Tuple

```
t=(10, 20, 30)
print(t, type(t))
```

```
(10, 20, 30) <class 'tuple'>
```

```
t[0]
```

```
10
```

```
for i in t:
  print(t)
```

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

```
t=(11, 22, 33) # allows overwriting, but does not allow modification
print(t)
```

```
(11, 22, 33)
```

```
t[0]=t[0]*100 # does not allow modification

```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-e83bd1d9a7a8> in <cell line: 1>()
----> 1 t[0]=t[0]*100

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

SEARCH STACK OVERFLOW

```
t=(10, 20, 30)
print(t, type(t), id(t))
t=(11, 22, 33)
print(t, type(t), id(t))
```

```
(10, 20, 30) <class 'tuple'> 140061672937600
(11, 22, 33) <class 'tuple'> 140061672938432
```

```
t=(10, 20, [11, 22, 33])
```

```
t[2].append(100)
```

```
print(t)
```

```
(10, 20, [11, 22, 33, 100])
```

```
t[2].append([1000, 2000, 3000])
```

```
print(t)
```

```
(10, 20, [11, 22, 33, 100, [1000, 2000, 3000]])
```

```
t[2][-1][-1]
```

```
3000
```

```
(t[2][-1])[-1] # nesting of indices
```

```
3000
```

Set

```
s={11, 22, 11, 33, 11, 22, 44, 11, 22}
```

```
print(s)
```

```
{33, 11, 44, 22}
```

```
s # only colab sorts it itself, but other platforms of python like jupyter will not sort
```

```
{11, 22, 33, 44}
```

```
s[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-17-f8bb2b116405> in <cell line: 1>()
----> 1 s[1]

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

SEARCH STACK OVERFLOW

```
for i in s:
  print(i)
```

```
33
11
44
22
```

```
a={1, 2, 3, 4}
b={3, 4, 5, 6}
```

```
a&b # intersection with ampersand
```

```
{3, 4}
```

```
a|b # union with pipe
```

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

```
a^b # symmertric difference with caret
```

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

```
a-b # difference
```

```
{1, 2}
```

```
b-a
```

```
{5, 6}
```

```
emp={1, 2, 3, 4, 5, 6, 7, 8, 9}
drama={2, 4, 6, 7}
sing={1, 3, 6, 8}
sports={1, 3, 6, 8}
# find who aprticipated in all
drama&sing&sports
```

```
{6}
```

```
# didn't participate in anything
emp-drama-sing-sports
```

```
{5, 9}
```

```
# particiapted in aleast one thing
drama|sing|sports
```

```
{1, 2, 3, 4, 6, 7, 8}
```

```
s=set()
```

```
s.add(int(input("Enter data: ")))
```

```
Enter data: 66
```

```
print(s, len(s))
```

```
{33, 66, 11, 6} 4
```

```
1 s
```

```
{6, 11, 33, 66}
```

```
1 print(s)
```

```
{33, 66, 11, 6}
```

```
1 s.remove(6)
```

```
1 print(s)
```

```
{33, 66, 11}
```

```
1 s.remove(6)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-59-077f15baad77> in <cell line: 1>()
----> 1 s.remove(6)

KeyError: 6
```

SEARCH STACK OVERFLOW

```
1 print(s)
```

```
{33, 66, 11}
```

```
1 s.discard(6)
```

```
1 print(s)
```

```
{33, 66, 11}
```

```
1 s.pop()
```

```
33
```

```
1 print(s)
```

```
{66, 11}
```

```
1 s.pop()
```

```
66
```

```
1 print(s)
```

```
{11}
```

```
1 s.pop()
```

```
11
```

```
1 print(s)
```

```
set()
```

```
1 s.pop()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-69-c88c8c48122b> in <cell line: 1>()
----> 1 s.pop()

KeyError: 'pop from an empty set'
```

SEARCH STACK OVERFLOW

```
1 s={11, 22, 33, 44, 55}
```

```
1 s
```

```
{11, 22, 33, 44, 55}
```

```
1 s.pop()
```

```
33
```

```
1 s.pop()
```

```
22
```

```
1 # enter all elements in a list first blank
2 # then print sum of all , use sum()
3 list1=[]
4 while True:
5   ele=input("Enter element: ")
6   if ele=="":
7     break
8   list1.append(float(ele))
9 print("unique data\n")
10 for i in sorted(set(list1)):
11   print(i)
```

```
Enter element: 55
Enter element: 12
Enter element: 12
```

```
Enter element: 45
Enter element: 23
Enter element:
unique data

12.0
23.0
45.0
55.0
```

```
# list all elements in sorted manner with their count
list1=[]
while True:
  ele=input("Enter element: ")
  if ele=="":
    break
  list1.append(float(ele))
print("unique data\n")
for i in sorted(set(list1)):
  print(i, "frequency: ", list1.count(i))
```

```
Enter element: 56
Enter element: 56
Enter element: 56
Enter element: 45
Enter element: 12
Enter element: 45
Enter element: 89
Enter element:
unique data

12.0 frequency:  1
45.0 frequency:  2
56.0 frequency:  3
89.0 frequency:  1
```

```
# list all elements in sorted manner with their count, also print the one with highest count
list1=[]
while True:
  ele=input("Enter element: ")
  if ele=="":
    break
  list1.append(float(ele))
print("unique data\n")
hig=0
higel=0
for i in sorted(set(list1)):
  print(i, "frequency: ", list1.count(i))
  if list1.count(i)>hig:
    hig=list1.count(i)
    higel=i
print("Highest frequency:", hig, "element:", higel)
```

```
Enter element: 89
Enter element: 45
Enter element: 12
Enter element: 75
```

```
Enter element: 89
Enter element: 45
Enter element: 69
Enter element: 89
Enter element: 45
Enter element: 23
Enter element: 89
Enter element: 75
Enter element:
unique data

12.0 frequency:  1
23.0 frequency:  1
45.0 frequency:  3
69.0 frequency:  1
75.0 frequency:  2
89.0 frequency:  4
Highest frequency: 4 element: 89.0
```

dict: Dictionary

```
d={1:"One", 2:"Two", 3:"Three", 4:"Four"}
```

```
print(d)
```

```
{1: 'One', 2: 'Two', 3: 'Three', 4: 'Four', 5: 'Five'}
```

```
d[3]
```

```
'Three'
```

```
d[2]
```

```
'Two'
```

```
d[5]="Five"
```

```
print(d)
```

```
{1: 'One', 2: 'Two', 3: 'Three', 4: 'Four', 5: 'Five'}
```

```
d[1]="Ek number"
```

```
print(d)
```

```
{1: 'Ek number', 2: 'Two', 3: 'Three', 4: 'Four', 5: 'Five'}
```

```
# print number in words, digit by digit
# e.g. 123 --> one two three
n=int(input("Enter a number: "))
```

```
di={0:"zero", 1:"one", 2:"two", 3:"three", 4:"four", 5:"five", 6:"six", 7:"seven", 8:"eight", 9:"nine"}
no=n
rev=0
while no>0:
  rev=(rev*10)+(no%10)
  no=no//10
while rev>0:
  d=rev%10
  rev=rev//10
  print(di[d], end=" ")
```

```
Enter a number: 8194
eight one nine four
```

```
# take a number and print unique digits only in sorted manner in words
# input: 361421288
# output: one two three four six eight
n=int(input("Enter a number: "))
di={0:"zero", 1:"one", 2:"two", 3:"three", 4:"four", 5:"five", 6:"six", 7:"seven", 8:"eight", 9:"nine"}
se=set()
no=n
while no>0:
  d=no%10
  rev=(rev*10)+(no%10)
  no=no//10
  se.add(d)
for i in sorted(se):
  print(di[i], end=" ")
```

```
Enter a number: 361421288
one two three four six eight
```

```
di={0:"zero", 1:"one", 2:"two", 3:"three", 4:"four", 5:"five", 6:"six", 7:"seven", 8:"eight", 9:"nine"}
for i in di:
  print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

```
print(di.keys())
```

```
dict_keys([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
print(di.values())
```

```
dict_values(['zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine'])
```

```
print(di.items())
```

```
dict_items([(0, 'zero'), (1, 'one'), (2, 'two'), (3, 'three'), (4, 'four'), (5, 'five'), (6, 'six'), (7, 'seven'), (8, 'eight'), (9, 'nine')])
```

```
for k in di.keys():
  print("Key:", k) # print all keys
```

```
Key: 0
Key: 1
Key: 2
Key: 3
Key: 4
Key: 5
Key: 6
Key: 7
Key: 8
Key: 9
```

```
for v in di.values():
  print("value:", v) # print all values
```

```
value: zero
value: one
value: two
value: three
value: four
value: five
value: six
value: seven
value: eight
value: nine
```

```
for k, v in di.items():
  print("Key:", k, "value:", v) # print all key-value pairs
```

```
Key: 0 value: zero
Key: 1 value: one
Key: 2 value: two
Key: 3 value: three
Key: 4 value: four
Key: 5 value: five
Key: 6 value: six
Key: 7 value: seven
Key: 8 value: eight
Key: 9 value: nine
```

string

```
s='amar'
s2="amar"
s3='''amar'''
s4="""amar"""
print(s, s2, s3, s4)
```

```
amar amar amar amar
```

```
# '+' is used to concatenate strings
s="am"
s2="ar"
s3=s+s2
```

```
print(s3)
```

```
amar
```

```
s="python"
print(s, id(s))
s=s+" language"
print(s, id(s))
```

```
python 140062268704944
python language 140061010665328
```

```
data=[10, 20]
print(data, id(data)) # data structures once assigned memory space, keeps that space
data.append(1000)
print(data, id(data))
```

```
[10, 20] 140061010450304
[10, 20, 1000] 140061010450304
```

```
s="amar"
s2="amar"
if s==s2:
  print("yes")
```

```
yes
```

```
# indexed access +ve and -ve
for i in range(0, len(s)):
  print(i, s[i])
```

```
0 a
1 m
2 a
3 r
```

```
s*3
```

```
'amaramaramar'
```

```
s="god_"
s*11
```

```
'god_god_god_god_god_god_god_god_god_god_god_'
```

```
for i in range(1, 6):
  print("* "*i)
```

```
*
* *
* * *
* * * *
* * * * *
```

```
for i in range(1, 10, 2):
  print("* "*i)
```

```
*
* * *
* * * * *
* * * * * * *
* * * * * * * * *
```

```
for i in range(5, 0, -1):
  print(" "*(5-i),"*"*i)
```

```
 *****
  ****
   ***
    **
     *
```

```
for i in range(1,6):
  print(" "*(5-i), "* "*i)
```

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
for i in range(5,0,-1):
  print(" "*(5-i), "* "*i)
```

```
* * * * *
 * * * *
  * * *
   * *
    *
```

```
s="abcdefghijk"
print(s[3:6])
print(s[:3]) # [start:3] [0:3]
print(s[4:]) # [4:end] [4:len(s)]
print(s[::-1]) # reverse
print(s[::]) # [start:end]
print(s[::2]) # alternate elements
```

```
def
abc
```

```
efghijk
kjihgfedcba
abcdefghijk
acegik
```

```
# take a word and print word triangle
# input: amar
#op:  a
#     am
#     ama
#     amar
inp=input("Enter a word: ")
for i in range(0, len(inp)+1):
  print(inp[:i])
```

```
Enter a word: satellite

s
sa
sat
sate
satel
satell
satelli
satellit
satellite
```

```
# take a word and print word triangle
# input: amar
#op:  a
#     am
#     ama
#     amar
inp=input("Enter a word: ")
for i in range(0, len(inp)+1):
  print(" "*(len(inp)-i),inp[:i])
```

```
Enter a word: satellite

         s
        sa
       sat
      sate
     satel
    satell
   satelli
  satellit
 satellite
```

```
# check whether given word is pallindrome or not
inp=input("Enter a word to check pallindrome: ")
if(inp==inp[::-1]):
  print(inp, "is pallindrome")
else:
  print(inp, "is not pallindrome")
```

```
Enter a word to check pallindrome: nitin
nitin is pallindrome
```

```
# check substring exists in string
# in & not in operators are case-sensitive
print("amar" in "amarendra bahubali")
print("amar" not in "amarendra bahubali")
```

```
True
False
```

```
s="pyhton"
dset=set(s)
print(dset)
```

```
{'t', 'p', 'n', 'o', 'h', 'y'}
```

```
# wap to check no of vowels in a given string
inp=input("Enter a string: ")
vcnt=0
for i in inp:
  if i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U':
    vcnt+=1
print(vcnt)
```

```
Enter a string: testing line to check vowel count
10
```

```
s="tHis IS tEST fOr Us"
```

```
s.swapcase()
```

```
'ThIS is Test FoR uS'
```

```
s.lower()
```

```
'this is test for us'
```

```
s.upper()
```

```
'THIS IS TEST FOR US'
```

```
s
```

```
'tHis IS tEST fOr Us'
```

```
s.title()
```

```
'This Is Test For Us'
```

```
s.capitalize()
```

```
'This is test for us'
```

```
s
```

```
'tHis IS tEST fOr Us'
```

```
s="assassination"
s.count('s')
```

```
4
```

```
s.find('s') # goes from start to end, +ve index
```

```
1
```

```
s.rfind('s') # reverse find, goes from end to start, +ve index
```

```
5
```

```
s.rjust(30, '*')
```

```
'*****************assassination'
```

```
s.ljust(30, '*')
```

```
'assassination*****************'
```

```
s.center(30, '*')
```

```
'********assassination*********'
```

```
s="he came by taxi"
s.replace("taxi", "ola")
```

```
'he came by ola'
```

```
# convert to past tense
s="this is done by me"
s.replace(" is", " was")
```

```
'this was done by me'
```

```
s="this is a test line"
wordlist=s.split()
print(len(wordlist))
print(wordlist)
```

```
5
['this', 'is', 'a', 'test', 'line']
```

```
s="    this    is    a    test    line  "
wordlist=s.split()
print(len(wordlist))
print(wordlist)
```

```
5
['this', 'is', 'a', 'test', 'line']
```

```
s="1,12,123,1234"
wlist=s.split(',')
wlist
```

```
['1', '12', '123', '1234']
```

```
# take a input from user, print word count
print(len(input("enter data: ").split()))
```

```
enter data: this is for testing a program
6
```

```
# use internal method to count vowel
inp=input("Enter data: ").lower()
c=inp.count('a')+inp.count('e')+inp.count('i')+inp.count('o')+inp.count('u')
print(c)
```

```
Enter data: this is A TEST
4
```

```
# interview challange, 3i, morningstar
# reverse line , but word by words
line="this is what was needed to do"
li=line.split()
li.reverse()
rline=" ".join(li)
print(rline)
```

```
do to needed was what is this
```

```
# alt solution
line="this is what was needed to do"
li=line.split()
li.reverse()
rline=""
for word in li:
  rline=rline+" "+word
print(rline.strip())
```

```
do to needed was what is this
```

```
# line has names of employees, reverse it and sort it
emp="Ramesh Suresh Paresh Mahesh"
li=emp.split()
li.sort()
for word in li:
  print(word)
```

```
Mahesh
Paresh
Ramesh
Suresh
```

```
for i in nlist:
  c+=1

```

```
# count frequencey of each word in a line of string
line=input("Enter a line:")
wlist=line.split()
ulist=sorted(set(wlist))
for w in ulist:
  print(w, ":", wlist.count(w))
```

```
Enter a line:this is is this line is
is : 3
line : 1
this : 2
```

```
# keep only forst occurence of is, remove rest
line = "this is so good so is to is is is"
wlist=line.split()
#print(wlist)
nlist=[]
for word in wlist:
  if word not in nlist:
    nlist.append(word)
print(" ".join(nlist))
```

```
this is so good to
```

```
# rotate all characters of a word till it becomes same word again
stri=input("Enter a word: ")
chlist=[]
print(stri)
for ch in stri:
  chlist.append(ch)
for i in range(len(stri)):
  c=chlist.pop(0)
  chlist.append(c)
  print("pass:", i+1, "".join(chlist))
```

```
Enter a word: india
india
pass: 1 ndiai
pass: 2 diain
```

```
pass: 3 iaind
pass: 4 aindi
pass: 5 india
```

```
# print a-z with ascii
# ord() retruns ascii
# chr() returns ascii to char
for i in range(ord('a'), ord('z')+1):
  print(chr(i), "is", i)
```

```
a is 97
b is 98
c is 99
d is 100
e is 101
f is 102
g is 103
h is 104
i is 105
j is 106
k is 107
l is 108
m is 109
n is 110
o is 111
p is 112
q is 113
r is 114
s is 115
t is 116
u is 117
v is 118
w is 119
x is 120
y is 121
z is 122
```

```
# print chars with ascii, user specifies starting & end char
for i in range(ord(input("Enter start char: ")), ord(input("Enter end char: "))+1):
  print(chr(i), "is", i)
```

```
Enter start char: E
Enter end char: J
E is 69
F is 70
G is 71
H is 72
I is 73
J is 74
```

```
# interview challange, eClerx, accenture
# line1="hi how are you"
# line2="this is how it is"
#     OUTPUT           line3="hi this how is are how you it is"
line1="hi how are you"
line2="this is how it is"
wlist1=line1.split()
```

```
wlist2=line2.split()
line3=""
i=0
j=0
while i<len(wlist1) or j<len(wlist2):
  if i<len(wlist1):
    line3=line3+" "+wlist1[i]
    i+=1
  if j<len(wlist2):
    line3=line3+" "+wlist2[j]
    j+=1
print(line3)
```

```
 hi this how is are how you it is
```

dict

```
# interview challange
'''
{id:[name, cost]}
{1:["Milk", 90], 2:["bread",35], 3:["Jam", 100], 4:["redbull", 125]}
get qty of each item from user using menu driven program, generate final bill after adding a GST of 5% on total bill
'''

```

```
# combine two lists to create a dictionary
klst=[1, 2, 3, 4]
vlst=['aa', 'bb', 'cc', 'dd']
dll={}
for i in range(len(klst)):
  dll[klst[i]]=vlst[i]
print(dll)
```

```
{1: 'aa', 2: 'bb', 3: 'cc', 4: 'dd'}
```

```
# combine two dict into third one
di1={1:"one", 2:"two", 3:"three"}
di2={4:"four", 5:"five"}
di1.update(di2)
print(di1)
```

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

```
dll2={10:"aaa", 33:"er", 1:"aa", 2:"bb", 3:"cc", 4:"dd"}
print(dll2)
todel=[1, 33, 4]
for i in todel:
  print("deleted:", dll2.pop(i))
print(dll2)
```

```
{10: 'aaa', 33: 'er', 1: 'aa', 2: 'bb', 3: 'cc', 4: 'dd'}
deleted: aa
```

```
deleted: er
deleted: dd
{10: 'aaa', 2: 'bb', 3: 'cc'}
```

```
# check if some value exists in dict
dll2={10:"aaa", 33:"er", 1:"aa", 2:"bb", 3:"cc", 4:"dd"}
print(dll2)
chk=int(input("enter key to check: "))
if chk in dll2.keys():
  print("yes")
else:
  print("no")
```

```
{10: 'aaa', 33: 'er', 1: 'aa', 2: 'bb', 3: 'cc', 4: 'dd'}
enter key to check: 5
no
```

```
# print key for max marks
d={"m1":77, "m2":67, "m3":57, "m4":97, "m5":7, "m6":47}
kys=d.keys()
max=0
msub=''
min=9999
mnsub=''
for i in kys:
  if d[i]>max:
    max=d[i]
    msub=i
  if d[i]<min:
    min=d[i]
    mnsub=i
print("Maximum marks:", max, "in subject:", msub)
print("Minimum marks:", min, "in subject:", mnsub)
```

```
Maximum marks: 97 in subject: m4
Minimum marks: 7 in subject: m5
```

```

```