|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва методу | **Що робить** | **Приймає** | **Повертає** | **Callback: in, out** |
| ***.push(arg)*** | Додає аргумент в кінець масиву | Аргумент | Довжину нового масиву | ----------------- |
| ***.pop()*** | Удаляет последний элемент из массива и возвращает этот элемент. | Нічого | Видалений елемент або undefined якщо він пустий | ------------------ |
| ***.unshift(arg)*** | Додає вказані елементи на початок масиву та повертають нову довжину масиву. | Аргумент | Довжину нового масиву | -------------------- |
| ***.shift()*** | Видаляє перший елемент із масиву | Нічого | повертає цей видалений елемент | ------------------- |
| ***.splice(start, deleteCount, item)*** | Змінює вміст масиву, видаляючи або замінюючи існуючі елементи та/або додаючи нові елементи | **Start** - Індекс від нуля, з якого починається зміна массиву, **deleteCount(opt)-** що вказує кількість елементів у масиві, які потрібно видалити з початку  **Item1...n**-елементи які треба додати | Масив, що містить видалені елементи або пустий масив |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва методу | **Що робить** | **Приймає** | **Повертає** | **Callback: in,out** |
| ***.sort(callback)*** | Сортує елементи масиву на місці та повертає посилання на той самий масив, тепер відсортований | Fn**(a** - Перший елемент для порівняння. Ніколи не буде undefined, **b**-другий елемент  ) | Посилання на вихідний масив, тепер відсортований. Зауважте, що масив сортується на місці, а копія не створюється. | **In** – a, b  **Out** -число, де:  Від’ємне значення означає, що a має бути перед b.  Додатне значення вказує на те, що a має стояти після b.  Нуль або NaN означає, що a і b вважаються рівними. |
| ***\*.slice(start-o, end-o)*** | Повертає вирізану частину масиву від start до end(не включається) | 2 параметри(індекси) | Масив або строку з вибраних елементів | --------------------- |
| ***\*.reduce(calb)*** | Виконує функцію для кожного елемента | Функцію калбек, початкове значення **acc** | Одне значення | In: acc-Значення, отримане в результаті попереднього виклику callbackFn. Під час першого виклику його значення дорівнює initialValue, якщо вказано останнє; інакше його значенням є array[0], currentValue-Значення поточного елемента |
| ***\*.map(calb)*** | Створює новий масив, заповнений результатами виклику наданої функції для кожного елемента в масиві, що викликає. | Функ-калбек, thisArg(optional) | Новий масив | **In:** element, index, array  **Out**: Повернене значення додається як один елемент у новий масив. |
| ***\*.filter(fn, this)*** | Створює поверхневу копію частини даного масиву, відфільтровану лише до елементів із даного масиву, які пройшли перевірку | Callback, thisArg(optional) | Масив | **In:** element, index, array  **Out:** boolean |
| ***\*.find(fn, this)*** | Повертає перший елемент у наданому масиві, який задовольняє надану функцію | Callback, thisArg(optional) | Element | **In:** element, index, array  **Out:** boolean |
| ***\*.every(fn, this)*** | Перевіряє, чи всі елементи в масиві проходять перевірку, реалізовану наданою функцією. | Callback, thisArg(optional) | Boolean | **In:** element, index, array  **Out:** boolean |
| ***\*.join()*** | Сворює та повертає новий рядок шляхом об’єднання всіх елементів цього масиву, розділених комами або вказаним рядком-роздільником | Separator: , space, ; etc | Рядок із об’єднаними всіма елементами масиву. | -------------------------- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **.forEach()** | Виконує вказану функцію один раз для кожного елемента масиву. | Callback, thisArg(optional) | undefined | **In:** element, index, array |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |