

Arrays are used both on their own and to implement many other data structures that place additional restrictions on how the data is manipulated.

Algorithmic complexity, strict arrays compared with JavaScript

In terms of data retrieval arrays have a good runtime since retrieving an element or storing an element takes constant time. Also an entire array takes up $O(n)$ space.

This only applies in the case of strict arrays that exist in strictly typed and more advanced languages such as Java and C++. In JavaScript an array is really closer to a list in other languages. That is to say, its size (and of course the types it holds) does not need to be known in advance or specified at all. In stricter languages, you would cast the type for the array and specify its length, for example:

```
int anArray[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }
```

Whilst storage and retrieval take place in constant time, deleting and inserting elements through methods like `pop` and `shift` are more time consuming, since the index of every item has to be updated in response.

Random access

As we can access any element of the array at any time through index notation, arrays offer random and non-sequential access. This is in contrast to stacks and queues (where only the top/bottom can be accessed) and linked lists.