Heading

Bold
italic
Bold italic

- tab space is called indentation in python>

- bullet point 1
- bullet point 2

1.number 1
2.number 2

> point 1

In [1]:

```
10+20
```

Out[1]:

```
30
```

Welcome to python

In [ ]:

```
print("Welcome to Python programming")
```

```
Welcome to Python programming
```

Input and output

In [13]:

```
a=input()
print(a)
b=input()
print(b)
c=input()
print(c)
```

```
10
10
2
2
3
3
```

In [14]:

```
n=10.10
print(type(n))
print(str(n))
```

```
<class 'float'>
10.1
```

Input method

In [ ]:

```
n=int(input("Enter an integer : "))
print(n)
```

In [15]:

```
n=10
a=6
print(n/a)
print(n**a)
print(n//a)
```

```
1.6666666666666667
1000000
1
```

In [16]:

```
a=10
b=10 #Single variable assignment
print(a)
a=b=c=d=120 #Multiple variable assignment
print(a)
print(d)
```

```
10
120
120
```

Conditional Statements

- If
- if-else
- elif
- Nested if

In [17]:

```
a="IIIT Srikakulam"
b="IIIT Srikakulam"
if(a==b):
    print("Same")
else:
    print("Different")
```

```
Same
```

In [18]:

```
if a is b:
    print("same")
else:
    print("Different")
```

```
Different
```

print ID of a variable

In [19]:

```
print(id(a)," ",id(b))
```

```
1808839741744   1808839593712
```

In [20]:

```
a="IIITSrikakulam"
b="IIITSrikakulam"
print(id(a)," ",id(b))
```

```
1808839741808   1808839741808
```

Largest of three

In [ ]:

```
a=int(input("Enter a number : "))
b=int(input("Enter another number : "))
c=int(input("Enter another number : "))
if(a>b and a>c):
    print(a)
elif(b>a and b>c):
    print(b)
else:
    print(c)
```

Prime number

In [10]:

```
n=int(input("Enter a number : "))
count=0
for i in range(2,n):
    if(n%i==0):
        count=count+1
if(count==0):
    print("Prime number ")
else:
    print("Not a prime number ")
```

```
Enter a number : 7
Prime number
```

Multiplication Table

In [9]:

```
n=int(input("Enter a number : "))
l=int(input("Enter lower limit : "))
u=int(input("Enter upper limit : "))
for i in range(l,u+1):
    print(n,"*",i,"=",n*1)
```

```
Enter a number : 12
Enter lower limit : 1
Enter upper limit : 10
12 * 1 = 12
12 * 2 = 12
12 * 3 = 12
12 * 4 = 12
12 * 5 = 12
12 * 6 = 12
12 * 7 = 12
12 * 8 = 12
12 * 9 = 12
12 * 10 = 12
```

Leap Year

In [8]:

```
year=int(input("Enter a year :"))
if((year%4==0 and year%100!=0) or (year%400==0)):
    print("Leap year")
else:
    print("Not a leap year")
```

```
Enter a year :2020
Leap year
```

Palindrome number

In [5]:

```
n=int(input("Enter a number : "))
num=0
k=n
p=0
while(k>0):
    rem=k%10
    p=p*10+rem
    k=k//10
if(n==p):
    print("Palindrome")
else:
    print("Not a palindrome")
```

```
Enter a number : 121
Palindrome
```

Factorial of number

In [7]:

```
n=int(input("Enter a number : "))
fact=1
for i in range(1,n+1):
    fact=fact*i
print("Factorial of ",n," is ",fact)
```

```
Enter a number : 6
Factorial of  6  is  720
```

Factorials upto

In [6]:

```
n=int(input("Enter a number : "))
fact=1
for i in range(1,n+1):
    fact=fact*i
    print("factorial of ",i," is ",fact)
```

```
Enter a number : 5
factorial of  1  is  1
factorial of  2  is  2
factorial of  3  is  6
factorial of  4  is  24
factorial of  5  is  120
```

Sorting a String

In [4]:

```
string=input("Enter the string : ")
words=string.split()
words.sort()
print(" ".join(words))
```

```
Enter the string : laxnman b
b laxnman
```

Count of integers , characters, and special symbols

In [2]:

```
string=input("Enter a string : ")
integer=0
ch=0
ss=0
length=0
for i in string:
    if(i.isdigit()):
        integer=integer+1
    if(i.isalpha()):
        ch=ch+1
    if(i.isspace()):
        ss=ss+1
    length=length+1
print("Total no of integers are ",integer)
print("Total no of characters are ",ch)
print("Total no of spaces are ",ss)
print("Total no of symbols are ",length-integer-ch-ss)
```

```
Enter a string : laxman@123 ra#0
Total no of integers are  4
Total no of characters are  8
Total no of spaces are  1
Total no of symbols are  2
```

Print list,tuple of a string

In [5]:

```
string=input("Enter a string : ")
words=string.split(',')
print(list(words))
print(tuple(words))
```

```
Enter a string : laxman,lokesh,venky,devil
['laxman', 'lokesh', 'venky', 'devil']
('laxman', 'lokesh', 'venky', 'devil')
```

Even or Odd

In [1]:

```
a=int(input("Enter a number : "))
if(a%2==0):
    print("Even number")
else:
    print("Odd number")
```

```
Enter a number : 10
Even number
```

Positive negative or zero

In [2]:

```
a=int(input("Enter a number : "))
if(a>0):
    print("Positive ")
elif(a==0):
    print("Zero")
else:
    print("Negative")
```

```
Enter a number : -3
Negative
```

Positive negative or zero using nested if

In [3]:

```
a=int(input("Enter a number : "))
if(a>=0):
    if(a>0):
        print("Positive")
    else:
        print("Zero")
else:
    print("Negative")
```

```
Enter a number : 0
Zero
```

Vote eligibility

In [4]:

```
a=int(input("Enter your age : "))
if(a>18):
    print("Your are eligible for voting")
else:
    print("Your not eligible for voting")
```

```
Enter your age : 19
Your are eligible for voting
```

Password checking

In [7]:

```
u="laxmanarao"
p="laxmana123"
ui=input("Enter your username : ")
if(ui==u):
    pi=input("Enter your password : ")
    if(pi==p):
        print("Welcome ",u)
    else:
        print("Invalid password")
else:
    print("User not registered")
```

```
Enter your username : laxmanarao
Enter your password : laxmana123
Welcome  laxmanarao
```

Registration

In [1]:

```
f_name=input("Enter your first name : ")
l_name=input("Enter your last name : ")
def passwd():
    p=input("Enter your password : ")
    pv=input("Confirm your password : ")
    if(p==pv):
        c=0
    else:
        print("Password does not match")
        passwd()
passwd()
def phno():
    p_no=int(input("Enter your phone number : "))
    count=0
    temp=p_no
    while(temp>0):
        rem=temp%10
        count+=1
        temp=temp//10
    if(count!=10):
        print("Invalid phone number")
        phno()
phno()
print("Registration sucessful")
```

```
Enter your first name : laxman
Enter your last name : b
Enter your password : laxmana123
Confirm your password : laxmana123
Enter your phone number : 8367
Invalid phone number
Enter your phone number : 8367027220
Registration sucessful
```

Print digits in reverse order

In [1]:

```
for i in range(10,0,-1):
    print(i,end=" ")
```

```
10 9 8 7 6 5 4 3 2 1
```

Pattern 1

In [11]:

```
n=int(input("Enter number of rows : "))
for i in range(1,n+1):
    for j in range(1,n+1):
        print("*",end=" ")
    print()
```

```
Enter number of rows : 5
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Pattern 2

In [13]:

```
n=int(input("Enter number of rows : "))
for i in range(1,n+1):
    for j in range(n+1,1,-2):
        print("*",end=" ")
    print()
```

```
Enter number of rows : 6
* * *
* * *
* * *
* * *
* * *
* * *
```

Pattern 3

In [27]:

```
n=int(input("Enter number of rows : "))
for i in range(1,n+1):
    for j in range(1,n+1):
        if i==1 or i==n or j==1 or j==n :
            print("*",end=" ")
        else:
            print(end="  ")
    print()
```

```
Enter number of rows : 6
* * * * * *
*         *
*         *
*         *
*         *
* * * * * *
```

Pattern 4

In [45]:

```
n=int(input("Enter number of rows : "))
for i in range(1,n+1):
    for j in range(1,n+1):
        if i==j or i+j==n+1:
            print("*",end=" ")
        else:
            print(" ",end="  ")
    print()
```

```
Enter number of rows : 5
*          *
   *    *
      *
   *    *
*          *
```

Calender month and using format

In [1]:

```
n=int(input("Enter a number : "))
k=n//7
k=k+1
p=0
for i in range(1,k+1):
    for j in range(1,7+1):
        p=p+1
        if p==n+1:
            break
        print("{:02}".format(p),end=" ")
    print()
```

```
Enter a number : 31
01 02 03 04 05 06 07
08 09 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Web browser and time packages

In [4]:

```
import time
import webbrowser
print("System time is ",time.ctime())
print("1.Youtube")
print("2.Gmail")
print("3.Cricbuzz")
print("4.State Bank INB")
print("5.Amazon")
print("6.Google classroom")
count=int(input("Enter your choice : "))
if count==1:
    webbrowser.open("https://www.youtube.com/")
elif count==2:
    webbrowser.open("https://mail.google.com/mail/u/0/#inbox")
elif count==3:
    webbrowser.open("https://www.cricbuzz.com/")
elif count==4:
    webbrowser.open("https://retail.onlinesbi.com/retail/login.htm")
elif count==5:
    webbrowser.open("https://www.amazon.in/")
elif count==6:
    webbrowser.open("https://classroom.google.com/u/2/h/")
else:
    print("Inavalid input")
```

```
System time is  Tue Feb 11 13:49:07 2020
1.Youtube
2.Gmail
3.Cricbuzz
4.State Bank INB
5.Amazon
6.Google classroom
Enter your choice : 3
```

Sum,Count,Number of even indexed digits of a number

In [1]:

```
n=int(input("Enter a number : "))
count=0
sum=0
num=0
k=0
p=n
while(p>0):
    rem=p%10
    count=count+1
    sum=sum+rem
    k=k*10+rem
    p=p//100
while(k>0):
    rem=k%10
    num=num*10+rem
    k=k//10
print("Count is ",count)
print("Sum is ",sum)
print("number is ",num)
```

```
Enter a number : 1534674657
Count is  5
Sum is  29
number is  54767
```

Sum,Count,Number of odd indexed digits of a number

In [3]:

```
n=int(input("Enter a number : "))
count=0
sum=0
num=0
k=n
p=0
while(k>0):
    rem=k%10
    p=p*10+rem
    k=k//10
while(p>0):
    rem=p%10
    count=count+1
    sum=sum+rem
    num=num*10+rem
    p=p//100

print("Count is ",count)
print("Sum is ",sum)
print("number is ",num)
```

```
Enter a number : 15346746574
Count is  6
Sum is  23
number is  136454
```

Sum,Count,Number of even digits of a number

In [9]:

```
n=int(input("Enter a number : "))
p=n
count=0
sum=0
num=0
k=0
while p>0:
    rem=p%10
    if(rem%2==0):
        k=k*10+rem
        sum=sum+rem
        count=count+1
    p=p//10
while k>0:
    rem=k%10
    num=num*10+rem
    k=k//10
print("Count is ",count)
print("Sum is ",sum)
print("Number is ",num)
```

```
Enter a number : 1246354
Count is  4
Sum is  16
Number is  2464
```

Strings

Acessing string

In [42]:

```
s="HELLO"
print(s[3])
print(s[0:4:1])
print(s)
print(s[::2])
print(s[-1:-6:-1])
s="IIIT Srikakulam"
print(s[0:4],end=" ")
print(s[-1:-11:-1])
print(s[0:4]+s[4]+s[-1:-11:-1])
print(s[-8:-16:-1])
if(len(s)%2==0):
    length=len(s)//2
    length+=1
    print(s[len(s)//2],end=" ")
    print(s[length])
else:
    print(s[len(s)//2])
```

```
L
HELL
HELLO
HLO
OLLEH
IIIT malukakirS
IIIT malukakirS
irS TIII
i
```

Palindrome

In [32]:

```
s=input("Enter a string : ")
count=0
s1=s[-1:-len(s)-1:-1]
if s==s1:
    print("Palindrome")
else:
    print("Not a palindrome")
```

```
Enter a string : madam
Palindrome
```

Length

In [33]:

```
len(s)
```

Out[33]:

```
5
```

Zfill and center

In [48]:

```
s=input("Enter a string : ")
print(s.zfill(10))
print(s.center(25,"*"))
```

```
Enter a string : laxmanarao
laxmanarao
********laxmanarao*******
```

All operations

In [55]:

```
print(dir(s))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul_
_', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitaliz
e', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'fi
nd', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'is
decimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintabl
e', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartiti
on', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'sw
apcase', 'title', 'translate', 'upper', 'zfill']
```

Split using white space

In [49]:

```
s=input("Enter a number : ")
print(s.split())
```

```
Enter a number : laxmana rao boddepalli
```

Out[49]:

```
['laxmana', 'rao', 'boddepalli']
```

Split using coma

In [53]:

```
s=input("Enter a string : ")
print(s.split(","))
```

```
Enter a string : laxmana,rao,boddepalli
['laxmana', 'rao', 'boddepalli']
```

Split using character

In [54]:

```
s=input("Enter a string : ")
print(s.split("k"))
```

```
Enter a string : lakshmankav
['la', 'shman', 'av']
```

Join

In [58]:

```
s=input("Enter a string : ")
print("@".join(s))
print(" ".join(s))
```

```
Enter a string : laxman
l@a@x@m@a@n
l a x m a n
```

Split

In [62]:

```
s=input("Enter a string : ")
s4=s.split()
print("@".join(s4))
```

```
Enter a string : laxman rao
```

Out[62]:

```
'laxman@rao'
```

Split and join

In [63]:

```
s=input("Enter a string : ")
s4=s.split()
print("".join(s4))
```

```
Enter a string : laxman rao
laxmanrao
```

Replace

In [64]:

```
s=input("Enter a string : ")
print(s.replace(" ",""))
```

```
Enter a string : laxmana rao
laxmanarao
```

Strip

In [67]:

```
s=input("Ente a string : ")
print(s.strip())
print(s.rstrip())
print(s.lstrip())
```

```
Ente a string :       laxmana rao
laxmana rao
      laxmana rao
laxmana rao
```

Count

In [69]:

```
s=input("Enter a string : ")
print(s.count("lax"))
print(s.count("a"))
```

```
Enter a string : laxmana rao
1
4
```

Functions

In-build functions

In [4]:

```
print(abs(-4))
print("hi")
print(bin(52892))
print(hex(52892))
print(oct(52892))
```

```
4
hi
0b1100111010011100
0xce9c
0o147234
```

User defined functions

- Without arguments without return type
- Without arguments with return type
- with argument without return type
- with argument with return type #### Arguments
- Required arguments
- Defalt arguments
- Keyword arguments
- Variable length arguments

In [5]:

```
# required arguments
def fn(a,b):
    return a+b
fn(10,20)
fn(10)
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-5-761d7d1b78a9> in <module>
      3     return a+b
      4 fn(10,20)
----> 5 fn(10)

TypeError: fn() missing 1 required positional argument: 'b'
```

In [8]:

```
# Default Arguments
def fn(a,b=5):
    return a+b
print(fn(10,20))
print(fn(10))
```

```
30
15
```

In [10]:

```
#keyword arguments
def fn(name,age):
    return name,age
fn(age=25,name="laxman")
```

Out[10]:

```
('laxman', 25)
```

In [11]:

```
# Variable length arguments
def fn(*a):
    return sum(a)
print(fn(1,2,3,4,5))
print(fn(10,20,30))
```

```
15
60
```

In [30]:

```
def prime(a):
    count=0
    for i in range(2,a):
        if(a%i==0):
            count=count+1
    if(count==0):
        return True
```

In [34]:

```
def primel(l,u):
    count=0
    if(l<2):
        l=2
    for i in range(l,u+1):
        if prime(i):
            print(i,end=" ")
            count=count+1
    print("\nTotal prime numbers are ",count)
```

In [32]:

```
a=int(input("Enter a number : "))
d=prime(a)
if(d==1):
    print("Prime")
else:
    print("Not prime")
```

```
Enter a number : 5
Prime
```

In [35]:

```
a=int(input("Enter lower limit :"))
b=int(input("Enter upper limit : "))
primel(a,b)
```

```
Enter lower limit :-5
Enter upper limit : 100
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
Total prime numbers are  25
```

Data structures

- List
- Tuple
- Set
- Dictionaries

In [44]:

```
li = [1,2,"hi","hello"]
l1=li.copy()
l2=[5,2,8,9,2]
l1.append(l2)
print(l1)
l1=l1+l2
print(l1)
n=input("Enter string : ")
l1.insert(2,n)
print(l1)
```

```
[1, 2, 'hi', 'hello', [5, 2, 8, 9, 2]]
[1, 2, 'hi', 'hello', [5, 2, 8, 9, 2], 5, 2, 8, 9, 2]
Enter string : lax
[1, 2, 'lax', 'hi', 'hello', [5, 2, 8, 9, 2], 5, 2, 8, 9, 2]
```

In [51]:

```
l=[1,2,3,4]
k=[5,6,7,8]
l.extend(k)
print(l)
l=l+k
print(l)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 5, 6, 7, 8]
```

Delete element - index based

In [53]:

```
l=[1,2,3,4]
k=[5,6,7,8]
l.pop()
print(l)
l.pop(1)
print(l)
```

```
[1, 2, 3]
[1, 3]
```

Remove element using value

In [54]:

```
l=[123,143,153,163]
l.remove(143)
print(l)
```

```
[123, 153, 163]
```

Length of List

In [57]:

```
len(l)
```

Out[57]:

```
3
```

Count an element

In [70]:

```
l=[5,1,2,1,2,3,1,4,1]
l.count(1)
```

Out[70]:

```
1
```

Index of element

In [69]:

```
print(l.index(4))
l.index(1)
```

```
7
```

Out[69]:

```
1
```

Clear a list

In [61]:

```
l.clear()
l
```

Out[61]:

```
[]
```

Input a list

In [1]:

```
n=input("Enter elements : ")
a=n.split(",")
li=[]
print(a)
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
print(li)
```

```
Enter elements : 1,4,2,6,3,7,8
['1', '4', '2', '6', '3', '7', '8']
[1, 4, 2, 6, 3, 7, 8]
```

Second largest element

In [2]:

```
n=input("Enter elements : ")
a=n.split(",")
li=[]
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
li.sort(reverse = True)
print("second largest element is ",li[1])
```

```
Enter elements : 1,3,5,2,6,3,7,8
second largest element is  7
```

Unique List

In [103]:

```
n=input("Enter elements : ")
a=n.split(",")
li=[]
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
print(li)
l2=[]
for i in li:
    if i not in l2:
        l2.append(i)
print("Unique list")
print(l2)
```

```
Enter elements : 1,2,3,4,5,1,2,6,7,2
[1, 2, 3, 4, 5, 1, 2, 6, 7, 2]
Unique list
[1, 2, 3, 4, 5, 6, 7]
```

Search an element

In [101]:

```
def search(li,value):
    if value in li:
        return True
    else:
        return False
n=input("Enter elements : ")
a=n.split(",")
li=[]
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
search(li,5)
```

```
Enter elements : 1,2,3,4,5,6
```

Out[101]:

```
True
```

Tuples

Input from keyboard

In [11]:

```
n=input("Enter elements : ")
a=n.split(",")
li=[]
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
t=tuple(li)
t
```

```
Enter elements : 1,2,5,3,7,5
```

Out[11]:

```
(1, 2, 5, 3, 7, 5)
```

Length

In [4]:

```
len(t)
```

Out[4]:

```
5
```

Count

In [5]:

```
t.count(2)
```

Out[5]:

```
1
```

Index

In [6]:

```
t.index(6)
```

Out[6]:

```
3
```

Min

In [12]:

```
min(t)
```

Out[12]:

```
1
```

Max

In [13]:

```
max(t)
```

Out[13]:

```
7
```

Delete

In [8]:

```
del(t)
t
```

```
--------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-8-f12e3a84697c> in <module>
----> 1 del(t)
      2 t

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

Sets

set={}

Input from keyboard

In [16]:

```
n=input("Enter elements : ")
a=n.split(",")
li=[]
for i in a:
    if i.isdigit():
        i=int(i)
        li.append(i)
s=set(li)
s
```

```
Enter elements : 1,2,4,3,5,6
```

Out[16]:

```
{1, 2, 3, 4, 5, 6}
```

Difference

In [18]:

```
l={1,2,3,4,5,5,77,9,8}
n={5,0,8,6,5,5}
l.difference(n)
```

Out[18]:

```
{1, 2, 3, 4, 9, 77}
```

Difference update

In [20]:

```
print(l)
l.difference_update(n)
print(l)
```

```
{1, 2, 3, 4, 5, 8, 9, 77}
{1, 2, 3, 4, 9, 77}
```

Intersection

In [21]:

```
l={1,2,3,4,5,5,77,9,8}
n={5,0,8,6,5,5}
l.intersection(n)
```

Out[21]:

```
{5, 8}
```

Intersection update

In [22]:

```
l.intersection_update(n)
l
```

Out[22]:

```
{5, 8}
```

Is disjoint

In [23]:

```
l={1,2,3,4,5,5,77,9,8}
n={5,0,8,6,5,5}
l.isdisjoint(n)
```

Out[23]:

```
False
```

Is subset

In [25]:

```
l={1,2,3,4,5}
k={1,2,3,4,5,6,7,8,9}
l.issubset(k)
```

Out[25]:

```
True
```

Is superset

In [26]:

```
k.issuperset(l)
```

Out[26]:

```
True
```

Symetric difference

In [28]:

```
l={1,2,3,4,5,5,77,9,8}
n={5,0,8,6,5,5}
l.symmetric_difference(n)
```

Out[28]:

```
{0, 1, 2, 3, 4, 6, 9, 77}
```

Union

In [29]:

```
l.union(n)
```

Out[29]:

```
{0, 1, 2, 3, 4, 5, 6, 8, 9, 77}
```

Remove

In [30]:

```
l.remove(77)
l
```

Out[30]:

```
{1, 2, 3, 4, 5, 8, 9}
```

Discard

In [32]:

```
l.discard(6)
l
```

Out[32]:

```
{1, 2, 3, 4, 5, 8, 9}
```

Remove VS Discard

In [34]:

```
l.remove(9)
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
<ipython-input-34-2e4f732c7741> in <module>
----> 1 l.remove(9)

KeyError: 9
```

In [37]:

```
print(l.discard(9))
```

```
None
```

Dictionaries

- collection of items which has the pair of key and values
- It is unordered
- Keys must be unique but values allows duplicates
- Mutable
- Represented with
- keys and values are seperated by column(:

In [3]:

```
a=set()
print(type(a))
d={}
type(d)
```

```
<class 'set'>
```

Out[3]:

```
dict
```

In [5]:

```
d={1:"one",2:"tooo","three":3}
d
```

Out[5]:

```
{1: 'one', 2: 'tooo', 'three': 3}
```

In [8]:

```
d[2]="two"
d
```

Out[8]:

```
{1: 'one', 2: 'two', 'three': 3}
```

In [10]:

```
d[2]="one"
d
```

Out[10]:

```
{1: 'one', 2: 'one', 'three': 3}
```

In [12]:

```
d2={5:"five",6:"six",7:"seven"}
d.update(d2)
d
```

Out[12]:

```
{1: 'one', 2: 'one', 'three': 3, 5: 'five', 6: 'six', 7: 'seven'}
```

In [13]:

```
d.get(2)
```

Out[13]:

```
'one'
```

In [14]:

```
d.items()
```

Out[14]:

```
dict_items([(1, 'one'), (2, 'one'), ('three', 3), (5, 'five'), (6, 'six'),
(7, 'seven')])
```

In [16]:

```
d.keys()
```

Out[16]:

```
dict_keys([1, 2, 'three', 5, 6, 7])
```

In [15]:

```
d2={5:"five",6:"six",7:"seven"}
d.values()
```

Out[15]:

```
dict_values(['one', 'one', 3, 'five', 'six', 'seven'])
```

Another way to declare dictionary

In [18]:

```
c=dict([(1,"one"),(2,"two")])
c
```

Out[18]:

```
{1: 'one', 2: 'two'}
```

In [19]:

```
d.pop(2)
d
```

Out[19]:

```
{1: 'one', 'three': 3, 5: 'five', 6: 'six', 7: 'seven'}
```

In [21]:

```
# Recently inserted element will be deleted
d.popitem()
d
```

Out[21]:

```
{1: 'one', 'three': 3, 5: 'five', 6: 'six'}
```

In [24]:

```
d2={5:"five",6:"six",7:"seven"}
d.fromkeys(d2,"laxman")
```

Out[24]:

```
{5: 'laxman', 6: 'laxman', 7: 'laxman'}
```

In [25]:

```
d2.setdefault(4,"hi")
```

Out[25]:

```
'hi'
```

Create Contacts

In [12]:

```
contacts={}
import re
pattern="(0|91)?[6-9][0-9]{9}$"
def pv(p):
    if re.match(pattern,p):
        return 1
    else:
        return 0
def create(name,num):
    if name not in contacts:
        if pv(str(num)):
            contacts[name]=int(num)
            print("Contact created sucessfully")
        else:
            print("Invalid phone number")
            j=int(input("Enter phone number :"))
            create(name,j)
    else:
        print("Name already exists")
n=int(input("No of inputs : "))
for i in range(n):
    a=input().split()
    b=a[0]
    c=a[1]
    d=int(c)
    create(b,d)
print("Contacts : ")
print(contacts)
```

```
No of inputs : 3
laxman 918367027220
Contact created sucessfully
dileep 6300189625
Contact created sucessfully
dad 09052083710
Contact created sucessfully
Contacts :
{'laxman': 918367027220, 'dileep': 6300189625, 'dad': 9052083710}
```

Edit contact

In [13]:

```
import re
pattern="(0|91)?[6-9][0-9]{9}$"
def edit(name,num):
    if re.match(pattern,num):
        contacts[name]=int(num)
        print("Contact updated sucessfully")
    else:
        print("Invalid phone number")
        s=input("Enter phone number : ")
        edit(name,s)
x=input().split()
l=x[0]
k=x[1]
edit(l,k)
contacts
```

```
dad 09849297205
Contact updated sucessfully
```

Out[13]:

```
{'laxman': 918367027220, 'dileep': 6300189625, 'dad': 9849297205}
```

Delete Contact

In [14]:

```
def delete(name):
    if name in contacts:
        contacts.pop(name)
        print("Contact deleted sucessfully")
    else:
        print("Contact does not exist")
x=input("Enter name to delete : ")
delete(x)
contacts
```

```
Enter name to delete : dileep
Contact deleted sucessfully
```

Out[14]:

```
{'laxman': 918367027220, 'dad': 9849297205}
```

In [1]:

```
import keyword
print(keyword.kwlist,end=" ")
```

```
['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']
```

Regular expressions

- Pattern matching
- Match()
- Search()
- Findall()

Matching

In [1]:

```
import re
re.match('lax','laxman')
```

Out[1]:

```
<re.Match object; span=(0, 3), match='lax'>
```

Search

In [2]:

```
# First occurence
import re
re.search('a','laxman')
```

Out[2]:

```
<re.Match object; span=(1, 2), match='a'>
```

Print all required string

In [3]:

```
import re
re.findall('a','laxman')
```

Out[3]:

```
['a', 'a']
```

Pattern matching

^ - Begining of the line
$ - Ending of the line
\s - white spaces are allowed
\S - not Allow white spaces
\d - allow digits \D - not allow digits only . - Allow any one character

- - repeat
- - repeat zero or more character [a,b,r] - must begin with any element [a-o] range {5} - only 5 {3,9} min 3 or max 9

In [7]:

```
import re
re.findall("\s","lakshmana rao")
```

Out[7]:

```
[' ']
```

In [8]:

```
import re
re.findall("\S","lakshmana rao")
```

Out[8]:

```
['l', 'a', 'k', 's', 'h', 'm', 'a', 'n', 'a', 'r', 'a', 'o']
```

In [9]:

```
import re
re.findall("\d","123lakshmana rao")
```

Out[9]:

```
['1', '2', '3']
```

In [10]:

```
import re
re.findall("\D","123lakshmana rao")
```

Out[10]:

```
['l', 'a', 'k', 's', 'h', 'm', 'a', 'n', 'a', ' ', 'r', 'a', 'o']
```

In [11]:

```
import re
re.findall("^.a","lakshmana rao")
```

Out[11]:

```
['la']
```

In [13]:

```
import re
re.findall("^..a","lakshmana rao")
```

Out[13]:

```
[]
```

In [14]:

```
import re
re.findall("^l.+","lakshmana rao")
```

Out[14]:

```
['lakshmana rao']
```

In [16]:

```
import re
re.findall("k.*","lakshmana rao")
```

Out[16]:

```
['kshmana rao']
```

In [17]:

```
import re
re.findall("o.+","lakshmana rao")
```

Out[17]:

```
[]
```

In [18]:

```
import re
re.findall("o.*","lakshmana rao")
```

Out[18]:

```
['o']
```

In [21]:

```
# must begin with
import re
re.match("^[a-o]","lakshmana rao")
```

Out[21]:

```
<re.Match object; span=(0, 1), match='l'>
```

In [22]:

```
# must not begin with
import re
re.match("[^a-o]","lakshmana rao")
```

phone number validation

- 10 digits
- start with 6-9
  8367027220

In [10]:

```
import re
pattern="(0|91)?[6-9][0-9]{9}$"
def pv(p):
    if re.match(pattern,str(p)):
        print("Phone number is valid")
    else:
        print("Phone number not valid")
n=int(input("Enter phone number : "))
pv(n)
```

```
Enter phone number : 918367027220
Phone number is valid
```

Email verification

- username ^[0-9a-z][a-z0-9._]{5,35}
- @ [@]
- domain name [a-z.]{4,10}
- .[.]
- extension[a-z]{2,5]

In [1]:

```
import re
pattern="^[0-9a-z][a-z0-9._]{5,35}[@][a-z.]{4,10}[.][a-z]{2,5}$"
def ev(mail):
    if re.match(pattern,mail):
        print("Email is valid")
    else:
        print("Email not valid")
n=input("Enter Email : ")
ev(n)
```

```
Enter Email : lakshmana5296@gmail.com
Email is valid
```

In [ ]: