

The GitHub Edition - Python(Basic)

In [2]:

```
print("Hello! Welcome to The GitHub Edition - Python(Basic)")
```

```
Hello! Welcome to The GitHub Edition - Python(Basic)
```

In [3]:

```
print(input())
```

```
Hi  
Hi
```

In [4]:

```
a=input().split(",")
print(a)
```

```
1,2,3,4
['1', '2', '3', '4']
```

some built-in functions

In [6]:

```
##split
a="1>2>3>4>5>56"
b=a.split(">")
print(b)
print(a.split('>'))
```

```
['1', '2', '3', '4', '5', '56']
['1', '2', '3', '4', '5', '56']
```

In [7]:

```
##Len
a=10
print(len(a))
```

```
-----  
TypeError                                     Traceback (most recent call last)  
<ipython-input-7-8a3c348f32dc> in <module>
      1 ##len
      2 a=10
----> 3 print(len(a))

TypeError: object of type 'int' has no len()
```

In [8]:

```
#Len  
a="the github edition"  
print(len(a))  
print(len("a"))
```

```
18  
1
```

In [9]:

```
#id  
a=100  
print(id(100))  
print(a)  
print(id(a))
```

```
140725948265360  
100  
140725948265360
```

In [10]:

```
#type casting  
a=100  
b=12.4  
print(float(a))  
print(int(b))  
print(float(b))  
print(str(a))  
print(str(b))
```

```
100.0  
12  
12.4  
100  
12.4
```

Conditions

In [12]:

```
#if  
a=int(input())  
if(a>100):  
    print("the number is greater than hundred")  
elif(a<100):  
    print("the number is lower than hundred")  
else:  
    print("the number is 100")
```

```
10  
the number is lower than hundred
```

In [14]:

```
#for
a='the github edition'
for I in range(len(a)):
    print(a[I])
print("-----")
for i in range(10):
    print(a[i])
print("-----")
for i in range(4,13):
    print(a[i])
print("-----")
for i in a:
    print(i)
print("-----")
for i in a.split():
    print(i)
for i in a.split():
    print(i,end="-->")
print("-----")
for i in a.split():
    print(i,end=" ")
```

t
h
e

g
i
t
h
u
b

e
d
i
t
i
o
n

t
h
e

g
i
t
h
u
b

g
i
t
h
u
b

```
e  
d  
-----  
t  
h  
e  
  
g  
i  
t  
h  
u  
b  
  
e  
d  
i  
t  
i  
o  
n  
-----  
the  
github  
edition  
the-->github-->edition-->-----  
the github edition
```

In [15]:

```
#while  
a=2  
while(a%2==0 and a<=10):  
    print(a)  
    a+=1
```

2

In [16]:

```
a=2
while(0<=a<=10):
    if(a%2==0):
        print(a)
    a+=1
print('_____')
for i in range(1,11):
    if(i%2==1):
        print(i)
```

2
4
6
8
10

1
3
5
7
9

Ascii Values

In [18]:

```
print(ord('a'))
print(ord("g"))
print(ord('z'))
print(ord("A"))
print(ord("G"))
print(ord("Z"))
print(chr(97))
print(chr(103))
print(chr(122))
print(chr(65))
print(chr(71))
print(chr(90))
print(ord('a'), '-->', chr(97))
print(ord('A'), '-->', chr(65))
print(ord('z'), '-->', chr(122))
print(ord('Z'), '-->', chr(90))
print(ord("1"), '-->', ord('9'))
```

```
97
103
122
65
71
90
a
g
z
A
G
Z
97 --> a
65 --> A
122 --> z
90 --> Z
49 --> 57
```

Operators

Arthemetic Operators

In [22]:

```
a=1
b=10
```

In [23]:

```
a+b
```

Out[23]:

11

In [24]:

```
a-b
```

Out[24]:

-9

In [25]:

```
a*b
```

Out[25]:

10

In [26]:

```
a/b
```

Out[26]:

0.1

In [27]:

```
a//b
```

Out[27]:

0

In [28]:

```
a%b
```

Out[28]:

1

In [29]:

```
a**b
```

Out[29]:

1

Bitwise Operators

In [31]:

```
a&b
```

Out[31]:

0

```
1-->0001  
10-->1010  
a&b-->0000  
a|b-->1011  
a^b-->1011
```

In [32]:

```
a|b
```

Out[32]:

11

In [33]:

```
a^b
```

Out[33]:

11

In [34]:

```
a=1
```

In [35]:

```
~a
```

Out[35]:

-2

```
a>>n--->a//2-->n times  
a<<n--->a*2--->n times
```

In [36]:

```
a>>2
```

Out[36]:

0

In [37]:

```
a=10  
a>>2
```

Out[37]:

2

In [38]:

```
a<<2
```

Out[38]:

40

In [39]:

```
a<<3
```

Out[39]:

80

In [40]:

```
a=2  
b=6
```

In [41]:

```
a&b
```

Out[41]:

2

In [42]:

```
a|b
```

Out[42]:

6

Local Operators

In [61]:

```
a<10 and b==10
```

Out[61]:

False

In [62]:

```
a==10 and b==10
```

Out[62]:

False

In [63]:

```
a==10 or b==10
```

Out[63]:

False

In [64]:

```
a and b
```

Out[64]:

1

In [65]:

```
a or b
```

Out[65]:

12

In [66]:

```
a=10  
b=1  
a and b
```

Out[66]:

1

In [67]:

```
a or b
```

Out[67]:

10

In [68]:

```
not a
```

Out[68]:

False

Assignment Operators

In [69]:

```
a==b
```

Out[69]:

False

In [70]:

```
a>b
```

Out[70]:

True

In [71]:

```
a>a
```

Out[71]:

False

In [72]:

```
a==a
```

Out[72]:

True

In [73]:

```
a>=b
```

Out[73]:

True

In [74]:

```
a<=b
```

Out[74]:

False

In [75]:

```
a+=b
```

```
a
```

Out[75]:

11

In [76]:

```
a-=b  
a
```

Out[76]:

10

In [77]:

```
a&=b  
a
```

Out[77]:

0

In [78]:

```
a=10  
b=1  
a|=b  
a
```

Out[78]:

11

Identity Operators

is, not is

In [80]:

```
a="the github edition"  
if(a not is "the github edition"):  
    print("true")  
else:  
    print("false")
```

```
File "<ipython-input-80-e90b0c9cd794>", line 2  
  if(a not is "the github edition"):  
          ^
```

SyntaxError: invalid syntax

In [81]:

```
a="the github edition"
if(a is not "the github edition"):
    print("true")
else:
    print("false")
```

true

```
<>:2: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:2: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<ipython-input-81-138f1584b9f7>:2: SyntaxWarning: "is not" with a literal. D
id you mean "!="?
    if(a is not "the github edition"):
```

In [82]:

```
#Membership operators are in ,not in.
l=[1,2,3,4,5]
if(9 in l):
    print("9 in l")
elif(3 in l):
    print("3 in l")
else:
    print("Check the number")
```

3 in l

In [83]:

```
9 in l
```

Out[83]:

False

In [84]:

```
9 not in l
```

Out[84]:

True

In [85]:

```
3 in l
```

Out[85]:

True

Operator Presedence

Operators	Meaning

```
() Parentheses
** Exponent
+x, -x, ~x Unary plus, Unary minus, Bitwise NOT
*, /, //, % Multiplication, Division, Floor division, Modulus
+, - Addition, Subtraction
<<, >> Bitwise shift operators
& Bitwise AND
^ Bitwise XOR
| Bitwise OR
==, !=, >, >=, <, <=, is, is not, in, not in Comparisons, Identity, Membership
operators
not Logical NOT
and Logical AND
or Logical OR
```

In [87]:

```
a=10
b=2
c=4
d=5
e=3
f=8
a+b//c-d*e/f
#15/8=1.,010-1
```

Out[87]:

8.125

In [88]:

```
3/8
```

Out[88]:

0.375

In [89]:

```
2//4
```

Out[89]:

0

In [90]:

```
15/8
```

Out[90]:

1.875

In [91]:

```
8.125+1.875
```

Out[91]:

```
10.0
```

Lists

In [93]:

```
l=[1,2,3,4,5]
```

In [94]:

```
l[0]
```

Out[94]:

```
1
```

In [95]:

```
l[10]
```

IndexError

Traceback (most recent call last)

```
<ipython-input-95-f97e037cf13> in <module>
----> 1 l[10]
```

IndexError: list index out of range

In [97]:

```
l[-1]
```

Out[97]:

```
5
```

In [98]:

```
l[-2]
```

Out[98]:

```
4
```

In [99]:

```
1[-10]
```

```
-----  
IndexError                                                 Traceback (most recent call last)  
<ipython-input-99-080d95ce6835> in <module>  
----> 1 l[-10]
```

IndexError: list index out of range

In [100]:

```
l[len(l)-1]
```

Out[100]:

5

In [101]:

```
l=[]  
for i in range(10):  
    l.append(i)
```

In [102]:

```
l
```

Out[102]:

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

In [103]:

```
g=[]  
for i in range(10):  
    g.extend(input())
```

```
the  
github  
edition  
-  
python  
basic  
hello  
world  
welcome  
to
```

In [104]:

```
g
```

Out[104]:

```
['t',
 'h',
 'e',
 'g',
 'i',
 't',
 'h',
 'u',
 'b',
 'e',
 'd',
 'i',
 't',
 'i',
 'o',
 'n',
 '-',
 'p',
 'y',
 't',
 'h',
 'o',
 'n',
 'b',
 'a',
 's',
 'i',
 'c',
 'h',
 'e',
 'l',
 'l',
 'o',
 'w',
 'o',
 'r',
 'l',
 'd',
 'w',
 'e',
 'l',
 'c',
 'o',
 'm',
 'e',
 't',
 'o']
```

In [105]:

```
q=[11,12,13,14]
```

In [106]:

```
#L=[0,1,2,3,4,5,6,7,8,9]
#q=[11,12,13,14]
#L=[0,1,2,3,4,5,6,7,8,9,11,12,13,14]
```

adding elements in list

In [108]:

```
l.append(q)
```

In [109]:

```
l
```

Out[109]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [11, 12, 13, 14]]
```

In [110]:

```
l.extend(q)
```

In [111]:

```
l
```

Out[111]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [11, 12, 13, 14], 11, 12, 13, 14]
```

In [112]:

```
l[1]
```

Out[112]:

```
1
```

In [113]:

```
l[10]
```

Out[113]:

```
[11, 12, 13, 14]
```

deleting elements in list

In [115]:

```
l.remove(13)
```

In [116]:

```
1
```

Out[116]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [11, 12, 13, 14], 11, 12, 14]
```

In [117]:

```
l.pop()
```

Out[117]:

```
14
```

In [118]:

```
1
```

Out[118]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [11, 12, 13, 14], 11, 12]
```

In [119]:

```
l.pop(4)
```

Out[119]:

```
4
```

In [120]:

```
del l[4]
```

In [121]:

```
1
```

Out[121]:

```
[0, 1, 2, 3, 6, 7, 8, 9, [11, 12, 13, 14], 11, 12]
```

In [122]:

```
l.remove(1)
```

In [123]:

```
1
```

Out[123]:

```
[0, 2, 3, 6, 7, 8, 9, [11, 12, 13, 14], 11, 12]
```

In [124]:

```
l.pop(7)
```

Out[124]:

```
[11, 12, 13, 14]
```

In [125]:

```
del l[10]
```

```
-----  
IndexError
```

```
Traceback (most recent call last)
```

```
<ipython-input-125-5783abe449d2> in <module>
```

```
----> 1 del l[10]
```

```
IndexError: list assignment index out of range
```

count

In [130]:

```
l=[]
for i in range(10):
    if(i%2==0):
        l.append(2)
    elif(i%2==1):
        l.append(1)
```

In [131]:

```
l
```

Out[131]:

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

In [132]:

```
l.count(1)
```

Out[132]:

```
5
```

In [133]:

```
l.count(2)
```

Out[133]:

```
5
```

In [134]:

```
l.count(3)
```

Out[134]:

```
0
```

In [135]:

```
#sorting
```

In [136]:

```
l=[1,2,3,1000,89,97,65,116,111,"E4","E3",108]
```

In [137]:

```
l.sort()
```

```
-----  
TypeError                                                 Traceback (most recent call last)  
<ipython-input-137-fb07ac7c73ab> in <module>  
----> 1 l.sort()  
-----
```

```
TypeError: '<' not supported between instances of 'str' and 'int'
```

In [138]:

```
l=[1,2,116,189,9987,1098,156,116,108,104,100,1008,1024,2779]
```

In [139]:

```
l
```

Out[139]:

```
[1, 2, 116, 189, 9987, 1098, 156, 116, 108, 104, 100, 1008, 1024, 2779]
```

In [142]:

```
l.sort()  
l
```

Out[142]:

```
[1, 2, 100, 104, 108, 116, 116, 156, 189, 1008, 1024, 1098, 2779, 9987]
```

In [144]:

```
l.sort(reverse=True)  
l
```

Out[144]:

```
[9987, 2779, 1098, 1024, 1008, 189, 156, 116, 116, 108, 104, 100, 2, 1]
```

In [145]:

```
#reversing elements in a List  
l.reverse()  
l
```

Out[145]:

```
[1, 2, 100, 104, 108, 116, 116, 156, 189, 1008, 1024, 1098, 2779, 9987]
```

In [146]:

```
l=[1,2,3,4,5,6,7,8,9,10]
```

In [150]:

```
#sum of elements in list  
c=0  
for i in range(len(l)):  
    c+=l[i]
```

In [151]:

```
c
```

Out[151]:

```
55
```

In [153]:

```
d=0  
for i in l:  
    d+=i  
d
```

Out[153]:

```
55
```

In [154]:

```
sum(l)
```

Out[154]:

```
55
```

In [155]:

```
#inserting a particular element in a list
```

In [335]:

```
l=["the 1","github 2","edition 3"]
l
```

Out[335]:

```
['the 1', 'github 2', 'edition 3']
```

In [336]:

```
l.insert("the github edition 4")
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-336-4dad239e47d9> in <module>
----> 1 l.insert("the github edition 4")
```

```
TypeError: insert expected 2 arguments, got 1
```

In [337]:

```
l.insert("the github edition 1",1)
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-337-f7c3b30f093d> in <module>
----> 1 l.insert("the github edition 1",1)
```

```
TypeError: 'str' object cannot be interpreted as an integer
```

In [338]:

```
l.insert(1,"the github edition 1")
l
```

Out[338]:

```
['the 1', 'the github edition 1', 'github 2', 'edition 3']
```

In [339]:

```
l.insert(4,"this is the new element added in the list")
l
```

Out[339]:

```
['the 1',
 'the github edition 1',
 'github 2',
 'edition 3',
 'this is the new element added in the list']
```

In [340]:

```
len(1)
```

Out[340]:

```
5
```

In [341]:

```
l.insert(10,"this will show some error")  
l
```

Out[341]:

```
['the 1',  
 'the github edition 1',  
 'github 2',  
 'edition 3',  
 'this is the new element added in the list',  
 'this will show some error']
```

In [342]:

```
l[10]
```

IndexError Traceback (most recent call last)
<ipython-input-342-f97e037cf13> in <module>
----> 1 l[10]

IndexError: list index out of range

In [343]:

```
l[5]
```

Out[343]:

```
'this will show some error'
```

finding index of a particular element

In [178]:

```
l=[111,222,333,444,555]
```

In [179]:

```
l.index(555)
```

Out[179]:

```
4
```

In [180]:

```
l.index(1555)
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-180-47e90d0b80f6> in <module>  
----> 1 l.index(1555)
```

ValueError: 1555 is not in list

In [181]:

```
l=[111,123,123,1111,111,123,111]
```

In [182]:

```
l.index(111)
```

Out[182]:

```
0
```

In [188]:

```
d=l.count(111)
print("intially the list is",l)
for i in range(d):
    c=l.index(111)
    print(c)
    del l[c]
print('after deleting a particular element in list',l)
```

intially the list is [123, 123, 1111, 123]

In [189]:

```
for i in range(d):
    c=l.index(111)
    print(c)
    del l[c]
print('after deleting a particular element in list',l)
```

after deleting a particular element in list [123, 123, 1111, 123]

In [187]:

```
l=[111,123,123,1111,111,123,111]
print("intially l is",l)
a=l.count(111)
for i in range(a):
    b=l.index(111)
    print("the index of 111 is",b)
    print("before poping the elements in the list",l)
    l.pop(b)
    print("after poping the elements in the list",l)
```

```
intially l is [111, 123, 123, 1111, 111, 123, 111]
the index of 111 is 0
before poping the elements in the list [111, 123, 123, 1111, 111, 123, 111]
after poping the elements in the list [123, 123, 1111, 111, 123, 111]
the index of 111 is 3
before poping the elements in the list [123, 123, 1111, 111, 123, 111]
after poping the elements in the list [123, 123, 1111, 123, 111]
the index of 111 is 4
before poping the elements in the list [123, 123, 1111, 123, 111]
after poping the elements in the list [123, 123, 1111, 123]
```

In [190]:

```
#inputs:1 2 3 4
a=input().split()
```

```
1 2 3 4
```

In [191]:

```
a
```

Out[191]:

```
['1', '2', '3', '4']
```

In [192]:

```
l=[]
for i in a:
    l.append(int(i))
```

In [193]:

```
l
```

Out[193]:

```
[1, 2, 3, 4]
```

In [194]:

```
l=[]
for i in range(len(a)):
    l.append(int(a[i]))
```

In [195]:

```
1
```

Out[195]:

```
[1, 2, 3, 4]
```

In [196]:

```
l=[]
for i in input().split():
    l.append(int(i))
```

```
1 2 3 4
```

In [197]:

```
1
```

Out[197]:

```
[1, 2, 3, 4]
```

In [198]:

```
l=[]
for i in range(len(input().split())):
    l.append(int(a[i]))
```

```
1 2 3 4
```

In [199]:

```
1
```

Out[199]:

```
[1, 2, 3, 4]
```

In [200]:

```
#using list() in -built function
```

In [201]:

```
a="1234567"
l=list(a)
```

In [202]:

```
1
```

Out[202]:

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

In [203]:

```
a[0]
```

Out[203]:

```
'1'
```

removing repeated elements in a list

In [205]:

```
l=[1,2,3,1,2,4,5,6,1,2,3,4,5,6]
```

In [206]:

```
g=[]
```

In [207]:

```
g
```

Out[207]:

```
[]
```

In [208]:

```
for i in l:  
    if(i not in g):  
        g.append(i)
```

In [209]:

```
g
```

Out[209]:

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

In [210]:

```
c=0  
for i in g:  
    d=l.count(i)  
    if(d>1):  
        c+=1
```

In [211]:

```
c
```

Out[211]:

```
6
```

slicing list elements

In [213]:

```
l=[1,2,3,4,5]
```

In [214]:

```
l[0:2]
```

Out[214]:

```
[1, 2]
```

In [215]:

```
l[1:3]
```

Out[215]:

```
[2, 3]
```

In [216]:

```
l=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
```

In [217]:

```
l[2:14:1]
```

Out[217]:

```
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

In [218]:

```
l[2:14:]
```

Out[218]:

```
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

In [219]:

```
l[2:14:3]
```

Out[219]:

```
[3, 6, 9, 12]
```

In [220]:

```
l[::-1]
```

Out[220]:

```
[20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [221]:

```
1[-1:6:0]
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-221-f092b71cdb7f> in <module>  
----> 1 1[-1:6:0]
```

ValueError: slice step cannot be zero

In [222]:

```
1[len(1)-1::-2]
```

Out[222]:

```
[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]
```

In [223]:

```
1[::-2]
```

Out[223]:

```
[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]
```

In [224]:

```
1[:4:-2]
```

Out[224]:

```
[20, 18, 16, 14, 12, 10, 8, 6]
```

In [225]:

```
1[4:-1:-1]
```

Out[225]:

```
[]
```

I[starting_index:destination:difference]

In [227]:

```
1[4:len(1)-1:-1]
```

Out[227]:

```
[]
```

In [228]:

```
l[0:len(l)-2]
```

Out[228]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
```

In [229]:

```
l[0:len(l)-1:1]
```

Out[229]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [230]:

```
a=int(input())#---> to tell how many elements in the List
b=[]
for i in range(a):
    b.append(input())
c=b
print(b)
print(c)
```

```
5
the
github
edition
python
basics
['the', 'github', 'edition', 'python', 'basics']
['the', 'github', 'edition', 'python', 'basics']
```

Tuples

In [232]:

```
t=(1,2,3,4,5)
```

In [233]:

```
t
```

Out[233]:

```
(1, 2, 3, 4, 5)
```

In [234]:

```
t[1]
```

Out[234]:

```
2
```

In [235]:

```
t[6]
```

```
-----  
IndexError                                     Traceback (most recent call last)  
<ipython-input-235-d806090743c9> in <module>  
----> 1 t[6]
```

IndexError: tuple index out of range

In [236]:

```
t[1]=2
```

```
-----  
TypeError                                     Traceback (most recent call last)  
<ipython-input-236-77a05d522e8c> in <module>  
----> 1 t[1]=2
```

TypeError: 'tuple' object does not support item assignment

In [237]:

```
l=[1,2,4,5]
```

In [238]:

```
l
```

Out[238]:

```
[1, 2, 4, 5]
```

In [239]:

```
l[1]
```

Out[239]:

```
2
```

In [240]:

```
l[1]=10
```

In [241]:

```
l
```

Out[241]:

```
[1, 10, 4, 5]
```

In [242]:

```
t[2]=4
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-242-eaddc54f0039> in <module>  
----> 1 t[2]=4
```

TypeError: 'tuple' object does not support item assignment

In [243]:

```
list(t)
```

Out[243]:

```
[1, 2, 3, 4, 5]
```

In [244]:

```
t
```

Out[244]:

```
(1, 2, 3, 4, 5)
```

In [246]:

```
q=list(t)  
q
```

Out[246]:

```
[1, 2, 3, 4, 5]
```

In [247]:

```
q[3]=10  
q
```

Out[247]:

```
[1, 2, 3, 10, 5]
```

In [249]:

```
t=tuple(q)  
t
```

Out[249]:

```
(1, 2, 3, 10, 5)
```

In [250]:

```
t.pop()
```

```
-----  
AttributeError                                     Traceback (most recent call last)  
<ipython-input-250-f05cb48c71e5> in <module>  
----> 1 t.pop()  

```

AttributeError: 'tuple' object has no attribute 'pop'

In [251]:

```
del t[0]
```

```
-----  
TypeError                                     Traceback (most recent call last)  
<ipython-input-251-542f9bb8f15e> in <module>  
----> 1 del t[0]  

```

TypeError: 'tuple' object doesn't support item deletion

In [252]:

```
remove(0)
```

```
-----  
NameError                                     Traceback (most recent call last)  
<ipython-input-252-57feed64aff9> in <module>  
----> 1 remove(0)  

```

NameError: name 'remove' is not defined

In [253]:

```
t.remove(0)
```

```
-----  
AttributeError                                     Traceback (most recent call last)  
<ipython-input-253-b8ff8c81a830> in <module>  
----> 1 t.remove(0)  

```

AttributeError: 'tuple' object has no attribute 'remove'

In [254]:

```
t.index(1)
```

Out[254]:

0

In [255]:

```
t.index(5)
```

Out[255]:

```
4
```

In [256]:

```
t.index(51)
```

```
-----
ValueError                                     Traceback (most recent call last)
<ipython-input-256-a4ad29ef6f22> in <module>
----> 1 t.index(51)

ValueError: tuple.index(x): x not in tuple
```

In [257]:

```
t
```

Out[257]:

```
(1, 2, 3, 10, 5)
```

In [258]:

```
t.count()
```

```
-----
TypeError                                     Traceback (most recent call last)
<ipython-input-258-9ccfa7eed41a> in <module>
----> 1 t.count()

TypeError: count() takes exactly one argument (0 given)
```

In [259]:

```
t.count(1)
```

Out[259]:

```
1
```

In [262]:

```
t=(1,2,3,4,6,5)
q=(10,11,12,13,14,15)
```

In [264]:

```
w=t+q  
w
```

Out[264]:

```
(1, 2, 3, 4, 6, 5, 10, 11, 12, 13, 14, 15)
```

In [265]:

```
sum(t)
```

Out[265]:

```
21
```

In [266]:

```
t=(1,2,3,4,5,6,7,8,9)  
print("The tuples is",t)  
q=list(t)  
print("the list is",q)  
print("the first indexed value",q[1])  
q[1]=10  
print("After assigning new value",q)  
q.append(11)  
print("After appending",q)  
del q[4]  
print("After deleting",q)  
q.remove(10)  
print("After removing element",q)  
q.pop()  
print("After poping",q)  
print("Finally making the list to tuple")  
t_upd=tuple(q)  
print("the updated tuple is",t_upd)
```

```
The tuples is (1, 2, 3, 4, 5, 6, 7, 8, 9)  
the list is [1, 2, 3, 4, 5, 6, 7, 8, 9]  
the first indexed value 2  
After assigning new value [1, 10, 3, 4, 5, 6, 7, 8, 9]  
After appending [1, 10, 3, 4, 5, 6, 7, 8, 9, 11]  
After deleting [1, 10, 3, 4, 6, 7, 8, 9, 11]  
After removing element [1, 3, 4, 6, 7, 8, 9, 11]  
After poping [1, 3, 4, 6, 7, 8, 9]  
Finally making the list to tuple  
the updated tuple is (1, 3, 4, 6, 7, 8, 9)
```

In [267]:

```
t1=(1,2,3,4,5)
t2=(1,2,3)
t3=t1-t2
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-267-e31d9c044ceb> in <module>
      1 t1=(1,2,3,4,5)
      2 t2=(1,2,3)
----> 3 t3=t1-t2
```

TypeError: unsupported operand type(s) for -: 'tuple' and 'tuple'

In [269]:

```
t1
```

Out[269]:

```
(1, 2, 3, 4, 5)
```

In [270]:

```
t2
```

Out[270]:

```
(1, 2, 3)
```

In [271]:

```
t3=t1*t2
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-271-f8685444531f> in <module>
----> 1 t3=t1*t2
```

TypeError: can't multiply sequence by non-int of type 'tuple'

In [272]:

```
len(t1)
```

Out[272]:

```
5
```

Dictionaries

In [274]:

```
d={'t':1,'g':10}
```

In [275]:

```
d
```

Out[275]:

```
{'t': 1, 'g': 10}
```

In [277]:

```
d={'t':10,'g':20,'t':25}  
d
```

Out[277]:

```
{'t': 25, 'g': 20}
```

In [278]:

```
d['t']
```

Out[278]:

```
25
```

In [279]:

```
d={"t":1,"h":2,"e":3}
```

In [281]:

```
q=d.keys()#to access keys values  
q
```

Out[281]:

```
dict_keys(['t', 'h', 'e'])
```

In [282]:

```
type(q)
```

Out[282]:

```
dict_keys
```

In [283]:

```
type(d)
```

Out[283]:

```
dict
```

In [284]:

```
len(d)
```

Out[284]:

3

In [285]:

```
#accessing key values
for i in d:
    print(i)
```

t
h
e

In [286]:

```
type(i)
```

Out[286]:

str

In [288]:

```
#accesing the values of keys in dictionaries
a=d.values()
a
```

Out[288]:

```
dict_values([1, 2, 3])
```

In [289]:

```
type(a)
```

Out[289]:

```
dict_values
```

In [290]:

```
from keys in d:
    print(keys)
```

File "<ipython-input-290-a591a5cee550>", line 1
 from keys in d:
 ^

SyntaxError: invalid syntax

In [291]:

```
#accesing elements in dictionaries  
d["t"]
```

Out[291]:

1

In [292]:

```
#accesing elements in dictionaries  
d.get('t')
```

Out[292]:

1

In [293]:

```
d.items()
```

Out[293]:

```
dict_items([('t', 1), ('h', 2), ('e', 3)])
```

In [295]:

```
d={"1":"the","2":"github","3":"edition"}  
d
```

Out[295]:

```
{'1': 'the', '2': 'github', '3': 'edition'}
```

In [296]:

```
d.keys()
```

Out[296]:

```
dict_keys(['1', '2', '3'])
```

In [297]:

```
d.values()
```

Out[297]:

```
dict_values(['the', 'github', 'edition'])
```

In [298]:

```
d[1]
```

```
-----  
KeyError                                                 Traceback (most recent call last)  
<ipython-input-298-abe283337115> in <module>  
----> 1 d[1]
```

KeyError: 1

In [299]:

```
d['1']
```

Out[299]:

'the'

In [300]:

```
d.get("1")
```

Out[300]:

'the'

In [301]:

```
d1={"4":"the github edition"}
```

In [302]:

```
type(d1)
```

Out[302]:

dict

In [303]:

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

1

2

3

In [304]:

```
for i in d:  
    print(d[i])
```

the
github
edition

In [306]:

```
d["github"]=5  
d
```

Out[306]:

```
{'1': 'the', '2': 'github', '3': 'edition', 'github': 5}
```

In [308]:

```
d['g']=10  
d
```

Out[308]:

```
{'1': 'the', '2': 'github', '3': 'edition', 'github': 5, 'g': 10}
```

In [310]:

```
d1={"the":6,"the github":7,"python":8}  
d1
```

Out[310]:

```
{'the': 6, 'the github': 7, 'python': 8}
```

In [311]:

```
d,d1
```

Out[311]:

```
({'1': 'the', '2': 'github', '3': 'edition', 'github': 5, 'g': 10},  
{'the': 6, 'the github': 7, 'python': 8})
```

In [312]:

```
d.update(d1)  
d
```

Out[312]:

```
{'1': 'the',  
 '2': 'github',  
 '3': 'edition',  
 'github': 5,  
 'g': 10,  
 'the': 6,  
 'the github': 7,  
 'python': 8}
```

In [313]:

```
del d["g"]
```

In [314]:

```
d
```

Out[314]:

```
{'1': 'the',
'2': 'github',
'3': 'edition',
'github': 5,
'the': 6,
'the github': 7,
'python': 8}
```

In [315]:

```
d.pop("o")
```

KeyError

Traceback (most recent call last)

```
<ipython-input-315-0ceda8290266> in <module>
----> 1 d.pop("o")
```

KeyError: 'o'

In [316]:

```
d.pop("the")
```

Out[316]:

6

In [317]:

```
d
```

Out[317]:

```
{'1': 'the',
'2': 'github',
'3': 'edition',
'github': 5,
'the github': 7,
'python': 8}
```

In [318]:

```
clear(d)
```

In [319]:

```
d
```

Out[319]:

```
{'1': 'the',
'2': 'github',
'3': 'edition',
'github': 5,
'the github': 7,
'python': 8}
```

In [320]:

```
d.clear()
```

In [321]:

```
d
```

Out[321]:

```
{}
```

In [322]:

```
d1={"the":6,"the github":7,"python":8}
d1
```

Out[322]:

```
{'the': 6, 'the github': 7, 'python': 8}
```

In [323]:

```
d.copy(d1)
```

```
-----  
TypeError                                     Traceback (most recent call last)
<ipython-input-323-da7131399020> in <module>
----> 1 d.copy(d1)
```

TypeError: copy() takes no arguments (1 given)

In [324]:

```
d.copy()
```

Out[324]:

```
{}
```

In [325]:

```
d1.copy()
```

Out[325]:

```
{'the': 6, 'the github': 7, 'python': 8}
```

In [326]:

```
d={"g":1,"i":2,"t":3,"h":4}
d1={"the":6,"the github":7,"python":8}
print("the type of d is",d)
q=d.keys()
print("the keys in the d are",q)
q1=d1.keys()
print("the keys in the d1 are",q1)
q2=d.values()
print("the values in the d are",q2)
q3=d1.values()
print("the values in the d1 are",q3)
d.update(d1)
print("the updated dictionary is",d)
print("the key-g value is",d["g"])
del d["g"]
print("After deleting the d becomes",d)
d.pop("t")
print("After popping the d becomes",d)
d["the github edition"]=1000
print("after assigning new value",d)
d.clear()
print("After clearing the elements ",d)
```

```
the type of d is {'g': 1, 'i': 2, 't': 3, 'h': 4}
the keys in the d are dict_keys(['g', 'i', 't', 'h'])
the keys in the d1 are dict_keys(['the', 'the github', 'python'])
the values in the d are dict_values([1, 2, 3, 4])
the values in the d1 are dict_values([6, 7, 8])
the updated dictionary is {'g': 1, 'i': 2, 't': 3, 'h': 4, 'the': 6, 'the gi
thub': 7, 'python': 8}
the key-g value is 1
After deleting the d becomes {'i': 2, 't': 3, 'h': 4, 'the': 6, 'the githu
b': 7, 'python': 8}
After popping the d becomes {'i': 2, 'h': 4, 'the': 6, 'the github': 7, 'pyth
on': 8}
after assigning new value {'i': 2, 'h': 4, 'the': 6, 'the github': 7, 'pytho
n': 8, 'the github edition': 1000}
After clearing the elements  {}
```

In [344]:

```
#Dictionaries example
#4
#1 2
#2 4
#3 6
#4 1
#o:{'1':2, '2':4, '3':6, '4':1}
a=int(input())
d={}
for i in range(a):
    q=input().split()
    d[q[0]]=int(q[1])
print(d)
```

```
4
1 2
2 4
3 6
4 8
{'1': 2, '2': 4, '3': 6, '4': 8}
```

In [345]:

```
a=int(input())
d={}
for i in range(a):
    q=input().split()
    d1={q[0]:int(q[1])}
    d.update(d1)
print(d)
```

```
6
the 2
git 3
hub 4
python 5
basic 6
thankyou 7
{'the': 2, 'git': 3, 'hub': 4, 'python': 5, 'basic': 6, 'thankyou': 7}
```

In [346]:

```
a=int(input())
d={}
for i in range(a):
    q=input().split()
    d1={q[0]:int(q[1])}
    d.update(d1)
print(d)
for i in d:
    if(d[i]%2==0):
        print("the even key is :",i)
    else:
        print("The odd key is :",i)
```

```
4
the 3
git 3
hub 3
python 6
{'the': 3, 'git': 3, 'hub': 3, 'python': 6}
The odd key is : the
The odd key is : git
The odd key is : hub
the even key is : python
```

In [347]:

```
"the" in d
```

Out[347]:

True

In [348]:

```
"qwerty" in d
```

Out[348]:

False

In [349]:

```
"git" is 2
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<ipython-input-349-069cd699df31>:1: SyntaxWarning: "is" with a literal. Did
you mean "=="?
    "git" is 2
```

Out[349]:

False

In [350]:

```
"qwerty" not in d
```

Out[350]:

True

```
tuples-->immutable  
lists,dict---> mutable
```

In [352]:

```
d={"t":123,"h":1234,"e":123456}  
print("d.keys()",d.keys())  
print("d.items()",d.items())  
print("d.values()",d.values())
```

```
d.keys() dict_keys(['t', 'h', 'e'])  
d.items() dict_items([('t', 123), ('h', 1234), ('e', 123456)])  
d.values() dict_values([123, 1234, 123456])
```

In [353]:

```
d["t"]
```

Out[353]:

123

In [354]:

```
d["t"] = 123456789  
d
```

Out[354]:

```
{'t': 123456789, 'h': 1234, 'e': 123456}
```

In [356]:

```
d1={"g":123}  
d.update(d1)  
d
```

Out[356]:

```
{'t': 123456789, 'h': 1234, 'e': 123456, 'g': 123}
```

In [357]:

```
"h" in d
```

Out[357]:

True

In [358]:

```
a={1,2,3,4}  
b={1,2,66,9}
```

In [359]:

```
a-b
```

Out[359]:

```
{3, 4}
```

In [360]:

```
a.intersection(b)
```

Out[360]:

```
{1, 2}
```

In [361]:

```
b-a
```

Out[361]:

```
{9, 66}
```

In [362]:

```
a.union(b)
```

Out[362]:

```
{1, 2, 3, 4, 9, 66}
```

In [363]:

```
a^b
```

Out[363]:

```
{3, 4, 9, 66}
```

In [364]:

```
a.issubset(b)
```

Out[364]:

```
False
```

In [365]:

```
a.issubset(b)
```

Out[365]:

```
False
```

List Comprehension

In [367]:

```
a=8
l=[]
for i in range(a):
    if(i%2==0):
        l.append(i)
print(l)
```

```
[0, 2, 4, 6]
```

In [369]:

```
l1=[i for i in range(a) if(i%2==0)]
l1
```

Out[369]:

```
[0, 2, 4, 6]
```

In [371]:

```
l2=[int(i) for i in input().split()]
l2
```

```
1 2 3 4 5 6 7 8 9 10 11 12 12
```

Out[371]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 12]
```

In [372]:

```
'''  
1 2 3  
4 5 6  
7 8 9  
'''  
l=[]  
for i in range(3):  
    l1=[]  
    for j in range(3):  
        l1.append(int(input()))  
        #l1.append(j)  
    l.append(l1)
```

```
60  
120  
140  
192  
300  
15  
25  
30  
40
```

In [373]:

```
l
```

Out[373]:

```
[[60, 120, 140], [192, 300, 15], [25, 30, 40]]
```

In [374]:

```
l[0][2]
```

Out[374]:

```
140
```

In [375]:

```
l=[]
l1=[]
for i in range(3):
    #l1=[]
    for j in range(3):
        l1.append(int(input()))
    #l1.append(j)
l.append(l1)
```

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

In [376]:

```
1
```

Out[376]:

```
[[1, 2, 3, 5, 6, 7, 8, 9, 0],
 [1, 2, 3, 5, 6, 7, 8, 9, 0],
 [1, 2, 3, 5, 6, 7, 8, 9, 0]]
```

In [378]:

```
d={"t":12,"h":13,"e":14,"g":15,"h":17}
for i in range(len(d)):
    for i in range(2):
        print(d[i][i])
```

```
-----  
KeyError                                                 Traceback (most recent call last)  
<ipython-input-378-0952b73673f4> in <module>  
      2 for i in range(len(d)):  
      3     for i in range(2):  
----> 4         print(d[i][i])
```

KeyError: 0

In [379]:

```
print("the github edition"+" "*5+"the github edition")
```

```
the github edition      the github edition
```

In [380]:

```
print("the github edition"+ " "+ "the github edition")
```

the github edition the github edition

In [381]:

```
for i in range(9):
    for j in range(7):
        if(j==3 or (i==0 and j!=3) or (i==8 and j<3)):
            print("*",end=' ')
        else:
            print(end=" ")
    print()
```

*

*

*

*

*

*

*

Dir

In [382]:

```
dir(input())
```

a

Out[382]:

```
['__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__',  
 'capitalize',  
 'casefold',  
 'center',  
 'count',  
 'encode',  
 'endswith',  
 'expandtabs',  
 'find',  
 'format',  
 'format_map',  
 'index',  
 'isalnum',  
 'isalpha',  
 'isascii',  
 'isdecimal',  
 'isdigit',  
 'isidentifier',  
 'islower',  
 'isnumeric',  
 'isprintable',
```

```
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'replace',
'rfind',
'rindex',
'rjust',
'rpartition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'zfill']
```

In [383]:

```
a=1  
dir(a)
```

Out[383]:

```
['__abs__',  
 '__add__',  
 '__and__',  
 '__bool__',  
 '__ceil__',  
 '__class__',  
 '__delattr__',  
 '__dir__',  
 '__divmod__',  
 '__doc__',  
 '__eq__',  
 '__float__',  
 '__floor__',  
 '__floordiv__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__index__',  
 '__init__',  
 '__init_subclass__',  
 '__int__',  
 '__invert__',  
 '__le__',  
 '__lshift__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__neg__',  
 '__new__',  
 '__or__',  
 '__pos__',  
 '__pow__',  
 '__radd__',  
 '__rand__',  
 '__rdivmod__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rfloordiv__',  
 '__rlshift__',  
 '__rmod__',  
 '__rmul__',  
 '__ror__',  
 '__round__',  
 '__rpow__',  
 '__rrshift__',  
 '__rshift__',  
 '__rsub__',  
 '__rtruediv__',  
 '__rxor__']
```

```
'__setattr__',
'__sizeof__',
'__str__',
'__sub__',
'__subclasshook__',
'__truediv__',
'__trunc__',
'__xor__',
'as_integer_ratio',
'bit_length',
'conjugate',
'denominator',
'from_bytes',
'imag',
'numerator',
'real',
'to_bytes']
```

In [384]:

```
l=[]
dir(l)
```

Out[384]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [385]:

```
dir(int())
```

Out[385]:

```
['__abs__',  
 '__add__',  
 '__and__',  
 '__bool__',  
 '__ceil__',  
 '__class__',  
 '__delattr__',  
 '__dir__',  
 '__divmod__',  
 '__doc__',  
 '__eq__',  
 '__float__',  
 '__floor__',  
 '__floordiv__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__index__',  
 '__init__',  
 '__init_subclass__',  
 '__int__',  
 '__invert__',  
 '__le__',  
 '__lshift__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__neg__',  
 '__new__',  
 '__or__',  
 '__pos__',  
 '__pow__',  
 '__radd__',  
 '__rand__',  
 '__rdivmod__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rfloordiv__',  
 '__rlshift__',  
 '__rmod__',  
 '__rmul__',  
 '__ror__',  
 '__round__',  
 '__rpow__',  
 '__rrshift__',  
 '__rshift__',  
 '__rsub__',  
 '__rtruediv__',  
 '__rxor__',  
 '__setattr__']
```

```
'__sizeof__',
'__str__',
'__sub__',
'__subclasshook__',
'__truediv__',
'__trunc__',
'__xor__',
'as_integer_ratio',
'bit_length',
'conjugate',
'denominator',
'from_bytes',
'imag',
'numerator',
'real',
'to_bytes']
```

In [386]:

```
d={}
dir(d)
```

Out[386]:

```
['__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'clear',
 'copy',
 'fromkeys',
 'get',
 'items',
 'keys',
 'pop',
 'popitem',
 'setdefault',
 'update',
 'values']
```

In [387]:

```
t=()
dir(t)
```

Out[387]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'count',
 'index']
```

Help

In [388]:

```
help(int)
```

```
__divmod__(self, value, /)
    Return divmod(self, value).

__eq__(self, value, /)
    Return self==value.

__float__(self, /)
    float(self)

__floor__(...)
    Flooring an Integral returns itself.

__floordiv__(self, value, /)
    Return self//value.

__format__(self, format_spec, /)
    Default object formatter.

__ge__(self, value, /)
```

In [391]:

```
help(input)
```

Help on method raw_input in module ipykernel.kernelbase:

```
raw_input(prompt='') method of ipykernel.ipkernel.IPythonKernel instance
    Forward raw_input to frontends
```

Raises

```
StdinNotImplementedError if active frontend doesn't support stdin.
```

In [392]:

```
l=[]  
help(l)
```

```
| If a key function is given, apply it once to each list item and s  
ort them,  
| ascending or descending, according to their function values.  
  
| The reverse flag can be set to sort in descending order.  
  
-----  
  
| Static methods defined here:  
  
| __new__(*args, **kwargs) from builtins.type  
|     Create and return a new object. See help(type) for accurate sign  
ature.  
  
-----  
  
| Data and other attributes defined here:  
  
| __hash__ = None
```

In [393]:

```
help(d)
```

Help on dict object:

```
class dict(object)  
| dict() -> new empty dictionary  
| dict(mapping) -> new dictionary initialized from a mapping object's  
|   (key, value) pairs  
| dict(iterable) -> new dictionary initialized as if via:  
|   d = {}  
|   for k, v in iterable:  
|       d[k] = v  
| dict(**kwargs) -> new dictionary initialized with the name=value pair  
s  
|   in the keyword argument list. For example: dict(one=1, two=2)  
  
| Built-in subclasses:  
|   StgDict  
  
| Methods defined here:  
  
----- /---\
```

In [394]:

```
help(t)
```

Help on tuple object:

```
class tuple(object)
| tuple(iterable=(), /)

| Built-in immutable sequence.

| If no argument is given, the constructor returns an empty tuple.
| If iterable is specified the tuple is initialized from iterable's ite
ms.

| If the argument is a tuple, the return value is the same object.

Built-in subclasses:
    asyncgen_hooks
    UnraisableHookArgs

Methods defined here:

__add__(self, value, /)
    Return self+value.

__contains__(self, key, /)
    Return key in self.

__eq__(self, value, /)
    Return self==value.

__ge__(self, value, /)
    Return self>=value.

__getattribute__(self, name, /)
    Return getattr(self, name).

__getitem__(self, key, /)
    Return self[key].

__getnewargs__(self, /)

__gt__(self, value, /)
    Return self>value.

__hash__(self, /)
    Return hash(self).

__iter__(self, /)
    Implement iter(self).

__le__(self, value, /)
    Return self<=value.

__len__(self, /)
    Return len(self).

__lt__(self, value, /)
    Return self<value.
```

```
__mul__(self, value, /)
    Return self*value.

__ne__(self, value, /)
    Return self!=value.

__repr__(self, /)
    Return repr(self).

__rmul__(self, value, /)
    Return value*self.

count(self, value, /)
    Return number of occurrences of value.

index(self, value, start=0, stop=9223372036854775807, /)
    Return first index of value.

    Raises ValueError if the value is not present.
```

Static methods defined here:

```
__new__(*args, **kwargs) from builtins.type
    Create and return a new object. See help(type) for accurate signature.
```

In [395]:

```
help(help)
```

Help on _Helper in module _sitebuiltins object:

```
class _Helper(builtins.object)
| Define the builtin 'help'.
|
| This is a wrapper around pydoc.help that provides a helpful message
| when 'help' is typed at the Python interactive prompt.
|
| Calling help() at the Python prompt starts an interactive help session.
| Calling help(thing) prints help for the python object 'thing'.
|
| Methods defined here:
|
| __call__(self, *args, **kwds)
|     Call self as a function.
|
| __repr__(self)
|     Return repr(self).
|
|-----|
| Data descriptors defined here:
|
| __dict__
|     dictionary for instance variables (if defined)
|
| __weakref__
|     list of weak references to the object (if defined)
```

Thank You!