

Толмачов Н.В.

студент,

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ФУНКЦІЇ, ЩО ВИКЛИКАЮТЬСЯ НЕГАЙНО (IMMEDIATELY-INVOKED FUNCTION EXPRESSION) У MOBI JAVASCRIPT. ЇХ ПРИЗНАЧЕННЯ ТА ПЕРЕВАГИ

Функція, що викликається негайно в JavaScript (IIFE) – це синтаксична конструкція, що дозволяє викликати функцію безпосередньо в місці її визначення. Вона має такий вигляд:

```
(function() {  
  //код  
})();  
Або так:  
(function() {  
  //код  
})();
```

Обидва варіанти еквівалентні. Коли ми визначаємо функцію, наприклад, `function func () {}` (Function Expression) або `var func = function() {}` (Function Declaration), створюється ідентифікатор функції, за допомогою якого можна викликати функцію, додавши круглі скобки: `func()`. Таким чином, щоб негайно викликати функцію, необхідно після функціонального виразу поставити круглі скобки. Але, якщо не загорнути сам функціональний вираз в круглі скобки, буде викликано синтаксичну помилку [1, с. 274]. Щоб розібратись, навіщо слід загортати функціональний вираз у круглі скобки, слід розглянути декілька прикладів і розібратися, чому вони неправильні.

```
function () {/ * код * /} (); / * SyntaxError:  
function statement requires a name * /
```

В цьому прикладі було отримано вилючення `SyntaxError`. Коли інтерпретатор зустрічає ключове слово `function`, за замовчуванням він приймає його за визначення функції (а не функціональний вираз). А оголошення функції вимагає індикації імені. Наступний приклад:

```
function func () {/ * код * /} (); / * SyntaxError:  
syntax error * /
```

Було додано ім'я функції. Тепер оголошення функції є правильним з синтаксичної точки зору, але це оголошення функції, а не вираз, тож скобки в цьому випадку будуть виконувати роль оператора групування (він керує пріоритетом операцій). Такі скобки не можуть бути пустими, тому буде викликано виключення. Тож спробуємо заповнити ці скобки:

```
function func () {/ * код * /} (1 + 1);
```

Цей приклад не викличе помилок, бо функцію визначено правильно, а оператор групування має вираз. Але функція не буде викликана негайно, бо не є функцією-виразом. Тому, простий спосіб повідомити інтерпретатор про те, що ключове слово `function` описує функціональний вираз – загорнути його в круглі скобки. Також слід звернути увагу на те, що якщо функція була присвоєна змінній, то вона є функціональним виразом і не потребує додаткових скобок:

```
var f = function () {/ * код * /} ();
```

Даний приклад написаний правильно, але для зручності читання коду рекомендується використовувати круглі скобки в даному випадку. Тоді людина, що читає код, зверне увагу на те, що змінній присвоюється значення функції, а не функціональний вираз [2].

Передати параметри функції типу IIFE можна таким чином:

```
(function print (firstWord, secondWord) {  
    alert (firstWord + secondWord); // Hello World  
}) ('Hello', ' World');
```

Функції такого типу зазвичай використовують для фіксації значень, що змінюються в циклі. Розглянемо наступний приклад:

```
var elems = document.getElementsByTagName ( 'a' );  
for (var i = 0; i < elems.length; i ++) {  
    elems [i] .addEventListener ( 'click', function () {  
        console.log( 'Number #' + i)  
    } False)  
}
```

В цьому прикладі вибирається масив посилань, і кожний з них присвоюємо обробник події «click». Очікуваний результат – при кліку на посилання буде виведено її порядковий номер в масиві. Але даний приклад не буде працювати правильно, бо поки відбудеться подія «click», значення змінної «i» зміниться, і клік по всім посиланням буде виводити останнє значення індексу «i».

Для виправлення цієї помарки слід використати функцію, що негайно викликається. Передається поточне значення «i» в цю функцію, таким чином воно «фіксується»:

```
var elems = document.getElementsByTagName('a');
for (var i = 0; i < elems.length; i++) {
    (function(elemNum){
        elems[i].addEventListener('click', function() {
            console.log('Number #' + elemNum);
        }, false);
    })(i);
}
```

Також за допомогою IEFE можна згенерувати обробник подій:

```
var elems = document.getElementsByTagName('a');
for (var i = 0; i < elements.length; i++) {
    elems[i].addEventListener('click',
    (function(elemNum){
        return function() {
            console.log('Number #' + elemNum);
        };
    })(i), false);
}
```

За допомогою функції, що негайно викликається в JavaScript реалізується патерн-шаблон «модуль». Суть цього патерна полягає в створенні локальної області видимості, в якій визначаються локальні («приватні») змінні та функції [3].

```
var basketModule = (function() {
    var basket = []; // приватна змінна
    return { // публічні методи
        addItem: function(values) {
            basket.push(values);
        },
        getItemCount: function() {
            return basket.length;
        },
        getTotal: function() {
            var q = this.getItemCount(), p=0;
            while(q--){
                p+= basket[q].price;
            }
            return p;
        }
    }
})();
```

За допомогою функції, що негайно викликається можна робити такі дії, для яких потрібно було б додавати додаткові змінні у глобальну область видимості.

Наприклад, «фіксація» змінної, яка протягом циклу приймає різні значення. Також паттерн-шаблон «модуль» заснований на даній функції, який дозволяє залишати глобальну область видимості без додаткових змінних.

Список використаних джерел:

1. Флэнеган Д. JavaScript. Подробное руководство, 6-е издание / Дэвид Флэнеган. – Санкт-Петербург: Символ-Плюс, 2012. – 1080 с.
2. Модули через замыкания [Електронний ресурс]. – Режим доступу до ресурсу: <https://learn.javascript.ru/closures-module>
3. Паттерны для масштабируемых JavaScript-приложений [Електронний ресурс]. – Режим доступу до ресурсу: <http://largescalejs.ru/module-pattern/>.

Томашівська М.В.

викладач, викладач-методист;

Согор А.А.

викладач,

Екологічний коледж

Львівського національного аграрного університету

МОДЕРНІЗАЦІЯ ІНФОРМАЦІЙНО-РЕСУРСНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНО-ВИХОВНОЇ РОБОТИ У ВНЗ УКРАЇНИ, ЯКІ ГОТУЮТЬ ФАХІВЦІВ ЗА КВАЛІФІКАЦІЄЮ «МОЛОДШИЙ СПЕЦІАЛІСТ»

Метою нашого дослідження є аналіз основного інформаційно-ресурсного забезпечення навчальної роботи у ВНЗ I-II рівнів акредитації та його модернізація.

У Постанові Кабінету Міністрів України від 30 грудня 2015 року № 1187 «Про затвердження Ліцензійних умов провадження освітньої діяльності закладів освіти» [6] значно розширюється поняття інформаційних ресурсів.

Жоден ВНЗ не спроможний створити самостійно потужну інформаційну базу, яка б відповідала потребам сьогодення. Тому застосування мережових технологій, можливість доступу до електронних інформаційних ресурсів суттєво підсилюють інформаційну складову потенціалу вищої школи.. Останнім часом активно впроваджується державна програма «Інформаційні та комунікаційні технології в освіті і науці», одним із завдань якої є розбудова інфраструктури науково-освітньої телекомунікаційної мережі (УРАН), підключення до неї наукових установ, наукових бібліотек, центрів науково-технічної інформації, інтеграція її з європейською науково-дослідницькою мережею (GEANT) [1].

На теренах європейських країн створено Європейську цифрову бібліотеку Europeana (<http://www.europeana.eu/portal>) [8].

Прикладами якісних вітчизняних відкритих освітніх ресурсів. є бібліотека українських шкільних підручників (<http://pidruchniki.net>), електронна бібліотека