

## Методы массива

- // квадратные скобки (обычно).  
***let arr = [item1, item2...];***
- // new Array (очень редко).  
***let arr = new Array(item1, item2...);***
- Вызов ***new Array(number)*** создаёт массив с заданной длиной.
- Свойство ***length*** отражает длину массива или, если точнее, его индекс плюс один. Длина корректируется автоматически методами.
- Если мы уменьшаем ***length*** вручную, массив укорачивается.

### ➤ Чтобы пройти по элементам массива:

1. ***for (let i = 0; i < arr.length; i++)*** – работает быстрее всего со старыми браузерами.
2. ***for (let item of arr)*** – современный синтаксис только для значений, к индексам нет доступа).
3. ***for (let i in arr)*** – никогда не используйте для массивов.

### ➤ Для добавления/удаления элементов:

1. ***push (...items)*** – добавляет элементы в конец.
2. ***pop()*** – извлекает элемент с конца.
3. ***shift()*** – извлекает элемент с начала.
4. ***unshift(...items)*** – добавляет элементы в начало.
5. ***splice(pos, deleteCount, ...items)*** – начиная с индекса *pos* удаляет *deleteCount* элементов и вставляет *items*.
6. ***slice(start, end)*** – создаёт новый массив, копируя в него элементы с позиции *start* до *end* (не включая *end*).
7. ***concat(...items)*** – возвращает новый массив: копирует все члены текущего массива и добавляет к нему *items*. Если какой-то из *items* является массивом, тогда берутся его элементы.

### ➤ Для поиска среди элементов:

1. ***indexOf/lastIndexOf(item, pos)*** – ищет *item*, начиная с позиции *pos*, и возвращает его индекс или *-1*, если ничего не найдено.
2. ***includes(value)*** – возвращает *true*, если массив имеет значение *value*, в противном случае *false*.
3. ***find/filter(func)*** – фильтрует элементы через функцию и отдаёт первые/все значения, которые при прохождении через неё возвращают *true*.
4. ***findIndex*** – похож на *find*, но возвращает индекс вместо значения.

➤ Для перебора элементов:

1. ***forEach(func)*** – вызывает *func* для каждого элемента. Ничего не возвращает.

➤ Для преобразования массива:

1. ***map(func)*** – создаёт новый массив из результатов вызова *func* для каждого элемента.
2. ***sort(func)*** – сортирует массив на месте, а потом возвращает его.
3. ***reverse()*** – на месте меняет порядок следования элементов на противоположный и возвращает изменённый массив.
4. ***split/join*** – преобразует строку в массив и обратно.
5. ***reduce(func, initial)*** – вычисляет одно значение из всего массива, вызывая *func* для каждого элемента и передавая промежуточный результат между вызовами.

➤ Дополнительно:

1. ***Array.isArray(arr)*** - проверяет, является ли *arr* массивом.
2. Обратите внимание, что методы ***sort***, ***reverse*** и ***splice*** изменяют исходный массив.
3. ***arr.some(fn)/arr.every(fn)*** - проверяет массив. Функция *fn* вызывается для каждого элемента массива аналогично *map*. Если какие - либо/все результаты являются *true*, метод возвращает *true*, иначе *false*.
4. ***arr.fill(value, start, end)*** – заполняет массив повторяющимися *value*, начиная с индекса *start* до *end*.
5. ***arr.copyWithin(target, start, end)*** – копирует свои элементы, начиная со *start* и заканчивая *end*, в собственную позицию *target* (перезаписывает существующие).