

Функции. Часть 2

№ урока: 12 **Курс:** Функции. Часть 2

Средства обучения: Visual Studio Code
Web Browser

Обзор, цель и назначение урока

Изучить продвинутые техники работы с функциями, разобраться с понятием области видимости, поднятием функции и поднятием переменной. Научиться определять функции разными способами и работать с функциями обратного вызова.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое область видимости.
- Понимать разницу между ключевыми словами `let` и `var`.
- Работать со стрелочными функциями.
- Работать с рекурсией.
- Использовать функции обратного вызова.
- Использовать методы для работы с массивами ECMAScript5.

Содержание урока

1. Области видимости, локальные и глобальные переменные
2. Ключевое слово `var`
3. Variable & Function hoisting
4. Стрелочные функции и анонимные функции
5. Рекурсия
6. Функции обратного вызова
7. Методы для работы с массивами ES5

Резюме

- **Область видимости (scope)** - часть программного кода, в пределах которого созданный идентификатор позволяет обратиться к привязанной к нему сущности.

Областью видимости выступает функция, если используется ключевое слово **var** для определения переменных. Если переменная создана с использованием `var`, к ней можно обратиться в любой части функции. Если к такой переменной обращение произойдет до ее определения в функции – значение переменной будет равным `undefined`.

Если при создании переменной используется ключевое слово **let** – работают блочные области видимости. Такую переменную можно использовать только после ее определения и только в текущем области видимости (блоке) и во вложенных областях видимости (блоках). Область видимости определяется с помощью операторных скобок `{ и }`.

- **Локальная область видимости** – область видимости, определенная функцией или операторными скобками. До введения ключевого слова `let`, каждая функция представляла локальную область видимости, блочной области видимости не существовало.
- **Глобальная область видимости** – область видимости, которая представляется объектом `window`. Любой код, который находится за пределами функции в элементе `script`, становится частью объекта `window`.
При инициализации переменной без ее определения, например, `name = 10`; такая переменная будет создана как глобальная.
В строгом режиме создание переменной без ключевого слова `let` или `var` приводит к ошибке.
- **Строгий режим (strict mode)** – режим выполнения сценария, который заставляет разработчика использовать более ограниченный синтаксис языка, способствующий написанию более качественного кода, через запрет проблемных конструкций языка.
- **Рекурсия** – вызов функции из этой же функции. Рекурсивный вызов, обычно используется для того, чтобы обработать значения в древовидной структуре данных. Как например, файловая система – папки, вложенные в папки и т.д. Но, если рекурсия используется для того, чтобы организовать обычный цикл — это может привести к ошибкам. Вызывая бесконечно (или достаточно большое количество раз) функцию, можно добиться ошибки, связанной с отсутствием свободного места в стеке – специальной области памяти, куда записываются адреса возврата вызываемых функций. Также, рекурсивный вызов читается в коде намного сложнее, чем обычный цикл, поэтому, реализовывая рекурсивный вызов, подумайте о возможности заменить такой код на циклическую конструкцию.
- **Поднятие или hoisting** — это механизм, который заставляет переменные и объявления функций передвигаться в начало своей области видимости перед тем, как код будет выполнен. Работает для функций – функцию можно вызвать, хотя в коде она объявлена ниже, и для переменных созданных с помощью `var`.

Закрепление материала

- Что такое область видимости?
- Что такое локальная область видимости?
- Что такое глобальная область видимости?
- Что такое рекурсия, когда стоит ее использовать и когда стоит избегать?
- Что такое strict mode?
- Что такое hoisting

Дополнительное задание

Задание

Создайте одну глобальную переменную с произвольным значением и две функции.

Продemonстрируйте, что глобальная переменная может использоваться в этих двух функциях.

Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\12 Functions в материалах к этому уроку.

Рекомендуемые ресурсы

Ключевое слово let

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/let>

Ключевое слово var

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/var>

Области видимости

<https://developer.mozilla.org/en-US/docs/Glossary/Scope>

Поднятие или hoisting

<https://developer.mozilla.org/ru/docs/Словарь/Поднятие>

Рекурсия

<https://habr.com/ru/post/337030/>