Контекст функции

№ урока:	13	**Курс:**	JavaScript Базовый

Средства обучения:	Visual Studio Code Web Browser

Обзор, цель и назначение урока

Рассмотреть, что такое контекст функции и как он изменяется в зависимости от расположения или способа вызова функции. Научиться привязывать контекст с помощью метода bind. Изучить способы планирования запуска функций в JavaScript.

Изучив материал данного занятия, учащийся сможет:

- Понимать, какое значение в данном случае находится в this запускаемой функции.
- Использовать методы call и apply.
- Привязывать контекст функции с помощью метода bind.
- Использовать методы setInterval и setTimeout для планирования запуска функции.

Содержание урока

1. Контекст функции
2. Смена контекста через apply и call
3. Привязка контекста через метод bind
4. Планирование вызова функции

Резюме

- Контекст функции – значение ключевого слова this. Зависит от места расположения функции в коде и от способа запуска функции.

- Если функция определена как глобальная (находится в теге script), при вызове такой функции контекстом функции является глобальный объект (window, если код выполняется в браузере).

- При включенном строгом режиме (use strict) глобальная функция не будет иметь контекста. Значение this будет равным undefined.

- Если функция определена как метод объекта – контекстом функции является этот объект. Если функция будет скопирована в новый объект - контекстом функции станет новый объект.

- Если функция является конструктором, то контекст устанавливается в новый объект при использовании ключевого слова new. Если используется функция конструктор и ключевое слово new не указывалось, контекстом будет объект, согласно правилам описанных выше. При использовании class конструктор не может быть запущен без использования ключевого слова new.

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

- Если используется getter или setter this указывает на объект, в котором определена конструкция.

- Если this используется в функции, которая находится в цепочке прототипов то this - указывает на тот объект, на котором изначально произошел вызова, а не на прототип, в котором определена функция.

- Если this используется в функции обработчике события DOM – this указывает на элемент, который инициировал событие, но данное поведение может работать не во всех браузерах.

- Если при запуске функции нужно изменить контекст функции – используются методы call или apply. Так как функция в JavaScript является объектом – функция содержит методы и свойства.
  function test() {}
  Для установки нужного контекста при запуске такой функции можно использовать:
  test.call(context);
  test.apply(context);

  Отличие между call и apply заключается в способе передачи параметров в функцию для которой меняется контекст. Call принимает параметры через запятую, apply принимает параметры в виде массива. В остальном функции ничем не отличаются и используется та, которая более удобна в конкретном случае.

- Используя метод bind на функции можно создать новую функцию, но привязать к этой функции указанный контекст. Также можно привязать параметры, для того чтобы при вызове новой функции не нужно было передавать значения, для этих привязанных параметров.

- Стрелочные функции не имеют своего контекста и всегда используют контекст окружения, в котором были определены. Стрелочные функции идеально подходят для функций обратного вызова. Если для функций обратного вызова используются обычные функции – необходимо предусмотреть возможность указания контекста если он используется в функции.

- Для планирования вызова функций, используется два глобальных метода setTimeout и setInterval. setTimeout позволяет запустить функцию после задержки один раз, а setInterval запускает функцию многократно, через указанный интервал времени.

- Оба метода возвращают id таймера. Используя clearTimeout и clearInterval и передавая в методы Id таймера, можно остановить запланированный вызов функции.

Закрепление материала

- Какой будет контекст у глобальной функции?
- Какой контекст у функции-конструктора, если вызов произошел без ключевого слова new?
- Какой контекст у функции, которая определена в прототипе?
- Какие методы позволяют поменять контекст вызываемой функции?
- В чем разница между call и apply?
- Какие есть методы планирования вызова функций, в чем их разница?
- Как можно отменить запланированный вызов функции?

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

Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\013 Function Context. Текст задач находится в комментариях в тегах script.

Рекомендуемые ресурсы

Ключевое слово This
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/this

setTimeout
https://developer.mozilla.org/ru/docs/Web/API/WindowOrWorkerGlobalScope/setTimeout

setInterval
https://developer.mozilla.org/ru/docs/Web/API/WindowOrWorkerGlobalScope/setInterval

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