Класи

№ уроку:	8	**Курс:**	JavaScript Базовий

Засоби навчання:	Visual Studio Code Web Browser

Огляд, мета та призначення уроку

Вивчити синтаксис роботи з класами. Навчитися користуватися класами, конструкторами, методами, відкритими та закритими полями для оформлення об'єктно-орієнтованого коду у сучасному стилі.

Вивчивши матеріал цього заняття, учень зможе:

- Працювати з ключовим словом class для створення сімейства об'єктів з однаковою структурою.
- Розуміти різницю між застосуванням ключового слова class і функцій конструкторів.
- Використовувати constructor у класі для ініціалізації екземпляру.
- Створювати властивості та методи у класі.
- Використовувати гетери та сетери для контролю над значеннями, які потрапляють до екземпляра класу.
- Працювати з відкритими та закритими полями.

Зміст уроку

1. Класи.
2. Використання конструкторів і методів.
3. Getter і setter.
4. Закриті та відкриті поля класів.
5. Транспайлери.

Резюме

- **Клас** – шаблон створення об'єкта. Містить дані та методи, які ці дані мають обробляти.

  Коли в сценарії повторно створюються об'єкти з однаковою структурою (властивості та методи об'єктів збігаються), щоб позбутися дублювання коду та зробити роботу з об'єктами зручнішою, ми можемо скористатися класами. Класи мають низку переваг у порівнянні з фабричними функціями та функціями конструкторами.

  Класи можна розглядати як аналог типів даних в інших мовах програмування. Але в JavaScript, створюючи клас, ми тільки визначаємо структуру об'єкта, яка має вийти, коли клас використовуватиметься для створення екземпляра. Навіть якщо для створення екземпляра використовувався специфічний клас, тип даних екземпляра буде object.

  Класи можна розглядати як синтаксичний цукор, який спрощує використання функцій конструкторів. Насправді визначаючи клас і додаючи в клас методи, ми визначаємо функцію конструктора та додаємо методи в прототип цієї функції, роблячи це коротшим і зручнішим способом.

- Ключове слово **class** введене в **ECMAScript 2015 або ES6**.
- Є два способи створення класу: **class declaration** та **class expression**.

  Class declaration:

  class MyClass {

  myMethod() {}

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine

Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

}

Class Expression:

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

Обидва варіанти дають можливість визначити клас і на основі цього класу створювати екземпляри. Проте варіант class expression дає можливість створити клас і передати його як параметр у метод, який викликається.

- **Класи не підтримують спливання (hoisting)**. Використовувати клас можна тільки після того, як його буде оголошено в коді.
- Клас може містити один спеціальний метод – **constructor**. Цей метод відповідає за ініціалізацію екземпляра класу.
- У класі може бути будь-яка кількість методів. Методи, які визначені у класі, будуть додані до прототипу функції конструктора.
- **Клас може містити властивості**. Властивості зазвичай додаються у конструкторі через ініціалізацію необхідної властивості на контексті (this) конструктора.
- Клас може містити спеціальні методи **get** (**getter**) і **set** (**setter**), які використовуються для того, щоб зв'язати властивість екземпляра класу з певною функціональністю, яка запускається під час читання або запису в цю властивість.
- Під час роботи з класами можна використовувати експериментальні можливості JavaScript, які підтримуються досить великою кількістю сучасних браузерів – **відкриті та закриті поля**. Поля можна розглядати як альтернативу властивостей, які додаються через функцію конструктор. Водночас поля можуть бути закритими, тоді як додана через конструктор властивість завжди доступна для використання як у класі, так і за його межами.
- **Відкрите поле** – поле класу, яке може використовуватися в класі та за його межами.

  Приклад визначення відкритого поля у класі:

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

- **Закрите поле** – поле класу, яке можна використовувати лише у класі. Якщо спробувати звернутися до закритого поля за межами класу, це призведе до помилки на етапі виконання.

  Приклад визначення закритого поля у класі (використовується # як перший символ імені поля):

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

- **Транспайлер** – тип компілятора, який виконує транспіляцію програми. Транспіляція – перетворення програмного коду, за якого вихідний код однією мовою програмування перетворюється на еквівалентний програмний код іншою мовою програмування.

CyberBionic systematics

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine

Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Babel** – безоплатний транспайлер із відкритим вихідним кодом.

  Використовується для конвертації ECMAScript 2015+ (ES6+) коду назад у сумісний код, який може бути запущений у старих версіях браузерів.

- **Polyfill** – код, що реалізує будь-яку функціональність, яка не підтримується в деяких версіях браузерів.

- Якщо в проєкті використовується функціональність, яка не підтримується в браузерах, для яких має бути підтримка, зазвичай використовується транспіляція як частина процесу розробки застосунку.

  Застосунок створюється з використанням сучасних функцій мови, які мало підтримуються, після чого відбувається **транспіляція** (часто цей процес також називають **компіляцією**). Застосунок перетворюється на аналогічний за поведінкою застосунок, але написаний з використанням старішої версії JavaScript, яка підтримується цільовим браузером. У результаті до HTML-розмітки під'єднується JavaScript-код, який вийшов після компіляції (транспіляції).

- Якщо під час розробки необхідно під'єднати підтримку певної функціональності, іноді використовуються **файли polyfill** – спеціальні JavaScript-сценарії, які імітують функціональність, якої бракує браузеру. Такі файли мають під'єднуватися до того, як буде додано будь-який код, що використовує функціональність, яка додається файлом polyfill.

Закріплення матеріалу

- Що таке клас?
- Які ви знаєте способи створення класів?
- Як використовуються ключові слова get і set у тілі класу?
- У чому різниця між закритим та відкритим полем? Як визначити відкрите та закрите поле у класі?
- Що таке транспайлер, для чого він використовується?
- Що таке файл polyfill, які завдання розв'язують такі файли?

Самостійна діяльність учня

Виконайте завдання у директорії Tasks\011 Classes. Текст завдань розташований у коментарях, у тегах script.

Рекомендовані ресурси

Класи у JavaScript
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Classes

Відкриті та закриті поля
https://github.com/tc39/proposal-class-fields

Практика побудови об'єктів
https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_building_practice

Babel
https://babeljs.io/

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine

Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com