



Microsoft Partner
Silver Learning



JavaScript Базовый

Контекст функции

JavaScript Базовый

Introduction



Охрименко Дмитрий
MCT

 _okhrimenko

 dmitriy.okhrimenko

 dokhrimenko



MCID: 9210561

JavaScript Базовый

Тема урока

Контекст функции

JavaScript Базовый

План урока

Контекст функции

Смена контекста с помощью методов apply и call

Привязка контекста функции через метод bind

Планирование вызова функции

Примеры использования

JavaScript Базовый

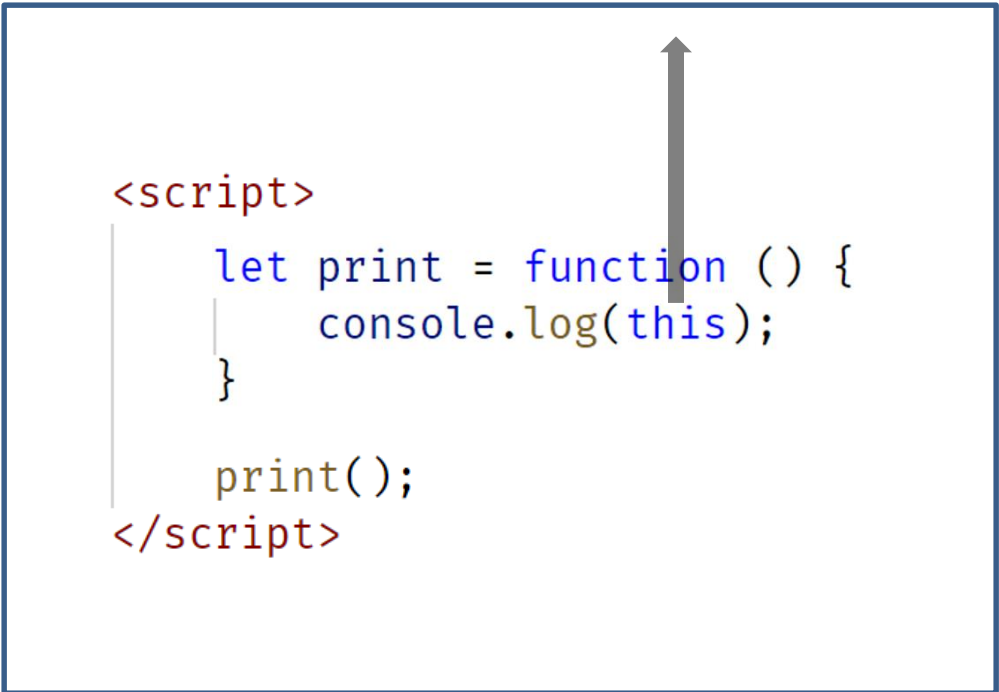
Контекст функции

this – ключевое слово, определяющее контекст выполнения кода. Указывает на объект, которому принадлежит выполняемый метод.

Window

```
<script>
  let print = function () {
    console.log(this);
  }

  print();
</script>
```



```
<script>
  let user1 = {
    login: "admin",
    email: "admin@example.com",
    print: function ()
    {
      console.log(this);
    }
  }

  user1.print();
</script>
```



JavaScript Базовый

Контекст

Расположение функции

Метод объекта

Глобальная функция

Глобальная функция (строгий режим)

Конструктор

Обработчик события

В цепочке прототипов

getter/setter

Значение this

Объект

Глобальный объект (window)

undefined

Новый создаваемый объект

Элемент выдавший событие

Объект, на котором был вызван метод

Объект

При вызове функции через методы **call** и **apply**, можно указать контекст вызова. С помощью метода **bind** можно создать новую функцию с привязанным контекстом.

JavaScript Базовый

Планирование

setTimeout – запланировать один вызов функции после указанной задержки

setInterval – запланировать многократный вызов функции с указным интервалом

Оба метода возвращают ID таймера

clearTimeout и **clearInterval** позволяют отменить запланированный вызов по указанному таймеру.

<i>Параметр</i>	<i>Описание</i>
handler	Функция или код, которые должны выполняться.
timeout	Задержка между вызовом handler.
...arguments	Список параметров для передачи при вызове handler.

JavaScript Базовый

Итоги

Контекст функции - это значение ключевого слова **this**.

Контекст функции зависит от расположения функции в коде и способа запуска функции.

Поменять контекст можно с помощью методов **apply**, **call** и **bind**.

Планирование запуска кода осуществляется с помощью функций **setInterval** и **setTimeout**.

Стрелочные функции (arrow functions) не имеют своего контекста, их контекст привязан к окружению, в котором была создана функция.

JavaScript Базовый

Спасибо за внимание! До новых встреч!



Охрименко Дмитрий
МСТ



MCID: 9210561

Информационный видеосервис для разработчиков программного обеспечения

