

ECMAScript 6 У ПОВСЯКДЕННОМУ ЖИТТІ

Вінницький національний технічний університет

Анотація

Представлено структури даних нового стандарту ECMAScript 6 з метою кращого абстрагування, забезпечення стійкості структури додатку до зовнішніх та внутрішніх чинників.

Ключові слова: ECMAScript 6, ECMAScript 5, Single page application.

Abstract

Presented data structure of the new standard ECMAScript 6 to better abstraction, ensuring the stability of the structure of the annex to external and internal factors.

Keywords: ECMAScript 6, ECMAScript 5, Single page application.

Розробка web-додатків таких, як Single page application [1], та серверної частини NodeJs [1] набирають все більшої популярності, замінюючи додатки, які були розроблені виключно для специфічних операційних систем. ECMAScript 6 [3], який прийшов на зміну ECMAScript 5 [1], дозволяє зберегти функціональність вже існуючого стандарту та розширити його, що дасть можливість розробляти системи підвищеної складності, здатні до масштабування.

У ES-2015 [3] передбачені нові способи оголошення змінних, а саме через `let` і `const` замість `var`: `let a = 5`;

При цьому, стосовно змінні `let` характерно таке.

1. Змінні видно тільки після оголошення.
2. Змінні видно тільки в поточному блоці.
3. Змінні видно не можна пере оголошувати в тому ж блоці.
4. При оголошенні змінної в циклі `for (let ...)` її видно тільки в поточному циклі. При цьому, в кожній ітерації оголошується своя змінна `let`.
5. Константа `const` оголошується лише раз для кожної операції.

Деструктуризація (destructuring assignment) - це особливий синтаксис присваювання, що дозволяє виконати призначену операцію стосовно масив або об'єкту одночасно кільком змінним, розбивши його на частини:

```
let {prop : varName = default, ...} = object.
```

Двокрапка задає відображення властивості `prop` в змінну `varName`, а значення `default` задає вираз, який буде використано, якщо значення відсутнє.

Для масивів, де має значення порядок, можна використовувати значення за замовчуванням:

```
let [var1 = default, var2, ...rest] = array.
```

Оголошення змінної на початку конструкції не обов'язково. Можна використовувати існуючі змінні, однак при деструктуризації об'єкта може знадобитися взяти вираз в дужки.

Опис вкладених об'єктів і масивів є класичним але при деструктуризації потрібно зберегти туж саму структуру що й вихідний об'єкт / масив.

У функціях основні зміни стосуються передачі параметрів, але, потрібно ввести додатковий короткий запис через стрілку `=>` [3].

```
function showMenu(title = "Без заголовку", width = 100, height = 200) {  
  alert(`${title} ${width} ${height}`);  
}
```

При передачі будь-якого значення, крім undefined, включаючи порожній рядок, нуль або null, параметр вважається переданим, і значення, за замовчуванням, не використовується.

В стандарті ECMAScript 6 покращена підтримка Unicode [3]. Внутрішнє кодування рядків в JavaScript - це UTF-16, тобто під кожен символ відводиться рівно два байта. Проте 2 байт під будь-які символи мов світу не вистачає, що призводить до використання двох Юнікодних символів (4 байта) на один символ мови. Таке поєднання називають «сурогатною парою» [1]. Найчастіший приклад сурогатної пари - це китайські ієрогліфи. Однак не всякий китайський ієрогліф - сурогатна пара. Значна частина “основного” юнікод-діапазону орієнтована на китайську мову, а тому деякі ієрогліфи представляються одним юнікод-символом, а всі інші – двома:

```
alert( '我'.length ); // 1
alert( '鯨'.length ); // 2
```

У тексті вище для першого ієрогліфа є окремий юнікод-символ, а тому довжина рядка 1, а для другого - використовується сурогатна пара (довжина – 2). Китайськими ієрогліфами сурогатні пари, як правило, не обмежуються. Їх використовують для представлення непоширених математичних символів, а також деяких символів для емоцій, наприклад:

```
alert( 'X'.length ); // 2, MATHEMATICAL SCRIPT CAPITAL X
alert( '☹'.length ); // 2, FACE WITH TEARS OF JOY
```

В сучасний JavaScript додані методи String.fromCodePoint і str.codePointAt - аналоги String.fromCharCode і str.charCodeAt, що коректно працюють з сурогатними парами [2]. Наприклад, charCodeAt вважає сурогатну пару двома різними символами і повертає код кожної.

У багатьох мовах є символи, які формуються як поєднання основного символу і позначки над ним/під ним. Найбільш часто зустрічаються такі поєднання, що мають окремий Юнікодний код. Для генерації довільних поєднань використовуються кілька Юнікодних символів.

Стандарт ECMAScript 6 є актуальним, оскільки популярність web-додатків таких, як Single page, на просторі інтернет зростає крім того стандарт ECMAScript 6 не конфліктує з ECMAScript 5, це дає змогу використовувати існуючі модулі.

Отже, новий стандарт ECMAScript 6, що підтримується близько 80% браузерів, можна використовувати, транспілюючи його в ECMAScript 5 за допомогою Babel.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Крокфорд Д. JavaScript сильні сторони / Дуглас Крокфорд. - Санкт-Петербург, Мир книг, 2013. 147 ст.
2. Босуел Д. Читаемый код или Программирование как искусство / Д. Босуел, Т. Фаучел. - Мир книг, 2012. 152 ст.
3. MOZILLA DEVELOPER NETWORK [Електронний ресурс] : [JavaScript reference] Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

Савчук Тамара Олександрівна — к.т.н., доцент, професор кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця

Куліковський Ігор Іванович — студент кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: kulikovsky93@gmail.com.

Tamara O. Savchuk — Cand. Sc. (Eng), Assistant Professor, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia;

Igor I. Kylikovsky — student of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, email: kulikovsky93@gmail.com.