Day 3: Arrays

by AvmnuSng

Arrays in JavaScript

Arrays

The JavaScript Array object is a global object that is used in the construction of arrays; which are high-level, list-like objects.

ARRAYS Recommended Article

View

Arrays

New to working with Arrays? Learn more about this data structure in our *Arrays* article.

Click the *Run* button in the examples below to execute the sample code.

1. Create an Array

EXAMPLE

```
var a = ['first', 'second'];

console.log('a\'s contents:', a);
console.log('a\'s length:', a.length);
```

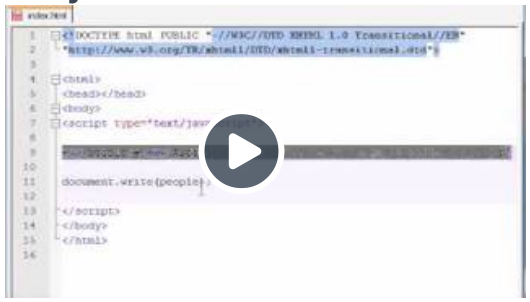
Output

Run

2. Access (Index Into) an Array Item

OK

Arrays



```
let a = ['first', 'second'];

// first = 'first'
let first = a[0];

// last = 'second'
let last = a[a.length - 1];

console.log('a[0]:', first);
console.log('a[a.length - 1]:', last);
```

Output

Run

3. Loop Over an Array

\- EXAMPLE

```
var a = ['first', 'second'];

a.forEach(function(e, i, array) {
    // 'i' is the index
    // 'e' is the element
    console.log(i + ' ' + e);
```

Output

Run

4. Append to the End of an Array

EXAMPLE

```
var a = ['first', 'second'];

// Append 'third' to array 'a'
a.push('third');

console.log('a:', a);
```

Output

Run

5. Remove From the end of an Array

```
var a = ['first', 'second', 'third'];
console.log('Original Array:', a);

// Remove the last element from the array
let removed = a.pop();

console.log('Modified Array:', a);
console.log('Removed Element:', removed);
```

Output

Run

6. Remove From the Front of an Array

\- EXAMPLE

```
var a = ['first', 'second', 'third'];
console.log('Original Array:', a);

// Remove the first element from the array
let removed = a.shift();

console.log('Modified Array:', a);
console.log('Removed Element:', removed);
```

Output

Run

7. Add to the Front of an Array

- EXAMPLE

```
var a = ['first', 'second', 'third'];
console.log('Original Array:', a);

// Insert element at the beginning of the array
a.unshift('fourth');

console.log('Modified Array:', a);
```

Output

Run

8. Find the Index of an Item in the Array

```
var a = ['first', 'second', 'third', 'fourth'];

let position = a.indexOf('second');

console.log('a:', a);
console.log('position:', position);
```

Output

Run

9. Remove an Item by Index Position

\- EXAMPLE

```
var a = ['first', 'second', 'third', 'fourth', 'fifth'];
console.log('Original Array:', a);

let position = 1;
let elementsToRemove = 2;
// Remove 'elementsToRemove' element(s) starting at 'position'
a.splice(position, elementsToRemove);

console.log('Modified Array:', a);
```

Output

Run

10. Copy an Array

- EXAMPLE

```
var a = ['first', 'second', 'third', 'fourth'];
console.log('a:', a);

// Shallow copy array 'a' into a new object
let b = a.slice();

console.log('b:', b);
```

Output

Run

11. Sort an Array

```
var a = ['c', 'a', 'd', 'b', 'aa'];
var b = [9, 2, 13, 7, 1, 12, 123];

// Sort in ascending lexicographical order using a built-in
a.sort();
b.sort();

console.log('a:', a);
console.log('b:', b);
```

Output

Run

- EXAMPLE

```
var a = ['c', 'a', 'd', 'b', 'aa'];
var b = [9, 2, 13, 7, 1, 12, 123];

// Sort in descending lexicographical order using a compare function
a.sort(function(x, y) { return x < y; } );
b.sort(function(x, y) { return x < y; } );

console.log('a:', a);
console.log('b:', b);
```

Output

- EXAMPLE

```
var a = ['c', 'a', 'd', 'b', 'aa'];

// Sort in descending lexicographical order using a compare arrow fu
a.sort((x, y) => x < y);

console.log('a:', a);
```

Output

Run

12. Iterate Over an Array

We can use the *for...of* statement to iterate over an array. This type of statement creates a loop that lets you iterate over iterable objects such as *Array*, *String*, *Set*, and *Map*.

- EXAMPLE

```
var a = ['first', 'second', 'third', 'fourth'];
```

```
5 }
```

Output

Run