**Array methods**

**Методи добавлення-видалення елементів:**

1. arr**.push(**item**)** – добавляє елемент в кінець масива, **повертають довжину оновленого масиву**
2. arr**.pop()** – видаляє останній елемент масива, **повертає видалений елемент**
3. arr**.shift()** -видаляє перший елемент масива, **повертає видалений елемент**
4. arr**.unshift(**item**)** – добавляє елемент на початок масива, **повертають довжину оновленого масиву**

**Методи формування-зміни елементів:**

1. str**.split(**str**)** – **метод розбиває строку на масив елементів**, без параметрів по буквенно, з параметром, розбиває строку по вказаному параметру
2. arr**.join()** – **повертає строку** складену із усіх елементів масива
3. arr.**reverse()** – **повертає масив перевертиш**

**Методи редагування структури масива:**

1. [arr**.splice(**start-index, count, param1, param2,…**)**](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/splice) -редагує поточний масив. **Повертає масив видалених елементів**

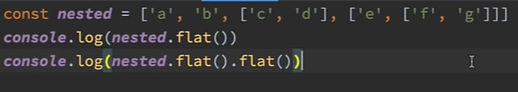
**а. Для заміни поточних елементів** масива новими параметрами – вказуємо діапазон на якому потрібно зробити заміну і задаємо нові параметри.

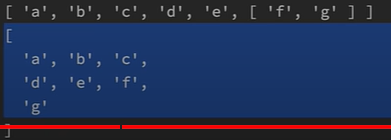
1. arr**.slice(**start-index, to-index**) – Повертає масив скопійованих елементів поточного масиву.** Не міняє структури поточного масива.

**а.** Якщо задати лише start-index – скопіює решту масива.

**б.** Якщо задати від’ємні значення – start-index – задаватиметься з кінця

1. arr**.concat(**arr1 || param1, arr2 || param2,…**) - Повертає новий масив елементів інших масивів або додаткових елементи.**
2. arr**.flet(num) –** застосовується до масивів, що містять в собі багаторівневі масиви**. Поветрає новий масив без першого рівня масивної вложеності. num** – глубина вложеності, яку потрібно розпарсити

****

****

1. **[…newArr]** = arr – копіювання масиву із деструктуризацією

**Методи перебору масива:**

1. arr**.forEach((**item, index, arr**)** => {}); - метод перебору з встроєною callback функцією для взаємодії з масивом. **Не міняє поточний масив.**
2. **for(**letvalue **of** arr**){} –** перебір елементів масива

**Array methods**

**Методи пошуку в масиві:**

**Пошук позиції шуканого елемента:**

1. arr**.indexOf(**item, from**)** – шукаємо **item**, починаючи з позиції **from.** Найкраще працює із строчними і числовими масивами. **Повертає індекс знайденого елемента або -1.**
2. arr**.lastIndexOf(**item, from**) –** шукаємо **item**, починаючи з позиції **from**, але з кінця. **Повертає індекс знайденого елемента або -1.**
3. arr.**findIndex((**item, index, arr**)** => { вираз-умова }); - перевірка наявності елемента із вказаною характеристикою. Найкраще підходить для роботи із об’єктними масивами. **Повертає індекс знайденого елемента або -1.**

**Визначення наявності шуканого елемента:**

1. arr**.includes(**item, from**)** - шукаємо **item**, починаючи з позиції **from. Повертає true / false. Може підтвердити наявність NaN**, як елемента масиву, навідмінно відінших перевірок

**Пошук множини необхідних елементів:**

1. arr**.filter((**item, index, arr**)** => {вираз-умова}); - **Повертає масив з усіх підходящих умові ітеруємих елементів.** Оригінальний масив не міняється
2. arr.**find((**item, index, arr**)** => { вираз-умова }); - **Повертає ітеруємий об’єкт, який пройшов провірку.** Знайшовши елемент – перевірка зупиняється.

**Array methods**

**Методи перетворення масива:**

1. arr**.map((**item, index, arr**)** => {вираз-умова}); - **Повертає масив результатів виклику callback** методу із виразом-умовою для **кожного елемента масиву.** Старий масив залишається незмінним. Може приймати як іменовані функції, так і анонімні
2. arr**.flatMap**((item, index, arr) => {вираз умова}) – діє за принципом map, але при утворені вложеності в масив масивів, розпаршує їх, як **flat()**
3. arr**.sort(**callback(a, b) || a, b**)** – **Повертає відсортований поточний масив.** Масив сортується за вказаною в callback умовою. Масив сортується по різниці між параметром a – поточним елементо і b – насупним елементом (а-b – від більшогодо меньшого, b-a – від меньшого до більшого).
4. arr**.reverce() – повертає масив перевертиш**

**Методи перетворення масив – строка:**

1. arr = str**.split(**str-divider, count**)** – розділяє строку на масив, по вказаній str-divider (‘, ’). **Повертає масив із елементами строки розділеними str-divider.** Пуста строка str-divider дасть нам масив букв слова
2. str = arr.**join(**str**) –** об’єднує масив в строку, склеюючи її строкою переданою, як параметр

**Методи перетворення масиву в складнішу структуру:**

1. arr**.reduce((**previousValue, item, index, array**) => {},** accum**); -** метод акумулятор для створення нових структур на основі масиву

previousValue – містить в собі результат виклику callback, на старті містить в собі значення accum

**Array methods**

**Метод перевірки масиву на тип даних arr:**

1. **Array.isArray(**data**)** – перевіряє чи є дані передані в метод масивом

**Додаткова інформація по методах масивів:**

1. <https://learn.javascript.ru/array-methods>
2. <https://www.freecodecamp.org/news/lets-clear-up-the-confusion-around-the-slice-splice-split-methods-in-javascript-8ba3266c29ae/>
3. <https://medium.com/@stasonmars/%D0%BE%D0%B1%D1%8A%D1%8F%D1%81%D0%BD%D1%8F%D0%B5%D0%BC-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B-%D0%B2-%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D0%B0%D1%85-js-filter-vs-map-vs-reduce-vs-foreach-995ef7468c4a>
4. <https://learn.javascript.ru/array-iteration>
5. <https://docs.google.com/presentation/d/1iRUQLMYVqmkI74mnsrjbDniS4VbsUplF6fipaUdV3Dc/edit#slide=id.g5db36dd8f1_2_58>