Array

Declaration and initialization
```
let array = ['cat', 'dog', 'fish'];
let array2 = new Array(3);
```
Adding an element
```
array[3] = 'dog';
```
Setting a value
```
array[1] = 'cat';
```
Deleting an element
```
delete array[3];
```
Looping
```
array.f o r E ac  h(f  un c t io
n(item) {
    consol  e.l  o g(  item);
}
```
Checking existence of a value
```
array.i nc lud e(' cat');
```

Dictionary

Creation
```
let dict = new Dictio  na ry();
```
Adding an key-value pair
```
dict.a  dd ( ' cat', 10);
```
Setting a value for a key
```
dict.s  et ( ' pe  rce  nt age',
 5.10);
```
Removing a key
```
dict.r  em o v e(  'pe  rc e n t
a  ge');
```
Checking existence of a key
```
dict.h  as K e y(  'fr  eq u e n
cy');
```
Checking existence of a value
```
dict.h  as V a lu  e(10);
```

HashSet

Creation
```
let hashset = new HashSet();
```
Adding an element
```
hashse  t.a  d d(1);
```
Removing an element
```
hashse  t.d  e le  te(1);
```
Checking the existence of an element
```
hashse  t.h  a s(3);
```

HashSet (cont)

Looping
```
let keys = hashse  t.v  a lu  es();
for(let i = 0; i < keys.l  ength; i++
){
    consol  e.l  o g(  key s[i]);
}
```

List

Creation
```
const list = new List();
```
Adding an element
```
list.a  dd(1);
list.a  dd A t He  ad(0);
```
Removing an element
```
list.r  em o v e(2);
```
Setting a value
```
list.s  et(0, 251);
```
Checking existence of a value
```
list.c  on t a in  s(1);
```
Looping
```
for(let i = 0; i < list.s  ize(); i++
){
    consol  e.l  o g(  lis  t.g  e t(
i));
}
```

Stack

Creation
```
let stack = new Stack();
```
Adding an element
```
stack.p  u s h(1);
```
Fetching top of stack
```
stack.p  e ek();
```
Removing an element
```
stack.p  op();
```

SortedSet

Creation
```
let sortedSet = new Sorted  Set(
);
```
Adding an element
```
sorted  Se t.a d d(3);
```
Removing an element
```
sorted  Se t.d e l et  e(3);
```

SortedSet (cont)

Checking the existence of an element
```
sorted  Se t.h a s(1);
```
Looping
```
for(el  ement in sorted  Se t.t o
rray) {
    consol  e.l  o g(  ele  ment);
}
```

Queue

Creation
```
let queue = new Queue();
```
Adding an element
```
queue.e  n q u eu  e(1);
```
Fetch of the first element
```
queue.f  r o n t();
```
Removing an element
```
queue.d  e q u eue();
```
Checking the existence of an element
```
queue.c  o n t ai  ns(7);
```

LinkedList

Creation
```
let linkedList = new Linked  Li st
();
```
Adding an element
```
linked  Li s t.a  ppe  nd(1);
linked  Li s t.p  rep  en d(1);
linked  Li s t.a  ddAt(1, 1);
```
Fetch of the head value
```
linked  Li s t.f  etc  hH e ad();
```
Get the value of a node
```
linked  Li s t.g  etA  t(1);
```
Removing an element
```
linked  Li s t.r  emo  ve A t(1);
linked  Li s t.d  ele  te(1);
```
Check existence of a value
```
linked  Li s t.c  ont  ai n s(1);
```
Looping
```
for (const e of linked  List) {
    consol  e.l  o g(  e);
}
```