

Тема 1. Основи JavaScript

Тема: масиви

*Морозов Андрій Васильович,
к.т.н, доц.,
декан факультету інформаційно-комп'ютерних
технологій ЖДТУ*

Методи для роботи з масивами

- Запис елементів масиву у рядок

```
var arr = ['Маша', 'Петя', 'Марина', 'Василий'];  
var str = arr.join(';');  
console.log(str); // Маша;Петя;Марина;Василий
```

- Видалення елементів масиву

```
var arr = ["Я", "вивчаю", "JavaScript"];  
delete arr[1]; // значення з індексом 1 видалене  
// тепер arr = ["Я", undefined, "JavaScript"];  
console.log(arr[1]); // undefined
```

- Копіювання частини масиву

```
var arr = ["Чому", "потрібно", "вчити", "JavaScript"];  
var arr2 = arr.slice(1, 3);  
    // елементи 1, 2 (не включаючи 3)  
console.log(arr2); // потрібно,вчити
```

```
var arr = ["Чому", "потрібно", "вчити", "JavaScript"];  
console.log( arr.slice(1) );  
    // взяти усі елементи, починаючи з номера 1  
    // ["потрібно", "вчити", "JavaScript"]
```

```
var arr2 = arr.slice(-2);  
    // копіювати від 2-го елемента з кінця і далі вправо  
    // arr2 = ["вчити", "JavaScript"]
```

```
var fullCopy = arr.slice();  
    // скопіювати весь масив
```

- Інвертування масиву

```
var arr = [1, 2, 3];  
arr.reverse();  
console.log(arr); // 3,2,1
```

- Сортування елементів масиву

```
// по замовчуванню елементи масиву порівнюються як рядки
var arr = [1, 2, 15];
arr.sort();
console.log(arr); // 1,15,2
```

Для порівняння елементів як чисел, потрібно оголосити власну функцію-компаратор:

```
function compareNumeric(a, b) {
  if (a > b) return 1;
  if (a < b) return -1;
}
var arr = [ 1, 2, 15 ];
arr.sort(compareNumeric);
console.log(arr); // 1,2,15
```

- Об'єднання масивів

Синтаксис використання:

```
var newArr = arr.concat(value1, value2, ... valueN)
```

```
var arr = [1, 2];  
var newArr = arr.concat(3, 4);  
console.log(newArr); // 1,2,3,4
```

```
var arr = [1, 2];  
var newArr = arr.concat([3, 4], 5);  
    // те саме, що і arr.concat(3,4,5)  
console.log(newArr); // 1,2,3,4,5
```

- Видалення і додавання елементів масиву одночасно

Синтаксис використання:

```
var newArr = arr.splice(index[, deleteCount, elem1, ..., elemN])
```

```
// видалити deleteCount елементів, починаючи з індексу index  
// вставити elem1, ..., elemN, починаючи з індексу index  
// повернути масив з видалених елементів
```

```
var arr = ["Я", "вивчаю", "JavaScript"];  
arr.splice(1, 1); // починаючи з індексу 1, видалити 1 елемент  
console.log(arr); // залишилось ["Я", "JavaScript"]
```

```
var arr = ["Я", "зараз", "вивчаю", "JavaScript"];  
arr.splice(0, 3, "Ми", "вивчаємо")  
// видалити 3 перших елементи і додати інші замість них  
console.log( arr ) // теперь ["Ми", "вивчаємо", "JavaScript"]
```

```
var arr = ["Я", "зараз", "вивчаю", "JavaScript"];  
    // видалити 2 перших елементи  
var removed = arr.splice(0, 2);  
console.log( removed );  
    // "Я","зараз" <-- масив видалених елементів
```

```
var arr = ["Я", "вивчаю", "JavaScript"];  
    // з позиції 2  
    // видалити 0 елементів і  
    // додати елементи "складну","мову"  
arr.splice(2, 0, "складну", "мову");  
console.log( arr );  
    // "Я","вивчаю","складну","мову","JavaScript"
```

Об'єкти як асоціативні масиви у JavaScript

- **Асоціативний масив** – структура даних, яка може зберігати будь-які дані у формі «ключ – значення».
- Структури даних «асоціативний масив» у JavaScript немає.
- Можливості зберігання пар «ключ – значення» надають об'єкти:

```
1. o = new Object();
```

```
2. o = { };
```

До об'єкта можна додавати властивості:

```
var person = { };
```

```
person.name = "Sergey";
```

```
person.age = 25;
```

І надалі використовувати як звичайні змінні:

```
var str = person.name + ": " + person.age;
```

```
// Sergey: 25
```


Для перевірки, чи є властивість в об'єкті використовується операція `in`:

```
if ("name" in person) {  
    alert( "Властивість name існує!" );  
}
```

Якщо звернутися до неіснуючої властивості, отримаємо значення `undefined`:

```
console.log(person.abcdef); // undefined
```

До властивостей можна звертатися за допомогою квадратних дужок:

```
console.log(person["name"]);
```

Цю форму використовують у випадках: 1) коли назва властивості містить пробіли або службові символи:

```
person["місце народження"] = "Житомир";  
console.log(person["місце народження"]);
```

2) якщо потрібно отримати значення за ім'ям властивості, яке зберігається у змінній.

```
var key = "age";  
console.log(person[key]); // 25
```

При створенні об'єкту, можна одразу задавати його властивості:

```
var obj = { ключ1: знач1, ключ2: знач2, ... };
```

Приклад:

```
var menuSetup = {  
  width: 300,  
  height: 200,  
  title: "Menu"  
};
```

```
var menuSetup = {};  
menuSetup.width = 300;  
menuSetup.height = 200;  
menuSetup.title = 'Menu';
```

Для проходів по властивостям об'єкта використовують цикл **for.in**:

```
for (var key in menuSetup)
{
    console.log("Ключ: " + key + ", значення: " +
        menuSetup[key]);
}
```

Результат:

Ключ: width, значення: 300

Ключ: height, значення: 200

Ключ: title, значення: Menu