

Классы

№ урока: 11 **Курс:** JavaScript Базовый

Средства обучения: Visual Studio Code
Web Browser

Обзор, цель и назначение урока

Изучить синтаксис работы с классами. Научиться пользоваться классами, конструкторами, методами и открытыми и закрытыми полями для оформления объектно-ориентированного кода в современном стиле.

Изучив материал данного занятия, учащийся сможет:

- Работать с ключевым словом `class` для создания семейства объектов с одинаковой структурой.
- Понимать разницу между применением ключевого слова `class` и функций конструкторов.
- Использовать `constructor` в классе для инициализации экземпляра.
- Создавать свойства и методы в классе.
- Использовать геттеры и сеттеры для контроля над значениями, которые попадают в экземпляр класса.
- Работать с открытыми и закрытыми полями.

Содержание урока

1. Классы
2. Использование конструкторов и методов
3. Getter и setter
4. Закрытые и открытые поля классов
5. Транспайлеры

Резюме

- **Класс** – шаблон для создания объекта. Содержит данные и методы, которые должны эти данные обрабатывать.
Когда в сценарии повторно создаются объекты с одинаковой структурой (свойства и методы объектов совпадают), для того чтобы избавиться от дублирования кода и сделать работу с объектами более удобной, мы можем воспользоваться классами. Классы имеют ряд преимуществ по сравнению с фабричными функциями и функциями конструкторами. Классы можно рассматривать как аналог типов данных в других языках программирования, но в JavaScript, создавая класс, мы только определяем структуру объекта, которая должна получиться, когда класс будет использоваться для создания экземпляра. Даже если для создания экземпляра использовался специфический класс, тип данных экземпляра будет `object`.
Классы можно рассматривать как синтаксический сахар, который упрощает использование функций конструкторов. На самом деле, определяя класс и добавляя в

класс методы, мы определяем функцию конструктор и добавляем методы в прототип этой функции, делая это более коротким и удобным способом.

- Ключевое слово `class` введено в ECMAScript 2015 или ES6.
- Есть два способа создания класса, `class declaration` и `class expression`:

Class declaration:

```
class MyClass {  
    myMethod() {}  
}
```

Class Expression:

```
let myClass = class {  
    myMethod() {}  
};
```

Оба варианта дают возможность определить класс и на основе данного класса создавать экземпляры, но вариант `class expression` дает возможность создать класс и передать его в качестве параметра в вызываемый метод.

- Классы не поддерживают всплытие (hoisting). Использовать класс можно только после того, как тот будет объявлен в коде.
- Класс может содержать один специальный метод – `constructor`. Этот метод отвечает за инициализацию экземпляра класса.
- В классе может находиться любое количество методов. Методы, определенные в классе, будут добавлены в прототип функции конструктора.
- Класс может содержать свойства. Свойства обычно добавляются в конструкторе через инициализацию необходимого свойства на контексте (`this`) конструктора.
- Класс может содержать специальные методы `get` (getter) и `set` (setter), которые используются для того, чтобы связать свойство экземпляра класса с определенной функциональностью, которая запускается при чтении или записи в данное свойство.
- При работе с классами можно использовать экспериментальные возможности JavaScript, которые поддерживаются достаточно большим количеством современных браузеров – открытые и закрытые поля. Поля можно рассматривать как альтернативу свойств, которые добавляются через функцию конструктор, но при этом поля могут быть закрытыми, в то время как добавленное через конструктора свойство всегда доступно для использования как в классе, так и за его пределами.
- Открытое поле – поле класса, которое может использоваться в классе и за его пределами.

Пример определения открытого поля в классе

```
class MyClass {  
    publicField = 1;  
}
```

- Закрытое поле – поле класса, которое может использоваться только в классе. Если попытаться обратиться к закрытому полю за пределами класса, это приведет к ошибке на этапе выполнения.

Пример определения закрытого поля в классе (используется # как первый символ имени поля)

```
class MyClass {  
    #privateField = 1;  
}
```

- Транспайлер – тип компилятора, который выполняет транспилиацию программы. Транспилиация – преобразование программного кода, при которой исходный код на одном языке программирования преобразовывается в эквивалентный программный код на другом языке программирования.
- **Babel** – бесплатный транспайлер с открытым исходным кодом. Используется для конвертации ECMAScript 2015+ (ES6+) кода в обратно совместимый код, который может быть запущен в старых версиях браузеров.
- Polyfill – код, реализующий какую-либо функциональность, не поддерживаемую в некоторых версиях браузеров.
- Если в проекте используется функциональность, которая не поддерживается в браузерах, для которых должна присутствовать поддержка, обычно используется транспилиция, как часть процесса разработки приложения. Приложение создается с использованием современных функций языка, которые мало поддерживаются, после чего происходит транспилиция (часто этот процесс также называют компиляцией) и приложение преобразовывается в аналогичное по поведению приложение, но написанное с использованием более старой версии JavaScript, которая поддерживается целевым браузером. В итоге к HTML разметке подключается JavaScript код, который получился после компиляции (транспилиции).
- Если при разработке необходимо включить поддержку определенной функциональности, иногда используются polyfill файлы – специальные JavaScript сценарии, которые имитируют функциональность, недостающую браузеру. Такие файлы должны подключаться до того, как будет добавлен, какой-либо код, использующий функциональность добавляемую polyfill файлом.

Закрепление материала

- Что такое класс?
- Какие способы создания классов вы знаете?
- Как используются ключевые слова get и set в теле класса?
- В чем разница между закрытым и открытым полем? Как определить открытое и закрытое поле в классе?

- Что такое транспайлер, зачем он используется?
- Что такое polyfill файл, какие задачи решают подобные файлы?

Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\011 Classes. Текст задач находится в комментариях в тегах script.

Рекомендуемые ресурсы

Классы в JavaScript

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Classes>

Открытые и закрытые поля

<https://github.com/tc39/proposal-class-fields>

Практика построения объектов

https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_building_practice

Babel

<https://babeljs.io/>