

## СПОСОБИ ВІДБОРУ ДАНИХ ДО JAVASCRIPT СЦЕНАРІЮ ПОБУДОВИ ДІАГРАМИ

Качурівська Ганна, Качурівський Володимир

Відокремлений підрозділ Національного університету біоресурсів і природокористування України  
«Бережанський агротехнічний інститут»

### Анотація

Розглянуто питання відбору числових даних із незалежних носіїв даних для автоматизації побудови діаграм засобами CANVAS API. Подано фрагмент Javascript коду відбору даних із зовнішнього файлу. Реалізовано метод відбору даних з рядка та стовпця таблиці даних, вкладеної в HTML-сторінку.

### Abstract

Issues of numerical data selection from independent data carriers for automation diagrams construction by means of the CANVAS API are considered. The fragment of data selection code from an external file is given. The data selection method from a row and a column of a data table embedded in the HTML page is implemented.

### Вступ

Усе частіше в аналітичних матеріалах сайтів використовують супровід числових даних їх візуальним поданням (інфографіка) за допомогою різного роду графічних об'єктів, призначених для швидкого та доступного сприйняття усієї інформації комплексно. Одним із засобів створення інформаційної графіки є побудова діаграм засобами CANVAS API.

### Постановка практичної проблеми

У публікаціях Danny Goodman [3] та Володимира Дронова [5] розглянуто питання створення комп'ютерної графіки в HTML-документах за допомогою CANVAS. У працях Steve Fulton та Jeff Fulton [2] та Джоша Маріначі [4] детально висвітлено оператори для побудови геометричних фігур, які можна використовувати для комплексної побудови інфографіки. Основи роботи з елементом HTML5 <canvas> описано в публікаціях [1; 6]. На даний час практично відсутні публікації, в яких описані методи та способи адаптації полотна побудови графіки під розміри екрану, а також не розглянуті способи передачі до Javascript сценарію числових даних, на основі яких будується інформаційний графік.

**Перелік вирішуваних завдань.** Визначити фрагмент коду JavaScript сценарію з побудови діаграми для відбору числових даних із зовнішнього файлу, а також із рядка або стовпця таблиці, вкладеної в HTML-документ.

### Виклад суті дослідження

Побудова діаграми засобами CANVAS API здійснюється на основі числових даних, сформованих у масиві, для прикладу data. Спосіб передачі числових даних безпосередньо при виклику JavaScript сценарію має певну незручність, а саме: при зміні числових даних необхідно проводити зміни в записаних параметрах виклику сценарію на нові числові дані в тексті Web-сторінки. Одним із способів вирішити дану проблему є реалізація відбору даних із зовнішнього файлу. Змінивши файл на новий, з актуальними даними, при активізації web-сторінки проводиться зчитування уже нових числових даних та здійснюється відповідна побудова діаграми.

Зосередимо свою увагу на формуванні числового масиву, використовуючи зовнішні джерела даних, який використовується для побудови діаграми.

***Відбір числових даних із зовнішнього файлу.***

Проаналізувавши різного типу файли збереження даних, зробили висновок, що найпростішим та зручним для формування числових даних є звичайний текстовий файл, створений в будь-якому текстовому редакторі з розширенням txt. Текстовий файл підпорядковується таким правилам запису: числові дані записуються у рядок, між даними ставиться розподільник «пробіл».

Наприклад, файл data.txt має такі числові дані: 20 30 50 40

Заголовок JavaScript сценарію побудови діаграми буде наступним:

function diagram\_2(path), де path – формальна змінна, яка прийме значення визначеного шляху доступу до файлу даних при його виклику.

Виклик JavaScript сценарію буде таким: diagram\_2('.../data.txt');

**Фрагмент коду.** Зчитування даних з файлу data.txt проведемо асинхронно за допомогою технології AJAX.

```
var oAJAX= new XMLHttpRequest();  
oAJAX.open("GET",path,false);  
oAJAX.send();
```

Результатом роботи операторів буде тестовий рядок. Для розподілу даних скористаємося функцією split() із вказаним розподільником «пробіл», згідно до формату файлу даних.

```
var arr=oAJAX.responseText.split(' ');
```

Результатом роботи функції буде масив arr із текстовими значеннями.

Для перетворення символічних рядків у числові дані використовуємо функції Number та наступну конструкцію.

```
var data=[];var j=0;  
for(i=0;i<arr.length;i++)  
if (!isNaN(Number(arr[i].innerHTML)))  
    {data[i]=Number(arr[i]);j++;}
```

Кількість переданих числових значень визначається як n=data.length;

Сформований числовий масив data використовується для побудови діаграми.

***Відбір числових даних таблиці, вкладки у HTML-сторінку.***

Ще одним способом відбору даних для побудови діаграми є використання значень з таблиці, яка розміщена у HTML-документі. Таблиця формується парними тегами <table> </table>, <tr> </tr> та <td> </td>. Для реалізації вибірки числових даних таблиця повинна бути простою (відсутні об'єднання комірок). Також врахуємо такий факт, що перший стовпець або рядок таблиці містять текстові підписи.

***Відбір даних з рядка таблиці***

Зазвичай джерелом даних для побудови діаграми є певний рядок таблиці. Рядок даних таблиці формується парними тегами <tr></tr> та <td></td>. Таблиця може містити декілька рядків. Для однозначного визначення рядка таблиці з якого необхідно взяти дані для побудови діаграми, доцільно задати ідентифікатор рядка. Наприклад (фрагмент HTML-документа конструювання таблиці):

```
<table>  
<tr id="data1"><td>20</td><td>30</td> <td>50</td><td>40</td>  
</tr> </table>, де data1 – ідентифікатор рядка даних.
```

Заголовок JavaScript сценарію буде наступним:

function diagram\_3(id\_tr), де id\_tr – формальна змінна-ідентифікатор рядка таблиці.

Виклик JavaScript сценарію буде таким: diagram\_3('data1');

**Фрагмент коду.** Вибір числових даних здійснюється такою конструкцією.

```
var data=[];
```

```
var row=document.getElementById(id_tr);
cell = row.getElementsByTagName('td');
var n=cell.length;var j=0;
for(i=0;i<n;i++)
    if ( !isNaN(Number(cell[i].innerHTML)))
        {data[j]=Number(cell[i].innerHTML);j++;}
```

*Примітка:* `isNaN()` – перевірка чи є вміст комірки числом.

Числовий масив `data` сформований і може використовуватися для побудови.

### *Відбір числових даних із вказаного стовпця таблиці*

Враховуючи тегову структуру побудови таблиці, однозначної ідентифікації стовпця, з якого необхідно взяти дані для побудови діаграми, провести неможливо. Для програмування відбору даних слід скористалися такою гіпотезою:

Із колекції вибраних комірок (за тегом `<td>`) необхідно вибрати кожну  $k$ -ту ( $k$  – кількість стовпців таблиці) починаючи з  $n$ -ої за індексом колекції.

Опишемо гіпотезу, детальніше:

1. Створити колекцію комірок таблиці;
2. Визначити кількість рядків та відповідно кількість стовпців;
3. Запустити процедуру відбору необхідних комірок даних.

Заголовок JavaScript сценарію буде наступним:

```
function diagram_4(id_table,n_coll), де id_table –
ідентифікатор таблиці, яка містить необхідні дані, n_coll – номер колонки, з якої
вибирати числові дані.
```

**Фрагмент коду.** Вибір числових даних з стовпця здійснюється такою конструкцією.

```
var data_td=document.getElementById(id_table);
var cell=data_td.getElementsByTagName('td');
var n=cell.length;
var row = data_td.getElementsByTagName('tr');
var k=n/row.length;
var data=[];
var j=0;
for(i=n_coll-1;i<n;i+=k)
    if ( !isNaN(Number(cell[i].innerHTML)))
        {data[j]=Number(cell[i].innerHTML);j++;}
```

Числовий масив `data` сформовано.

**Висновки.** Використання автономних носіїв числових даних дозволяє автоматизувати побудову діаграм для певної множини вхідних даних. Подальшої розробки потребують питання відбору числових даних з файлів формату `csv`.

### **Список використаних джерел:**

1. HTML5 Canvas. URL: <https://www.tutorialrepublic.com/html-tutorial/html5-canvas.php> (дата звернення: 15.01.2018).

2. Steve Fulton, Jeff Fulton. HTML5 Canvas. O'Reilly Media, 2011. 654 p.

3. Д. Гудман, JavaScript и DHTML. Сборник рецептов. Для профессионалов. Москва, 2015. 523 с.

4. Джош Мариначи. Основы рисования. URL: <https://webref.ru/dev/canvasdeepdive/chapter01> (дата звернення: 15.01.2018).

5. Дронов В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. Москва, 2013. 416 с.

6. Элемент Canvas HTML5. URL: [https://msdn.microsoft.com/uk-ua/gg589510\(v=vs.85\)](https://msdn.microsoft.com/uk-ua/gg589510(v=vs.85)) (дата звернення: 21.01.2018).