**Конвертиране на Масивите към Strings**

В JavaScript, всички обекти имат **valueOf ()** и **ToString ()** методи. **valueOf ()** метода е поведението по подразбиране за масив. Той връща масива като стринг:

Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*document.getElementById("demo").innerHTML = fruits.valueOf();*

Край на формуляра

За JavaScript масивите, **valueOf ()** и **ToString ()** са равни.Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*document.getElementById("demo").innerHTML = fruits.toString();*

**Край на формуляра**

**join()** метода също присъединява всички елементи на масива в [стринг](https://js.w3schools.bg/%D1%81%D1%82%D1%80%D0%B8%D0%BD%D0%B3%D0%BE%D0%B2%D0%B5/" \o "Урок за Стрингове). Той се държи точно като ToString (), но с него може да определим сепаратора (разделителя):

*var fruits = ["Banana", "Orange","Apple", "Mango"];*

*document.getElementById("demo").innerHTML = fruits.join(" \* ");*

**Край на формуляра**

**Pop и Push JavaScript методи**

Когато работите с [масиви в JavaScript](https://js.w3schools.bg/%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D0%B8/" \o "Урок за Масивите в езикът JavaScript - видове и типове) е много лесно да добавяте и премахвате елементи. Това се прави чрез **pop** и **push**. **Pop** премахва елемент, а ***push*** добавя такъв**. *pop()*** метода премахва последния елемент от масива.Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.pop(); // Removes the last element ("Mango") from fruits*

**Край на формуляра**

**push()** метода добавя нов елемент към масива (в края):

Забележка: [0] е първият елемент в масив. [1] е втория. Array индексите започват от 0.Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.push("Kiwi"); // Adds a new element ("Kiwi") to fruits*

Край на формуляра

Метода **pop()** връща извадения масив. Метода push() връща дължината на новия масив.

**Shifting Елементи**

**Shift()** е еквивалентно на **pop(),** но хваща първия вместо последния елемент. **shift()** метода премахва първия елемент на масива, и „сменя“ индекса на всички други елементи с едно място надолу.Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.shift(); // Removes the first element "Banana" from fruits*

**Край на формуляра**

**unshift()** метода добавя нов елемент към масива (в началото), и „unshift“-ва по-старите елементи:Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.unshift("Lemon"); // Adds a new element "Lemon" to fruits*Край на формуляра

Методът **shift()** връща стринга, който е „изместен“. Методът **unshift()** връща дължината на новия масив.

**Промяна на елементите**

Array елементите са достъпни чрез техния индекс :Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits[0] = "Kiwi"; // Changes the first element of fruits to "Kiwi"*

Край на формуляра

Пропъртито **length** предоставя лесен начин да добавите нов елемент към масив:Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits[fruits.length] = "Kiwi"; // Appends "Kiwi" to fruit*

Край на формуляра

**Изтриване на елементи**

Тъй като [JavaScript масивитe](https://js.w3schools.bg/%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D0%B8/) са обекти, елементитe могат да бъдат изтрити с помощта на оператора delete :Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*delete fruits[0]; // Changes the first element in fruits to undefined*

Край на формуляра

Използването на **delete** оставя недефинирани дупки в масива. Използвайте **pop()** или **splice()**, за да избегнете нежелани ефекти.

**Splice на Array**

**Splice()** методът може да се използва за добавяне на нови елементи към масив.Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.splice(2, 0, "Lemon", "Kiwi");*

Край на формуляра

Първият параметър (2) определя позицията, на която трябва да бъдат добавени новите елементи. Вторият параметър (0) определя колко елементи трябва да бъдат отстранени . Останалите параметри(„Lemon“, „Kiwi“) определят новите елементи, които ще бъдат добавени .

Използването на **splice(),** за да премахнете елементи.

Ако настроите параметрите както трябва, може да премахвате елементи без да оставяте дупки в масива:Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.splice(0,1); // Removes the first element of fruits*

Край на формуляра

Първият параметър (0) определя позицията, в която трябва да бъдат добавени новите елементи. Вторият параметър (1) определя колко елементи трябва да бъдат отстранени . Останалите параметри са пропуснати. Няма да се добавят нови елементи.

**Сортирането на Масиви**

sort() метода сортира масив по азбучен ред:Начало на формуляра

*var fruits = ["Banana", "Orange", "Apple", "Mango"];*

*fruits.sort(); // Sorts the elements of fruits*

Край на формуляра

Метода **sort()** взема функция като параметър.

**Обръщане на Масив**

reverse() метода обръща реда на елементите в масив. Можете да го използвате, за да сортирате масив в низходящ ред:

Начало на формуляра

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.sort(); // Sorts the elements of fruits

fruits.reverse(); // Reverses the order of the elements

**Край на формуляра**

**Сортиране на числа**

Метода **sort()** не може да се използва за сортиране на числа, тъй като сортира по азбучен ред. Един от начините да ги сортирате е чрез [JS функция](https://js.w3schools.bg/%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8/):

Начало на формуляра

*var points = [40, 100, 1, 5, 25, 10];*

*points.sort(function(a, b){return a-b});*

Край на формуляра

Използвайте същия трик, за да сортирате масив в низходящ ред:*Начало на формуляра*

*var points = [40, 100, 1, 5, 25, 10];*

*points.sort(function(a, b){return b-a});*

Край на формуляра

**Как да намерим най-високата (или най-ниска) стойност?**

Как да намерим най-високата стойност в масив?

**Пример за най-високата**

Начало на формуляра

*var points = [40, 100, 1, 5, 25, 10];*

*points.sort(function(a, b){return b-a});*

*// now points[0] contains the highest value*

Край на формуляра

**Пример за най-ниската**

Начало на формуляра

*var points = [40, 100, 1, 5, 25, 10];*

*points.sort(function(a, b){return a-b});*

*// now points[0] contains the lowest value*

**Край на формуляра**

**Присъединяване на Масиви**

**Concat()** методът създава нов масив от „присъединяването“ на два масива:Начало на формуляра

*var myGirls = ["Cecilie", "Lone"];*

*var myBoys = ["Emil", "Tobias","Linus"];*

*var myChildren = myGirls.concat(myBoys); // Concatenates (joins) myGirls and myBoys*

Край на формуляра

Методът **Concat()** може да вземе неограничен брой масиви:Начало на формуляра

*var arr1 = ["Cecilie", "Lone"];*

*var arr2 = ["Emil", "Tobias","Linus"];*

*var arr3 = ["Robin", "Morgan"];*

*var myChildren = arr1.concat(arr2, arr3); // Concatenates arr1 with arr2 and arr3*

Край на формуляра

**Нарязване на Масив**

**slice()** метода изрязва парче от масива:Начало на формуляра

*var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];*

*var citrus = fruits.slice(1,3);Край на формуляра*