Функции конструкторы

№ урока:	10	**Курс:**	JavaScript Базовый

Средства обучения:	Visual Studio Code Web Browser

Обзор, цель и назначение урока

Изучить способы создания множества объектов с одинаковой структурой с использованием конструкторов. Изучить механизмы наследования через прототипы. Научиться выносить методы объектов в прототипы.

Изучив материал данного занятия, учащийся сможет:

- Создавать и использовать функции конструкторы.
- Работать с прототипами для эффективного использования оперативной памяти.
- Понимать принципы наследования через прототипы.
- Использовать hasOwnProperty и ключевое слово instanceof.

Содержание урока

1. Функция-фабрика
2. Функция-конструктор
3. Работа с прототипами
4. For...in и прототипы
5. Ключевое слово instanceof

Резюме

- Для создания объектов в JavaScript не требуется определения типа данных. Для того чтобы создавать множество объектов с одинаковой структурой, можно применять функции-конструкторы или функции-фабрики.

- Функция-фабрика – функция, которая создает, конфигурирует и возвращает объект. При этом если создать несколько объектов через одну и туже функцию, структура объектов будет одинаковой. Такой подход позволяет избавиться от дублирования кода, и объединить все операции, связанные с конфигурации объекта в одной функции. Одним из главных недостатков такого подхода является то, что при наличии методов в создаваемых объектах все методы будут дублироваться в каждом экземпляре и занимать больше оперативной памяти.

- Функция-Конструктор - функция для создания и инициализации объектов с одинаковой структурой.
  По соглашению, функции должны именоваться с заглавной буквы, а при вызове функции - перед функцией конструктором нужно использовать ключевое слово new.

- Функцию конструктор можно рассматривать как аналог классов в строго типизированных языках программирования. Для создания нового «типа данных» в JavaScript используются

CyberBionic systematics

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

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

функции конструкторы (начиная с версии ECMAScript 2015 можно также использовать ключевое слово class, которое подобнее будет рассматриваться в следующем уроке).

- В теле функции конструктора работа должна быть нацелена на то, чтобы задать структуру новому объекту. Доступ к новому объекту происходит через ключевое слово this. В конце, значение this неявно возвращается.

- Ключевое слово new перед вызовом конструктора отвечает за создание нового пустого объекта и установку этого объекта в качестве контекста для функции конструктора (контекст функции – значение ключевого слова this). Если вызвать функцию конструктор без ключевого слова this, то контекстом функции будет глобальный объект или undefined - если код выполняется в строгом режиме (strict mode). Также в теле конструктора можно определить дополнительные проверки, чтобы конструктор срабатывал корректно даже если был вызван без new. Если функция называется с заглавной буквы (соглашения для именования конструкторов), то будет хорошей практикой запустить эту функцию с ключевым словом new, даже если функция поддерживает вызов без этого ключевого слова.

- Каждая функция в JavaScript связана с объектом – прототипом. Например, function Test() {} имеет прототип, и чтобы получить к нему доступ нужно обратиться к свойству Test.prototype.
  Прототип используется в момент создания нового объекта, ключевое слово new после создания объекта связывает этот объект с прототипом той функции, которая указывается после ключевого слова new.
  Таким образом, если с помощью конструктора создать, например, 10 экземпляров, то все эти экземпляры будут связаны с одним и тем же прототипом.

- При использовании оператора **.** для получения доступа к членам объекта, происходит поиск данного члена в объекте и, если поиск не даст результата, то поиск продолжается в связанном прототипе. Если у прототипа есть свой прототип, то поиск может продолжаться дальше.
  Общей практикой является вынесение в прототип методов. Так как методы не меняют свою работу от объекта к объекту, созданному через конструктор, то нет смысла тратить оперативную память для хранения методов в каждом объекте. Метод выносится в прототип, и когда на объекте происходит вызов данного метода, то интерпретатор, после неуспешной попытки найти метод в объекте, переходит в прототип и вызывает метод на прототипе.

- Если обратиться к экземпляру и записать свойство, которое уже есть в прототипе, то свойство появляется только в указанном экземпляре – прототип остается без изменений. Получить доступ к прототипу можно только через операции чтения. Операции записи будут изменять экземпляр, с которым идет взаимодействие.

- Instanceof - ключевое слово для проверки наличия в цепочке прототипов прототипа указанной функции. Пример использования – variable instanceof MyConstructor, если переменная с именем variable была создана через конструктор MyConstructor - ключевое слово вернет значение true иначе false.

Закрепление материала

CyberBionic systematics

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

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Что такое функция-фабрика?
- Что такое функция-конструктор?
- Что произойдёт если запустить конструктор без ключевого слова new?
- Какие есть соглашения по именованию конструктора?
- За что отвечает ключевое слово new при использовании его совместно с конструкторами?
- Что такое прототип, опишите сценарии использования прототипа?
- Для чего используется ключевое слово instanceof?

Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\010 Constructor Function. Текст задач находится в комментариях в тегах script.

Рекомендуемые ресурсы

Функции конструкторы
https://www.w3schools.com/js/js_object_constructors.asp

Прототипы объектов
https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/Object_prototypes

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

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com