

Static

Статические методы и свойства в JavaScript связаны с классами и не привязаны к экземплярам классов. Они существуют на самом классе и доступны непосредственно через имя класса, а не через созданные объекты этого класса.

Статические методы:

1. Определение статических методов:

Статический метод определяется внутри класса с использованием ключевого слова `static`.

```
class MyClass {  
  static myStaticMethod() {  
    // Логика статического метода  
  }  
}
```

2. Вызов статических методов:

Статический метод вызывается непосредственно через имя класса, без создания экземпляра класса.

```
MyClass.myStaticMethod();
```

3. Использование статических методов:

Статические методы часто используются для выполнения операций, которые не зависят от состояния экземпляра класса и не требуют доступа к его свойствам. Они могут использоваться как утилитарные методы, для выполнения вычислений, генерации данных и т. д.

Статические свойства:

1. Определение статических свойств:

В JavaScript статические свойства (классовые свойства) определены в классах с использованием предложения "static" и "get". Они могут быть только для чтения (т.е., они не могут быть изменены).

```
class MyClass {
  static get myStaticProperty() {
    return 'This is a static property';
  }
}
```

2. Доступ к статическим свойствам:

Статические свойства также вызываются непосредственно через имя класса, без создания экземпляра класса.

```
console.log(MyClass.myStaticProperty); // Выведет: "This is a static property"
```

Пример:

Вот пример класса с использованием статического метода и статического свойства:

```
class MathUtil {
  static square(x) {
    return x * x;
  }

  static get PI() {
    return 3.14159265359;
  }
}

console.log(MathUtil.square(4)); // Выведет: 16
console.log(MathUtil.PI); // Выведет: 3.14159265359
```

Статический метод `square` выполняет операцию возведения числа в квадрат, и его вызов не зависит от создания объектов класса `MathUtil`. Статическое свойство `PI` предоставляет значение числа `π` и также доступно без создания экземпляров класса.

Статические методы и свойства в классах JavaScript полезны для создания утилитарных функций и хранения констант, которые применимы ко всем экземплярам класса и не требуют доступа к их состоянию.