



JavaScript Базовий

Класи. Успадкування

JavaScript Базовий

Introduction



Охріменко Дмитро
MCT

 _okhrimenko

 dmitriy.okhrimenko

 dokhrimenko



MCID: 9210561

JavaScript Базовий

Тема уроку

Класи. Успадкування

JavaScript Базовий

План уроку

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

JavaScript Базовий

Успадкування

Успадкування – концепція об'єктно-орієнтованих мов програмування, за якою один тип даних (клас) може успадковувати дані та функціональність іншого типу даних (класу).

Успадкування сприяє повторному використанню наявного коду.

Прототипне успадкування – механізм успадкування, що підтримується у JavaScript, який ґрунтується на побудові ланцюжка прототипів і спільному використанню функцій між об'єктами.

Ключові слова **class** та **extends** є синтаксичним цукром прототипно-орієнтованої моделі успадкування.



JavaScript Базовий

Повторне використання коду

Шаблони успадкування з використанням прототипу:

- Ланцюжок прототипів.
- Запозичення конструкторів.
- Запозичення конструктора та встановлення прототипу.
- Спільне використання прототипу.
- Тимчасовий конструктор.

Шаблони повторного використання коду:

- Успадкування через прототип.
- Успадкування копіюванням властивостей.
- Змішування.
- Запозичення методів (call, apply).



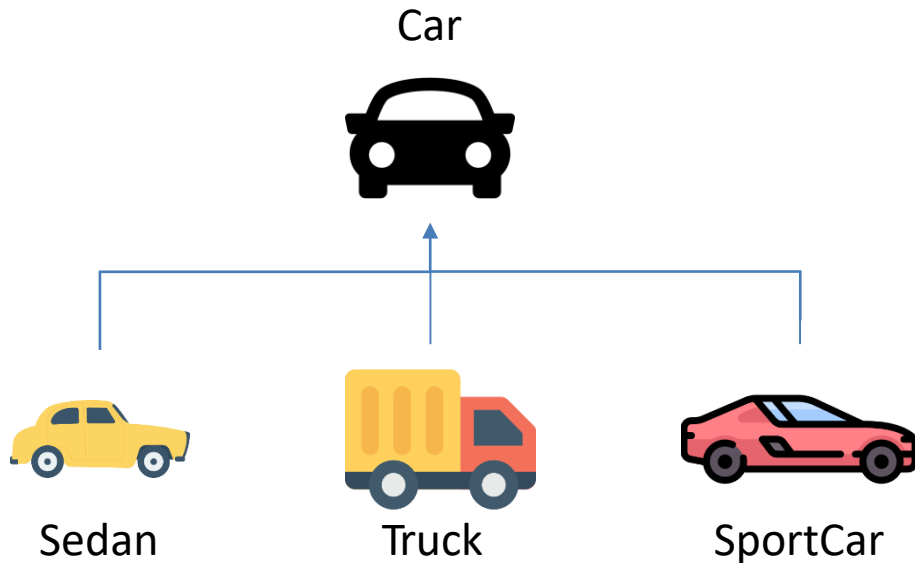
JavaScript Шаблини

Урок №4. Шаблини створення об'єктів і повторне використання коду

<https://itvdn.com/ru/video/javascript-patterns/create-patterns>

JavaScript Базовий

Успадкування класів



```
class Car {  
  |   move() {}  
}
```

```
class Sedan extends Car {  
  |   takePassengers() {}  
}
```

```
class Truck extends Car {  
  |   takeCargo() {}  
}
```

```
class SportCar extends Car {  
  |   goFast() {}  
}
```

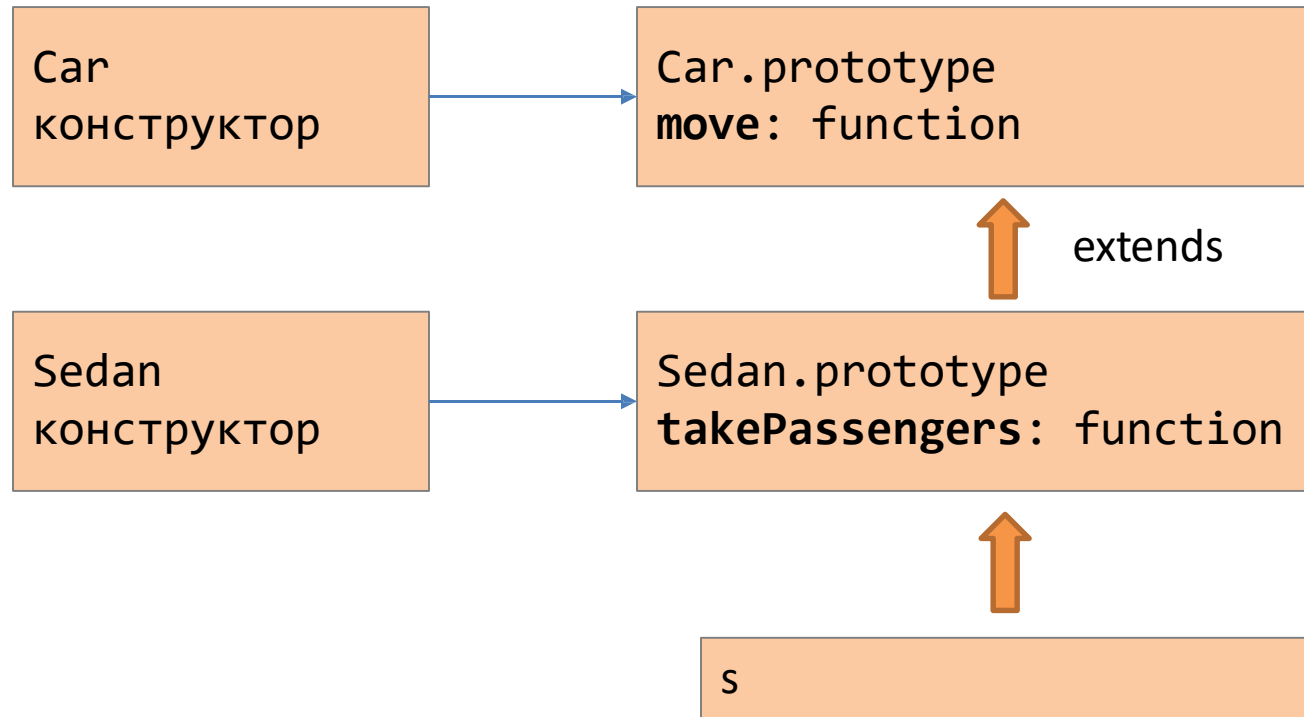
JavaScript Базовий

Успадкування класів

```
class Car {  
  move() {}  
}
```

```
class Sedan extends Car {  
  takePassengers() {}  
}
```

```
let s = new Sedan();  
s.takePassengers();  
s.move();
```



JavaScript Базовий

static

static – ключове слово визначення статичних полів і методів. Статичний член класу наявний в одному екземплярі для всього застосунку.

Статичні властивості або метод створюються у функції-конструкторі, а не в кожному екземплярі.

Для отримання доступу до статичних членів необхідно виконувати звернення через ім'я класу, а не через конкретний екземпляр.

```
class Sedan extends Car {  
    static totalCarsSold = 0;  
    takePassengers() {}  
}
```

```
Sedan.totalCarsSold ++;
```

Sedan
totalCarsSold = 0;

Sedan.prototype
takePassengers: function

JavaScript Базовий

Висновки

- JavaScript використовує **прототипно-орієнтовану** модель успадкування. Успадкування спрощує **повторне використання** коду.
- Якщо клас А успадковується від класу В, то прототип класу А використовує як свій прототип прототип класу В.
- Успадкування може ґрунтуватися безпосередньо через роботу з прототипом функцій конструкторів або через використання ключових слів **class** і **extends**.
- Ключове слово **super** дає змогу отримати доступ до членів базового класу (до прототипу конструктора класу, який встановлений як базовий).
- Якщо у похідному класі є конструктор, у цьому конструкторі треба обов'язково викликати конструктор батьківського класу, використовуючи **super (параметри)**.
- **Статична властивість або метод** – властивість чи метод функції конструктора.

JavaScript Базовий

Дякую за увагу! До нових зустрічей!



Охріменко Дмитро
МСТ



MCID: 9210561

Інформаційний відеосервіс для розробників програмного забезпечення

