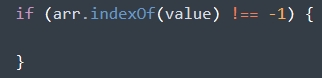
**Проверить наличие элемента в массиве**

**Индекс первого вхождения элемента в массиве.**

**arr.indexOf(searchElement, [startPos])** - возвращает индекс первого совпадения, по которому переданный элемент может быть найден в массиве или -1, если такого индекса нет.

Примечание:

* Использует строгое сравнение.
* Проверить наличие элемента в массиве:



* При поиске строкового значения, учитывать регистр символов.
* Применяется к строке как к массиву символов. **ВАЖНО**! Поиск значения в строке происходит след. образом: ищет подстроку в строке и возращет индекс или -1.
* Аргумент «startPos» - позиция, с которой включительно начать поиск.

**Индекс последнего совпадения элемента.**

**arr.lastIndexOf(searchElement, [from])** – возвращает индекс последнего вхождения элемента, по которому элемент находится в массиве. Иначе -1.

**Проверит наличие элемента в массиве**

**arr.includes(value, [pos])** - проверят, если массив содержит переданное значение, то возвращает true иначе - false.

Замечания:

* При работе с данными в массивах, имена переводят в нижний регистр, чтобы упростить последующую работу с данными. При выводе можно форматировать так как нужно.
* Проблемы с регистром символов у includes().
* Ищет значения NaN, (null, undefined).

**Найти значение в массиве и получить его.**

**arr.find(cb(item, index, arr))** – для каждого элемента массива вызывает функцию, если функция, по условию, возвращает true, то метод вернёт данный элемент массива и остановит цикл. Если элемент не будет найден, вернёт undefined.

Применение:

* Найти первое совпадение искомого элемента.
* Найти элемент массива, который первый попадает по условия if
* Поиск конкретного значения.
* Поиск значения в определённом диапазоне.
* Не работает то строками(String)

**Индекс первого совпадения элемента.**

**arr.findIndex(cb(item, index, arr)** – для каждого элемента массива вызывает функцию-предикат, если функция, по условию вернёт true, то метод вернёт индекс данного элемента массива и остановит цикл. Если элемент не будет найден, вернёт -1.

**Манипуляция с элементами массива**

**arr.push(element/elements)** - добавить элемент/ы в конец массива и возвращает новую длину массива – обновлённое свойство length. **Изменяет исходный массив.**

Примечания:

* Массив элементов добавится как 1 элемент.
* После применения метода, если обратиться к массиву после изменения его методом, массив будет изменным. А до применения методом - нет.

**arr.pop()** - удаляет последний элемент массива и возвращает его. **Изменяет исходный массив**.

Примечания:

* Удаление элементов в цикле с исп. pop(), нужно проверить возвращаемое значение на «undefined». В зависимости от условия остановить цикл. if (arr.pop() !== 'undefined') {}

**arr.shift()** - удаляет первый элемент из массива и возвращает его значение. **Изменяет исходный массив**. При удалении элемента, происходит переиндексация массива – изменение индексов массива(сдвигает все индексы и пересчитывает массив). Это затратная операция, не рекомендуется применять.

**arr.unshift(elem, elem)** - добавляет один или более элементов в начало массива и возвращает новую длину массива.

* **Изменяет исходный массив**.
* **Переиндексирует массив. Не рекомендуется к применению на больших структурах.**

**Скопировать массив.**

**arr.slice([begin, end])** - копирует значения массива с позиции «**begin**» включительно и до позиции «***end»***  не включая её. Возвращает новый массив. Если не указать аргументов, то скопирует полностью.

* Можно применять к строке

**Произвести какое-либо действие с каждым элементом массива**

**arr.map(callback())** - применяет функцию к каждому элементу массива и добавляет элемент в новый массив.

Применение:

* перебрать весь массив и над каждым элементом произвести операцию и возвратить.
* В map можно изменят элементы исходного массива до команды return, и метод будет работать с уже изменённым элементом.
* Метод возвращает: элемент массива, индекс, исходный массив.
* Если массив разряженный, то возвращаемый элемент и индекс будет empty(пустым). Не производит над разряженным элементом никаких действий, а просто возвращает его.

**Вернуть элементы массива, которые попадают под условие.**

**arr.filter(cb)** - создаст новый массив из элементов, для которых callback-функции вернула true.

* Если выражение в return будет интерпретировано как true, тогда элемент массива будет добавлен в новый массив. ВОЗВРАЩАТЬ нужно логическое выражение и использовать его в ***if () { return true },*** чтобы избежать ошибки. Если возвратить элемент массива, который равен 0, то он будет интерпретирован как false.
* Filter возвращает только элемент, а не index или array.

**Произвести какое-либо действие с элементами массива и вернуть единственное значение.**

**arr.reduce(cb(accum, item, index, arr))** - Возвращает значение, которое накопилось в accum.

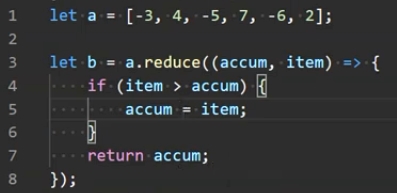
* accum – переменная накопитель, счётчик.
* item – текущий элемент массива.

Работа:

* В accum записывает первый элемент массива и запускает цикл со второго элемента массива. На каждой итерации цикла, можно накапливать переменную accum текущим элементом массива(или производить любое другое действие).
* В return нужно возвращать accum.
* Внутри reduce можно задавать начальное значение accum, arr.reduce(cb(), 0 - Значение acum по умолчанию), тогда цикл начнётся с первого элемента массива. В таком случае, если первый элемент массива будет отрицательным числом, это поможет исключить его попадание в accum и суммирование с последующими, например, положительными числами.

Применение:

* Найти максимальный/минимальный элемент массива.



* Склеить значения.
* Переиндексировать массив - создать на основе массива объект.
* Найти сумму/разность/произведение элементов, которые подходят под заданное условие.
* Переиндексировать массив - создать на основе массива объектов по какому-либо параметру упрощённый массив.
* Везде, где нужно получить одно число на основе массива.

**Заменить и/или удалить элементы в заданной позиции.**

**arr.splice(start, startCount, [item])** - удаляет или добавляет элементы в массив в заданной позиции и возвращает удалённые элементы.

* **start** - индекс, с которого начать изменять.
* **startCount** - количество удаляемых элементов.
* **item** – элемент(-ы), который нужно добавить.

Примечания:

* Изменяет исходный массив.
* Нагружает систему на больших массивах, т.к. при каждом удалении пересчитывает индексы.

**Соединить массивы**

**arr.concat(arr/value)** - возвращает новый массив, который состоит из массива на котором он был вызван, соединённого с массивом(-и) или значениями переданными в аргументе.

* Осуществляет поверхностное копирование:
  + Если массив содержит **простые значения**, то передача происходит по значению,
  + Если массив содержит вложенные массивы, объекты, то передача будет по ссылке. Это значит, если изменить исходный массив, новый массив тоже изменится.
* Соединяет разряженные массивы.
* Можно склеивать строки.

**Преобразовать массив в строку.**

**arr.join('')** - объединяет все элементы массива в строку.

Примечания:

* Если разделитель не указан, то разделение идёт через запятую.

let str = ['h', 'e', 'l', 'l', 'o'].join() // h,e,l,l,o

**Преобразовать строку в массив строк.**

**arr.split()** - разбивает строку на массив строк.

Применение:

* Можно указать второй аргумент - количество элемнтов(символов строки), которые нужно добавить в массив.
* В качестве разделителя может принимать регулярные выражения.
* Без аргумента, преобразует строку в 1 элемент массива.

'hello petr billy'.split(' ') // ['hello petr billy']

* Аргумент «''» - пустая строка, разобьёт строку на массив символов.
* Аргумент ' ' разобьёт строку на массив строк через пробел

'hello petr billy'.split(' ') // ['hello', 'petr', 'billy']

**arr.every(cb())** - для каждого элемента массива запускает функцию-предикат, если функция для всех элементов возвратит true, то метод вернёт значение true, иначе false. Если хоть один элемент будет равен false, тогда вернёт false.

Примечания:

* Для пустого массива возвращает true
* Если в методе до возвращения значения изменить исходный массив, например, добавить элемент. Исходный массив изменится, но метод every будет работать с неизменённым массивом.

**arr.some()** - для каждого элемента массива запускает функцию-предикат, если функция хоть для одного элемента возвратит true, то метод вернёт в переменную значение true, иначе false.

Примечания:

* Для пустого массива вернёт false