

Array	List	Queue
Declaration and initialization <pre>let array = ['cyber', 'sapientia']; let also = new Array(3);</pre>	Creation <pre>const list = new List();</pre>	Creation <pre>let queue = new Queue();</pre>
Adding an element <pre>array[3] = 'statistics';</pre>	Adding an element <pre>list.add(1); list.add('thead');</pre>	Adding an element <pre>queue.enqueue(1);</pre>
Setting a value <pre>array[1] = 'security';</pre>	Removing an element <pre>list.remove(2);</pre>	Fetch of the first element <pre>queue.front();</pre>
Deleting an element <pre>delete array[3];</pre>	Setting a value <pre>list.set(0, 1337);</pre>	Removing an element <pre>queue.dequeue();</pre>
Looping <pre>array.forEach(function(item) { console.log(item); });</pre>	Checking existence of a value <pre>list.contains(1);</pre>	Checking the existence of an element <pre>queue.contains(7);</pre>
Checking the existence of a value #1 <pre>array.includes('cyber');</pre>	Looping <pre>for(let i = 0; i < list.size(); i++) { console.log(list.get(i)); }</pre>	Stack
Checking the existence of a value #2 <pre>array.indexOf('statistics') !== -1</pre>	The type SortedList has the same methods, but ensures that the values are sorted.	Creation <pre>let stack = new Stack();</pre>
Checking the existence of a value #3 <pre>for (let i = 0; i < array.length; i++) { if (array[i] === searchValue) { found = true; break; } }</pre>	Dictionary	Adding an element <pre>stack.push(1);</pre>
Checking the existence of a value #4 <pre>array.find(item => item === searchValue);</pre>	Creation <pre>let dict = new Dictionary();</pre>	Fetching top of stack <pre>stack.peek();</pre>
<p>Arrays are also used to implement List, Queue, Stack. For these data structures I'll create a JS class that exposes useful methods.</p>	Adding an key-value pair <pre>dict.add('frequency', 20);</pre>	Removing an element <pre>stack.pop();</pre>
	Setting a value for a key <pre>dict.set('percentage', 40.10);</pre>	SortedSet
	Removing a key <pre>dict.remove('percentage');</pre>	Creation <pre>let sortedSet = new SortedSet();</pre>
	Checking existence of a key <pre>dict.hasKey('frequency');</pre>	Adding an element <pre>sortedSet.add(3);</pre>
	Checking existence of a value <pre>dict.hasValue(20);</pre>	Removing an element <pre>sortedSet.delete(3);</pre>
		Checking the existence of an element <pre>sortedSet.has(1);</pre>
		Looping <pre>for(element in sortedSet.toArray()) { console.log(element); }</pre>
LinkedList	HashSet	

Creation

```
let linkedList = new LinkedList();
```

Adding an element

```
linkedList.append(3);  
linkedList.prepend(1);  
linkedList.addAt(1, 3);
```

Fetch of the head value

```
linkedList.getFirst();
```

Get the value of a node

```
linkedList.getValueAt(2);
```

Removing an element

```
linkedList.removeAt(1);  
linkedList.delete(7);
```

Check existence of a value

```
linkedList.contains(7);
```

Looping

```
for (const element of linkedList)  
{  
    console.log(element);  
}
```

Creation

```
let hashSet = new HashSet();
```

Adding an element

```
hashSet.add(1);
```

Removing an element

```
hashSet.delete(1);
```

Checking the existence of an element

```
hashSet.has(3);
```

Looping

```
let keys = hashSet.values();  
for(let i = 0; i < keys.length; i++)  
{  
    console.log(keys[i]);  
};
```

The HashSet is pretty similar to the Dictionary, but it doesn't have values; it just uses the keys.



By Boge (username1)

cheatography.com/username1/

Published 16th October, 2023.

Last updated 16th October, 2023.

Page 1 of 2.

Sponsored by CrosswordCheats.com

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>