```

In [23]:
```
type(a)
```

Out[23]: str

In [24]:
```
b = 'abcd1234@!#$😋😋'
type(b)
```

Out[24]: str

In [ ]:

In [27]:
```
#single line string denoted by ' ' or " "
a = 'single line string'
b = "single line string"

print(a)
print(b)
```

```
single line string
single line string
```

In [26]:
```
type(a),type(b)
```

Out[26]: (str, str)

In [28]:
```
#multi line string is denoted with ''' ''' or """ """
a = '''this is multi
line string'''
b = """this is a
multi line string"""

print(a)
print(b)
```

```
this is multi
line string
this is a
multi line string
```

In [ ]:

In [29]:
```
a = 'Tasty Apple'
print(a)
```

```
Tasty Apple
```

In [30]:
```
type(a)
```

Out[30]: str

In [31]:
```
#indexing
#Syntax : var[index_pos]
a[4]
```

Out[31]: 'y'

In [32]:
```
a[-7]
```

Out[32]: 'y'

In [33]:
```
#slicing
#Forward Slicing
#syntax = var[start:stop+1:step]

#'sty A'
a[2:7:1]
```

Out[33]: 'sty A'

In [34]:
```
a[-9:-4:1]
```

Out[34]: 'sty A'

In [35]:
```
#"TsyApe"
a[0:11:2]
```

Out[35]: 'TsyApe'

In [39]:
```
a[-11:11:2]
```

Out[39]: 'TsyApe'

In [ ]:
```

```

In [40]:
```
#"Ttal"
a[0:10:3]
```

Out[40]: 'TtAl'

In [41]:
```
a[-11:-1:3]
```

Out[41]: 'TtAl'

In [42]:
```
a[::]
```

Out[42]: 'Tasty Apple'

In [43]:
```
a[::-1]
```

Out[43]: 'elppA ytsaT'

In [44]:
```
#Backward SLicing
#syntax = var[start:stop-1:step]

#"A yts"
a[6:1:-1]
```

Out[44]: 'A yts'

In [47]:
```
a[-5:-10:-1]
```

Out[47]: 'A yts'

In [ ]:
```

```

String Functions

```
capitalize() : converts the first character to upper case and the
rest lower
case.

upper()      : converts all the characters into upper case

lower()      : converts all the characters into lower case

casefold()   : converts all the characters into lower case

center()     : returns a centered string

count()      : it counts how many times a particular character is
repeated

index()      : returns index position of a particular character

strip()      : removes spaces present before and after the string
```

In [48]:
```
a = "heLlO eVeRYoNE"
print(a)
type(a)
```

```
heLlO eVeRYoNE
```

Out[48]: str

In [49]:
```
a.capitalize()
```

Out[49]: 'Hello everyone'

In [50]:
```
a.upper()
```

Out[50]: 'HELLO EVERYONE'

In [51]:
```
a.lower()
```

Out[51]: 'hello everyone'

In [52]:
```
a.casefold()
```

Out[52]: 'hello everyone'

In [54]:
```
a.swapcase()
```

Out[54]: 'HElLo EvEryOne'

In [55]:
```
a.title()
```

Out[55]: 'Hello Everyone'

In [56]:
```
a.center(50)
```

Out[56]: '   heLlO eVeRYoNE   '

In [59]:
```
a.center(50,'😳')
```

Out[59]: '😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳heLlO eVeRYoNE😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳😳'

In [60]:
```
a.index('e')
```

Out[60]: 1

In [61]:
```
a.count('e')
```

Out[61]: 3

In [64]:
```
'😃'.join(['john','gmail.com','Nizam'])
```

Out[64]: 'john😃gmail.com😃Nizam'

In [65]:
```
'Ayushman Shreyansh Gobo'.split()
```

Out[65]: ['Ayushman', 'Shreyansh', 'Gobo']

In [73]:
```
'Ayushman_Shreyansh_Gobo'.split('_')
```

Out[73]: ['Ayushman', 'Shreyansh', 'Gobo']

In [74]:
```
'              hello              '.strip()
```

Out[74]: 'hello'

In [80]:
```
'*********%%*****hello****%%%*******'.strip('*%')
```

Out[80]: 'hello'

In [81]:
```
'              hello              '.lstrip()
```

Out[81]: 'hello   '

In [82]:
```
'              hello              '.rstrip()
```

Out[82]: '   hello'

In [83]:
```
'abc'.isalpha()
```

Out[83]: True

In [84]:
```
'ABC'.isupper()
```

Out[84]: True

In [86]:
```
'abc'.islower()
```

Out[86]: True

In [87]:
```
'1234'.isdigit()
```

Out[87]: True

In [88]:
```
'1234'.isnumeric()
```

Out[88]: True

In [90]:
```
'abc123'.isalnum()
```

Out[90]: True

In [91]:
```
'The Fairy Tale'.istitle()
```

Out[91]: True

In [92]:
```
'var 123'.isidentifier()
```

Out[92]: False

In [93]:
```
'Var_123'.isidentifier()
```

Out[93]: True

In [ ]:

Calendar 🗓️

In [85]:
```
import calendar
print(calendar.calendar(2025))
```

```
                                  2025

      January               February               March
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
       1  2  3  4  5                  1  2                  1  2
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   3  4  5  6  7  8  9
13 14 15 16 17 18 19  10 11 12 13 14 15 16  10 11 12 13 14 15 16
20 21 22 23 24 25 26  17 18 19 20 21 22 23  17 18 19 20 21 22 23
27 28 29 30 31        24 25 26 27 28        24 25 26 27 28 29 30
                                            31

      April                 May                   June
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6               1  2  3  4                  1
 7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20  12 13 14 15 16 17 18   9 10 11 12 13 14 15
21 22 23 24 25 26 27  19 20 21 22 23 24 25  16 17 18 19 20 21 22
28 29 30              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

       July                 August              September
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6               1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17  15 16 17 18 19 20 21
21 22 23 24 25 26 27  18 19 20 21 22 23 24  22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31  29 30

      October              November              December
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
       1  2  3  4  5                  1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31
```

In [ ]:

LIST (list)

Sequence of items denoted with '[]' and separated with ','.

It is ordered(indexed with specific position), mutable(items can be changed) and heterogenous(supports multiple data-type).

In [3]:
```
l = [20,2.5,'cat',True]
print(l)
```

```
[20, 2.5, 'cat', True]
```

In [4]:
```
type(l)
```

Out[4]: list

In [5]:
```
#indexing syntax : var[index]
l[2]
```

Out[5]: 'cat'

In [6]:
```
l[-2]
```

Out[6]: 'cat'

In [7]:
```
#slicing syntax : var[start : stop+1 : step]
l[0:3:2]
```

Out[7]: [20, 'cat']

In [8]:
```
l[-4:-1:2]
```

Out[8]: [20, 'cat']

In [9]:
```
l[::]
```

Out[9]: [20, 2.5, 'cat', True]

In [10]:
```
l[::-1]
```

Out[10]: [True, 'cat', 2.5, 20]

In [ ]:
```

```

List Methods

Adding Items to a List

- var.append(obj) : add item to the end of the list

- var.insert(ind,obj) : add item to the given index of the list
- var.extend([sequence]) : add items from the given sequence at the end of the list

Removing Items from the List

- var.remove(obj) : removes the first matched instance of object from the list
- var.pop(index) : it pop or delete the item of the given index from the list
- var.clear() : empty the list

Miscellaneous methods

- var.copy() : copy the item
- var.count(obj)
- var.index(obj)
- var.sort() : sort or arrange in ascending order
- var.sort(reverse=True) : arrange in descending order
- var.reverse() or var[: : -1] can be use to reverse the list

In [11]:
```
l = [20,56,78,45,56,50]
print(l)
type(l)
```

```
[20, 56, 78, 45, 56, 50]
```

Out[11]: list

In [12]:
```
#adding items to a list
#append
print(l)
l.append(100)
print(l)
```

```
[20, 56, 78, 45, 56, 50]
[20, 56, 78, 45, 56, 50, 100]
```

In [13]:
```
#insert
print(l)
l.insert(2,60)
print(l)
```

```
[20, 56, 78, 45, 56, 50, 100]
[20, 56, 60, 78, 45, 56, 50, 100]
```

In [14]:
```
#extend
print(l)
l.extend([10,20,30])
print(l)
```

```
[20, 56, 60, 78, 45, 56, 50, 100]
[20, 56, 60, 78, 45, 56, 50, 100, 10, 20, 30]
```

In [15]:
```
l
```

Out[15]: [20, 56, 60, 78, 45, 56, 50, 100, 10, 20, 30]

In [16]:
```
#removing elements from a list
#pop
print(l)
l.pop()
print(l)
```

```
[20, 56, 60, 78, 45, 56, 50, 100, 10, 20, 30]
[20, 56, 60, 78, 45, 56, 50, 100, 10, 20]
```

In [17]:
```
print(l)
l.pop(2)
print(l)
```

```
[20, 56, 60, 78, 45, 56, 50, 100, 10, 20]
[20, 56, 78, 45, 56, 50, 100, 10, 20]
```

In [18]:
```
#remove
print(l)
l.remove(56)
print(l)
```

```
[20, 56, 78, 45, 56, 50, 100, 10, 20]
[20, 78, 45, 56, 50, 100, 10, 20]
```

In [19]:
```
#Miscellaneous methods in list
#copy
print(l)
m = l.copy()
print(m)
```

```
[20, 78, 45, 56, 50, 100, 10, 20]
[20, 78, 45, 56, 50, 100, 10, 20]
```

In [20]:
```
#count
l.count(20)
```

Out[20]: 2

In [21]:
```
#index
l.index(50)
```

Out[21]: 4

In [22]:
```
#sort - ascending order
print(l)
l.sort()
print(l)
```

```
[20, 78, 45, 56, 50, 100, 10, 20]
[10, 20, 20, 45, 50, 56, 78, 100]
```

In [23]:
```
print(l)
l.sort(reverse=True)  #descending order
print(l)
```

```
[10, 20, 20, 45, 50, 56, 78, 100]
[100, 78, 56, 50, 45, 20, 20, 10]
```

In [27]:
```
#reverse
a = [34,78,45,10]
print(a)
a.reverse()
print(a)
```

```
[34, 78, 45, 10]
[10, 45, 78, 34]
```

In [26]:
```
#clear
print(l)
l.clear()
print(l)
```

```
[100, 78, 56, 50, 45, 20, 20, 10]
[]
```

In [ ]:

Tuple (tuple)

An ordered but immutable(which cannot change size or permanent) collection of items.

It is denoted by parenthsis '()' and separated by ','.

In [28]:
```
t = (10,3.5,'dog',False)
print(t)
```

```
(10, 3.5, 'dog', False)
```

In [29]:
```
type(t)
```

Out[29]: tuple

In [30]:
```
#indexing
t[3]
```

Out[30]: False

In [31]:
```
#slicing
t[::]
```

Out[31]: (10, 3.5, 'dog', False)

In [32]:
```
t[::-1]
```

Out[32]: (False, 'dog', 3.5, 10)

Tuple Methods

In [33]:
```
#count
t.count(10)
```

Out[33]: 1

In [34]:
```
#index
t.index('dog')
```

Out[34]: 2

In [ ]:

In [ ]:

Set (set)

Unordered but mutable collection of unique items.

Denoted with curly brackets '{}' and separated by ','.

In [35]:
```
s = {20,'cat',50,4.5,True}
print(s)
```

```
{'cat', True, 50, 4.5, 20}
```

In [36]:
```
type(s)
```

Out[36]: set

In [37]:
```
a = {1,1,1,6,6,6,6,9,9,2,2,2}
print(a)
```

```
{1, 2, 6, 9}
```

In [ ]:

set methods

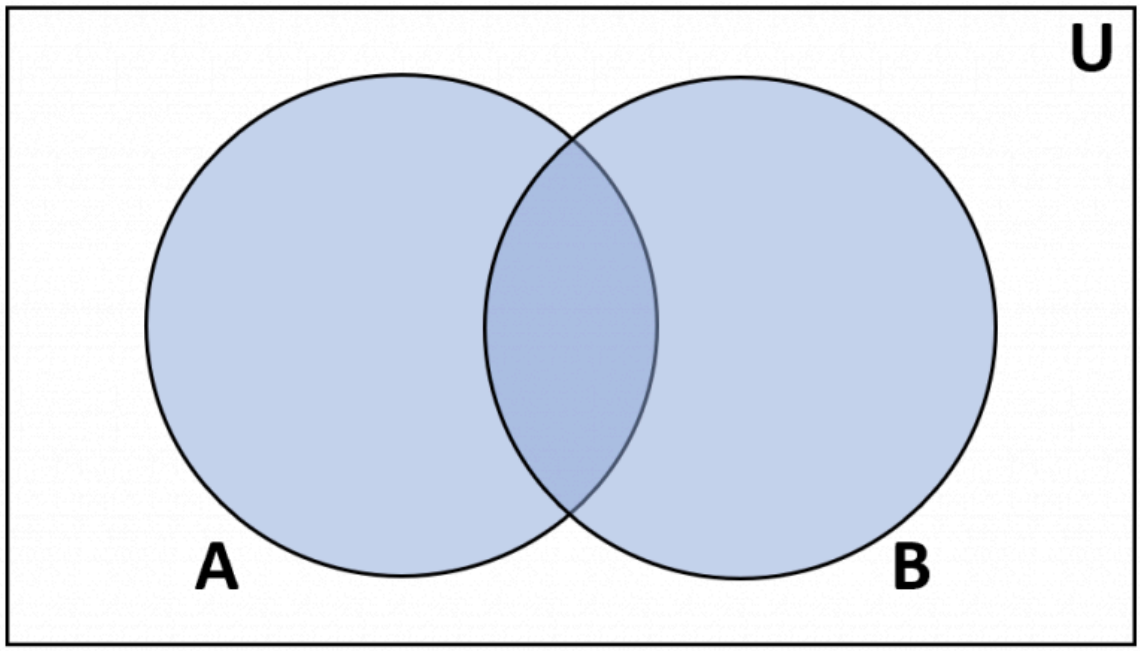
In [39]:
```
s
```

Out[39]: {20, 4.5, 50, True, 'cat'}

In [40]:
```
#add
print(s)
s.add(100)
print(s)
```

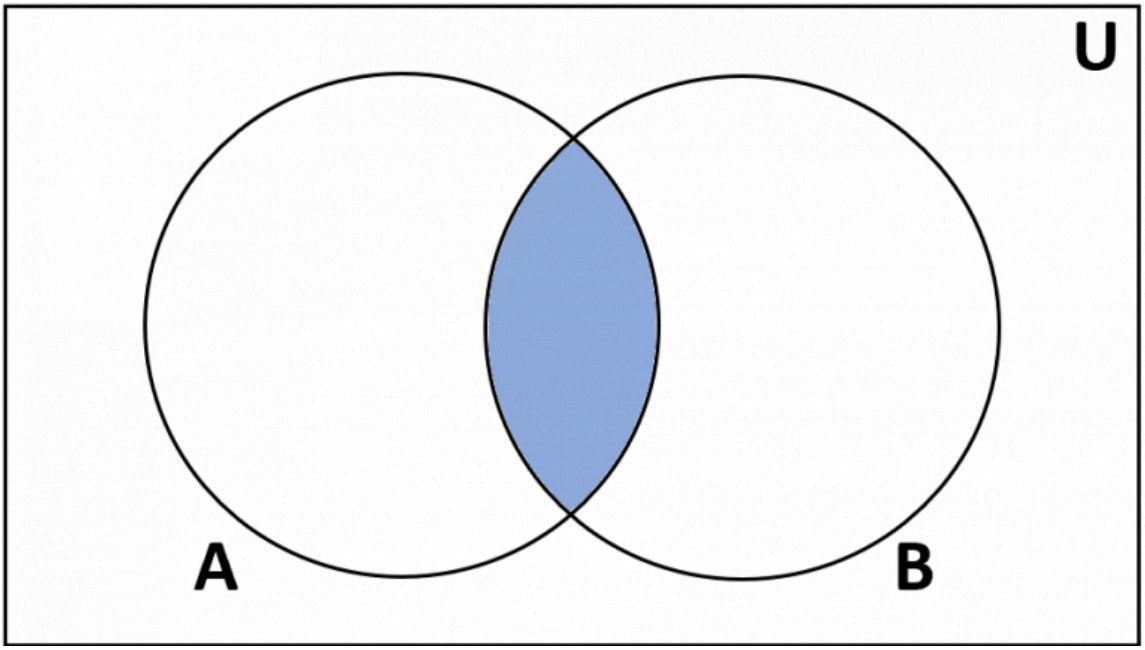
```
{'cat', True, 50, 4.5, 20}
{'cat', True, 50, 4.5, 20, 100}
```

In [41]:
```
a = {1,4,2,6,8,9}
b = {2,4,7}
```



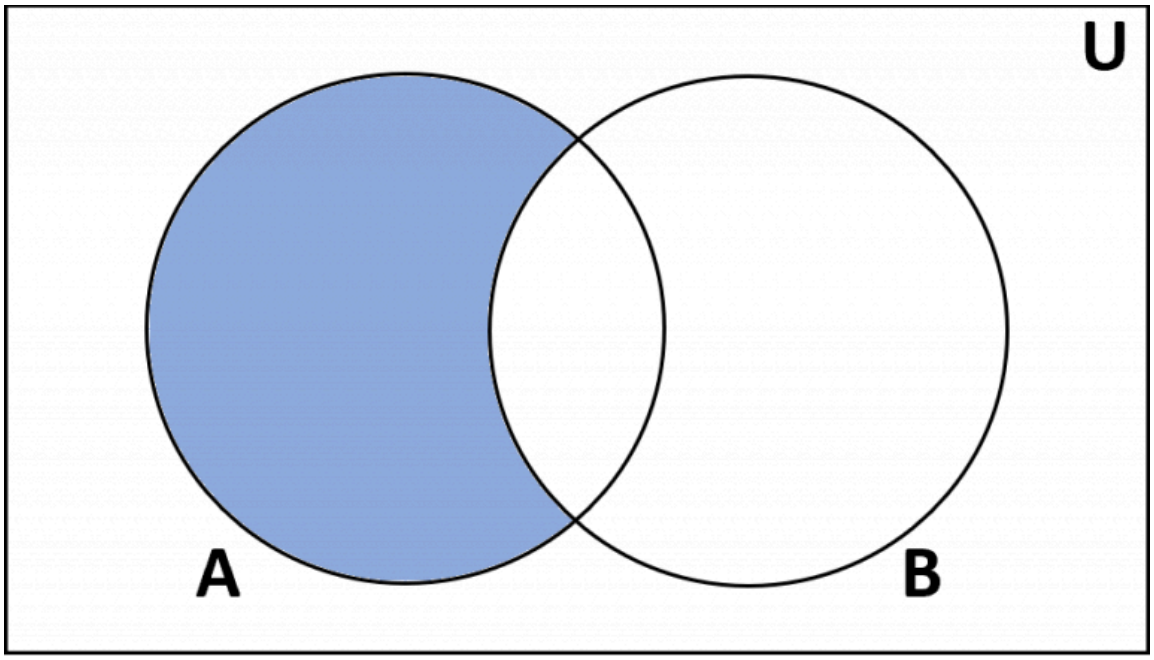
In [42]:
```
#union
a.union(b)
```

Out[42]:
```
{1, 2, 4, 6, 7, 8, 9}
```



In [43]:
```
#intersection
a.intersection(b)
```

Out[43]: {2, 4}



In [44]:
```
#difference
a.difference(b)  #A-B
```

Out[44]: {1, 6, 8, 9}

In [45]:
```
b.difference(a)  #B-A
```

Out[45]: {7}

In [46]:
```
#issuperset
a.issuperset(b)
```

Out[46]: False

In [47]:
```
#issubset
b.issubset(a)
```

Out[47]: False

In [48]:
```
#isdisjoint
a.isdisjoint(b)
```

Out[48]: False

In [50]:
```
s
```

Out[50]: {100, 20, 4.5, 50, True, 'cat'}

In [51]:
```
#pop
print(s)
```

```
s.pop()
print(s)
```

```
{'cat', True, 50, 4.5, 20, 100}
{True, 50, 4.5, 20, 100}
```

In [52]:
```
print(s)
s.pop()
print(s)
```

```
{True, 50, 4.5, 20, 100}
{50, 4.5, 20, 100}
```

In [ ]:
```

```

In [60]:
```
#remove
print(s)
s.remove(100)
print(s)
```

```
{50, 4.5, 20, 100}
{50, 4.5, 20}
```

In [61]:
```
#clear
print(s)
s.clear()
print(s)
```

```
{50, 4.5, 20}
set()
```

Dictionary (dict)

An ordered, mutable collection of key and value pairs.

Denoted with curly brackets '{}' and key must be unique.

key and value pairs are separated with comma and key and value are separated with colon

In [66]:
```
d = {"Name":"Sunny","Roll":25,"Course":"AI"}
print(d)
```

```
{'Name': 'Sunny', 'Roll': 25, 'Course': 'AI'}
```

In [63]:
```
type(d)
```

Out[63]:
```
dict
```

Dictionary methods

In [67]:
```
#update
print(d)
```

```
d.update({'Marks':99.9})
print(d)
```

```
{'Name': 'Sunny', 'Roll': 25, 'Course': 'AI'}
{'Name': 'Sunny', 'Roll': 25, 'Course': 'AI', 'Marks': 99.9}
```

In [68]:
```
print(d)
d.update({'Course':'Python'})
print(d)
```

```
{'Name': 'Sunny', 'Roll': 25, 'Course': 'AI', 'Marks': 99.9}
{'Name': 'Sunny', 'Roll': 25, 'Course': 'Python', 'Marks': 99.9}
```

In [69]:
```
# keys()
d.keys()
```

Out[69]:
```
dict_keys(['Name', 'Roll', 'Course', 'Marks'])
```

In [70]:
```
# values()
d.values()
```

Out[70]:
```
dict_values(['Sunny', 25, 'Python', 99.9])
```

In [71]:
```
# items()
d.items()
```

Out[71]:
```
dict_items([('Name', 'Sunny'), ('Roll', 25), ('Course', 'Python'), ('Marks', 99.
9)])
```

In [72]:
```
# get()
d.get('Name')
```

Out[72]:
```
'Sunny'
```

In [73]:
```
d.get('Roll')
```

Out[73]:
```
25
```

In [74]:
```
#popitem
print(d)
d.popitem()
print(d)
```

```
{'Name': 'Sunny', 'Roll': 25, 'Course': 'Python', 'Marks': 99.9}
{'Name': 'Sunny', 'Roll': 25, 'Course': 'Python'}
```

In [75]:
```
#pop
print(d)
d.pop('Roll')
print(d)
```

```
{'Name': 'Sunny', 'Roll': 25, 'Course': 'Python'}
{'Name': 'Sunny', 'Course': 'Python'}
```

In [76]:
```
#clear
print(d)
```

```
d.clear()
print(d)
```

```
{'Name': 'Sunny', 'Course': 'Python'}
{}
```

In [ ]:

pywhatkit

In [77]:
```
#to install pywhatkit
!pip install pywhatkit
```

```
Collecting pywhatkit
  Obtaining dependency information for pywhatkit from https://files.pythonhosted.or
g/packages/8c/58/e06d722ddaf7100765c5d0a34caf8d9fe92202bfabe632a9dea572312901/pywhat
kit-5.4-py3-none-any.whl.metadata
  Downloading pywhatkit-5.4-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: Pillow in c:\users\lab25\anaconda3\lib\site-packages
(from pywhatkit) (10.2.0)
Collecting pyautogui (from pywhatkit)
  Downloading PyAutoGUI-0.9.54.tar.gz (61 kB)
     ---------------------------------------- 0.0/61.2 kB ? eta -:--:--
     ---------------------------------------- 0.0/61.2 kB ? eta -:--:--
     ------ --------------------------------- 10.2/61.2 kB ? eta -:--:--
     ------------------------------- ------ 51.2/61.2 kB 525.1 kB/s eta 0:00:01
     ------------------------------- ------ 51.2/61.2 kB 525.1 kB/s eta 0:00:01
     ---------------------------------------- 61.2/61.2 kB 364.3 kB/s eta 0:00:00
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: requests in c:\users\lab25\anaconda3\lib\site-package
s (from pywhatkit) (2.31.0)
Collecting wikipedia (from pywhatkit)
  Downloading wikipedia-1.4.0.tar.gz (27 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: Flask in c:\users\lab25\anaconda3\lib\site-packages
(from pywhatkit) (2.2.5)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\lab25\anaconda3\lib\site-
packages (from Flask->pywhatkit) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\lab25\anaconda3\lib\site-pack
ages (from Flask->pywhatkit) (3.1.3)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\lab25\anaconda3\lib\sit
e-packages (from Flask->pywhatkit) (2.0.1)
Requirement already satisfied: click>=8.0 in c:\users\lab25\anaconda3\lib\site-packa
ges (from Flask->pywhatkit) (8.1.7)
Collecting pymsgbox (from pyautogui->pywhatkit)
  Downloading PyMsgBox-1.0.9.tar.gz (18 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Collecting pytweening>=1.0.4 (from pyautogui->pywhatkit)
  Downloading pytweening-1.2.0.tar.gz (171 kB)
     ---------------------------------------- 0.0/171.2 kB ? eta -:--:--
     ---------------------------------- --- 153.6/171.2 kB 4.6 MB/s eta 0:00:01
     ------------------------------------ 171.2/171.2 kB 3.4 MB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting pyscreeze>=0.1.21 (from pyautogui->pywhatkit)
  Downloading pyscreeze-1.0.1.tar.gz (27 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
```

```
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Collecting pygetwindow>=0.0.5 (from pyautogui->pywhatkit)
  Downloading PyGetWindow-0.0.9.tar.gz (9.7 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting mouseinfo (from pyautogui->pywhatkit)
  Downloading MouseInfo-0.1.3.tar.gz (10 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\lab25\anaconda3
\lib\site-packages (from requests->pywhatkit) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\lab25\anaconda3\lib\site-pac
kages (from requests->pywhatkit) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\lab25\anaconda3\lib\si
te-packages (from requests->pywhatkit) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lab25\anaconda3\lib\si
te-packages (from requests->pywhatkit) (2025.1.31)
Requirement already satisfied: beautifulsoup4 in c:\users\lab25\anaconda3\lib\site-p
ackages (from wikipedia->pywhatkit) (4.12.2)
Requirement already satisfied: colorama in c:\users\lab25\anaconda3\lib\site-package
s (from click>=8.0->Flask->pywhatkit) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\lab25\anaconda3\lib\site-
packages (from Jinja2>=3.0->Flask->pywhatkit) (2.1.3)
Collecting pyrect (from pygetwindow>=0.0.5->pyautogui->pywhatkit)
  Downloading PyRect-0.2.0.tar.gz (17 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: soupsieve>1.2 in c:\users\lab25\anaconda3\lib\site-pa
ckages (from beautifulsoup4->wikipedia->pywhatkit) (2.5)
Collecting pyperclip (from mouseinfo->pyautogui->pywhatkit)
  Downloading pyperclip-1.9.0.tar.gz (20 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Downloading pywhatkit-5.4-py3-none-any.whl (15 kB)
Building wheels for collected packages: pyautogui, wikipedia, pygetwindow, pyscreez
e, pytweening, mouseinfo, pymsgbox, pyperclip, pyrect
  Building wheel for pyautogui (pyproject.toml): started
  Building wheel for pyautogui (pyproject.toml): finished with status 'done'
  Created wheel for pyautogui: filename=pyautogui-0.9.54-py3-none-any.whl size=37706
sha256=74725d985ee09441c94e0ee62a3b37ca1e3ca4ab8ded79cd93f83e4d849c0413
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\95\dc\b1\fe122b
791e0db8bf439a0e6e1d2628e48f10bf430cae13521b
  Building wheel for wikipedia (setup.py): started
  Building wheel for wikipedia (setup.py): finished with status 'done'
  Created wheel for wikipedia: filename=wikipedia-1.4.0-py3-none-any.whl size=11707
sha256=5c3cf84ba3e90f9fef82637783dcac4df90465f991332bfed07fca34205bad89
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\8f\ab\cb\45ccc4
0522d3a1c41e1d2ad53b8f33a62f394011ec38cd71c6
  Building wheel for pygetwindow (setup.py): started
  Building wheel for pygetwindow (setup.py): finished with status 'done'
  Created wheel for pygetwindow: filename=PyGetWindow-0.0.9-py3-none-any.whl size=11
079 sha256=4a845812f2cefd830f4d7f732d33bb51984ce7e7be98a1de40b1c858d92b1392
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\07\75\0b\7ca0b5
```

```
98eb4c21d43ba4bcc78a0538dfcf803a5997da33bc19
  Building wheel for pyscreeze (pyproject.toml): started
  Building wheel for pyscreeze (pyproject.toml): finished with status 'done'
  Created wheel for pyscreeze: filename=pyscreeze-1.0.1-py3-none-any.whl size=14480
sha256=e0e8679fe7de06ef17e0df2f155d4f31e5b027374ba73d61e0f89354a20c37d3
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\cd\e3\dd\267b39
3d8e8f607e47194942740d080d9bfd835cd4375a3de1
  Building wheel for pytweening (setup.py): started
  Building wheel for pytweening (setup.py): finished with status 'done'
  Created wheel for pytweening: filename=pytweening-1.2.0-py3-none-any.whl size=8030
sha256=cb6dc5590cac6ebcf739c65c87081b4315ea8aa72020ff0798854439bc53ac86
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\db\81\dc\0d61a3
c9614f288e057ab63924e2a49edbeed4ffc916dcda1e
  Building wheel for mouseinfo (setup.py): started
  Building wheel for mouseinfo (setup.py): finished with status 'done'
  Created wheel for mouseinfo: filename=MouseInfo-0.1.3-py3-none-any.whl size=10906
sha256=1c5557227e6b03f68610fdaf9cd10530791d5ea994f4527d2bf5a8139c96c40d
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\20\0b\7f\939ac9
ff785b09951c706150537572c00123412f260a6024f3
  Building wheel for pymsgbox (pyproject.toml): started
  Building wheel for pymsgbox (pyproject.toml): finished with status 'done'
  Created wheel for pymsgbox: filename=pymsgbox-1.0.9-py3-none-any.whl size=7466 sha
256=ede5a7e0e5048241947c259f8a7f8c6a9f9155e2b01cadf53d4f4c0e23aef7a3
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\85\92\63\e126ee
5f33d8f2ed04f96e43ef5df7270a2f331848752e8662
  Building wheel for pyperclip (setup.py): started
  Building wheel for pyperclip (setup.py): finished with status 'done'
  Created wheel for pyperclip: filename=pyperclip-1.9.0-py3-none-any.whl size=11020
sha256=8a81a9bce9c94b0ce2bb7835deb1c83b717dd2843ca6e98b6a1de04026ea2215
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\e8\e7\56\591cb8
8ba1783b38c40d584026e766aac9c3a048e34128ce8b
  Building wheel for pyrect (setup.py): started
  Building wheel for pyrect (setup.py): finished with status 'done'
  Created wheel for pyrect: filename=PyRect-0.2.0-py2.py3-none-any.whl size=11205 sh
a256=be6496b38663381f05a3886bf07804f4949ed6b0b4ec193bc97b81f858004f28
  Stored in directory: c:\users\lab25\appdata\local\pip\cache\wheels\c4\e9\fc\b7a666
dd4f9a3168fb44d643079b41d36ddab52f470707e820
Successfully built pyautogui wikipedia pygetwindow pyscreeze pytweening mouseinfo py
msgbox pyperclip pyrect
Installing collected packages: pytweening, pyrect, pyperclip, pymsgbox, pyscreeze, p
ygetwindow, mouseinfo, wikipedia, pyautogui, pywhatkit
Successfully installed mouseinfo-0.1.3 pyautogui-0.9.54 pygetwindow-0.0.9 pymsgbox-
1.0.9 pyperclip-1.9.0 pyrect-0.2.0 pyscreeze-1.0.1 pytweening-1.2.0 pywhatkit-5.4 wi
kipedia-1.4.0
```

In [79]:
```python
#to import pywhatkit
import pywhatkit as kit
```

In [80]:
```python
#to search something on youtube
kit.playonyt("Tom and Jerry")
```

Out[80]:
```
'https://www.youtube.com/watch?v=t0Q2otsqC4I\\\\u0026pp=ygUNVG9tIGFuZCBKZXJyeQ%3D%
3D'
```

In [81]:
```
#to search on browser
kit.search('Python')
```

In [82]:
```
kit.info("What is Artificial Intelligence")
```

```
Artificial intelligence (AI) refers to the capability of computational systems to pe
rform tasks typically associated with human intelligence, such as learning, reasonin
g, problem-solving, perception, and decision-making. It is a field of research in co
mputer science that develops and studies methods and software that enable machines t
o perceive their environment and use learning and intelligence to take actions that
maximize their chances of achieving defined goals. Such machines may be called AIs.
```

In [ ]:

Operators : Operators are the symbols which are used to perform operations on values and variables

Types of Operators in Python :

- Arithmatic Operators
- Assignment Operators
- Comparision/Relational Operators
- Logical Operators
- Identity Operators
- Membership Operators

Arithmatic Operators

```
Arithmetic operators are used to perform mathematical operations
like addition, subtraction, multiplication, etc.
```

Operator	Operation	Example
`+`	Addition	`5 + 2 = 7`
`-`	Subtraction	`4 - 2 = 2`
`*`	Multiplication	`2 * 3 = 6`
`/`	Division	`4 / 2 = 2`
`//`	Floor Division	`10 // 3 = 3`
`%`	Modulo	`5 % 2 = 1`
`**`	Power	`4 ** 2 = 16`

In [83]:
```
a = 11
b = 2
```

In [84]:
```
#addition
a + b
```

Out[84]: 13

In [85]:
```
#subtraction
a - b
```

Out[85]: 9

In [86]:
```
#multiplication
a * b
```

Out[86]: 22

In [87]:
```
#division
a / b
```

Out[87]: 5.5

In [88]:
```
#floor division
a // b
```

Out[88]: 5

In [89]:
```
#modulo
a % b
```

Out[89]: 1

In [90]:
```
#power
2 ** 3
```

Out[90]: 8

In [91]:
```
2 ** 5 ** 3
```

Out[91]: 42535295865117307932921825928971026432

In [92]:
```
2 ** 125
```

Out[92]: 42535295865117307932921825928971026432

Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Name	Example
`=`	Assignment Operator	`a = 7`
`+=`	Addition Assignment	`a += 1 # a = a + 1`
`-=`	Subtraction Assignment	`a -= 3 # a = a - 3`
`*=`	Multiplication Assignment	`a *= 4 # a = a * 4`
`/=`	Division Assignment	`a /= 3 # a = a / 3`
`%=`	Remainder Assignment	`a %= 10 # a = a % 10`
`**=`	Exponent Assignment	`a **= 10 # a = a ** 10`

In [93]:
```
a = 10
a
```

Out[93]: 10

In [94]:
```
a += 1
a
```

Out[94]: 11

In [95]:
```
a -= 2
a
```

Out[95]: 9

In [96]:
```
a *= 3
a
```

Out[96]: 27

In [97]:
```
a /= 2
a
```

Out[97]: 13.5

In [98]:
```
a //= 2
a
```

Out[98]: 6.0

In [99]:
```
a %= 5
a
```

Out[99]: 1.0

In [100…
```
a **= 3
a
```

Out[100… 1.0

In [ ]:

Comparision/Relational Operators

Comparison operators compare two values/variables and return a boolean result: True or False

Operator	Meaning	Example
`==`	Is Equal To	`3 == 5` gives us **False**
`!=`	Not Equal To	`3 != 5` gives us **True**
`>`	Greater Than	`3 > 5` gives us **False**
`<`	Less Than	`3 < 5` gives us **True**
`>=`	Greater Than or Equal To	`3 >= 5` give us **False**
`<=`	Less Than or Equal To	`3 <= 5` gives us **True**

In [1]:
```
6 == 6
```

Out[1]: True

In [2]:
```
5 != 5
```

Out[2]: False

In [3]:
```
10 > 5
```

Out[3]: True

In [4]:
```
5 < 10
```

Out[4]: True

In [5]:
```
10 >= 10
```

Out[5]: True

In [6]:
```
20 <= 30
```

Out[6]: True

In [ ]:
```

```

Logical Operators

Logical operators are used to check whether an expression is True or False. They are used in decision-making.

Operator	Example	Meaning
`and`	a **and** b	**Logical AND:** `True` only if both the operands are `True`
`or`	a **or** b	**Logical OR:** `True` if at least one of the operands is `True`
`not`	**not** a	**Logical NOT:** `True` if the operand is `False` and vice-versa.

In [101…
```
#AND
10>5 and 5<10
```

Out[101… True

In [102…
```
10>5 and 5>10
```

Out[102… False

In [7]:
```
10<5 and 5>10
```

Out[7]: False

In [8]:
```
10<5 and 5<10
```

Out[8]: False

In [9]:
```
#OR
10>5 or 5>10
```

Out[9]: True

In [10]:
```
10>5 or 5<10
```

Out[10]: True

In [11]:
```
10<5 or 5>10
```

Out[11]: False

In [12]:
```
10<5 or 5>10
```

Out[12]: False

In [13]:
```
#NOT
not 10>5
```

Out[13]: False

In [15]:
```
not 5>10
```

Out[15]: True

Special Operators

In [103…
```
#identity operator - 'is' and 'is not'
a = 10
type(a)
```

Out[103… int

In [104…
```
id(a)   #to check the memory location
```
Out[104…
```
140708370170952
```

In [105…
```
a = 6
b = 6
```

In [106…
```
a is b
```
Out[106…
```
True
```

In [107…
```
id(a)
```
Out[107…
```
140708370170824
```

In [108…
```
id(b)
```
Out[108…
```
140708370170824
```

In [109…
```
x = [1,2,3]
y = [1,2,3]
```

In [110…
```
x is y
```
Out[110…
```
False
```

In [111…
```
a is not y
```
Out[111…
```
True
```

In [112…
```
id(x)
```
Out[112…
```
2332573003008
```

In [113…
```
id(y)
```
Out[113…
```
2332570417152
```

In [114…
```
#membership operator - 'in' and 'not in'
'a' in 'Apple'
```
Out[114…
```
False
```

In [116…
```
'a' not in 'Apple'
```
Out[116…
```
True
```

In [117…
```
10 in [24,78,10,99]
```
Out[117…
```
True
```

In [ ]:

Type Casting

Conversion of data types

In [16]:
```
a = '123'
print(a)
type(a)
```

```
123
```

Out[16]: str

In [17]:
```
i = int(a)
print(i)
type(i)
```

```
123
```

Out[17]: int

In [18]:
```
f = float(i)
print(f)
type(f)
```

```
123.0
```

Out[18]: float

In [19]:
```
a = 'Hello Everyone'
print(a)
type(a)
```

```
Hello Everyone
```

Out[19]: str

In [20]:
```
l = list(a)
print(l)
type(l)
```

```
['H', 'e', 'l', 'l', 'o', ' ', 'E', 'v', 'e', 'r', 'y', 'o', 'n', 'e']
```

Out[20]: list

In [21]:
```
t = tuple(a)
print(t)
type(t)
```

```
('H', 'e', 'l', 'l', 'o', ' ', 'E', 'v', 'e', 'r', 'y', 'o', 'n', 'e')
```

Out[21]: tuple

In [22]:
```
s = set(a)
print(s)
```