

CSC 231- The ArrayList Data Structure

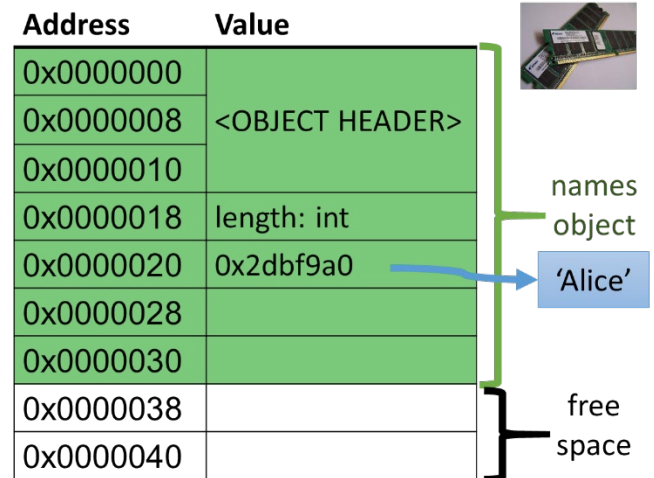
The Python `list[]` type implements an ArrayList data structure.

An array is a contiguous block of computer memory.

- HEADER is 24 bytes
- length is 8 bytes - returned when you call `len(list)`.
- Each object reference (values in the list) is 8 bytes
- The sizes depend on your computer, but will be constant

The main benefit of an array-based list is that we can access any element by index in $O(1)$ time by using arithmetic computations.

Python pre-allocates a larger block of memory for the list in anticipation you will add data to it



read by index

```
x = list[2]
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

write to index

```
list[2] = 'Eugene'
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

`list.append(value)`

```
list.append('Bart')
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

`list.pop()`

```
list.pop()
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

```
list.insert(index, value)
list.insert(0, 'Paris')
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

```
list.pop(index)
list.pop(0)
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

```
list.remove(value)
list.remove('Alice')
```

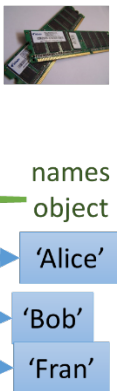
length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

```
list.remove('Horatio')
```

length = 6	
index	value
0	'Alice'
1	'Bob'
2	'Fran'
3	'John'
4	12345
5	10.0

```
list.append() and list.insert(index, value) – revisited
list.append('Bart')
```

Address	Value
0x00000000	<OBJECT HEADER>
0x00000008	
0x00000010	
0x00000018	length: int
0x00000020	0x2dbf9a0
0x00000028	0x42e7599
0x00000030	0x3f81634
0x00000038	<unrelated data>
0x00000040	<unrelated data>



names object

'Alice'

'Bob'

'Fran'