Функції. Частина 2

№ уроку:	12	**Курс:**	JavaScript Starter

Засоби навчання:	Visual Studio Code Web Browser

Огляд, ціль та призначення уроку

Вивчити просунуті техніки роботи з функціями, розібратися з поняттям області видимості, підняттям функції та змінної. Навчитися визначати функції різними способами та працювати з функціями зворотного виклику.

Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке область видимості.
- Розуміти різницю між ключовими словами let та var.
- Працювати зі стрілочними функціями.
- Працювати із рекурсією.
- Використовувати функції зворотного виклику.
- Використовувати методи роботи з масивами ECMAScript5.

Зміст уроку

1. Області видимості, локальні та глобальні змінні
2. Ключове слово var
3. Variable & Function hoisting
4. Стрілкові функції та анонімні функції
5. Рекурсія
6. Опції зворотного виклику
7. Методи роботи з масивами ES5

Резюме

- **Область видимості (scope)** - частина програмного коду, в межах якого створений ідентифікатор дозволяє звернутися до прив'язаної до нього сутності.

  Областю видимості є функція, якщо використовується ключове слово **var** для визначення змінних. Якщо змінна створена з використанням var, до неї можна звернутися в будь-якій частині функції. Якщо до такої змінної звернення відбудеться до її визначення у функції, значення змінної буде рівним undefined.
  Якщо під час створення змінної використовується ключове слово **let**, працюють блокові області видимості. Таку змінну можна використовувати тільки після її визначення і тільки в поточній області видимості (блоці), вкладених областях видимості (блоках). Область видимості визначається за допомогою операторних дужок {i}.

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

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

- **Локальна область видимості** – це область видимості, визначена функцією або операторними дужками. До введення ключового слова let кожна функція представляла локальну область видимості, блокової області видимості не існувало.

- **Глобальна область видимості –** область видимості, яка є об'єктом window. Будь-який код, що знаходиться за межами функції елемента script, стає частиною об'єкта window.
  При ініціалізації змінної без визначення, наприклад, name = 10; така змінна буде створена як глобальна.
  У строгому режимі створення змінної без ключового слова let чи var призводить до помилки.

- **Строгий режим (strict mode)** – режим виконання сценарію, який змушує розробника використовувати більш обмежений синтаксис мови, що сприяє написанню якіснішого коду через заборону проблемних конструкцій мови.

- **Рекурсія –** виклик функції з цієї функції. Рекурсивний виклик зазвичай використовується для того, щоб обробити значення в деревоподібній структурі даних. Як наприклад, файлова система - папки, вкладені папки і т. д. Але якщо рекурсія використовується для того, щоб організувати звичайний цикл, це може призвести до помилок. Викликаючи нескінченно (або досить багато разів) функцію, можна досягти помилки, пов'язаної з відсутністю вільного місця в стеку - спеціальної області пам'яті, куди записуються адреси повернення функцій, що викликаються. Також рекурсивний виклик читається в коді набагато складніше, ніж звичайний цикл, тому, реалізовуючи рекурсивний виклик, подумайте про можливість замінити такий код на циклічну конструкцію.

- **Підіймання або hoisting** — це механізм, який змушує змінні та оголошення функцій пересуватися на початок своєї області видимості перед тим, як код буде виконано. Працює для функцій – функцію можна викликати, хоча в коді вона оголошена нижче, і для змінних, створених за допомогою var

- Методи роботи з масивами:

  forEach – метод, який виконує зазначену функцію для кожного елемента масиву.
  Наприклад, наступний рядок коду виведе значення кожного елемента масиву в консоль
  array.forEach(x => console.log(x))

  filter – метод створює новий масив з усіма елементами, що пройшли перевірку, який задається за допомогою функції.
  Наприклад, наступний рядок коду поверне новий масив, в якому будуть значення більше 10 з елементів масиву array
  let newArray = array.filter(x => x > 10);

  map – метод створює новий масив, з результатами роботи функції. При цьому функція при кожному виклику отримує значення з вихідного масиву. Ця функція перетворює значення вихідного масиву на відповідні значення нового масиву.

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

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

Наприклад, наступний рядок коду створить новий масив, в якому значення будуть вдвічі більшими, ніж значення у вихідному масиві.
let newArray = array.map(x => x * 2);

every – повертає true якщо кожне значення масиву підходить під умову, що визначається функцією, яка передається.
Наприклад, наступний рядок коду поверне true, якщо в масиві немає елементів менше 0
let res = array.every(x => x > 0);

some – повертає true якщо хоча б одне значення масиву підходить під умову, що визначається функцією, яка передається.
Наприклад, наступний рядок коду поверне true, якщо в масиві є хоча б один елемент зі значенням 10
let res = array.some(x => x == 10);

Закріплення матеріалу

- Що таке область видимості?
- Що таке локальна область видимості?
- Що таке глобальна область видимості?
- Що таке рекурсія, коли її слід використовувати і коли варто уникати?
- Що таке strict mode?
- Що таке hoisting?

Додаткові завдання

Завдання
Створіть одну глобальну змінну з довільним значенням та дві функції. Покажіть, що глобальна змінна може використовуватися в цих двох функціях.

Самостійна діяльність учня

Виконайте завдання у директорії Exercises\Tasks\12 Functions у матеріалах до цього уроку.

Рекомендовані ресурси

Ключове слово let
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let

Ключове слово var
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var

Області видимості
https://developer.mozilla.org/en-US/docs/Glossary/Scope

Підняття або hoisting

https://developer.mozilla.org/en-US/docs/Glossary/Hoisting

CyberBionic systematics
CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Рекурсія
https://habr.com/en/post/337030/

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

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