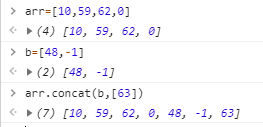
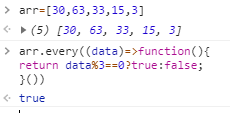
Functions performed on array

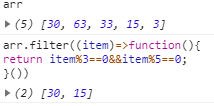
1. concat(): This function is performed on array. It takes one or more arrays as argument and concatenates them to the called array object and forms new array. The original array is not altered. Instead a new array is returned.



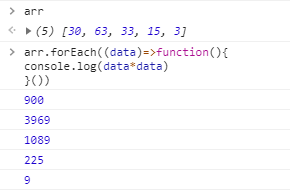
1. every(): This function works on each element of the array and returns true if and only if all the elements pass a test condition, otherwise it returns false. This function doesn’t test the condition for undefined values and key-value pairs in the array.



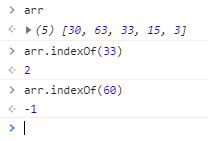
1. filter():This function is carried on array. It creates new array, whose elements are those which obey supplied condition.



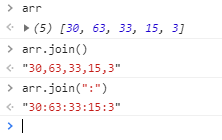
1. forEach(): This function is carried on array. It is used to traverse whole array. It doesn’t traverse the key-value pairs or undefined values.



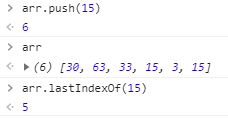
1. indexOf(): This returns the first index of the given element in array. If the element is not found in the array, this function returns -1.



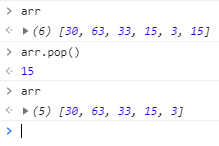
1. join(): This function is carried on array. It joins the array elements as a string and returns the string. We can supply a string or number to the function and it acts like joiner. That is between every elements the argument is inserted.



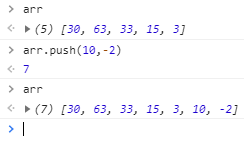
1. lastIndexOf(): This function returns the last index of the given element in the array. If the element is not found, it returns -1.



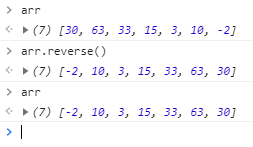
1. pop(): This function removes the last element in the array and returns the removed element. This also removes the empty/undefined element. But it doesn’t work on key-value pairs.



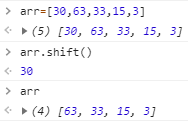
1. push():This function takes variable number of objects and adds those objects at the end of array. It returns the length of new array.



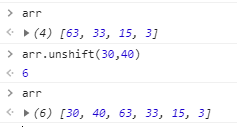
1. reverse(): This function is used to reverse the order of elements in the array. This function doesn’t reverse the key-value pairs in array. The changes are reflected in the original array.



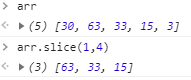
1. shift(): This function removes first element of the array, and returns the same.



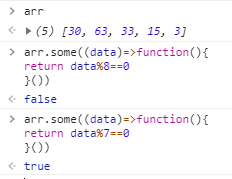
1. unshift(): This function takes variable number of arguments and adds those at the beginning of the array. It returns the length of new array.



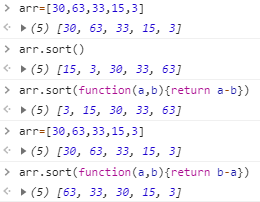
1. slice(): This function is used extract parts of an array. It takes 2 optional arguments, start and end index. If both are specified, it returns new array from the start position till the end-1 position. If end position is not specified, it returns a new array from start position till the last element of the array. If both are not specified, this function returns the same array, which invoked the function. This function doesn’t consider key-value pairs in the array if present.



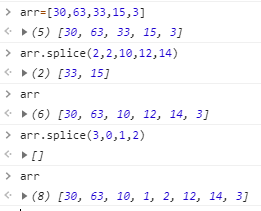
1. some(): This function returns true if at-least one element of the array obeys the given condition. Else it returns false.



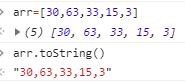
1. sort(): This function is used to sort the array. If the array contains integers, the integer elements are sorted by considering them as string. We can supply our custom comparator to sort the array. It doesn’t consider key-value pair for sorting.



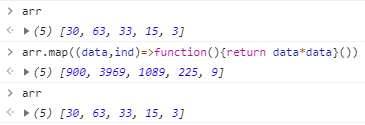
1. splice(): This function adds and/or deletes elements in an array. It takes 3 arguments. First argument tells the starting position, Second argument tells the number of elements to be deleted starting from the starting position, Third argument is of variable type which is the elements to be added at the starting position.



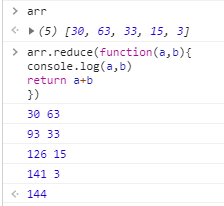
1. toString(): This function returns the string representation of the array. It doesn’t consider the key-value pair present in the array.



1. map(): This function is used to traverse the array, modify the array elements, and return a new array. The original array is unchanged. This function takes a callback function.



1. reduce(): This function is used to reduce the array to single value. This function traverses the array from left to right. It takes a callback function whose arguments are optional. The first argument is the resultant of the previous operation. Second argument is the next element in the array. Third argument is the index of the next element and fourth argument is the array itself. We can provide logic of reducing the array to single value inside the function.



1. reduceRight(): This function is used to reduce the array to single value. This function traverses the array from right to left. It works similar to reduce().

