

Технічні науки

УДК 004.852

Магас Дмитро Михайлович

студент

Івано-Франківського національного технічного університету нафти і газу

Магас Дмитрий Михайлович

студент

Ивано-Франковского национального технического

университета нефти и газа

Mahas Dmytro

Student of the

Ivano-Frankivsk National Technical University of Oil and Gas

**РЕАЛІЗАЦІЯ ОЧИЩЕННЯ ДИНАМІЧНОЇ ТАБЛИЦІ
РЕАЛИЗАЦИЯ ОЧИСТКИ ДИНАМИЧЕСКОЙ ТАБЛИЦЫ
REALIZATION OF CLEARING OF THE DYNAMIC TABLE**

Анотація. Динамічні таблиці, їх призначення та реалізація.
Очищення динамічної таблиці.

Ключові слова: динамічна таблиця, алгоритм, web, JS, DOM.

Аннотация. Динамические таблицы, их назначение и реализация.
Очистка динамической таблицы.

Ключевые слова: динамическая таблица, алгоритм, web, JS, DOM.

Summary. Dynamic tables, their purpose and realisation. Clearing of the
dynamic table.

Key words: dynamic table, algorithm, web, JS, DOM.

Вступ. Маніпулювання з динамічними даними є цілком буденним явищем для більшості програмістів, особливо якщо мова йде про веб-

орієнтовані мови програмування. Проте, для ефективного використання динамічних даних, потрібно враховувати чимало дрібниць, будувати допоміжний функціонал та реалізовувати механізми, без яких в звичайних умовах можна було би обійтись. Без сумніву, існують очевидні переваги використання динамічних даних – проект набуває певної інтелектуальності, підвищується рівень інтерактивності, іноді вдається скоротити час досягнення цільової дії. В даній статті ми розглянемо звичайну динамічну таблицю та кілька підходів до її очищення.

1. Поняття та побудова динамічної таблиці

Динамічна таблиця – це вид даних, який структурований в табличному вигляді, основною особливістю якого є механізм видачі (утворення) інформації «на льоту».

У веб-програмуванні є чимало креативних та складних способів побудувати динамічну таблицю. В найпростішому ж випадку розробники звертаються до DOM.

DOM (англ. Document Object Model, укр. Об'єктна модель документа) — специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами (як правило, документів HTML, XML). Визначається ця специфікація консорціумом W3C.

Разом із поширенням та розвитком веб-технологій і веб-переглядачів почали з'являться різні, часто несумісні інтерфейси роботи із HTML документами в інтерпретаторах JavaScript вбудованих в веб-переглядачі. Це спонукало World Wide Web Consortium (W3C) узгодити та визначити низку стандартів, які отримали назву W3C Document Object Model (W3C DOM). Специфікації W3C не залежать від платформи або мови програмування.

Перш за все потрібно вирішити як саме будуватиметься таблиця:

- 1) з самого початку;

2) шляхом наповнення існуючої порожньої таблиці.

Для другого варіанту необхідно, щоб у файлі .html містилась порожня таблиця вигляду `<table id = "some_id"> </table>`, будь-який блок фільтрів з кнопкою підтвердження (submit), яка викликатиме подію onclick при натисканні, що в свою чергу ініціюватиме виконання функції, яка будуватиме таблицю. Також у файлі стилів (.css) необхідно попередньо приховати таблицю :

```
#someid {  
    display: hidden;  
}
```

Приклад кнопки submit:

```
<input type="submit" class="btn btn-success" value="Підтвердити"  
onclick="getTable()">
```

Приклад функції динамічного заповнення таблиці:

```
function getTable() {  
  
    var mt = document.getElementById("someid");  
  
    //робимо таблицю видимою і додаємо їй рамку  
  
    mt.style.display = "visible";  
  
    mt.style.border = "1px solid lightgray";  
  
    //додаємо таблиці рядок з коміркою  
  
    var r1 = mt.insertRow(-1);  
  
    var r1c1 = r1.insertCell(0);  
  
    r1c1.innerHTML = "sometext";  
    // додаємо верхню рамку та внутрішні відступи для комірки
```

```
r1c1.style.borderTop = "1px solid lightgray";  
  
r1c1.style.padding = '3px';  
  
}
```

Таким чином можна отримати динамічно створену (заповнену) таблицю.

2. Очищення динамічної таблиці

Ми вже переконались, що побудувати динамічну таблицю доволі легко. Як щодо її очищення? Зазвичай, очищення динамічної таблиці потрібне для нової видачі результату. Якщо не здійснити очищення таблиці, то в таблиці будуть як нові так і старі результати. Реалізувати очищення таблиці можна шляхом створення кнопки refresh та відповідної функції очищення.

Приклад кнопки:

```

```

Можна цілком обійтись і без кнопки, очищаючи таблицю автоматично при кожному наступному натисканні клавіші «підтвердити» й видаючи новий результат. В ідеалі потрібно передбачити як ручне очищення (скидання) результату так і автоматичне. Також нерідко застосовується підхід при якому таблиця очищається не повністю, а лиш відповідно до змін у блоці параметрів, але така реалізація вимагає додаткових умовних структур і рідко є виправданою.

Функція ручного очищення майже ніколи не реалізується як безпосереднє очищення, а найчастіше як перезагрузка робочої області:

```
function makeRefresh() {
```

```
        window.location.reload();  
    }  
}
```

Автоматичне очищення таблиці можна реалізувати скориставшись лічильником (флагом стану). Він встановлюється перед основним тілом функції видачі результату і приймає значення 0. Після підтвердження видачі і побудови таблиці значення лічильника інкрементується. При повторній видачі результату здійснюється перевірка лічильника на рівність 0, у разі спрацювання false-сценарію таблиця очищається, а значення лічильника знову прирівнюється нулю, після чого ініціюється запуск функції видачі результату «зсередини».

Приклад реалізації автоматичного очищення засобами JS DOM:

```
var count = 0; //лічильник для оновлення результатів  
  
function getTable() {  
    if (count > 0) { //Перевірка на наявність результату, якщо є то очистка  
таблиці і запуск, якщо ні, то запуск  
  
        while (document.getElementById("someid").hasChildNodes()) {  
document.getElementById("someid").removeChild(document.getElementById("someid").  
lastChild);  
  
        }  
  
        count = 0;  
  
        getTable();  
  
    } else {  
  
        ... внутрішнє наповнення таблиці ...  
  
        Counter();  
  
    }  
  
    //функція інкременту (збільшення результату на 1)  
  
    function Counter() {
```

```
count++;  
  
}
```

Висновки. Як ми мали змогу побачити, очищення динамічних таблиць не є надто складним з точки зору програмування, проте вимагає уваги та розуміння особливостей проекту в контексті якого вирішується дане завдання. Поставивши себе на місце майбутнього користувача, можна реалізувати механізми взаємодії з динамічними даними (зокрема і очистки) найбільш зручним та передбачуваним шляхом.

Література

1. DOM [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/Об%27єктна_модель_документа.
2. Mark H. Overmars The Design of Dynamic Data Structures / Mark H. Overmars - Springer. -188 p.
3. W3C [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org/wiki/W3C>.