

# Классы. Наследование

**№ урока:** 12 **Курс:** JavaScript Базовый

**Средства обучения:** Visual Studio Code  
Web Browser

## Обзор, цель и назначение урока

Научиться использовать наследование через классы с использованием ключевого слова `extends`. Понимать принцип прототипного наследования в JavaScript. Научиться использовать ключевое слово `static`.

## Изучив материал данного занятия, учащийся сможет:

- Понимать принципы наследования в JavaScript.
- Использовать ключевое слово `extends` для наследования классов.
- Переопределять методы, полученные от родительского класса.
- Использовать ключевое слово `super`.
- Использовать ключевое `static`, понимать назначение статических методов классов.

## Содержание урока

1. Что такое наследование
2. Ключевое слово `extends`
3. Конструкторы и ключевое слово `super`
4. Переопределение методов
5. Статические методы и свойства

## Резюме

- **Наследование** – концепция объектно-ориентированных языков программирования, согласно которой, один тип данных может наследовать функциональность другого типа данных.  
В JavaScript нет возможности создавать пользовательские типы данных, поэтому наследование базируется на объектах, а не на типах. Один объект наследует функциональность другого объекта. При этом наследование происходит посредством установки связей между прототипами. Такое наследование называется прототипно-ориентированной моделью наследования.
- Наследование в первую очередь необходимо для того, чтобы избавиться от дублирования кода и упростить повторное использование кода. Также, используя наследования на основе классов, можно сделать приложение более структурированным.
- Для языков со строгой типизацией (Java, C#, C++ и другие) наследование дает много других дополнительных преимуществ и не ограничивается повторным использованием кода.

- Основная цель наследования – обеспечить повторное использование кода. Есть много разных шаблонов написания кода, которые позволяют добиться такого же результата, не используя наследование. Варианты наследования с помощью прототипов и другие приемы наследования кода, можно увидеть в уроке «Шаблоны создания объектов и повторного использования кода» <https://itvdn.com/ru/video/javascript-patterns/create-patterns>
- При использовании ключевых слов `class` и `extends` можно добиться наследования на основе прототипов, используя простой и понятный синтаксис.
- `class A extends B {}` - данное определение указывает, что класс A наследуется от класса B. Это означает что методы и свойства, которые были определены в классе B переходят “по наследству” классу A.
- Наследование через ключевое слово `extends` базируется на прототипах при `class A extends B {}` - для прототипа функции конструктора A будет установлен прототип конструктора B. `A.prototype.__proto__ = B.prototype`  
Такое поведение поддерживает сохранение в прототипе класса A методов, относящихся к классу A, но также дает доступ к методам, которые определены в классе B. `__proto__` это «системное» свойство, которое есть в каждом экземпляре и в котором хранится ссылка на прототип, связанный с этим экземпляром.
- Если в производном классе определен конструктор, то этот конструктор всегда должен вызывать конструктор базового класса, до того, как будет впервые произведено обращение к `this`. Вызов базового конструктора происходит через ключевое слово `super`.
- Если в производном классе нет конструктора – конструктор, вызывающий базовый конструктор, создается автоматически.
- Переопределение метода – создание метода копии метода из базового класса в производном классе. Новый метод в производном классе может полностью заменить функциональность метода из базового класса или расширить ее с помощью вызова `super.имя_метода(параметры)`.
- `static` – ключевое слово для определения статических свойств или методов. Статическое свойство или метод – существует в одном экземпляре для всего приложения.

`class My { static prop = 1 }` – в данном примере создается статическое свойство `prop` со значением 1, которое будет общим для всего приложения, в любой части кода, обращаясь к статическому свойству будет производиться обращение к одной и той же области памяти. Статическое свойство или метод можно рассматривать как глобальное, доступное и общее для всех.

Статическая конструкция хранится не в экземпляре или прототипе, а на функции-конструкторе, поэтому, чтобы к ней обратиться необходимо выполнить обращение на имени класса (функции-конструктора). Например, `My.prop = 10`;

- Статические методы, зачастую являются методами-утилитами, которые могут пригодиться в любой части кода для выполнения какого-либо действия.

## Закрепление материала

- Что такое наследование?
- Как выполнить наследование классов в JavaScript?
- Как работает наследование через ключевое слово extends?
- Что делает ключевое слово super?
- Что обязательно должен сделать конструктор производного класса?
- Что такое статическое свойство, как его определить и как к нему обратиться?

## Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\012 Classes. Inheritance. Текст задач находится в комментариях в тегах script

## Рекомендуемые ресурсы

Наследование в JavaScript

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

Наследование и цепочка прототипов

[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 построения объектов

[https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object\\_building\\_practice](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_building_practice)