JavaScript Array flat() Method

The flat() method is an inbuilt array method that flattens a given array into a newly created one-dimensional array. It concatenates all the elements of the given multidimensional array, and flats upto the specified depth. We can specify the depth limit to where we need to flatten the array. By default, the depth limit is 1.

Syntax

```
var newArr=arr.flat(<depth>);
```

Parameters

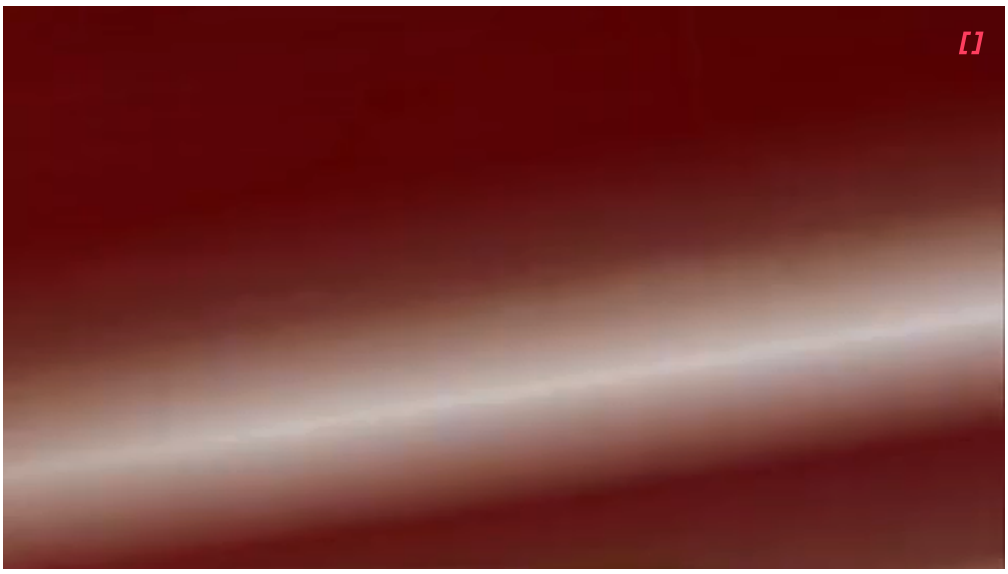
Depth: It is an optional parameter which specifies the depth to flatten an array. By default, its value is 1.

Return

It returns a newly created array containing all the sub-array elements concatenated into it.

JavaScript Array flat() Method Example

Let's see the below examples to understand better.



Example1

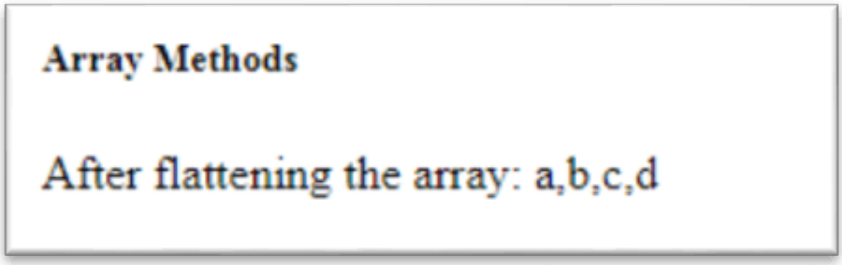
A simple working of flat() method on a two dimensional array.

```
<html>
<head> <h5> Array Methods </h5> </head>
```

```
var arr=['a','b',['c','d']]; //given 2D array
var newArr=arr.flat(); //using flat() method
document.write("After flattening the array:  "+newArr);
</script>
</body>
</html>
```

Test it Now

Output:



Example2

Testing a multidimensional array with flat() method.

```
<html>
<head> <h5> Array Methods </h5> </head>
<body>
<script>
var arr=[90,18,[89,56],[13,20,[67,17]]]; //given multidimensional array
var newArr=arr.flat(); //using flat() method
document.write("After flattening the array:  "+newArr);
</script>
</body>
</html>
```

Test it Now

Output:

It is clear that each element in the array is concatenated to the newly created 1D array.

⇧ SCROLL TO TOP

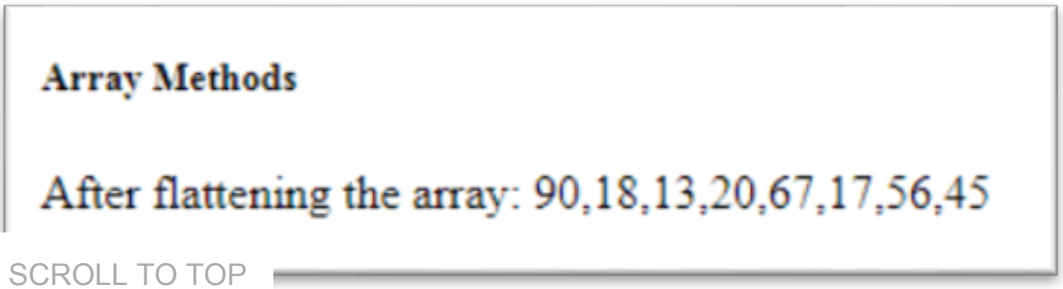
Example3

Let's flatten an array upto a specified depth.

```
<html>
<head> <h5> Array Methods </h5> </head>
<body>
<script>
var arr=[90,18,[13,20,[67,17,[56,45]]]]; //given multidimensional array
var newArr=arr.flat(3); //using flat() method with a specified depth value.
document.write("After flattening the array:  "+newArr);
</script>
</body>
</html>
```

Test it Now

Output:



⇧ SCROLL TO TOP

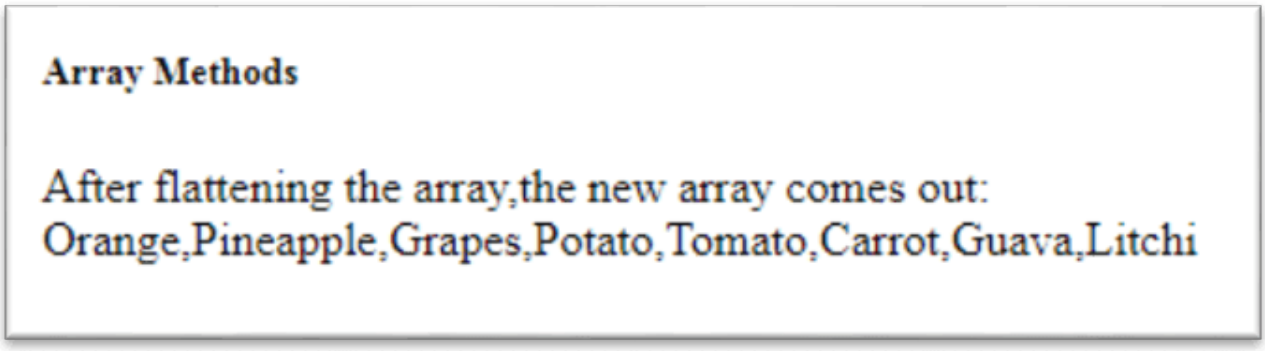
Example4

Using flat() method with depth value as infinity.

```
<html>
<head> <h5> Array Methods </h5> </head>
<body>
<script>
var arr=['Orange','Pineapple','Grapes',['Potato','Tomato','Carrot',['Guava','Litchi']]]; //given a multidimensional array.
var newArr=arr.flat(Infinity); //setting depth value as infinity.
document.write("After flattening the array,the new array comes out: <br> "+newArr);
</script>
</body>
</html>
```

Test it Now

Output:



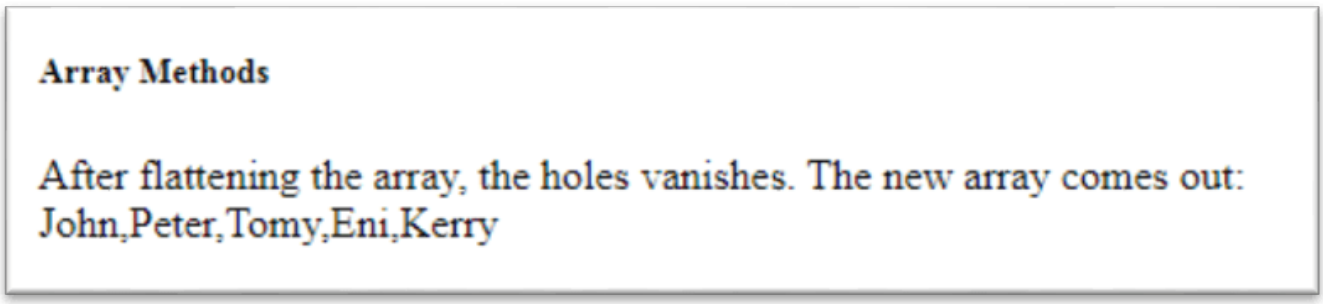
Example5

⇧ SCROLL TO TOP vith holes in between.

```
<html>
<head> <h5> Array Methods </h5> </head>
<body>
<script>
var arr=['John','Peter',,'Tomy',['Eni',,'Kerry']]; //given 2D array with holes in between.
var newArr=arr.flat(); //using flat() method.
document.write("After flattening the array, the holes vanishes. The new array comes out: <br> "+new
</script>
</body>
</html>
```

Test it Now

Output:



It is clear that all array elements are concatenated leaving the holes out, after applying the flat() method.

← Prev Next →

⇧ SCROLL TO TOP