

# Використання колбеків на практиці

Веб-програмування на стороні сервера 5.1.2024

# Робота з масивами

- Вбудований тип масиву (Array) має багато допоміжних методів, які приймають колбек для визначення результату чи умови операції
- Колбеки викликаються синхронно (тобто, послідовність операцій у програмі зберігається, наступна операція не буде виконана допоки метод не буде виконано повністю)
- Колбеки викликаються для кожного з елементів масиву

## Робота з масивами: `Array.prototype.forEach()`

```
1  const array1 = ['a', 'b', 'c'];
2
3  array1.forEach((element) => console.log(element));
4
5  // Expected output: "a"
6  // Expected output: "b"
7  // Expected output: "c"
```

## Робота з масивами: Array.prototype.find()

```
1  const array1 = [5, 12, 8, 130, 44];  
2  
3  const found = array1.find((element) => element > 10);  
4  
5  console.log(found);  
6  // Expected output: 12
```

# Робота з масивами: `Array.prototype.filter()`

```
1  const words = ['spray', 'elite', 'exuberant', 'destruction', 'present'];
2
3  const result = words.filter((word) => word.length > 6);
4
5  console.log(result);
6  // Expected output: Array ["exuberant", "destruction", "present"]
```

## Робота з масивами: `Array.prototype.map()`

```
1  const array1 = [1, 4, 9, 16];  
2  
3  // Pass a function to map  
4  const map1 = array1.map((x) => x * 2);  
5  
6  console.log(map1);  
7  // Expected output: Array [2, 8, 18, 32]
```

# Модуль fs з Node.js

Дає змогу:

- Читати\писати файли, отримувати інформацію про них
- Копіювати та видаляти файли
- Маніпулювати директоріями, символьними посиланнями, правами доступу до файлів

Підтримувані стилі роботи:

- Синхронний (минула лекція)
- **Асинхронний (колбеки)**
- Асинхронний (проміси) (див. наступну лекцію)

<https://nodejs.org/api/fs.html>

# Модуль fs з Node.js: Колбеки

- Приймає колбек, який виконує обробку за результатом завершення операції
- **Колбек виконується асинхронно!** Тобто програма продовжує виконання негайно, операція виконується паралельно з виконанням програми, а колбек буде викликаний як тільки операція буде завершена (тобто пізніше)
- Зазвичай кількість параметрів колбеку залежить від особливості операції, але:
  - Перший параметр резервують для збереження помилки
  - Якщо операція успішна то перший параметр null або undefined



# Модуль fs з Node.js: Колбеки

```
1  const fs = require("node:fs");
2
3  fs.writeFile('test.txt', 'Hello!', 'utf8', (err) => {
4      if (err) {
5          console.log(err);
6          return;
7      }
8      console.log('The file has been saved!');
9  });
```

# Модуль fs з Node.js: Колбеки

```
1  const fs = require("node:fs");
2
3  fs.readFile('test.txt', 'utf8', (err, data) => {
4      if (err) {
5          console.log(err);
6          return;
7      }
8      console.log(data);
9  });
```