



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍏 🍊 .splice(1, 1, 🍋) ➡ ➡ ➡ 🍎 🍋 🍊

Метод splice змінює вміст масиву, видаляючи існуючі елементи та/або додаючи нові. У цьому прикладі з масиву видаляється один елемент, починаючи з індексу 1 (🍏), і на його місце вставляється 🍋.

Результат: 🍎 🍋 🍊



## Основні методи масивів

[www.t.me/shvets\\_js](https://www.t.me/shvets_js)

🍎 🍏 🍊 .slice(1, 2) ➡ ➡ ➡ 🍎

Метод slice повертає новий масив, що містить елементи масиву, починаючи з індексу 1 до індексу 2 (не включаючи індекс 2). У цьому прикладі це елемент 🍎 .

Результат: 🍎



## Основні методи масивів

[www.t.me/shvets\\_js](http://www.t.me/shvets_js)

🍎 🍐 .concat([🍊🍐]) ➡ ➡ ➡ 🍎 🍐 🍊 🍐

Метод concat об'єднує два або більше масивів. У цьому прикладі об'єднується масив 🍎 🍐 з масивом 🍊 🍐.

Результат: 🍎 🍐 🍊 🍐



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍎 .forEach(🍎 => 🍐) ➡ ➡ ➡ 🍐 🍐

Метод `forEach` виконує вказану функцію один раз для кожного елемента масиву. У цьому прикладі кожен елемент `🍎` змінюється на `🍐`.

Результат: 🍐 🍐



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍎 .map(🍎 => 🍐) ➡ ➡ ➡ 🍐 🍐

Метод `map` створює новий масив з результатами виклику вказаної функції для кожного елемента масиву. У цьому прикладі кожен елемент 🍎 перетворюється на 🍐.

Результат: 🍐 🍐



## Основні методи масивів

[www.t.me/shvets\\_js](https://www.t.me/shvets_js)

🍎 🍏 🍊 .filter(🍎 🍏) ➡ ➡ ➡ 🍏 🍏

Метод filter створює новий масив з усіма елементами, що пройшли перевірку, яка задається переданою функцією. У цьому прикладі залишаються тільки 🍏 та 🍏.

Результат: 🍏 🍏



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод sort сортує елементи масиву на місці та повертає відсортований масив. У цьому прикладі масив сортується за алфавітом.

Результат:



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `reverse` змінює порядок елементів у масиві на зворотний. У цьому прикладі 🍎, 🍊 та 🍐 міняються місцями.

Результат: 🍎 🍊 🍐



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

apple orange apple .join(' ') → → → 'apple orange apple'

Метод `join` об'єднує всі елементи масиву в рядок, використовуючи вказаний розділювач (у даному випадку пробіл).

Результат: 'apple orange apple'



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍏 🍊 🍎 .split(' ') ➡ ➡ ➡ ['🍏','🍊','🍎']

Метод `split` розділяє рядок на масив підрядків за допомогою вказаного розділювача. Тут масив емодзі розділяється по пробілу.

Результат: ['🍏','🍊','🍎']



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `push` додає один або кілька елементів в кінець масиву і повертає нову довжину масиву. У цьому прикладі додається в кінець масиву.

Результат:



## Основні методи масивів

[www.t.me/shvets\\_js](https://www.t.me/shvets_js)

🍎 🍏 🍊 .pop() ➡ ➡ ➡ 🍎 🍏 + 🍊

Метод pop видаляє останній елемент масиву та повертає його. У цьому прикладі видаляється 🍊 .

Результат: 🍎 🍏 +



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `unshift` додає один або кілька елементів на початок масиву і повертає нову довжину масиву. У цьому прикладі додається на початок масиву.

Результат:



## Основні методи масивів

[www.t.me/shvets\\_js](https://www.t.me/shvets_js)

🍎 🍐 🍊 .shift() ➡ ➡ ➡ 🍐 🍊 + 🍏

Метод shift видаляє перший елемент масиву та повертає його. У цьому прикладі видаляється 🍏 .

Результат: 🍐 🍊 + 🍏



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



.includes( ) ➔ ➔ ➔ true

Метод `includes` перевіряє, чи містить масив вказаний елемент, і повертає `true` або `false`. У цьому прикладі перевіряється наявність в масиві.

Результат: true



## Основні методи масивів

[www.t.me/shvets\\_js](http://www.t.me/shvets_js)

🍎 🍏 🍊 .indexOf(🍏) ➡ ➡ ➡ 1

Метод `indexOf` повертає перший індекс, за яким  
елемент може бути знайдений в масиві, або -1, якщо  
елемент не знайдений. У цьому прикладі знайдений  
індекс для 🍏 .

Результат: 1



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `find` повертає перший елемент в масиві, який відповідає вказаній функції перевірки. У цьому прикладі знаходиться 🍎.

Результат: 🍎



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍏 🍊 .findIndex(🍏) ➔ ➔ ➔ 1

Метод `findIndex` повертає індекс першого елемента, який відповідає вказаній функції перевірки. У цьому прикладі повертається індекс для 🍏.

Результат: 1



JS

## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍏 🍊 .reduce((🍎,🍏) =>🍎 +🍏) ➔ ➔ ➔ "🍎🍏🍊"

Метод reduce виконує функцію редуктора, що включає всі елементи масиву в одне значення. У цьому прикладі всі елементи об'єднуються в один рядок.

Результат: "🍎🍏🍊"



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍏 🍊 .every(🍎 =>🍎 !== '🍊') ➔ ➔ ➔ false

Метод every перевіряє, чи відповідають всі елементи масиву вказаній функції перевірки. У цьому прикладі перевіряється, чи всі елементи не є 🍊 .

Результат: false



## Основні методи масивів

[www.t.me/shvets\\_js](http://www.t.me/shvets_js)

🍎 🍏 🍊 .some(🍏 =>🍏 === '🍏') ➔ ➔ ➔ true

Метод some перевіряє, чи відповідає хоча б один елемент масиву вказаній функції перевірки. У цьому прикладі перевіряється, чи є в масиві🍏.

Результат: true



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `fill` заповнює всі елементи масиву одним статичним значенням, починаючи з індексу 0 і закінчуючи останнім елементом. У цьому прикладі всі елементи заповнюються 🍐 .

Результат: 🍐 🍐 🍐



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍎 🍏 🍊 .copyWithin(1, 0) ➡ ➡ ➡ 🍎 🍎 🍏

Метод copyWithin копіює частину масиву в інше місце в цьому ж масиві без зміни його розміру. У цьому прикладі копіюється елемент з індексу 0 на індекс 1.

Результат: 🍎 🍎 🍏



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод flat зменшує вкладеність масиву, створюючи новий масив з усіх вкладених елементів. У цьому прикладі масив залишається таким же, оскільки елементи не вкладені.

Результат:



**JS**

## Основні методи масивів

[www.t.me/shvets\\_js](http://www.t.me/shvets_js)

apple orange apple .flatMap(apple => [apple, apple]) → → → apple orange apple

apple apple

Метод flatMap застосовує вказану функцію до кожного елемента масиву і створює новий масив, який потім розгладжує один рівень вкладеності. У цьому прикладі кожен елемент масиву перетворюється в подвійний елемент з додаванням apple.

Результат: apple orange apple apple



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍏 🍊 🍎 .entries() ➡ ➡ ➡ [0, 🍏 ], [1, 🍊 ], [2, 🍎 ]

Метод entries повертає новий об'єкт Array Iterator, який містить пари ключ/значення для кожного індексу в масиві. У цьому прикладі кожен елемент масиву перетворюється в пару індекс/значення.

Результат: [0, 🍏 ], [1, 🍊 ], [2, 🍎 ]



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



.keys() ➔ ➔ ➔ 0, 1, 2

Метод keys повертає новий об'єкт Array Iterator, який містить ключі для кожного індексу в масиві. У цьому прикладі повертаються всі індекси масиву.

Результат: 0, 1, 2



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `values` повертає новий об'єкт `Array Iterator`, який містить значення для кожного індексу в масиві. У цьому прикладі повертаються всі значення масиву.

Результат:  ,  , 



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍏 🍊 🍎 .findLast(🍏 => 🍏 === '🍏') ➔ ➔ ➔ 🍏

Метод `findLast` повертає останній елемент в масиві, який відповідає вказаній функції перевірки. У цьому прикладі знаходиться останнє значення 🍏.

Результат: 🍏



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)

🍏 🍊 🍎 .findLastIndex(🍊 => 🍊 === '🍊 ') ➔ ➔ ➔ 1

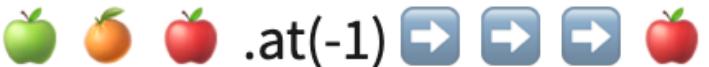
Метод `findLastIndex` повертає індекс останнього елемента, який відповідає вказаній функції перевірки. У цьому прикладі повертається індекс останнього значення 🍊 .

Результат: 1



## Основні методи масивів

[www.t.me/shvets\\_js](https://t.me/shvets_js)



Метод `at` повертає елемент за заданим індексом. В даному випадку повертається останній елемент, оскільки використовується індекс `-1`.

Результат: