

The ArrayList

"Better" arrays

Arrays

```
String[] values = new String[5];
```

- Great for
 - holding multiple items
- Not so great for
 - growing
 - inserting/removing
 - shifting

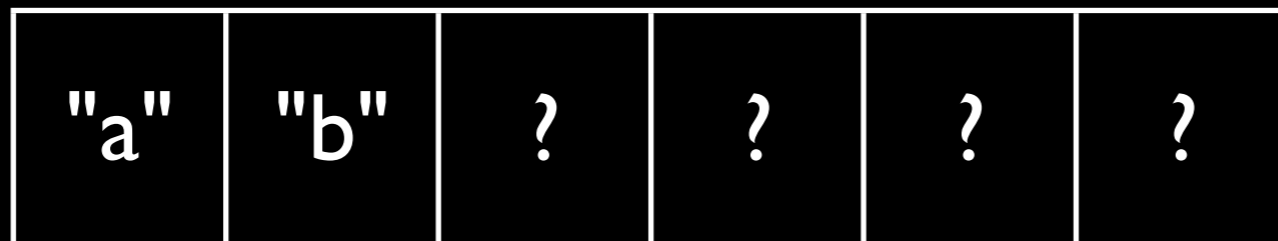
Solution: ArrayList

- An ArrayList is a class
 - uses an array to hold items
 - works like a grocery list
 - internal methods take care of nasty array operations behind the scenes



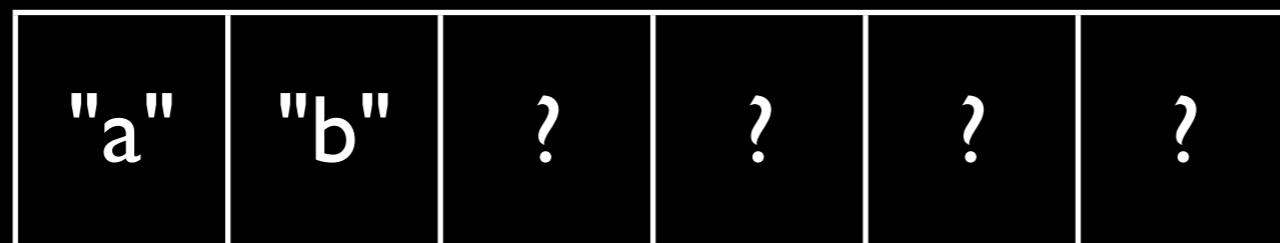
Capacity vs Size

- Use an array that is partially full
- Capacity is how big the array
- Size is how many usable items are in the list



Capacity vs Size

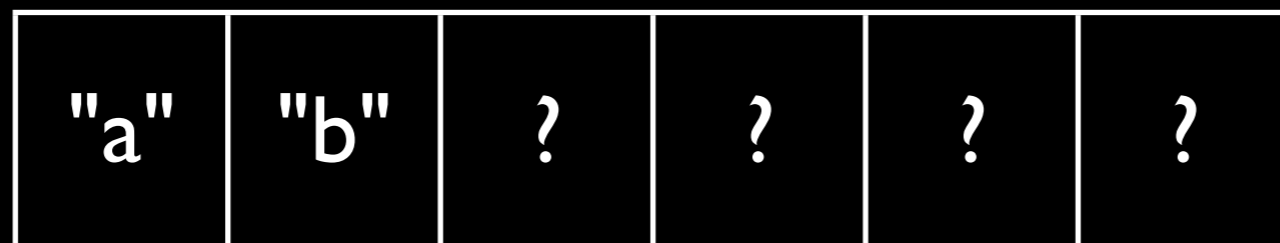
- Use an array that is partially full
- Capacity is how big the array
- Size is how many usable items are in the list



capacity: 6

Capacity vs Size

- Use an array that is partially full
- Capacity is how big the array
- Size is how many usable items are in the list



capacity: 6

size: 2

Getting Started

```
public class ArrayList {  
    private String[] array;  
    private int size;
```

```
    public ArrayList() {  
        array = new String[10];  
        size = 0;
```

```
    }
```

```
}
```

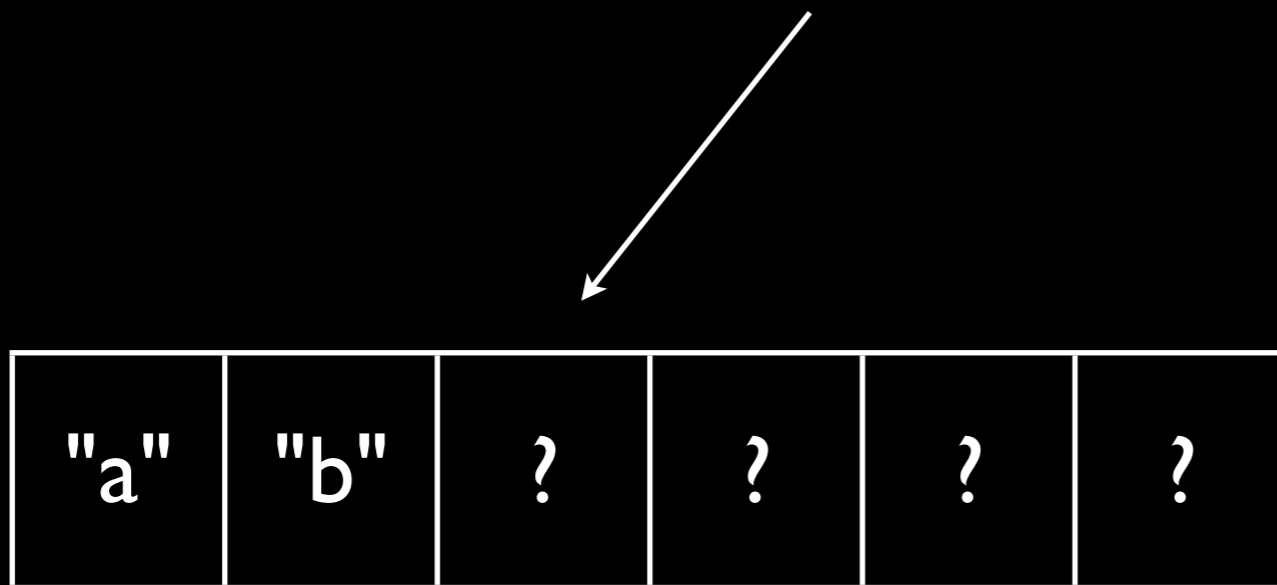
default capacity
of 10

Getting the Size

```
public int size() {  
    return size;  
}
```

How to add

- Place at the end (if there's room!)
- Where's the end?

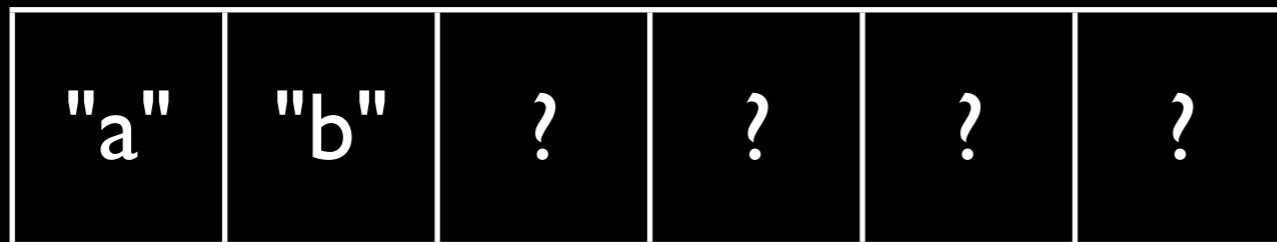


capacity: 6
size: 2

How to add

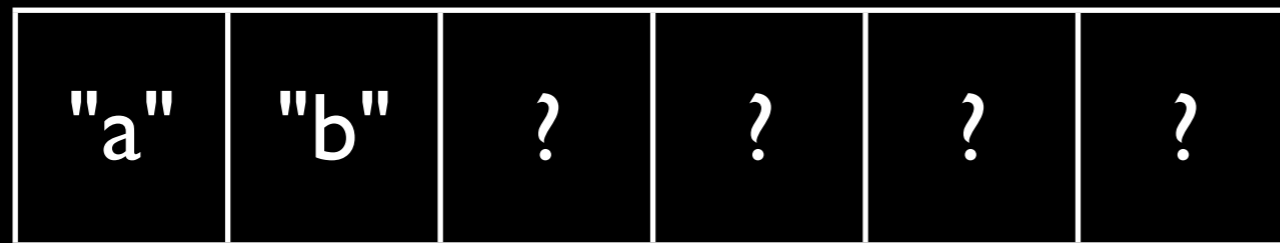
- Place at the end (if there's room!)
- Where's the end?

index is same as size!



capacity: 6
size: 2

How to add

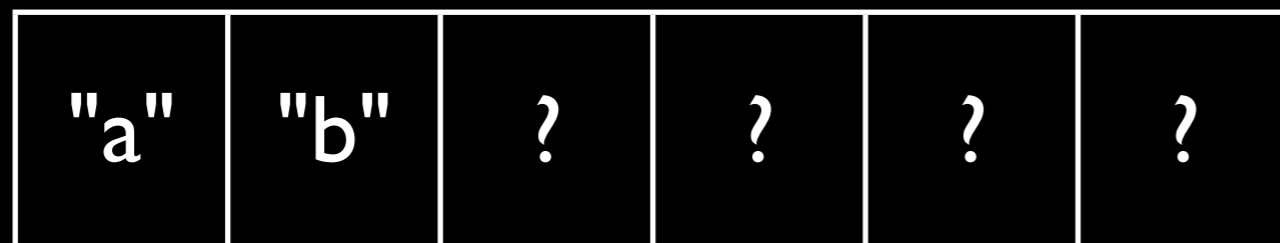


capacity: 6

size: 2

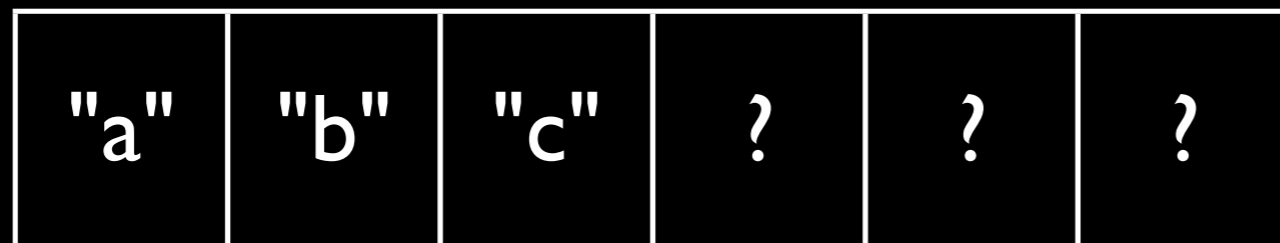


How to add

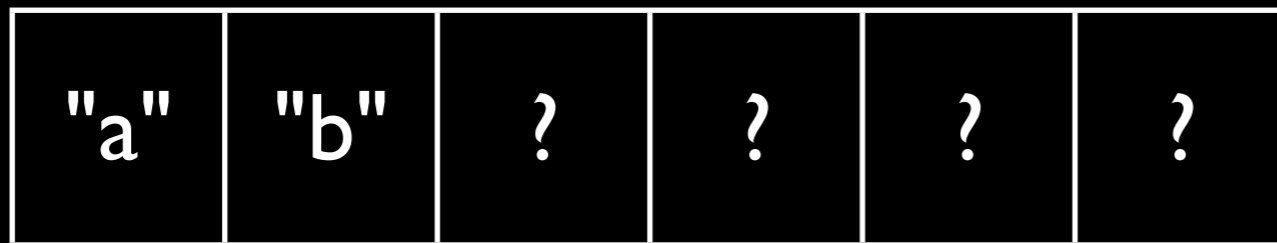


capacity: 6

size: 2

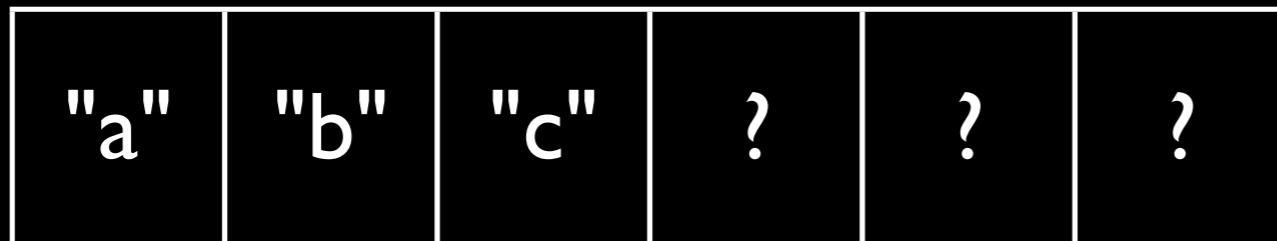


How to add



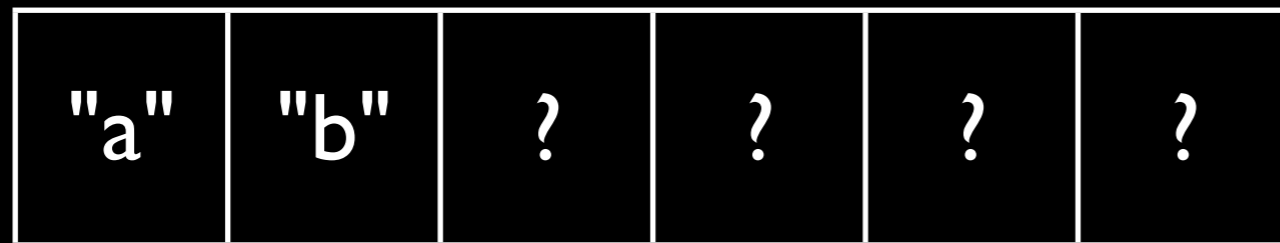
capacity: 6

size: 2



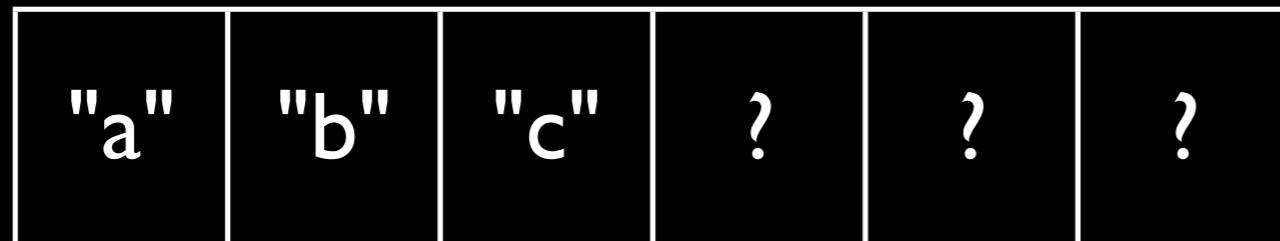
capacity: 6

How to add



capacity: 6

size: 2



capacity: 6

size: 3

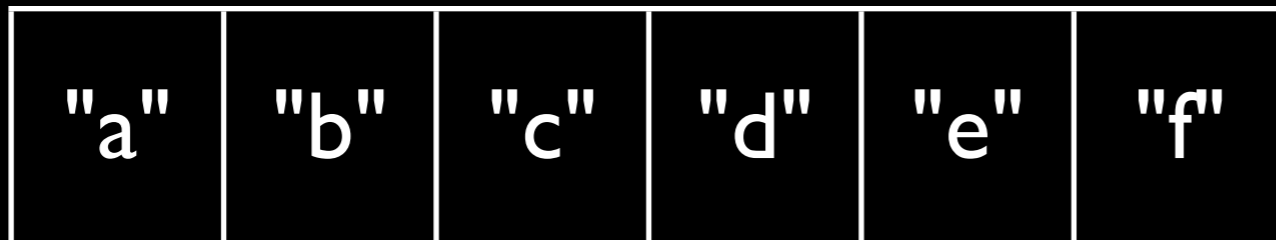
How to add

- When is the list full?

"a"	"b"	"c"	"d"	"e"	"f"
-----	-----	-----	-----	-----	-----

How to add

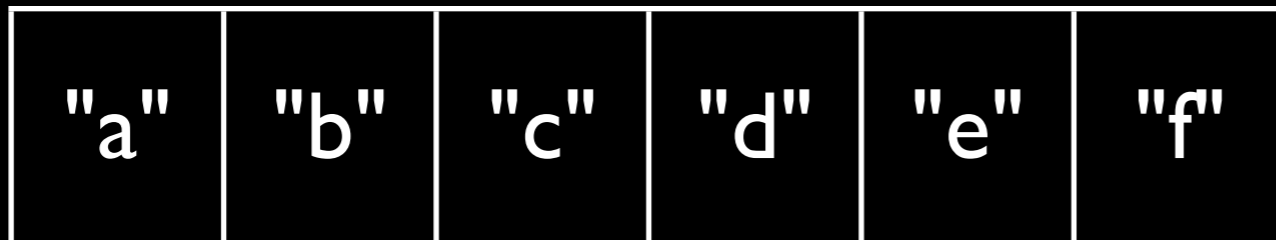
- When is the list full?



capacity: 6

How to add

- When is the list full?

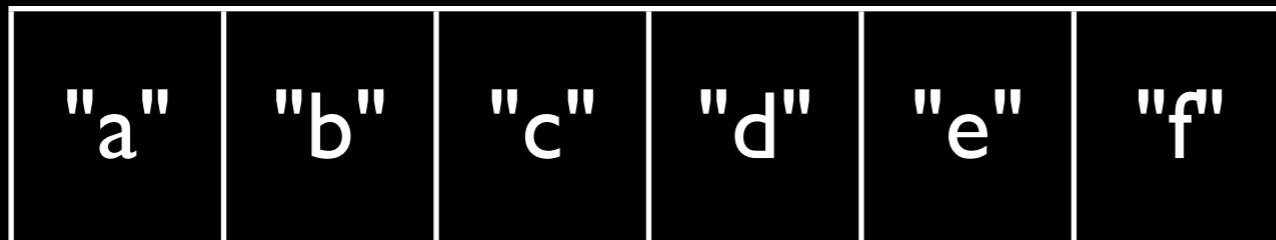


capacity: 6

size: 6

How to add

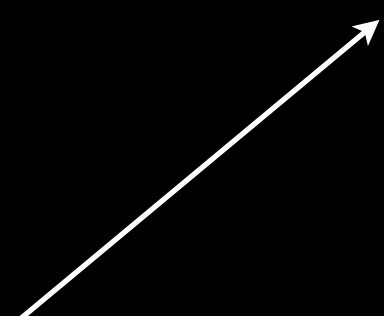
- When is the list full?



capacity: 6

size: 6

size == capacity



Add v. I

```
public void add(String s) {  
    if(size < array.length) {  
        array[size] = s;  
        size++;  
    }  
}
```

But what to do when full?

Grow

```
private void grow() {  
    String[] temp = new String[array.length*2];  
    for(int i = 0; i < array.length; i++)  
        temp[i] = array[i];  
    array = temp;  
}
```

I believe java actually multiplies by 3/2's

Add v.2

```
public void add(String s) {  
    if(size == array.length)  
        grow();  
    array[size] = s;  
    size++;  
}
```

Getting

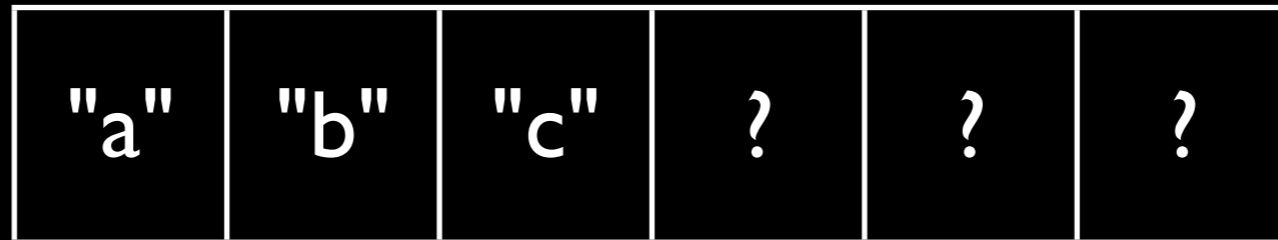
"a"	"b"	"c"	?	?	?
-----	-----	-----	---	---	---

get(0)

get(2)

get(3)

Getting



get(0)

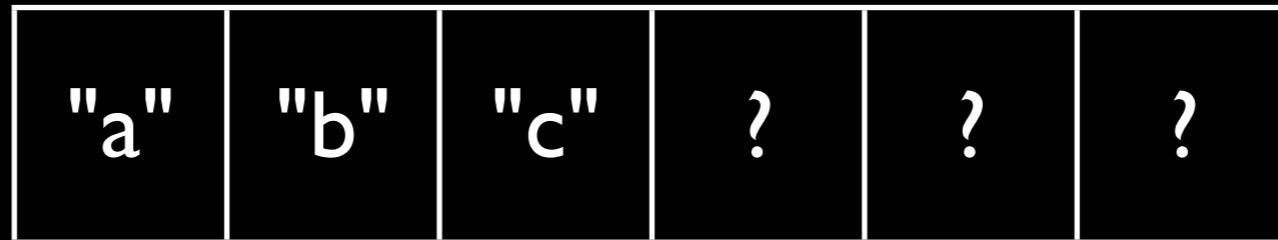
get(2)

get(3)



"a"

Getting



get(0)



"a"

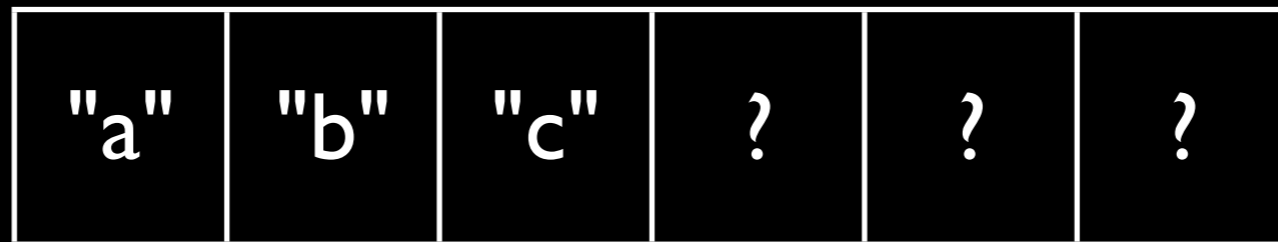
get(2)



"c"

get(3)

Getting



get(0)



"a"

get(2)



"c"

get(3)



CRASH

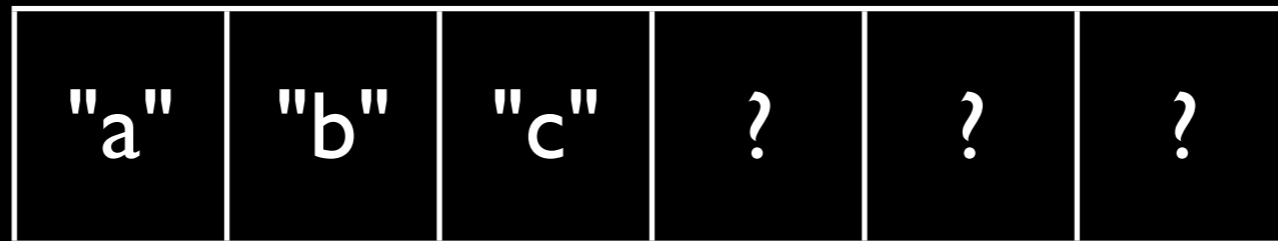
Getting

```
public String get(int i) {  
    if(i < 0 || i >= size)  
        throw new IllegalArgumentException("Index " + i + " is illegal!");  
  
    return array[i];  
}
```

Your program will CRASH if you try
to get outside of the valid range

Setting

- Replace function



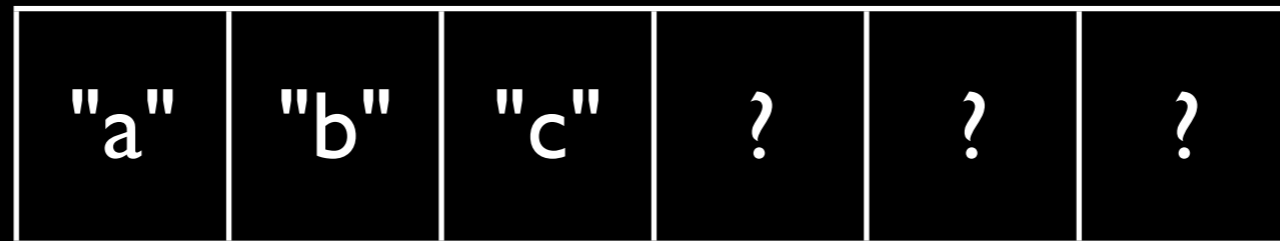
capacity: 6

size: 3

`set(0, "b")`

Setting

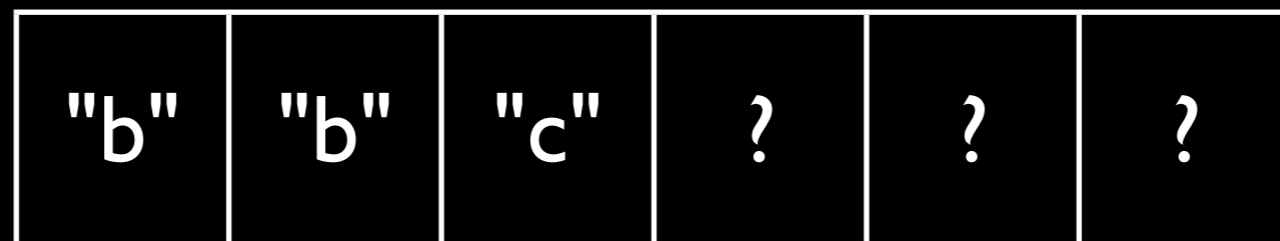
- Replace function



capacity: 6

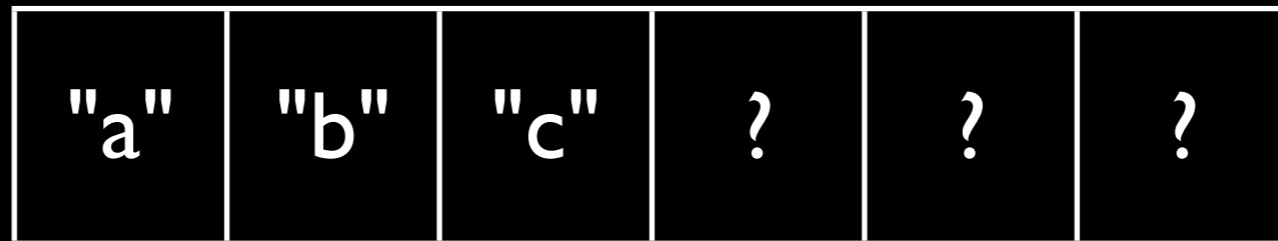
size: 3

set(0, "b")



Setting

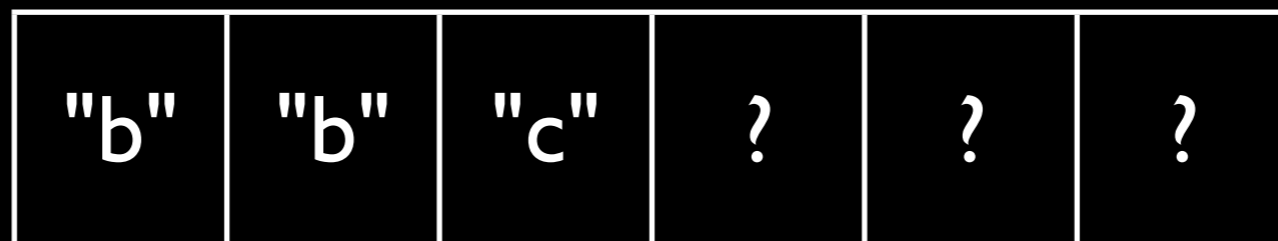
- Replace function



capacity: 6

size: 3

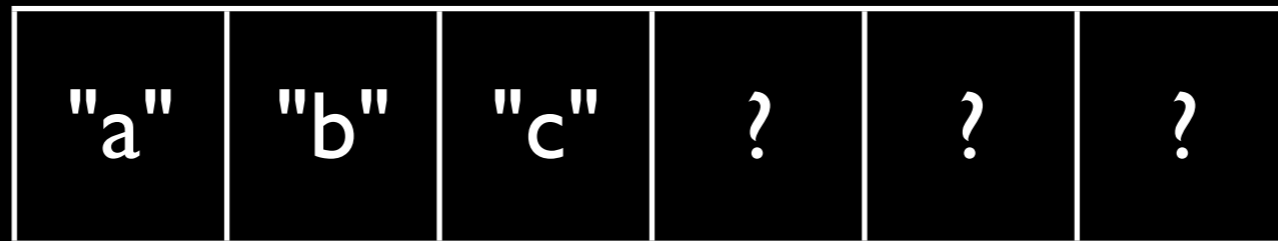
set(0, "b")



capacity: 6

Setting

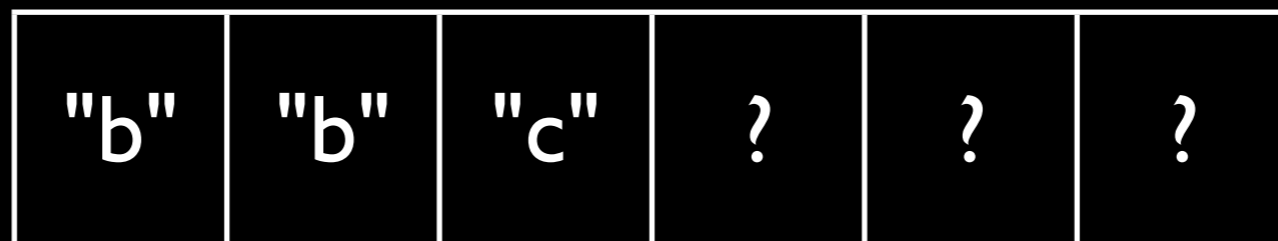
- Replace function



capacity: 6

size: 3

set(0, "b")



capacity: 6

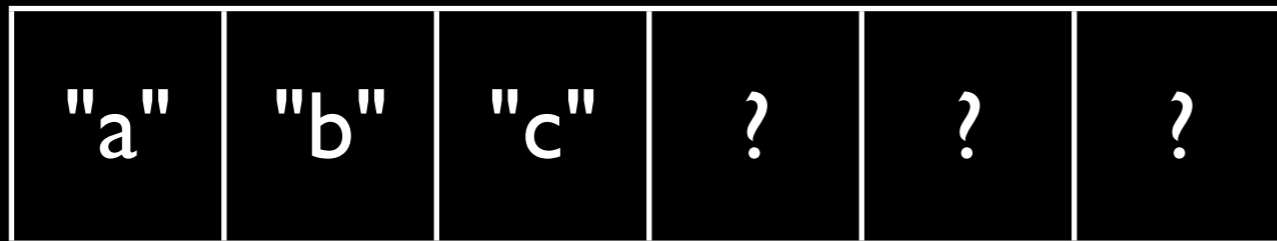
size: 3

Setting

```
public String set(int i, String s) {  
    if(i < 0 || i >= size)  
        throw new IllegalArgumentException("Index " + i + " is illegal!");  
  
    String old = array[i];  
    array[i] = s;  
    return old;  
}
```

Returns the value that used to be there!

Inserting

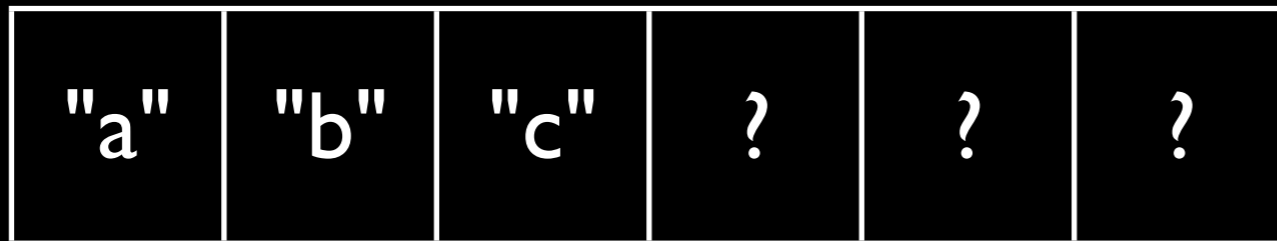


capacity: 6

size: 3

`add(0, "d")`

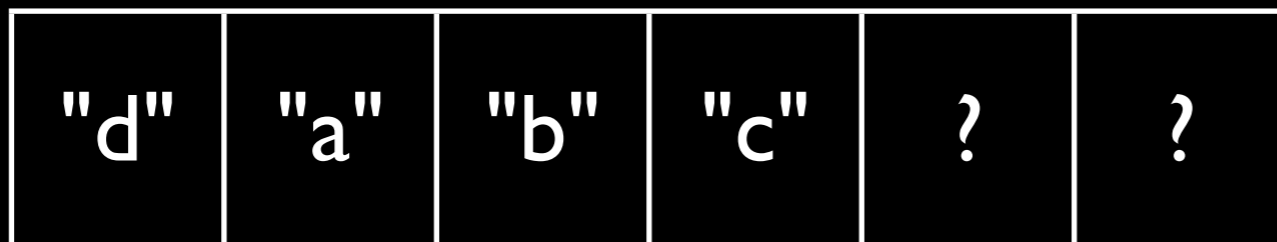
Inserting



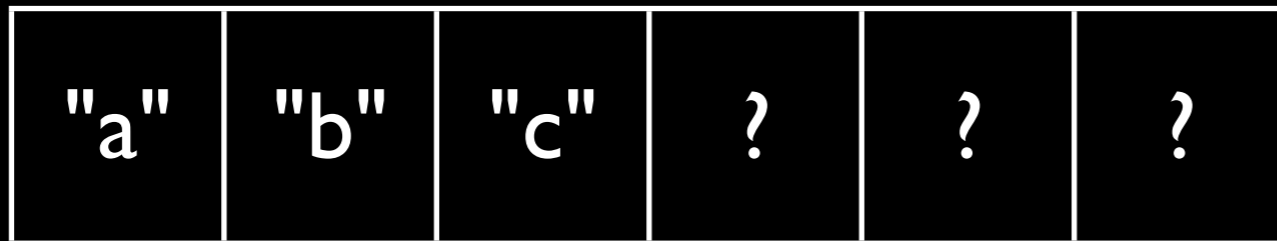
capacity: 6

size: 3

add(0, "d")



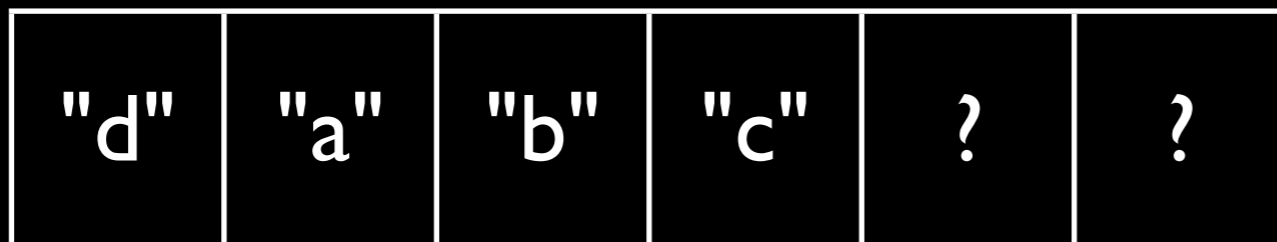
Inserting



capacity: 6

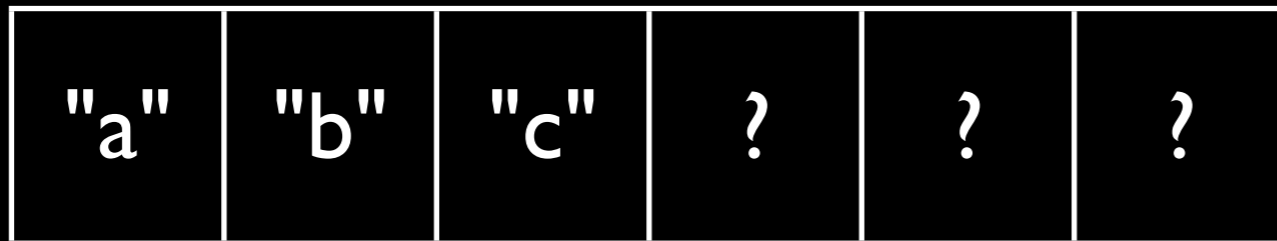
size: 3

add(0, "d")



capacity: 6

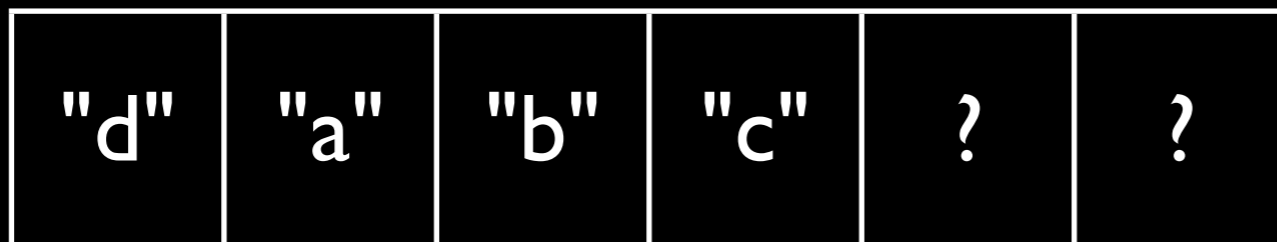
Inserting



capacity: 6

size: 3

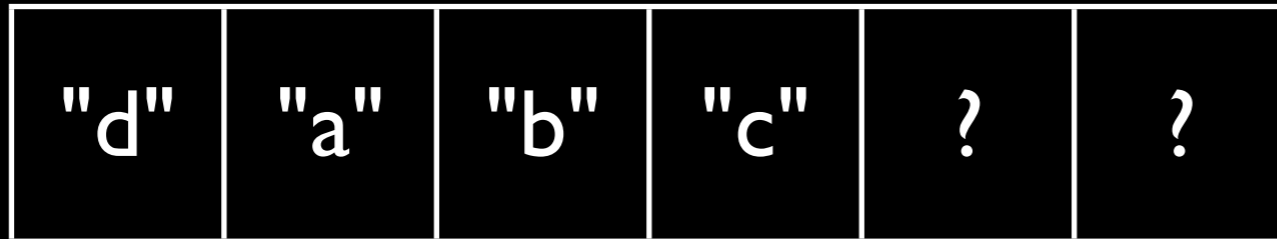
add(0, "d")



capacity: 6

size: 4

Inserting



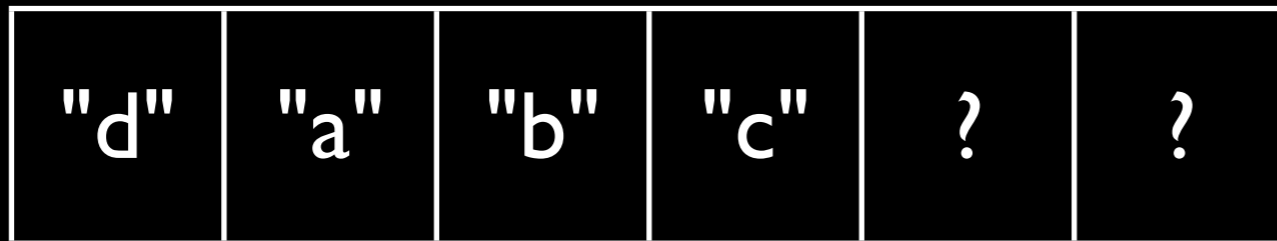
capacity: 6

size: 4

add(3, "e")



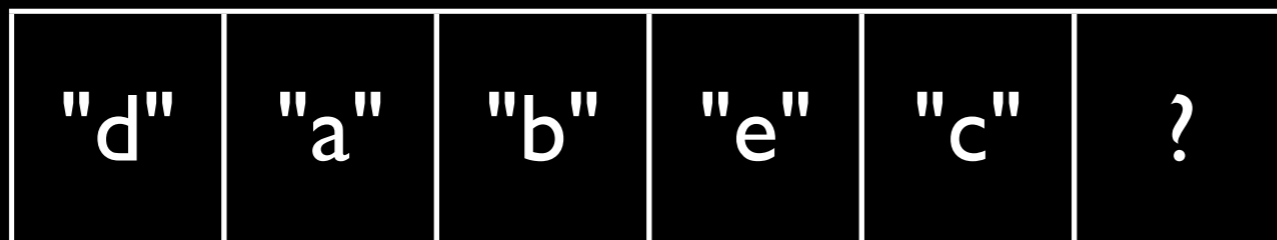
Inserting



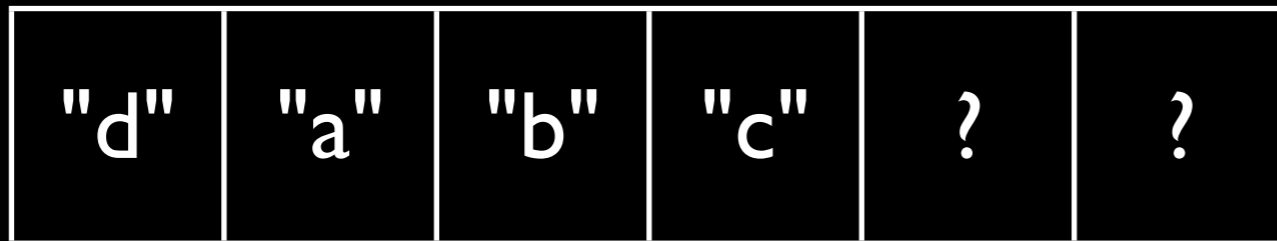
capacity: 6

size: 4

add(3, "e")



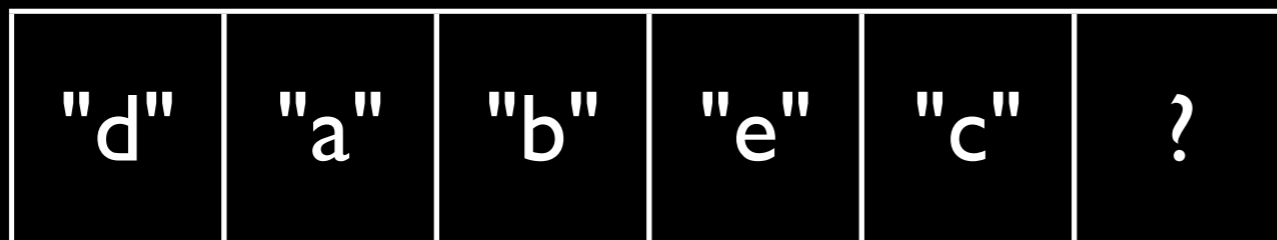
Inserting



capacity: 6

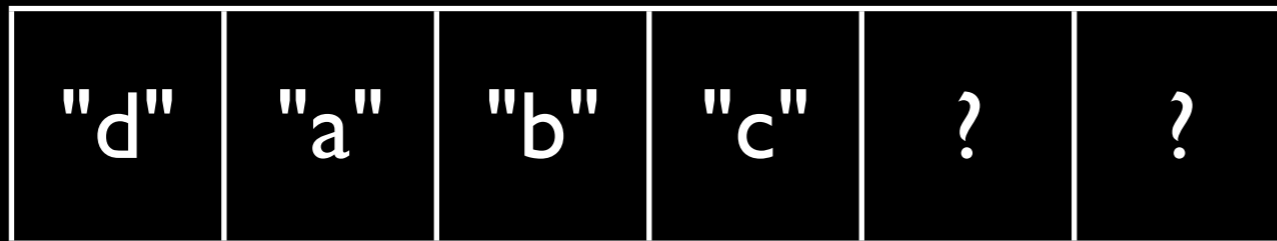
size: 4

add(3, "e")



capacity: 6

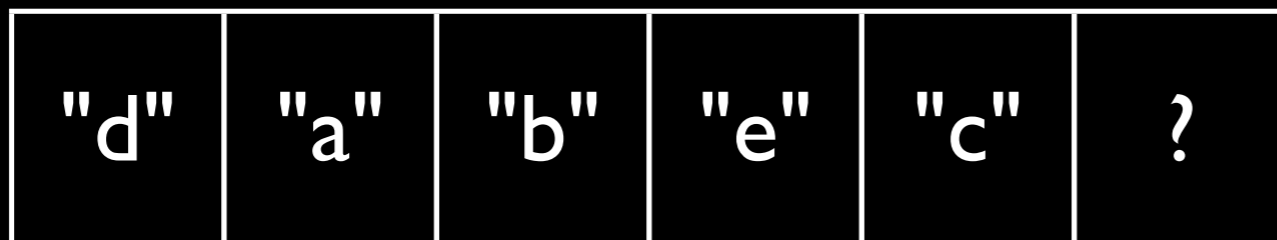
Inserting



capacity: 6

size: 4

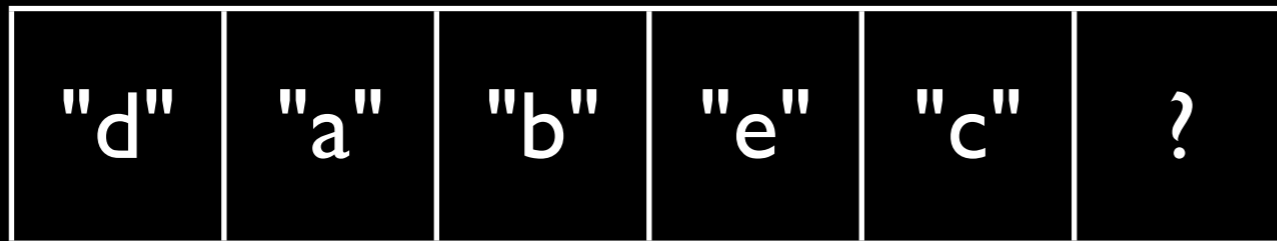
add(3, "e")



capacity: 6

size: 5

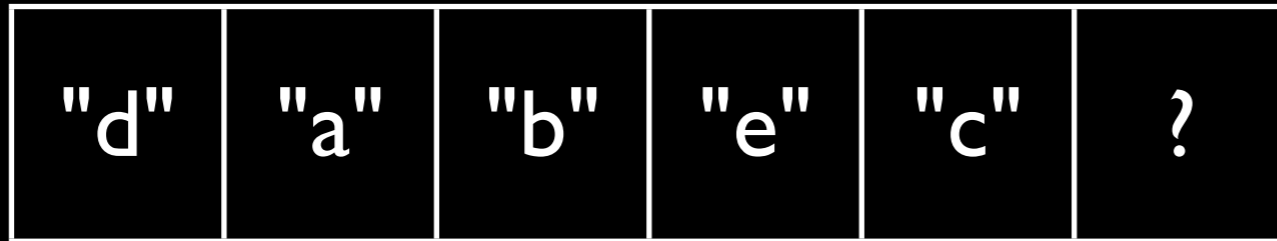
Inserting



capacity: 6

size: 5

Inserting



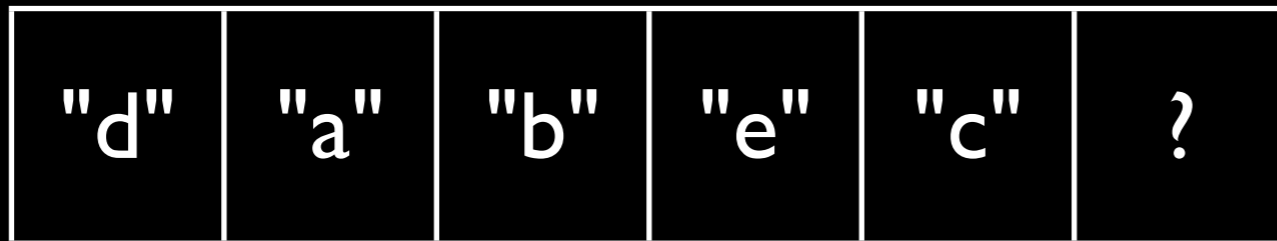
capacity: 6

size: 5

add(5, "f")



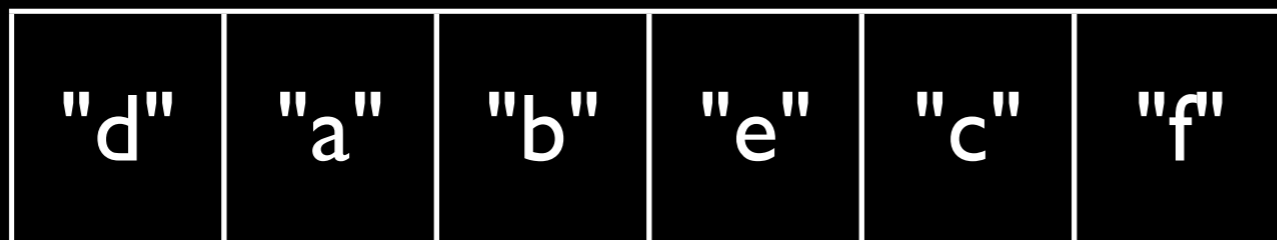
Inserting



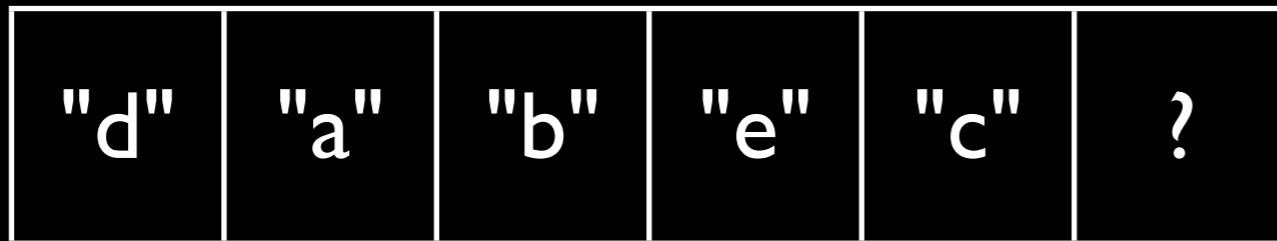
capacity: 6

size: 5

add(5, "f")



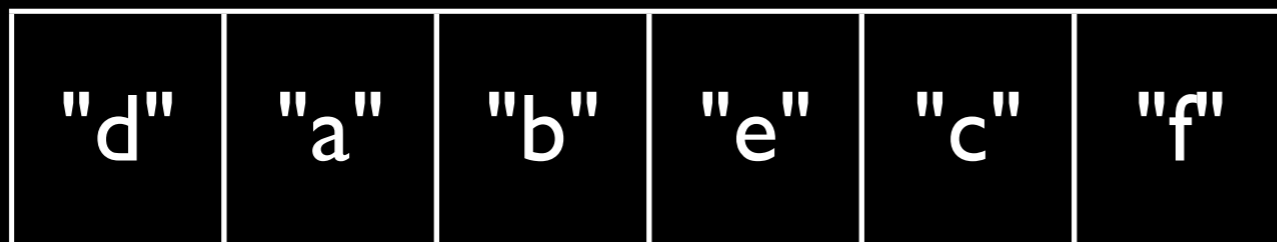
Inserting



capacity: 6

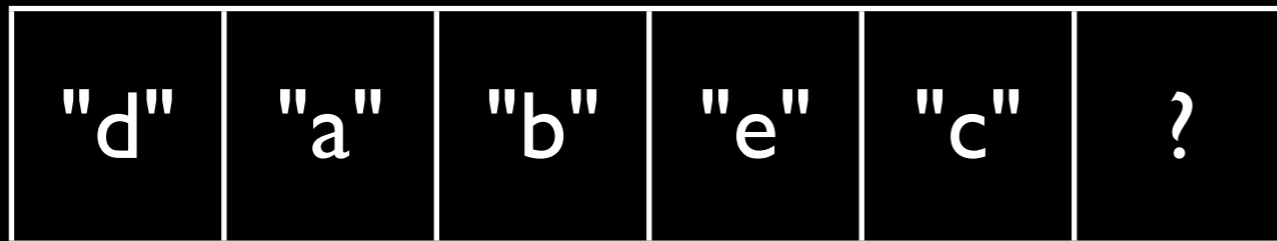
size: 5

add(5, "f")



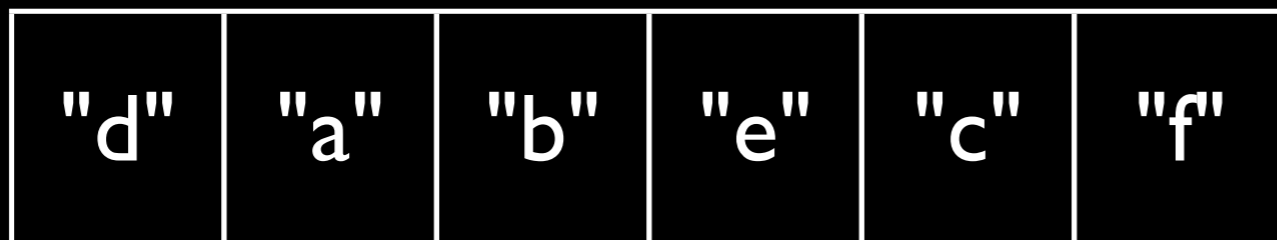
capacity: 6

Inserting



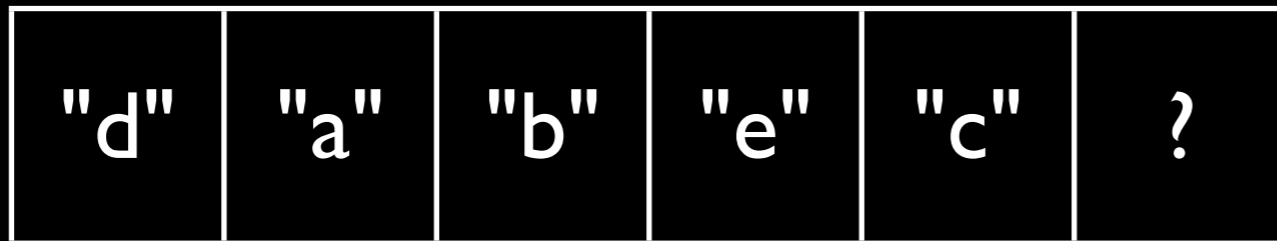
capacity: 6
size: 5

add(5, "f")



capacity: 6
size: 6

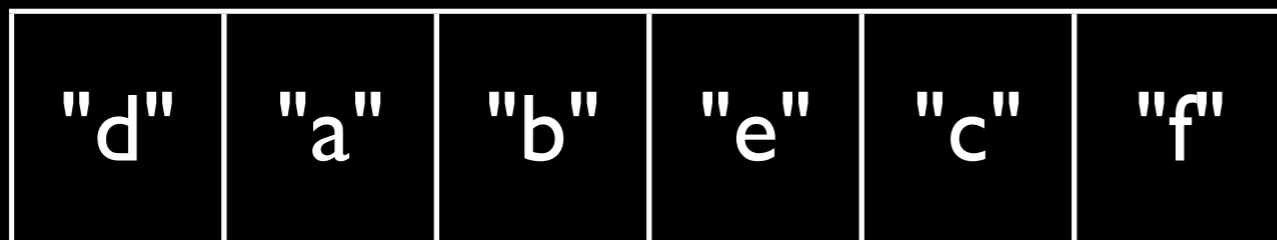
Inserting



capacity: 6

size: 5

add(5, "f")



capacity: 6

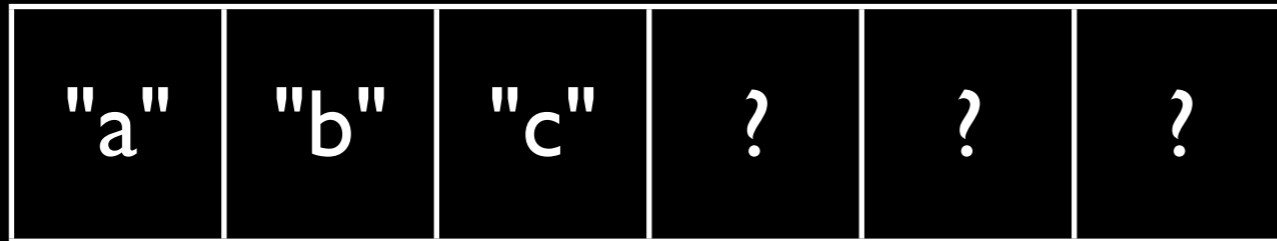
size: 6

You can insert at
the size!

Second Add

```
public void add(int i, String s) {  
    if(i < 0 || i > size)  
        throw new IllegalArgumentException("Index " + i + " is illegal!");  
  
    if(size == array.length)  
        grow();  
  
    for(int index = size; index > i; index--)  
        array[index] = array[index-1];  
  
    array[i] = s;  
    size++;  
}
```

Remove

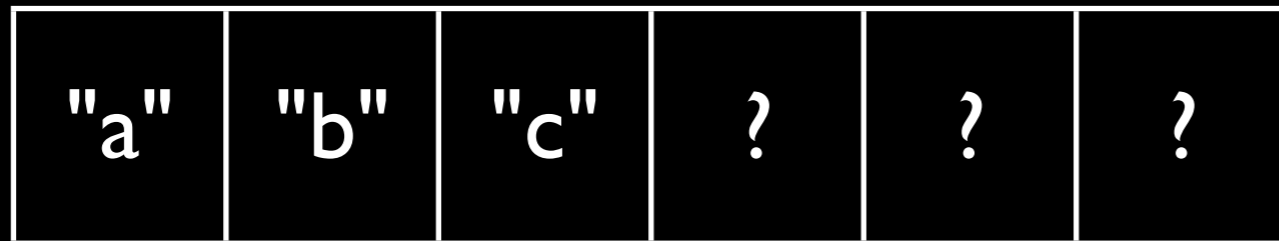


capacity: 6

size: 3

`remove(l)`

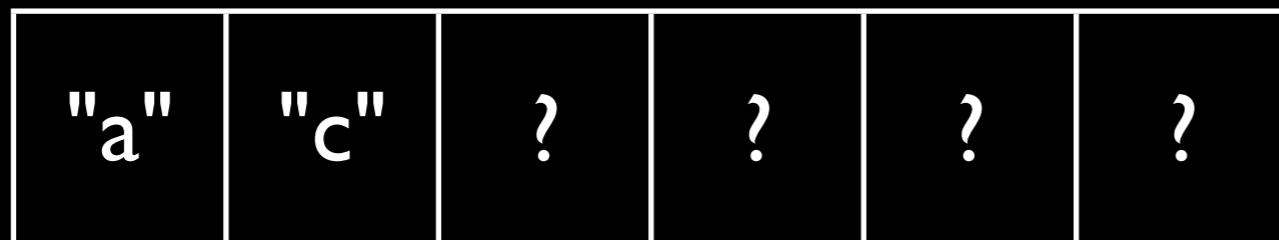
Remove



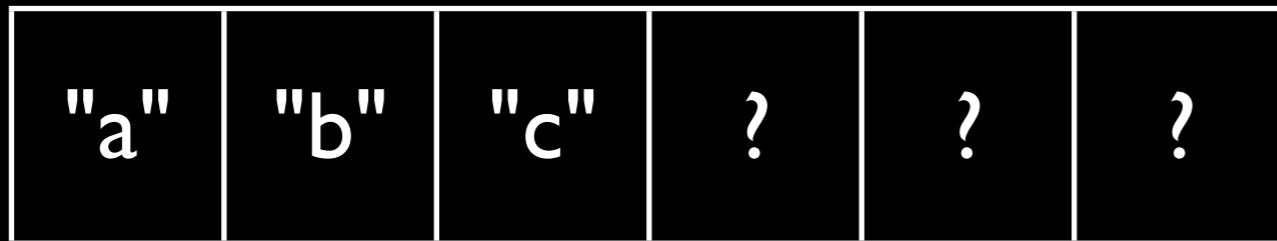
capacity: 6

size: 3

remove(1)



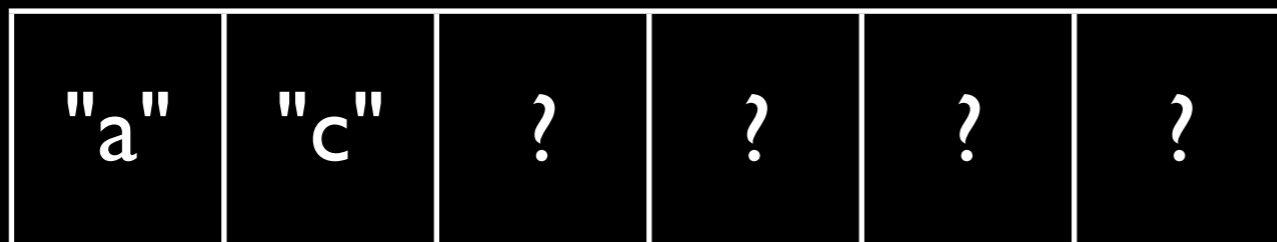
Remove



capacity: 6

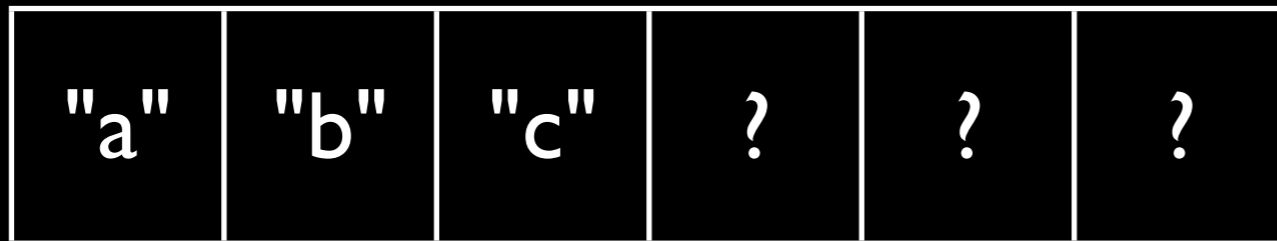
size: 3

remove(1)



capacity: 6

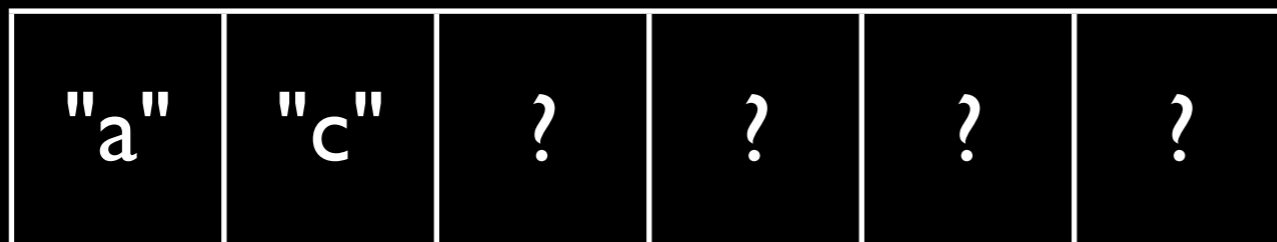
Remove



capacity: 6

size: 3

remove(1)



capacity: 6

size: 2

Remove

```
public String remove(int i) {  
    if(i < 0 || i >= size)  
        throw new IllegalArgumentException("Index " + i + " is illegal!");  
  
    String old = array[i];  
    for(int index = i; index < size-1; index++)  
        array[index] = array[index+1];  
  
    size--;  
    return old;  
}
```

Returns the removed value

ArrayList<E>

- Java has a built in ArrayList<E> class that can hold ANYTHING, not just String's
- The <E> indicates what you want to put in the list
- You need this declaration above your class

```
import java.util.ArrayList;
```

ArrayList Practically

```
public static void main(String[] args)
{
    ArrayList<String> list = new ArrayList<String>();
    list.add("a");
    list.add("b");
    list.add("c");
    list.add(0, "d");
    System.out.println(list.get(1));
    System.out.println(list.remove(2));
    System.out.println(list.get(2));
    list.set(0, "e");
    System.out.println(list);
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```

```
    list.add("b");
```

```
    list.add("c");
```

```
    list.add(0, "d");
```

```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```

```
    list.add("c");
```

```
    list.add(0, "d");
```

```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```

```
    list.add(0, "d");
```

```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```



```
    list.add(0, "d");
```

```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```



```
    list.add(0, "d");
```



```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```



```
    list.add(0, "d");
```



```
    System.out.println(list.get(1));
```

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

output

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```



```
    list.add(0, "d");
```



```
    System.out.println(list.get(1));
```



```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

output

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
    ArrayList<String> list = new ArrayList<String>();
```



```
    list.add("a");
```



```
    list.add("b");
```



```
    list.add("c");
```



```
    list.add(0, "d");
```



```
    System.out.println(list.get(1));
```



output

a

```
    System.out.println(list.remove(2));
```

```
    System.out.println(list.get(2));
```

```
    list.set(0, "e");
```

```
    System.out.println(list);
```

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
System.out.println(list.get(2));
```

```
list.set(0, "e");
```

```
System.out.println(list);
```

```
}
```

output

a

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

a

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

b

```
System.out.println(list.get(2));
```

```
list.set(0, "e");
```

```
System.out.println(list);
```

```
}
```

output

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

a

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

b

```
System.out.println(list.get(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.set(0, "e");
```

```
System.out.println(list);
```

```
}
```

output

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
System.out.println(list.get(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.set(0, "e");
```

```
System.out.println(list);
```

output

a

b

c

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

a

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

b

```
System.out.println(list.get(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

c

```
list.set(0, "e");
```

"e"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
System.out.println(list);
```

output

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```

?	?	?	?	?	...
---	---	---	---	---	-----

```
list.add("a");
```

"a"	?	?	?	?	...
-----	---	---	---	---	-----

```
list.add("b");
```

"a"	"b"	?	?	?	...
-----	-----	---	---	---	-----

```
list.add("c");
```

"a"	"b"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
list.add(0, "d");
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

```
System.out.println(list.get(1));
```

"d"	"a"	"b"	"c"	?	...
-----	-----	-----	-----	---	-----

a

```
System.out.println(list.remove(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

b

```
System.out.println(list.get(2));
```

"d"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

c

```
list.set(0, "e");
```

"e"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

```
System.out.println(list);
```

"e"	"a"	"c"	?	?	...
-----	-----	-----	---	---	-----

output

```
}
```

ArrayList Practically

```
public static void main(String[] args)
```

```
{
```

```
ArrayList<String> list = new ArrayList<String>();
```



```
list.add("a");
```



```
list.add("b");
```



```
list.add("c");
```



```
list.add(0, "d");
```



```
System.out.println(list.get(1));
```



a

```
System.out.println(list.remove(2));
```



b

```
System.out.println(list.get(2));
```



c

```
list.set(0, "e");
```



```
System.out.println(list);
```



{e, a, c}

output

```
}
```

Quick Reference Guide

```
class java.util.ArrayList<E>
```

- `int size()`
- `boolean add(E obj)` // appends `obj` to end of list; returns `true`
- `void add(int index, E obj)` // inserts `obj` at position `index` ($0 \leq \text{index} \leq \text{size}$),
// moving elements at position `index` and higher
// to the right (adds 1 to their indices) and adjusts size
- `E get(int index)`
- `E set(int index, E obj)` // replaces the element at position `index` with `obj`
// returns the element formerly at the specified position
- `E remove(int index)` // removes element from position `index`, moving elements
// at position `index + 1` and higher to the left
// (subtracts 1 from their indices) and adjusts size
// returns the element formerly at the specified position