Cheatography

# stathw-js-cheatsheet Cheat Sheet

by Boge (username1) via cheatography.com/195153/cs/40822/

## Array

**Declaration and initialization**

```
let array = ['cyber', 'insecurity', 'sapienza'];
let also = new Array(3);
```

**Adding an element**

```
array[3] = 'statistics';
```

**Setting a value**

```
array[1] = 'security';
```

**Deleting an element**

```
delete array[3];
```

**Looping**

```
array.forEach(function(item){
    console.log(item);
});
```

**Checking the existence of a value #1**

```
array.includes('cyber');
```

**Checking the existence of a value #2**

```
array.indexOf('statistics') !== -1;
```

**Checking the existence of a value #3**

```
for (let i = 0; i < array.length; i++) {
    if (array[i] === searchValue) {
        found = true;
        break;
    }
}
```

**Checking the existence of a value #4**

```
array.find(item => item === searchValue) !== undefined;
```

Arrays are also used to implement List, Queue, Stack. For these data structures I'll create a JS class that exposes useful methods.

## LinkedList

## List

**Creation**

```
const list = new List();
```

**Adding an element**

```
list.add(1);
list.addAtHead(0);
```

**Removing an element**

```
list.remove(2);
```

**Setting a value**

```
list.set(0, 1337);
```

**Checking existence of a value**

```
list.contains(1);
```

**Looping**

```
for(let i = 0; i < list.size(); i++)
{
    console.log(list.get(i));
});
```

The type SortedList has the same methods, but ensures that the values are sorted.

## Dictionary

**Creation**

```
let dict = new Dictionary();
```

**Adding an key-value pair**

```
dict.add('frequency', 20);
```

**Setting a value for a key**

```
dict.set('percentage', 40.10);
```

**Removing a key**

```
dict.remove('perceptage');
```

**Checking existence of a key**

```
dict.hasKey('frequency');
```

**Checking existence of a value**

```
dict.hasValue(20);
```

## HashSet

## Queue

**Creation**

```
let queue = new Queue();
```

**Adding an element**

```
queue.enqueue(1);
```

**Fetch of the first element**

```
queue.front();
```

**Removing an element**

```
queue.dequeue();
```

**Checking the existence of an element**

```
queue.contains(7);
```

## Stack

**Creation**

```
let stack = new Stack();
```

**Adding an element**

```
stack.push(1);
```

**Fetching top of stack**

```
stack.peek();
```

**Removing an element**

```
stack.pop();
```

## SortedSet

**Creation**

```
let sortedSet = new SortedSet();
```

**Adding an element**

```
sortedSet.add(3);
```

**Removing an element**

```
sortedSet.delete(3);
```

**Checking the existence of an element**

```
sortedSet.has(1);
```

**Looping**

```
for(element in sortedSet.toArray()) {
    console.log(element);
});
```

### Creation

```
let linkedList = new Linked List();
```

### Adding an element

```
linked Lis t.a ppe nd(3);
linked Lis t.p rep end(1);
linked Lis t.a ddAt(1, 3);
```

### Fetch of the head value

```
linked Lis t.f etc hHe ad();
```

### Get the value of a node

```
linked Lis t.g etA t(2);
```

### Removing an element

```
linked Lis t.r emo veA t(1);
linked Lis t.d ele te(7);
```

### Check existence of a value

```
linked Lis t.c ont ain s(7);
```

### Looping

```
for (const element of linked List)
{
    consol e.l og( ele ment);
}
```

### Creation

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

### Adding an element

```
hashse t.a dd(1);
```

### Removing an element

```
hashse t.d ele te(1);
```

### Checking the existence of an element

```
hashse t.h as(3);
```

### Looping

```
let keys = hashse t.v alu es();
for(let i = 0; i < keys.l ength; i++)
{
    consol e.l og( key s[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.