

# Класи. Успадкування

**№ уроку:** 9 **Курс:** JavaScript Базовий

**Засоби навчання:** Visual Studio Code  
Web Browser

## Огляд, мета та призначення уроку

Навчитися використовувати успадкування через класи за допомогою ключового слова `extends`. Розуміти принцип прототипного успадкування в JavaScript. Навчитися використовувати ключове слово `static`.

## Вивчивши матеріал цього заняття, учень зможе:

- Розуміти принципи успадкування в JavaScript.
- Використовувати ключове слово `extends` для успадкування класів.
- Перевизначати методи, які отримані від батьківського класу.
- Використовувати ключове слово `super`.
- Використовувати ключове слово `static`, розуміти призначення статичних методів класів.

## Зміст уроку

1. Що таке успадкування.
2. Ключове слово `extends`.
3. Конструктори та ключове слово `super`.
4. Перевизначення методів.
5. Статичні методи та властивості.

## Резюме

- **Успадкування** – концепція об'єктно-орієнтованих мов програмування, згідно з якою один тип даних може успадковувати функціональність іншого типу даних.

У JavaScript немає можливості створювати користувацькі типи даних, тому успадкування ґрунтується на об'єктах, а не на типах. Один об'єкт успадковує функціональність іншого об'єкта. Водночас успадкування відбувається через встановлення зв'язків між прототипами. Таке успадкування називається **прототипно-орієнтованою моделлю успадкування**.

- Передусім успадкування необхідне для того, щоб позбутися дублювання коду та спростити повторне використання коду. Також, використовуючи успадкування на основі класів, можна зробити застосунок структурованішим.
- Для мов із суворою типізацією (Java, C#, C++ та інші) успадкування дає багато інших додаткових переваг і не обмежується повторним використанням коду.
- **Основна мета успадкування – забезпечити повторне використання коду.** Є багато

різних шаблонів написання коду, які дають змогу досягти такого ж результату, не використовуючи успадкування. Варіанти успадкування за допомогою прототипів та інші прийоми успадкування коду можна побачити в уроці «Шаблони створення об'єктів та повторного використання коду» <https://itvdn.com/ru/video/javascript-patterns/create-patterns>.

- Під час використання ключових слів **class** і **extends** можна досягти успадкування на основі прототипів, використовуючи простий і зрозумілий синтаксис.
- **class A extends B {}** – це визначення вказує, що клас A успадковується від класу B. Це означає, що методи та властивості, які були визначені в класі B, переходять «у спадок» класу A.
- Успадкування через ключове слово **extends** ґрунтується на прототипах за **class A extends B {}** – для прототипу функції конструктора буде встановлений прототип конструктора B. A.prototype.proto = B.prototype.

Така поведінка підтримує збереження в прототипі класу A методів, які належать до класу A, але також дає доступ до методів, що визначені у класі B.

**proto** – це «системна» властивість, яка є в кожному екземплярі та в якій зберігається посилання на прототип, що пов'язаний із цим екземпляром.

- Якщо у похідному класі визначено конструктор, то цей конструктор завжди має викликати конструктор базового класу до того, як вперше буде зроблено звернення до **this**. Виклик базового конструктора відбувається через ключове слово **super**.
- Якщо в похідному класі немає конструктора, то конструктор, який викликає базовий конструктор, створюється автоматично.
- **Перевизначення методу** – створення методу копії методу з базового класу у похідному класі. Новий метод у похідному класі може повністю замінити функціональність методу базового класу або розширити її за допомогою виклику `super.ім'я_методу` (параметри).
- **static** – ключове слово визначення статичних властивостей чи методів. Статична властивість або метод є в одному екземплярі для всього застосунку.

`class My { static prop = 1 }` – у цьому прикладі створюється статична властивість `prop` зі значенням 1, яка буде загальною для всього застосунку. Звертаючись до статичної властивості, у будь-якій частині коду буде здійснюватися звернення до однієї й тієї ж області пам'яті. Статична властивість або метод можна розглядати як глобальне, доступне та загальне для всіх.

Статична конструкція зберігається не в екземплярі чи прототипі, а на функції-конструкторі. Щоб до неї звернутися, необхідно виконати звернення на ім'я класу (функції-конструктора). Наприклад, `My.prop = 10;`

- Статичні методи часто є методами-утилітами, які можуть стати в пригоді в будь-якій частині коду для виконання будь-якої дії.

## Закріплення матеріалу

- Що таке успадкування?
- Як виконати успадкування класів у JavaScript?
- Як працює успадкування через ключове слово extends?
- Що робить ключове слово super?
- Що обов'язково має зробити конструктор похідного класу?
- Що таке статична властивість, як визначити та як до нього звернутися?

## Самостійна діяльність учня

Виконайте завдання у директорії Tasks\012 Classes. Inheritance. Текст завдань розташований у коментарях, у тегах script.

## Рекомендовані ресурси

Успадкування у JavaScript

<https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Inheritance>

Наслідування та ланцюжок прототипів

[https://developer.mozilla.org/ru/docs/Web/JavaScript/Inheritance and the prototype chain](https://developer.mozilla.org/ru/docs/Web/JavaScript/Inheritance_and_the_prototype_chain)

Практика побудови об'єктів

[https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object building practice](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_building_practice)