



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍏 🍊 .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



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .find()    

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

Результат: 





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍎 🍏 🍌 .findIndex(🍏) ➡ ➡ ➡ 1

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

Результат: 1



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍏 🍊 .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: "🍏 🍏 🍊"





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

```
🍏 🍏 🍏 .every(🍏 => 🍏 !== '🍏') ➡ ➡ ➡ false
```

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


Результат: false



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .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






## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .fill( )      

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

Результат:   



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍏 🍏 .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: 🍏 🍏 🍏





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .flat()      














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


Результат:   



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .flatMap( => [,  ])         
 

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  .

Результат:      





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

   .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,  ]



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍊 🍏 .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





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍊 🍏 .values() ➡ ➡ ➡ 🍏 , 🍊 , 🍏

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

Результат: 🍏 , 🍊 , 🍏



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

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

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

Результат: 🍏





## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

```
🍏 🍊 🍎 .findLastIndex(🍊 => 🍊 === '🍊 ') ➡ ➡ ➡ 1
```

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

Результат: 1



## Методи масивів Javascript

[www.linkedin.com/in/shvetsm123](https://www.linkedin.com/in/shvetsm123)

🍏 🍊 🍎 .at(-1) ➡ ➡ ➡ 🍎

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

Результат: 🍎