Конструкторы и прототипы

№ урока: 1 **Курс:** JavaScript Advanced

Средства обучения: Компьютер с установленной Visual Studio Code, Visual Studio, Sublime Text

Обзор, цель и назначение урока

Язык JavaScript не имеет полноценной поддержки как другие языки программирования, например, C#, Java или C++. Тем не менее в JavaScript имеется возможность создавать псевдоклассы используя функции-конструкторы и прототипы. Цель урока – изучить техники определения псевдоклассов в JavaScript.

Изучив материал данного занятия, учащийся сможет:

- Создавать свои типы данных в JavaScript
- Определять функции-конструкторы
- Работать с прототипами
- Понимать назначение и использовать общие методы класса Object

Содержание урока

1. Конструкторы
2. Свойства и методы экземпляра
3. Свойства и методы функции-конструктора
4. Прототипы и наследование
5. Общие методы Object

Резюме

В языке JavaScript не существует классов, но с помощью функций-конструкторов и прототипов можно реализовать поведение подобное использованию классов в других ООП языках программирования. Если в контексте JavaScript используется термин «класс», под ним подразумевается использование прототипа или функции-конструктора.

Функции, которые применяются совместно с оператором new называются функциями-конструкторами. Задача оператора new – создать пустой объект и передать его функции конструктору в качестве значения ключевого слова this. Задача **функции-конструктора** – заполнить объект свойствами и методами.
Пример создания объекта – new Date().

Объект – структура данных, которая содержит именованные фрагменты данных, а также может содержать методы для работы с фрагментами данных. Объект группирует связанные значения и действия в единый набор, который, как правило, облегчает процесс программирования. В таких языках как C# или C++ класс определяет структуру объекта. В JavaScript вместо класса будут использоваться конструкторы, которые настраивают по одном и тому же принципу пустые объекты.

Каждый объект имеет свой набор **свойств экземпляра**. Если имеется 10 объектов, созданных функцией-конструктором, то у каждого объекта будет свое свойство (всего будет 10 копий свойств). Свойство экземпляра – свойство, которое было задано через функцию конструктор.

Метод экземпляра – то же самое, что и свойства экземпляра, но вместо значения в нем храниться функция.

Если в методе экземпляра нужно обратиться к свойствам экземпляра, обязательно нужно использовать ключевое слово this**.**

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

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

В языке C# есть понятие статический метод и статическое свойство, то есть значение или действе, которое является общим для всех, а не принадлежит, конкретному экземпляру. Для того, чтобы имитировать статические методы в JavaScript, метод или свойство нужно определить на функции-конструкторе, а не на объекте.

Методы функции-конструктора не могут использовать ключевое слово this, так как они не связаны с конкретным объектом.

Прототип – это объект, от которого наследуется создаваемый новый объект. Любой объект в JavaScript наследует свойства и методы своего прототипа. Если добавить значение в прототип оно будет доступно для всех объектов, которые были созданы через функцию конструктор. Для получения доступа к прототипу используется следующий синтаксис **–** Конструктор.prototype.

Когда в JavaScript создается новая функция-конструктор, она добавит в объект предопределенный набор свойств.

Общие методы прототипа Object:
toString() – метод для преобразования объекта в строковое представление.
valueOf() – во многом похож на toString(), но вызывается когда JavaScript требуется преобразовать объект к элементарному типу.
equals() – метод (не является методом Object) используется для сравнения двух объектов по значению.
compareTo(obj) – метод (не является методом Object) используется для определения отношения между объектами. Если метод возвращает значение меньше нуля, объект, на котором был произведен вызов меньше чем объект, которые передан в качестве параметра. Если метод вернет значение больше нуля – объект, на котором произведен вызов, больше чем тот, который передан в параметрах. Результат метода равен нулю если объекты равны.

Закрепление материала

1. Что такое функция-конструктор?
2. Правильно ли утверждать, что в JavaScript есть классы?
3. Какое назначение оператора new при использовании совместно с функцией конструктором?
4. В чем разница между свойством и методом?
5. В чем разница между свойством экземпляра и свойством функции-конструктора?
6. В чем заключается разница между toString и valueOf?
7. Опишите действия, которые нужно предпринять, если нужно создать объекты, которые поддерживают сравнение по значению.

Дополнительное задание

Задание для урока
Создайте функцию-конструктор Box (коробка) для создания объекта со свойствами Height (высота), Width (ширина), Depth (глубина) и Material (материал). Также, в объекте должны быть методы Volume – для получения объема и equals() – для сравнения объектов, объекты должны считаться равными, если содержат одинаковые значения в свойстве Material.

Самостоятельная деятельность учащегося

Задание 1
Создайте функцию конструктор, которая создает объект со свойством экземпляра, свойством функции-конструктора, с методом экземпляра и методом функции-конструктора. Имена выберите произвольно.

Задание 2

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

Разработайте функцию-конструктор, которая будет создавать объект Human (человек). Создайте массив объектов и реализуйте функцию, которая будет сортировать элементы массива по значению свойства Age по возрастанию или по убыванию.

Задание 3
Разработайте функцию-конструктор, которая будет создавать объект Human (человек), добавьте на свое усмотрение свойства и методы в этот объект. Подумайте, какие методы и свойства следует сделать уровня экземпляра, а какие уровня функции-конструктора.

Задание 4
Создайте пример демонстрирующий назначение метода toString().

Рекомендуемые ресурсы

Создание объектов в JavaScript
http://habrahabr.ru/post/119391/

Работа с объектами
https://developer.mozilla.org/en-US/docs/JavaScript/Guide/Working_with_Objects

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

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