|  |  |  |
| --- | --- | --- |
| **fill()** | array.fill(value, start, end) | **Заповнення** всіх (або деяких) елементів масиву фіксованим значенням |
|  |  |  |
| **for ... of** | **for** (let el **of** array) { .. } | **Перегляд** елементів ітерованого об'єкта **без їх зміни** |
| **for** |  | **Перегляд** та **внесення змін до елементів** масиву |
| **forEach** | array.**forEach(**  (el, index, baseArr) => умова **)** | **Перегляд** та **внесення змін до елементів** масиву **без** використання **індекса** |
| **every** | аrray.**every(**  (el, index, baseArr) => умова **)** | **Перегляд елементів**, повертає **true,** якщо усі задовільняють заданій умові |
| **some** | аrray.**some(**  (el, index, baseArr) => умова **)** | **Перегляд елементів**, повертає **true,** якщо хочаб один задовільняє заданій умові |
|  |  |  |
| **map** | аrray.**map(**  (el, index, baseArr) => умова **)** | **Створення нового масиву** на основі іншого, з таким же length, з можливою зміною елементів які задовільняють заданій умові |
| **filter** | аrray.**filter(**  (el, index, baseArr) => умова **)** | **Створення нового масиву** на основі іншого, з будь яким length, з можливою зміною елементів які задовільняють заданій умові |
|  |  |  |
| **reducе** | аrray.**reduce (**  (prevSum,el, index, baseArr) => умова, початкове значення **)** | **Обчислення** одного значення, на основі елементів масиву |
|  |  |  |
| **push** | аrray.**push (** нове значення **)** | **Додавання** нового значення (**елемента**) в кінець масиву |
| **unshift** | аrray.unshift**(** нове значення **)** | **Додавання** нового значення (**елемента**) у початок масиву |
| **splice** | аrray.splice**(**початковий індекс, скільки видалити, перелік нових значень **)** | **Додавання** нового значення (**елемента**) у будь-яке місце масиву |
|  |  |  |
| **pop** | аrray.pop**()** | **Видалення** та повернення останнього **елемента** з масиву |
| **shift** | аrray.shift**()** | **Видалення** та повернення першого **елемента** з масиву |
| **splice** | аrray.splice**(**початковий індекс, скільки видалити (повернути) **)** | **Видалення** та повернення **елементів** з масиву |
|  |  |  |
| **slice** | аrray.slice**()** | **Копіювання** всього масиву |
| **slice** | аrray.slice**(**початковий індекс, кінцевий індекс (не включаючи) **)** | **Копіювання** частини масиву |
|  |  |  |
| **indexOf** | аrray.indexOf **(**пошукове значення, початковий індекс (не обов'язково) **)** | **Поверне індекс** першого елемента значення якого **=** вказаному |
| **lastIndexOf** | аrray.lastIndexOf**(**пошукове значення, початковий індекс (не обов'язково)**)** | **Поверне індекс** останнього елемента значення якого **=** вказаному |
| **findIndex** | аrray.findIndex**(**  (el, index, baseArr) => умова **)** | **Поверне індекс** першого елемента, значення якого задовільняє умові |
| **findLastIndex** | аrray.findLastIndex**(**  (el, index, baseArr) => умова **)** | **Поверне індекс** останнього елемента, значення якого задовільняє умові |
|  |  |  |
| **find** | аrray.find**(**  (el, index, baseArr) => умова **)** | **Поверне** перший **елемент**, значення якого задовільняє умові |
| **findLast** | аrray.findLast**(**  (el, index, baseArr) => умова **)** | **Поверне** останній **елемент**, значення якого задовільняє умові |
|  |  |  |
| **includes** | аrray.includes**(**el**)** | Дозволяє визначити **чи є** **елемент** у масиві |
|  |  |  |
| **sort()** | аrray.sorts**()** | Сортування у порядку зростання. Усі елементи будуть перетворені у **string** |
| **sort(**el1, el2) | аrray.sorts**(**(el1, el2) => el1 - el2**)** | Сортування елементів згідно правилу порівняння |
|  |  |  |
| **at** | аrray.at**(** **- 5** **)** | Доступ до елмента за індексом, від **правого** краю масиву (починаючи з **-1**) |
|  |  |  |
| **split()** | аrray.includes**(**‘ **;** ’**)** | створює масив з елементів рядка, які розділяються **вказаним** **знаком** |
| **join()** | аrray.join**(**‘ **;** ’**)** | створює рядок з масиву, в якому елементи розділяються **вказаним** **знаком** |
|  |  |  |
| **concat** | аrray.concat**(**array2**)** | Повертає новий масив, який складається з елементів двох масивів |