



**00:15 Array
introduction**

03:26 Big O Analysis

**08:03 Static vs
Dynamic Array**

13:02 Exercise

Store apple's stock price for 5 days and answer,

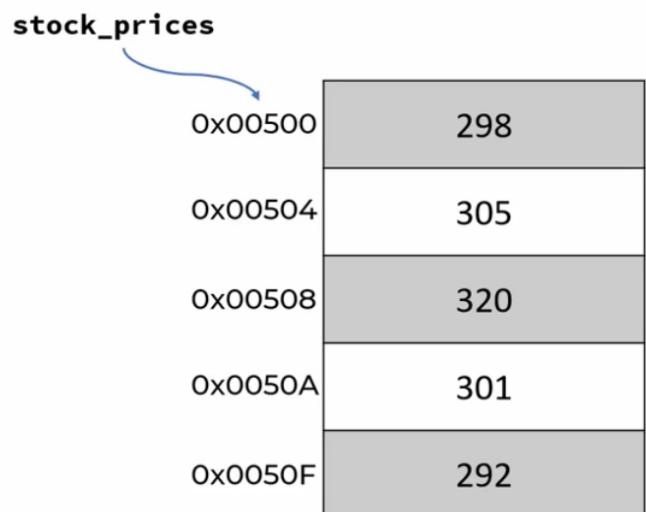
1. What was the price on day 1?
2. What was the price on day 3?

```
stock_prices = [298,305,320,301,292]  
stock_prices[0] ← 298 ← price on day 1  
stock_prices[2] ← 320 ← price on day 3
```

Store apple's stock price for 5 days and answer,

1. What was the price on day 1?
2. What was the price on day 3?

```
stock_prices = [298,305,320,301,292]  
stock_prices[0] ← 298 ← price on day 1  
stock_prices[2] ← 320 ← price on day 3
```





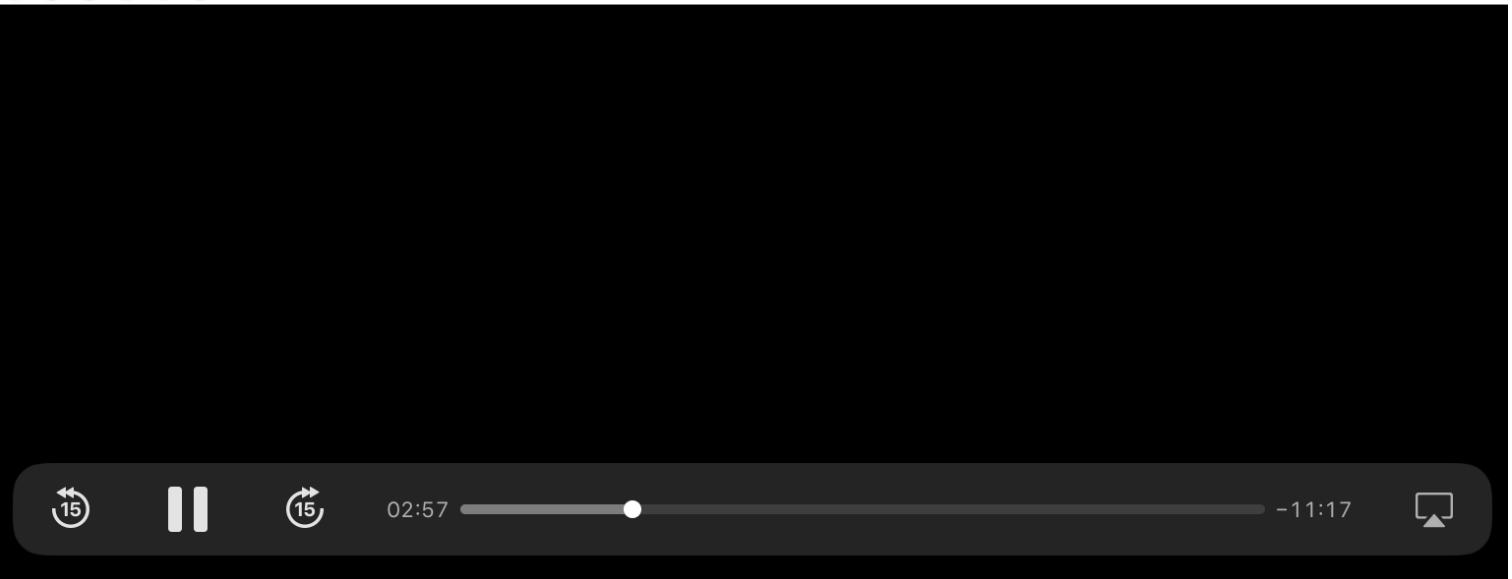
$$298 = 100101010$$

4 bytes

$$298 = 00000000\ 00000000\ 00000001\ 00101010$$

0x00500	00000000
0x00501	00000000
0x00502	00000001
0x00503	00101010

A blue bracket on the right side of the table groups the last two rows (0x00502 and 0x00503) and is labeled "298". A blue arrow points from the binary representation above to the value in the 0x00503 cell.



stock_prices

0x00500	298
0x00504	305
0x00508	320
0x0050A	301
0x0050F	292

0x00500	00000000	
0x00501	00000000	
0x00502	00000001	
0x00503	00101010	
0x00504	00000000	
0x00505	00000000	
0x00506	00000001	
0x00507	00110001	
	...	





Scenario 1: What was the price on day 3 ?

`stock_prices`

0x00500	298
0x00504	305
0x00508	320
0x0050A	301
0x0050F	292

`stock_prices[2] → 320 (price on day 3)`

`stock_prices[0] → 0x00500`

`stock_prices[2] → 0x00500 + 2 * sizeof(integer)`

`stock_prices[2] → 0x00500 + 2 * 4`

`stock_prices[2] → 0x00508`

Lookup by index = O(1)



05:23



-08:51



Scenario 2: On what day price was 301?

stock_prices

0x00500	298
0x00504	305
0x00508	320
0x0050A	301
0x0050F	292

```
for i in range(len(stock_prices)):  
    if stock_prices[i]==301:  
        return i
```

Lookup by value = O(n)



Scenario 3: Print all prices

stock_prices

0x00500	298
0x00504	305
0x00508	320
0x0050A	301
0x0050F	292

```
for price in stock_prices:  
    print(price)
```

Array traversal = O(n)



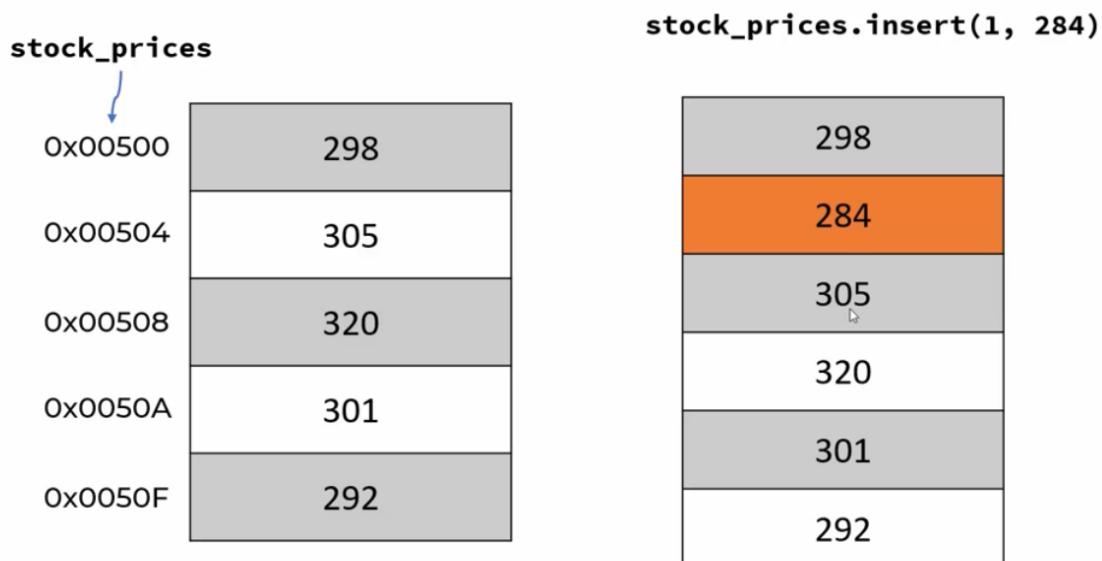
06:42



-07:32

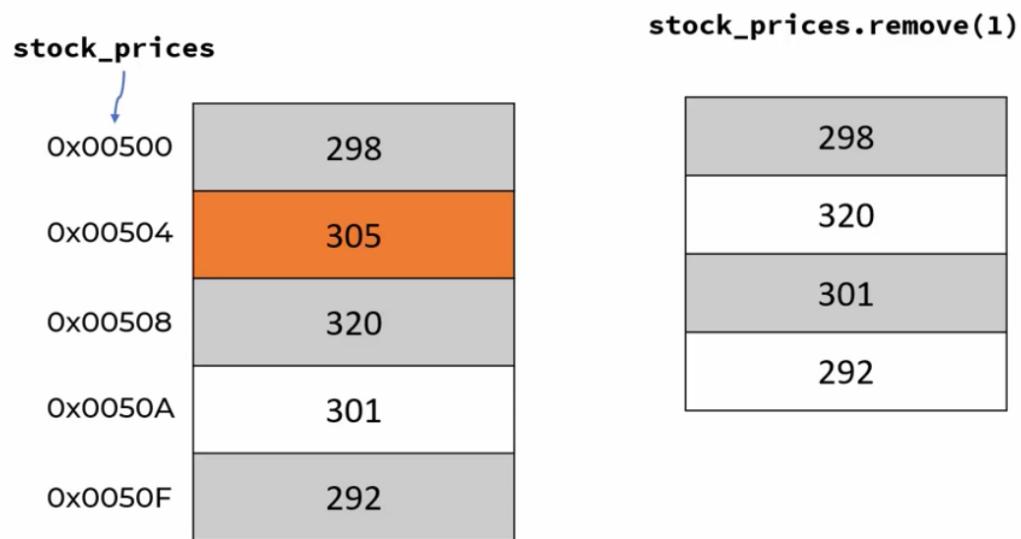


Scenario 4: Insert new price 284 at index 1



Array insertion = O(n)

Scenario 5: Delete element at index 1



Array deletion = O(n)



In python, **list** is implemented as **dynamic array**

In other languages like JAVA, C++ we have **static** and **dynamic arrays** both



08:12



-06:02



```
int[] stockPrices = new int[5]; Static Array
```

```
stockPrices[0] = 298;  
stockPrices[1] = 305;  
stockPrices[2] = 320;  
stockPrices[3] = 301;  
stockPrices[4] = 292;
```

```
stockPrices[6] = 400; ←———— Not allowed. Throws ArrayIndexOutOfBoundsException
```

JAVA

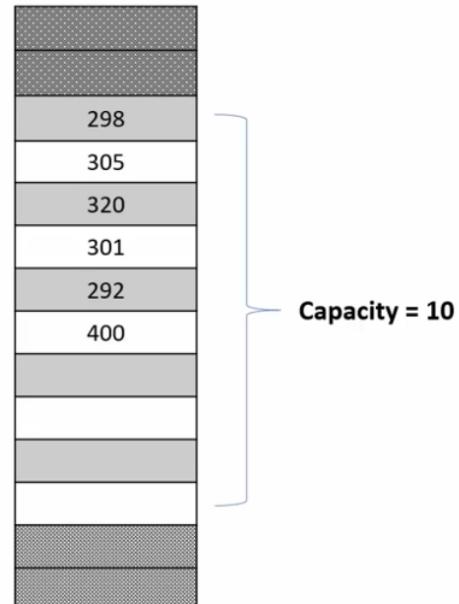
```
ArrayList<Integer> stockPrices = new ArrayList<Integer>(); Dynamic Array
```

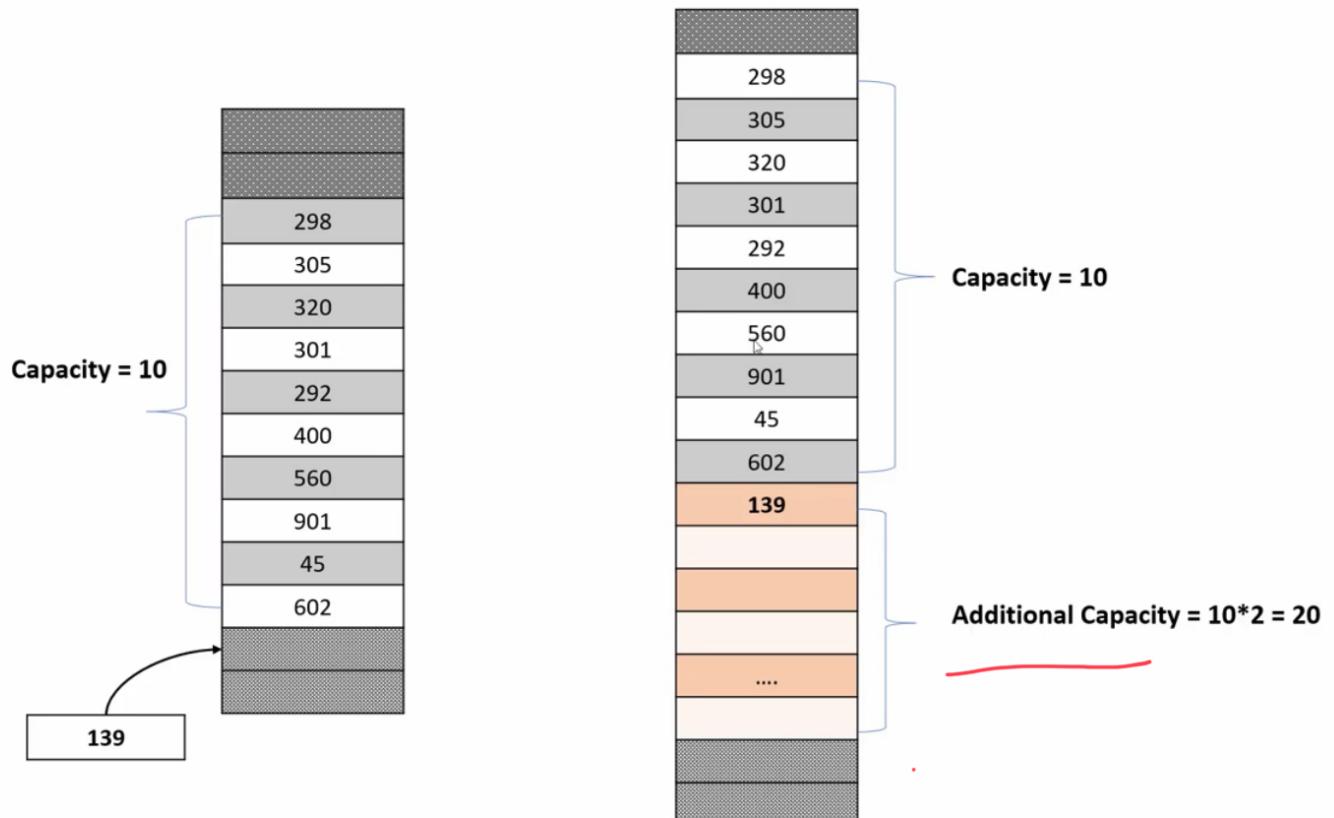
```
stockPrices.add(298);  
stockPrices.add(305);  
stockPrices.add(320);  
stockPrices.add(301);  
stockPrices.add(292);  
  
stockPrices.add(400);
```





```
ArrayList<Integer> stockPrices = new ArrayList<Integer>();  
  
stockPrices.add(298);  
stockPrices.add(305);  
stockPrices.add(320);  
stockPrices.add(301);  
stockPrices.add(292);  
↓  
stockPrices.add(400);
```





新增 capacity
= original capacity $\times 2$

$$\therefore \text{the third} = 30 \times 2 + 30 = 90$$

$$(10 + 20) \rightarrow 30 = 90$$



```
stock_prices = [2,3,5,6]  
stock_names = ["AAPL", "IBM", "TATA"]  
  
stock_data = [  
    {"ticker": "AAPL", "price": 302},  
    {"ticker": "TSLA", "price": 902},  
    {"ticker": "TATA", "price": 278},  
]
```

Arrays can store numbers, text or complex objects





```
stock_prices = [  
    [2,3,5,6],  
    [40,42,38,44],  
    [78,89,71,66]  
]
```

2 Dimensional Array



12:38



-01:36



	Python	Java	C++
Static array		Native array	Native array
Dynamic array	list	ArrayList	std::vector ↳

py/2_arrays_exercise.md at master · codebasics/py · GitHub

1. Let us say your expense for every month are listed below,
 - i. January - 2200
 - ii. February - 2350
 - iii. March - 2600
 - iv. April - 2130
 - v. May - 2190

Create a list to store these monthly expenses and using that find out,

1. In Feb, how many dollars you spent extra compare to January?
2. Find out your total expense in first quarter (first three months) of the year.
3. Find out if you spent exactly 2000 dollars in any month
4. June month just finished and your expense is 1980 dollar. Add this item to our monthly expense list
5. You returned an item that you bought in a month of April and got a refund of 200\$. Make a correction to your monthly expense list based on this

Solution

2. You have a list of your favourite marvel super heros.

```
heros=['spider man','thor','hulk','iron man','captain america']
```

Using this find out,



14:00

-00:14

