```
const counter1 = makeCounter();
const counter2 = makeCounter();

console.log(counter1()); // 1
console.log(counter2()); // 1
console.log(counter2()); // 2
console.log(counter1()); // 2
```

Приклад показує, що далі ми можемо створювати необмежену кількість лічильників, які будуть незалежно і надійно виконуватися один від одного.

В JavaScript у кожної виконуваної функції, блоку коду чи скрипта є пов'язаний з ними внутрішній (прихований) об'єкт, який називається лексичним оточенням (англ. Lexical Environment). Цей об'єкт складається з двох частин [1]:

1. об'єкт, в якому як властивості зберігаються всі локальні змінні, а також параметри та деяка інша інформація, така, наприклад, як значення this;
2. посилання на зовнішнє лексичне оточення – тобто те, яке відповідає коду зовні (зовні від поточних фігурних дужок).

Література

1. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание. – Пер. с англ. – СПб: СимволПлюс, 2012. – 1080 с., ил.

ПРАКТИКА ДЕСТРУКТУРИЗАЦІЇ МАСИВІВ В JAVASCRIPT

Одегова Є.О.
Науковий керівник – ***Григорьев О.В., канд. техн. наук, доцент***
(Харківський національний університет радіоелектроніки)

Нехай маємо наступний масив `arr1 = [1, 2, 3, 4, 5]`, елементи якого необхідно присвоїти окремим змінним, наприклад, `a, b, c, d` та `e`. Найочевиднішим вирішенням цієї проблеми буде, наприклад, такий код:

```
const arr1 = [1, 2, 3, 4, 5];

const a = arr1[0];
const b = arr1[1];
const c = arr1[2];
const d = arr1[3];
const e = arr1[4];
```

```
console.log(a); // 1
console.log(b); // 2
console.log(c); // 3
console.log(d); // 4
console.log(e); // 5
```

Але цей підхід значно ускладнюється, коли розмір масиву відразу не відомий, або він змінюється по ходу виконання програми. В такому випадку кращим рішенням буде використання особливості мови програмування JavaScript, яка називається деструктуризація [1]. Деструктуризація (або деструктуруюче присвоєння, англ. destructuring assignment) – це особливий синтаксис присвоювання, при якому можна присвоїти елементи масиву окремим змінним. Тоді, з використанням деструктуризації, попередній код буде виглядати наступним чином:

```
const arr1 = [1, 2, 3, 4, 5];

const [a, b, c, d, e] = arr1;

console.log(a); // 1
console.log(b); // 2
console.log(c); // 3
console.log(d); // 4
console.log(e); // 5
```

За допомогою деструктуризації можна присвоювати не всі елементи масиву a, наприклад, тільки перші два:

```
const [a, b] = [1, 2, 3, 4, 5];

console.log(a); // 1
console.log(b); // 2
```

або два, починаючи з другого:

```
const [, , a, b] = [1, 2, 3, 4, 5];

console.log(a); // 3
console.log(b); // 4
```

В поєднанні з rest-параметрами на основі існуючого масиву можна легко створити новий масив з декількох останніх елементів:

```
const [, , ...restArr] = [1, 2, 3, 4, 5];

console.log(restArr); // 3, 4, 5
```

Не існуючим елементам масиву можна присвоїти значення за замовчуванням або, взагалі, вирахувати, чи ввести з зовні:

```
const [, , a = 99, b = prompt('Input b?')] =
[1, 2, , , 5];

console.log(a); // 99
console.log(b); // ...
```

Оскільки дані типу `string` в певній мірі теж можна розглядати як масиви, то з ними також можна використовувати деструктуризацію. При цьому код програми становиться тільки яснішим та краще сприймається візуально.

Література

1.Хавербек М. Выразительный JavaScript, 2-е издание. – Пер. с англ. – М.: Вильямс, 2014 – 437 с.

АНАЛІЗ НЕОБХІДНОСТІ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ШКОЛЯРІВ

Данів С.Р.
Науковий керівник – ***Сенчук Т.С., ст. викладач***

Людині XXI віку важко уявити своє життя без високотехнологічних пристроїв, як-то смартфони, планшетні ПК, бо саме ці науково-технічні новинки дозволяють не тільки підтримувати зв’язок з людьми із різних куточків світу, а й допомагають у формуванні та розвитку сучасного інформаційного простору.

Сфера IT прогресує кожного дня. Розробки у сфері мобільних додатків стають дедалі різноманітнішими, бо навіть побутова техніка (телевізори, холодильники і т.д.) зараз керується під Android OS.

Сьогодні за допомогою смартфону можна не тільки робити високоякісні зображення, а й переглянути інформацію стосовно будь-кого або – чого (сторінки у соціальних мережах, блогах, інформаційних порталах і т.д.). Наявність сучасних технологій дає нам змогу не тільки полегшити життя, а ще й економить наш час. Інформація та дані все