2

In [3]:
```
print(8 / 3)
```

```
2.6666666666666665
```

In [5]:
```
print(-8//3)
```

```
-3
```

In [6]:
```
print(-8/3)
```

```
-2.6666666666666665
```

In [7]:
```
print(8//-3)
```

```
-3
```

In [8]:
```
print(-(8//3))
```

```
-2
```

In [ ]:

In [ ]:

In [ ]:

-3

In [6]:
```
print(-8/3)
```

```
-2.6666666666666665
```

In [7]:
```
print(8//-3)
```

```
-3
```

In [8]:
```
print(-(8//3))
```

```
-2
```

In [9]:
```
a = 60
if a > 50:
    print("FIRST")
if a > 40:
    print("SECOND")
```

```
FIRST
SECOND
```

In [ ]:

In [ ]:

In [ ]:


```
FIRST
```

Data Structures

Lists

In [11]:
```
a = ["akash", 5, True, print, 5.67, [1, 2, 3]]
```

In [12]:
```
type(a)
```

Out[12]:
```
list
```

In [13]:
```
len(a)
```

Out[13]:
```
6
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Lists

In [11]:
```
a = ["akash", 5, True, print, 5.67, [1, 2, 3]]
```

In [12]:
```
type(a)
```

Out[12]:
```
list
```

In [13]:
```
len(a)
```

Out[13]:
```
6
```

In [16]:
```
a[0] # 0th index
```

Out[16]:
```
'akash'
```

In [17]:
```
a[1] # 1st index
```

Out[17]:
```
5
```

In [18]:
```
a[-1]
```

Out[18]:
```
[1, 2, 3]
```

In [ ]:

In [ ]:

List Slicing

In [ ]:
```
a = [1,2,3,4,5,6,7,8,9,10]

a[3:8:-1]
```

In [28]:
```
l = [5, 1, 2, 3, 7, 8]

res = l[0:3:1]
print(res)
print(id(res))
```

```
[5, 1, 2]
140477777661760
```

In [29]:
```
l
print(id(l))
```

```
140477777771328
```

In [31]:
```
res = l[2:5] # default jump, inc = 1
print(res)
```

```
[2, 3, 7]
```

In [ ]:

In [ ]:
```
a =

a[3:8:-1]
```

In [28]:
```
l = [5, 1, 2, 3, 7, 8]

res = l[0:3:1]
print(res)
print(id(res))
```

```
[5, 1, 2]
140477777661760
```

In [29]:
```
l
print(id(l))
```

```
140477777771328
```

In [31]:
```
res = l[2:5] # default jump, inc = 1
print(res)
```

```
[2, 3, 7]
```

In [ ]:
```
res = l[:5] # default start = 0, default jump = 1
```

In [ ]:

In [ ]:

14(

In [31]:
```
res = l[2:5] # default jump, inc = 1
print(res)
```

```
[2, 3, 7]
```

In [33]:
```
res = l[:5] # default start = 0, default jump = 1
print(res)
```

```
[5, 1, 2, 3, 7]
```

In [35]:
```
res = l[:] # default start = 0, end = len(l), jump = 1
print(res)
```

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

In [36]:
```
l[::2]
```

Out[36]: [5, 2, 7]

In [ ]:

In [ ]:

In [ ]:

In [33]:
```
res = l[:5] # default start = 0, default jump = 1
print(res)
```

```
[5, 1, 2, 3, 7]
```

In [35]:
```
res = l[:] # default start = 0, end = len(l), jump = 1
print(res)
```

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

In [36]:
```
l[::2]
```

Out[36]: [5, 2, 7]

In [37]:
```
print(l)
```

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

In [38]:
```
l[::15]
```

Out[38]: [5]

In [ ]:

Out[38]: [5

Negative Indexing

In [39]:
```
numbers = [5, 1, 2, 3, 7, 8]*2
```

In [40]:
```
print(numbers)
```

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

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Out[38]: [5

10 / 10

Negative Indexing

In [39]:
```
numbers = [5, 1, 2, 3, 7, 8]*2
```

In [40]:
```
print(numbers)
```

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

In [41]:
```
numbers[-1:-5:-1]
```

Out[41]: [8, 7, 3, 2]

8, 7, 3, 2

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Out[38]: [5]

Negative Indexing

In [39]:
```
numbers = [5, 1, 2, 3, 7, 8]*2
```

In [40]:
```
print(numbers)
```

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

In [41]:
```
numbers[-1:-5:-1]
```

Out[41]: [8, 7, 3, 2]

In [42]:
```
numbers[-1::-1] # negative increment, default end = -len(l) - 1 (till the s
```

Out[42]: [8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1, 5]

In [ ]:

In [ ]:

In [ ]:

N

In [39]:
```
numbers = [5, 1, 2, 3, 7, 8]*2
```

In [40]:
```
print(numbers)
```

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

In [41]:
```
numbers[-1:-5:-1]
```

Out[41]:
```
[8, 7, 3, 2]
```

In [42]:
```
numbers[-1::-1] # negative increment, default end = -len(l) - 1 (till the s
```

Out[42]:
```
[8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1, 5]
```

In [45]:
```
numbers[::-1] # default start = -1, default end = -len(l) - 1, inc = -1
```

Out[45]:
```
[8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1, 5]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [39]:
```
numbers = [5, 1, 2, 3, 7, 8]*2
```

In [40]:
```
print(numbers)
```

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

In [41]:
```
numbers[-1:-5:-1]
```

Out[41]:
```
[8, 7, 3, 2]
```

In [42]:
```
numbers[-1::-1] # negative increment, default end = -len(l) - 1 (till the
```

Out[42]:
```
[8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1, 5]
```

In [45]:
```
numbers[::-1] # default start = -1, default end = -len(l) - 1, inc = -1
```

Out[45]:
```
[8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1, 5]
```

In [47]:
```
numbers[-1:-12:-1]
```

Out[47]:
```
[8, 7, 3, 2, 1, 5, 8, 7, 3, 2, 1]
```

In [ ]:

In [ ]:

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Q

14 / 14

In [50]:
```
numbers = [5, 1, 2, 3, 7, 8]*2

print(numbers)
```

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

default inc = 1

L → R

In [51]:
```
numbers[-1::]
```

Out[51]:
```
[8]
```

In [ ]:

default end = len(l)

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Out[47]: [8

In [ ]:

Quizzes

In [50]:
```
numbers = [5, 1, 2, 3, 7, 8]*2

print(numbers)
```

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

In [52]:
```
numbers[-1::]
```

Out[52]: [8]

In [56]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [57]:
```
numbers[3:5:-1]
```

Out[57]: []

In [ ]:

In [ ]:

In [52]: nu

Out[52]: [8]

In [56]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [57]:
```
numbers[3:5:-1]
```

Out[57]: []

Mix negative and postive index

In [59]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [60]:
```
numbers[-2:4:-1]
```

Out[60]: []

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [52]: nu

Out[52]: [8]

In [56]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [57]:
```
numbers[3:5:-1]
```

Out[57]: []

Mix negative and postive index

In [59]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [60]:
```
numbers[-2:4:-1]
```

Out[60]: []

In [61]:
```
numbers[-2:5:-1]
```

Out[61]: []

In [ ]:

In [ ]:

In [ ]:

In [57]: nu

Out[57]: []

Mix negative and postive index

In [59]:
```
numbers = [5, 1, 2, 3, 7, 8]
```

In [60]:
```
numbers[-2:4:-1]
```

Out[60]: []

In [61]:
```
numbers[-2:5:-1]
```

Out[61]: []

In [ ]:
```
numbers[-2:2:-1]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Out[60]: []

In [61]:
```
numbers[-2:5:-1]
```

Out[61]: []

In [62]:
```
numbers[-2:2:-1]
```

Out[62]: [7, 3]

In [63]:
```
number = [5, 1, 2, 3, 7, 8, 5, 1, 2, 3, 7, 8]
```

In [65]:
```
# see the index as a position not number
number[5:-3:1]
```

Out[65]: [8, 5, 1, 2]

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [75]:
```
numbers = [5, 1, 2, 3, 7, 8, 5, 1, 2, 3, 7, 8]

for i in numbers:
    print(i, end=' ')
```

```
5 1 2 3 7 8 5 1 2 3 7 8
```

Mutability

In [79]:
```
bank_accounts_used_by_me = [3000, 0, 5000]
```

In [80]:
```
bank_accounts_used_by_dad = bank_accounts
```

In [81]:
```
bank_accounts_used_by_me[0] -= 1000
```

In [82]:
```
print(bank_accounts_used_by_me)
```

```
[2000, 0, 5000]
```

In [ ]:

In [ ]:

In [ ]:

Mutability

In [85]:
```
bank_accounts_used_by_me = [3000, 0, 5000]
```

In [86]:
```
bank_accounts_used_by_dad = bank_accounts_used_by_me
```

In [87]:
```
bank_accounts_used_by_me[0] -= 1000
```

In [88]:
```
print(bank_accounts_used_by_me)
```

```
[2000, 0, 5000]
```

In [89]:
```
print(bank_accounts_used_by_dad)
```

```
[2000, 0, 5000]
```

Both list are same?

What?

In [83]:
```
l1 = [3000, 0, 5000] # me
l2 = l1 # dad

l1[0] -= 1000 # I used money
print(l1) # my copy changes
```

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java

Python 3.6
(known limitations)

```
1  l1 = [3000, 0, 5000] # me
2  l2 = l1 # dad
3
4  l1[0] -= 1000 # I used money
5  print(l1) # my copy changes
6  print(l2) # my dad
```

Edit this code

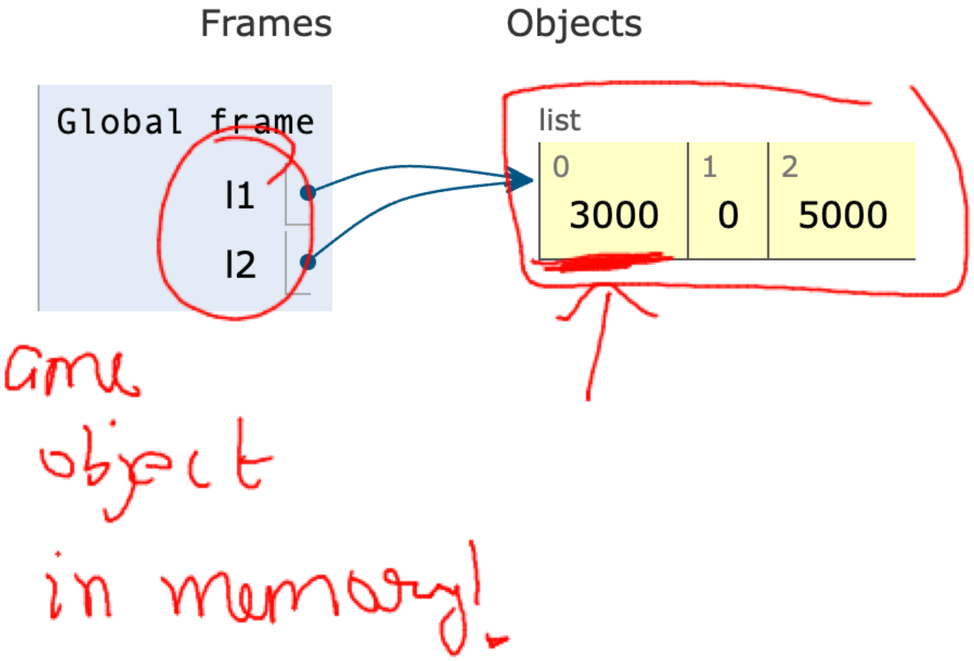
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 3 of 5

Customize visualization

Print output (drag lower right corner to resize)



In [92]: #

In [ ]:
```
l1 = [3, 4, 1, 2, 5]

l2 = l1

l1[0] = 7

print(l1)
print(l2)
```

In [93]:
```
a = 1234
b = a
print(id(a) == id(b))
```

```
True
```

In [94]:
```
a = 1234
b = a
b = 3
print(id(a) == id(b))
```

```
False
```

In [ ]:

In [ ]:

3

MCQ

In [103]:
```
my_teams = ['Raptors', 'Heat', 'Nets']
your_teams = my_teams
print(id(my_teams) == id(your_teams))
```

```
True
```

In [105]:
```
my_teams[1] = 'Lakers'
print(id(my_teams) == id(your_teams))
```

```
True
```

In [106]:
```
print('My teams are:', my_teams)
print('Your teams are:', your_teams)
```

```
My teams are: ['Raptors', 'Lakers', 'Nets']
Your teams are: ['Raptors', 'Lakers', 'Nets']
```

In [ ]:

HW

In [107]:
```
board = [["", "", ""], ["", "", ""], ["", "", ""]]
```

In [ ]:
```
board[0][0]
```

In [ ]:

In [ ]:

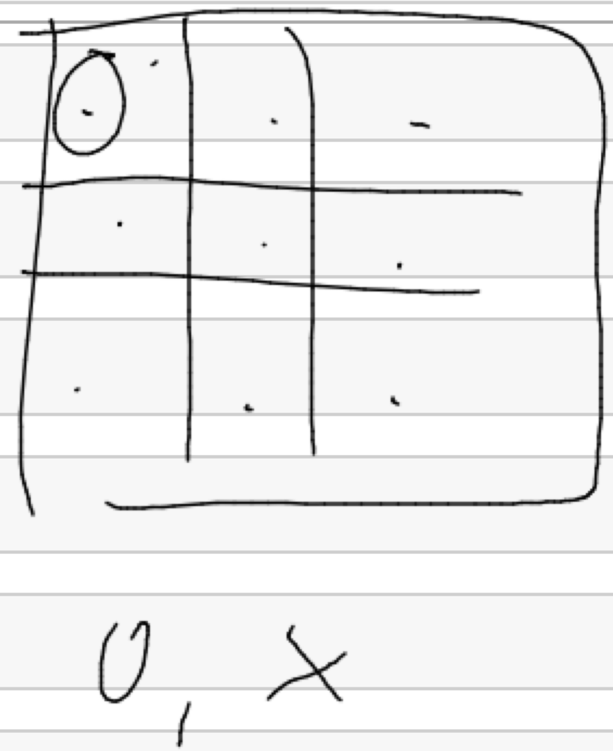
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Magic

In [120]:
```
board = [['']  * 3] * 3
```

In [121]:
```
print(board)
```

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

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Doubts

In [127]:
```
L = [10, 20, 30, 40, 50, 60]
print(L[2:4:15])
```

```
[30]
```

In [128]:
```
print(L[2:4:-1])
```

```
[]
```

In [129]:
```
print(L[-1:4:-1])
```

```
[60]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Doubts

In [127]:
```
L = [10, 20, 30, 40, 50, 60]
print(L[2:4:15])
```

```
[30]
```

In [128]:
```
print(L[2:4:-1])
```

```
[]
```

In [129]:
```
print(L[-1:4:-1])
```

```
[60]
```

In [133]:
```
print(L[-3:-1:-1])
```

```
[]
```

default end = -len(l)-1

In [134]:
```
print(L[-3:-1:1])
```

```
[40, 50]
```

In [135]:
```
L[0:-1]
```

Out[135]:
```
[10]
```

```
[60]
```

In [133]:
```
print(L[-3:-1:-1])
```

```
[]
```

In [134]:
```
print(L[-3:-1:1])
```

```
[40, 50]
```

In [136]:
```
L = [10, 20, 30, 40, 50, 60]

L[0::-1]
```

```
[10]
```

In [137]:
```
L[0:0:-1]
```

```
[]
```

In [138]:
```
L[0:(-len(L) - 1):-1]
```

```
[10]
```

In [139]:
```
L[0::1]
```

```
[10, 20, 30, 40, 50, 60]
```

In [133]: pr

```
[]
```

In [134]:
```
print(L[-3:-1:1])
```

```
[40, 50]
```

In [136]:
```
L = [10, 20, 30, 40, 50, 60]

L[0::-1]
```

Out[136]:
```
[10]
```

In [137]:
```
L[0:0:-1]
```

Out[137]:
```
[]
```

In [138]:
```
L[0:(-len(L) - 1):-1]
```

Out[138]:
```
[10]
```

In [139]:
```
L[0::1]
```

Out[139]:
```
[10, 20, 30, 40, 50, 60]
```

In [ ]:

```
b =
```



```
print(id(a) == id(b))
```

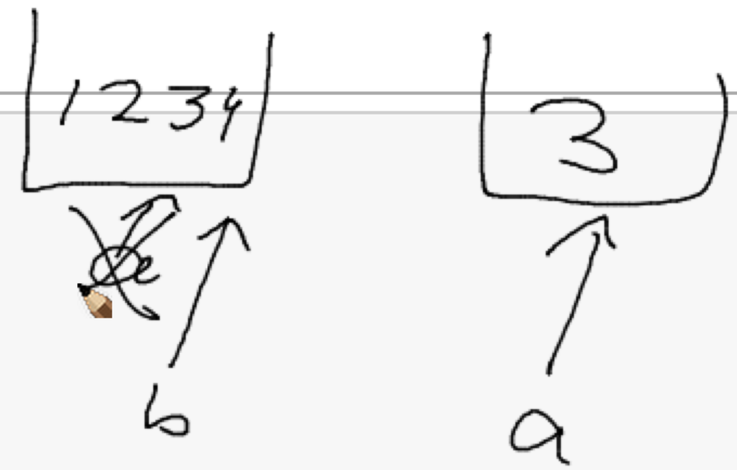
```
True
```

In [100]:
```
a = 1234
b = a
b = 3 # immutable, creating a new object
print(id(a) == id(b))

print(a)
print(b)
```

```
False
1234
3
```

In [144]:
```
a = 1234
b = a
a = 3
print(id(a) == id(b))
print(a)
print(b)
```




```
False
3
1234
```

In [160]:
```
print(a)
```

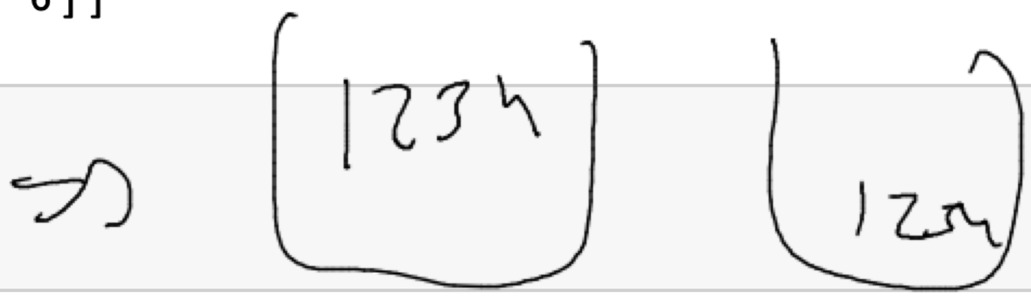
```
[1, 2, 3, 4, [4, 5, 6]]
```

In [161]:
```
print(b)
```

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

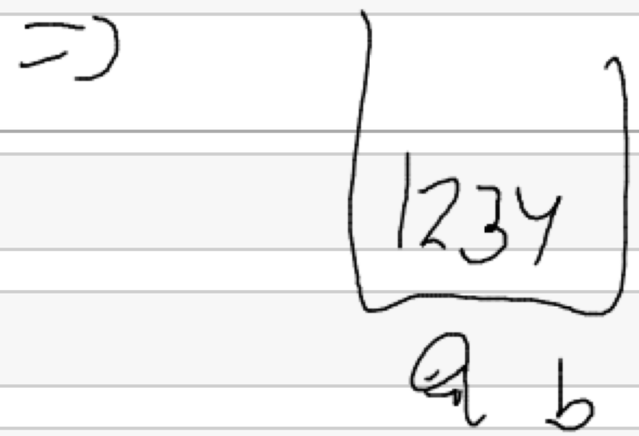
In [165]:
```
a=1234
b=1234
id(a)==id(b)
```

Out[165]: False



In [164]:
```
a = 1234
b = a
id(a) == id(b)
```

Out[164]: True



In [ ]:

In [ ]:

In [ ]:

```
b =
print(id(a) == id(b))
```

```
True
```

0 1 2 3 4 5

In [169]:
```
numbers = [5, 1, 2, 3, 7, 8]
numbers[6] = [100]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call las
t)
Input In [169], in <cell line: 2>()
      1 numbers = [5, 1, 2, 3, 7, 8]
----> 2 numbers[6] = [100]

IndexError: list assignment index out of range
```

In [ ]:

In [ ]:

DSML Intermediate : Python Refresher - 2

Recap

In [1]:
```
print('Hello world')
```

```
Hello world
```

In [2]:
```
print(8 // 3)
```

```
2
```

In [3]:
```
print(8 / 3)
```

```
2.666666666666665
```

In [5]:
```
print(-8//3)
```

```
-3
```

In [6]:
```
print(-8/3)
```

```
-2.666666666666665
```