

## Лабораторна робота №3

### Написання функцій. Робота з масивами.

**Мета:** навчитися створювати функції в мові JavaScript. Ознайомитися з методами для обробки масивів. Закріпити навички роботи з рядками.

#### Хід Роботи:

**Завдання 1.** 1. Напишіть функцію, яка приймає 2 рядки та порівнює їх довжину. Функція повертає 1, якщо в першому рядку більше символів, ніж у другому; -1 – якщо у другому більше символів, ніж у першому; 0 – якщо рядки однакової довжини.

```
function stringcomp(str1, str2){
    if(str1.length > str2.length){
        return 1
    }
    if(str1.length < str2.length){
        return -1
    }
    else {
        return 0
    }
}
```

```
let s1 = prompt("string 1:");
let s2 = prompt("string 2:");
alert(stringcomp(s1, s2))
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.125.11.000 – Лр.3							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Шарковський Р.О			Звіт з лабораторної роботи №3				Літ.	Арк.	Аркушів	
Перевір.		Болотіна В. В									1	13
Реценз.									ФІКТ, гр. КБ-21-2			
Н. Контр.												
Зав.каф.		Єфіменко А.А.										

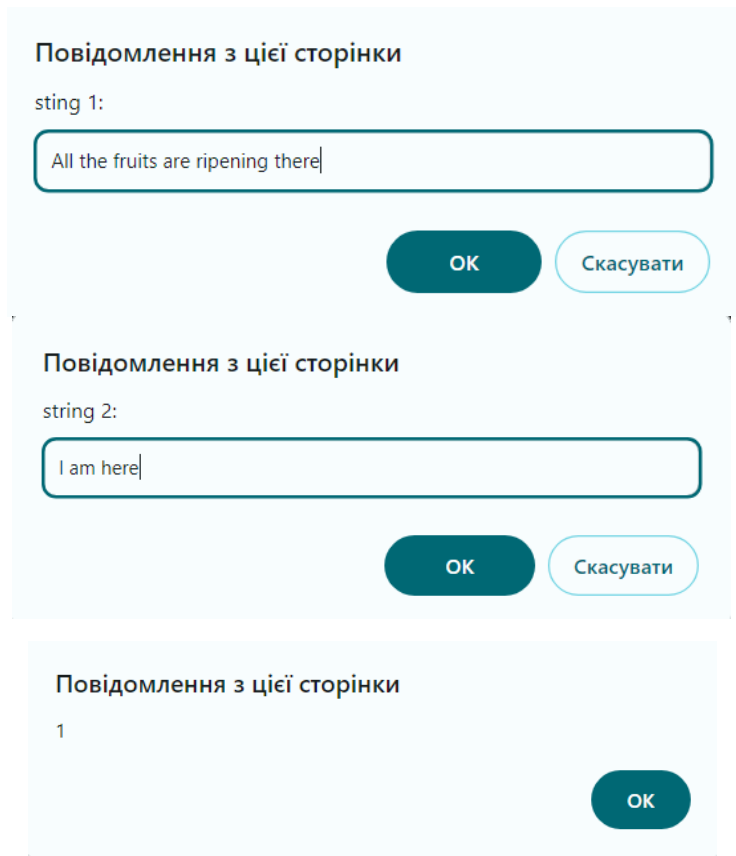


Рис. 1.1 Результат виконання функції

Напишіть функцію, яка переводить у верхній регістр перший символ переданого рядка.

```
function firstcap(str1){
    let first = str1.charAt(0).toUpperCase();
    let rem = str1.slice(1);
    return first + rem;
}

let s1 = prompt("string:")
alert(firstcap(s1))
```

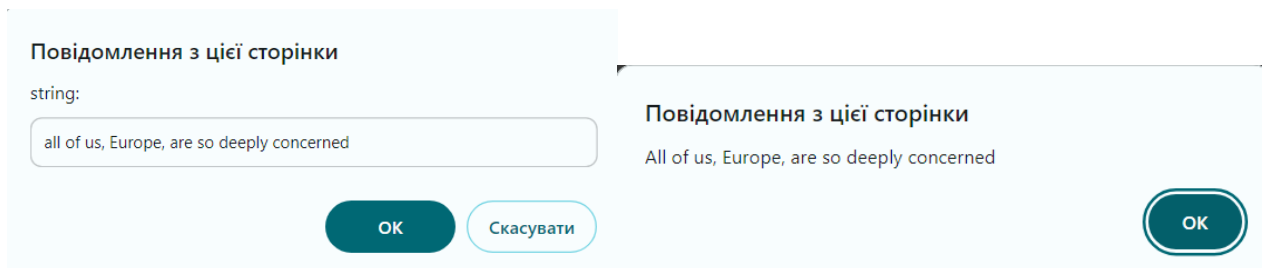


Рис. 1.2 Результат виконання функції

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.125.11.000 – Лр.3	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Напишіть функцію, яка підраховує кількість голосних літер у переданому рядку.

```
function numofvowels(str1){
    let vows = "AEIOUYaeoiuy"
    count = 0
    for (let i = 0; i <= str1.length; i++){
        for(let j = 0; j <= vows.length; j++){
            if (str1.charAt(i) == vows.charAt(j)){
                count += 1
            }
        }
    }
    return(count)
}

let s1 = prompt("string:")
alert(numofvowels(s1))
```

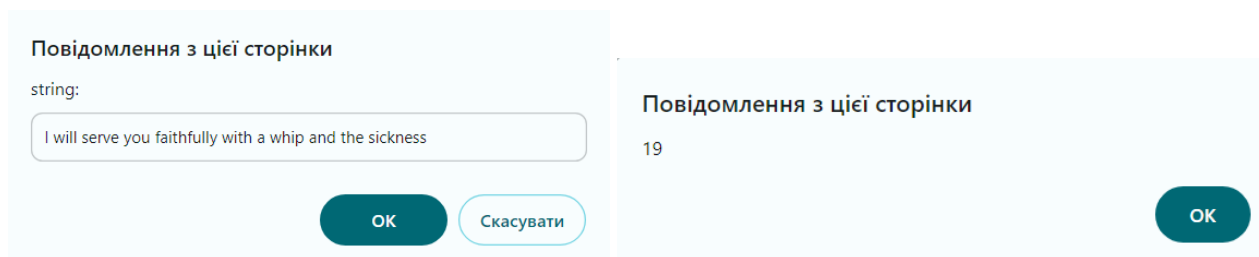


Рис. 1.3 Результат виконання функції

Напишіть функцію для перевірки спаму в переданому рядку. Функція повертає true, якщо рядок містить спам. Спамом вважати наступні слова: 100%безкоштовно, збільшення продажів, тільки сьогодні, не видаляйте, xxx.

```
function spamCheck(str1) {
    let spam = ["100%безкоштовно", "збільшення продажів", "тільки сьогодні", "не видаляйте", "xxx"];
    for (let i = 0; i < spam.length; i++) {
        let spamw = spam[i].toLowerCase();
        if (str1.toLowerCase().includes(spamw)) {
            return true;
        }
    }
    return false;
}

let s1 = prompt("string:")
alert(spamCheck(s1))
```

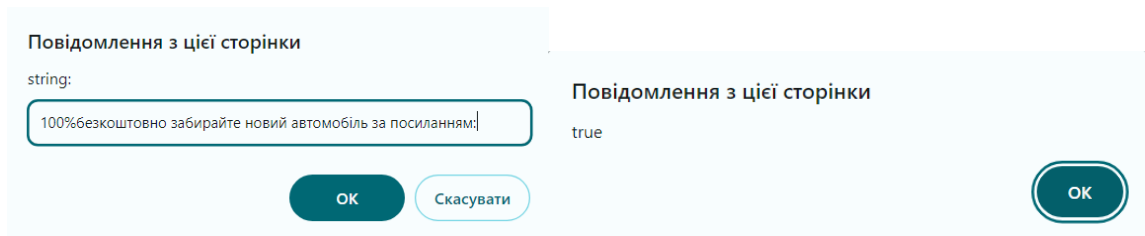


Рис. 1.4 Результат виконання функції

Напишіть функцію скорочення рядка. Функція приймає рядок та його максимальну довжину. Якщо довжина рядка більша, ніж максимальна, необхідно відкинути зайві символи, додавши замість них три крапки. Наприклад: `truncate("Hello, world!", 8)` має повернути `"Hello..."`.

```
function shortenString(str1, num){
  if (str1.length > num){
    return str1.slice(0, num) + "...";
  }
  else {
    return str1
  }
}

let s1 = prompt("string:")
let n = prompt("max length:")
alert(shortenString(s1, n))
```

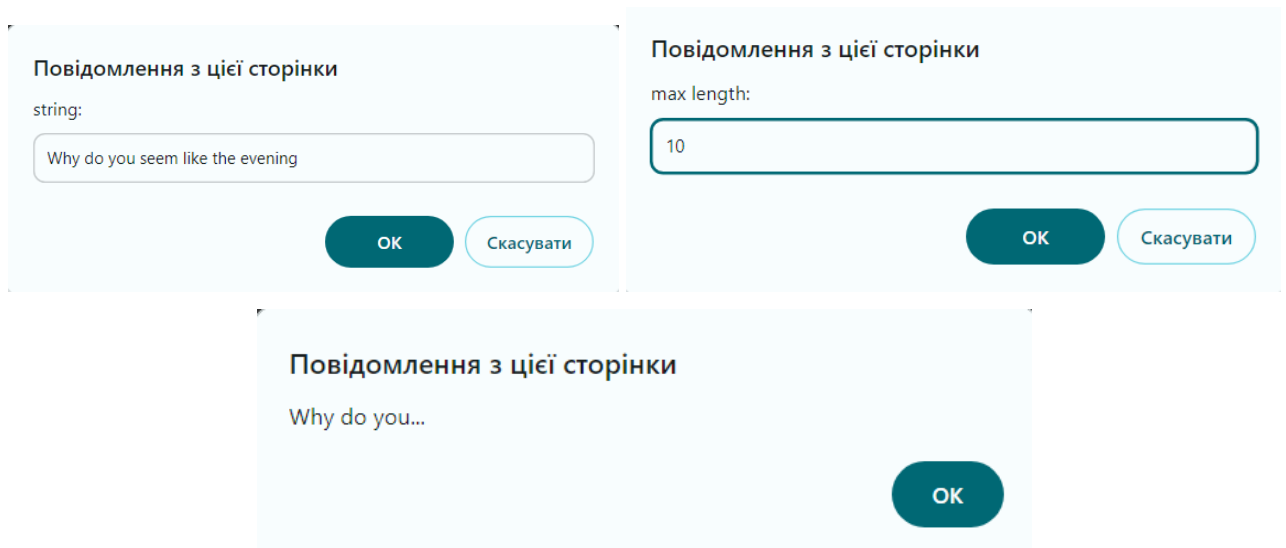


Рис. 1.5 Результат виконання функції

Напишіть функцію, яка приймає рядок і символ і виводить індекси, за якими знаходиться цей символ у рядку. Також виведіть, скільки разів зустрічається цей символ у рядку.

```
function symbol(str1, a){
    let indexes = ""
    count = 0
    for (let i = 0; i <= str1.length; i++){
        if (str1.charAt(i) == a){
            count += 1
            indexes += i + ", "
        }
    }
    return "indexes: " + indexes + "count: " + count
}

let s1 = prompt("string:")
let a = prompt("symbol:")
alert(symbol(s1, a))
```

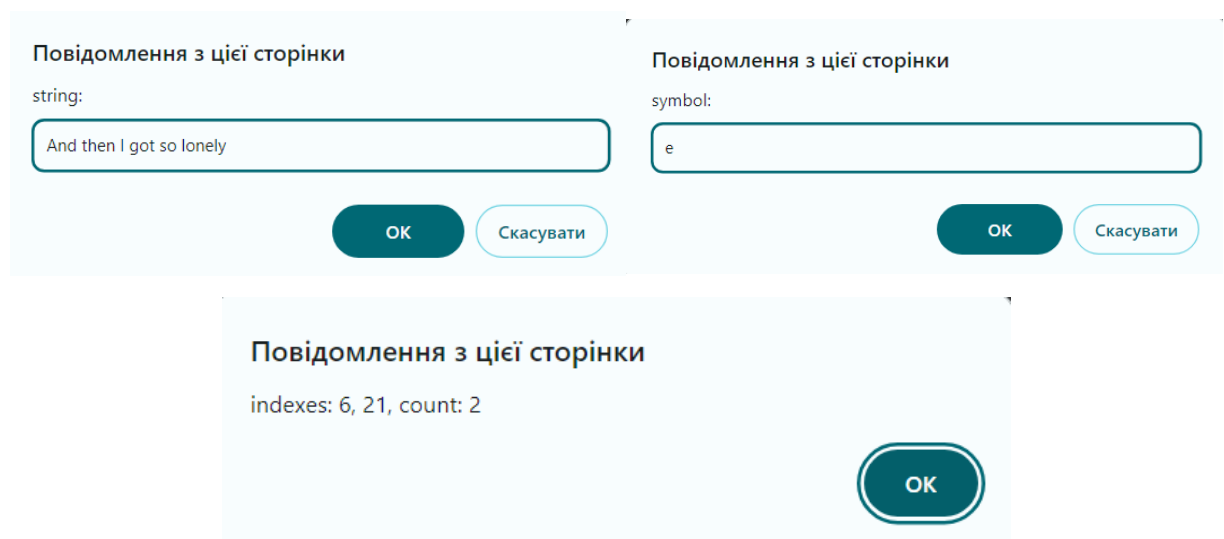


Рис. 1.6 Результат виконання функції

Створіть об'єкт, що описує час (години, хвилини, секунди), і наступні функції для роботи з цим об'єктом.

1. Функція виведення часу на екран.
2. Функція зміни часу на передану кількість секунд.
3. Функція зміни часу на передану кількість хвилин.
4. Функція зміни часу на передану кількість годин.

Враховуйте те, що в останніх 3 функціях при зміні однієї частини часу, може змінитися й інша. Наприклад: якщо до часу «20:30:45» додати 30 секунд, то має вийти «20:31:15», а не «20:30:75».

```
let time = {
  hours: 0,
  minutes: 0,
  seconds: 0,

  showtime : function(){
    let displayhours = this.addzero(this.hours);
    let displayminutes = this.addzero(this.minutes);
    let displayseconds = this.addzero(this.seconds);
    alert(displayhours + ":" + displayminutes + ":" + displayseconds);
  },

  addzero : function(num){
    if (num < 10){
      return "0" + num
    }
    else{
      return num
    }
  },
}

time.hours = parseInt(prompt("Hours:"));
time.minutes = parseInt(prompt("Minutes:"));
time.seconds = parseInt(prompt("Seconds:"));
time.showtime();
```

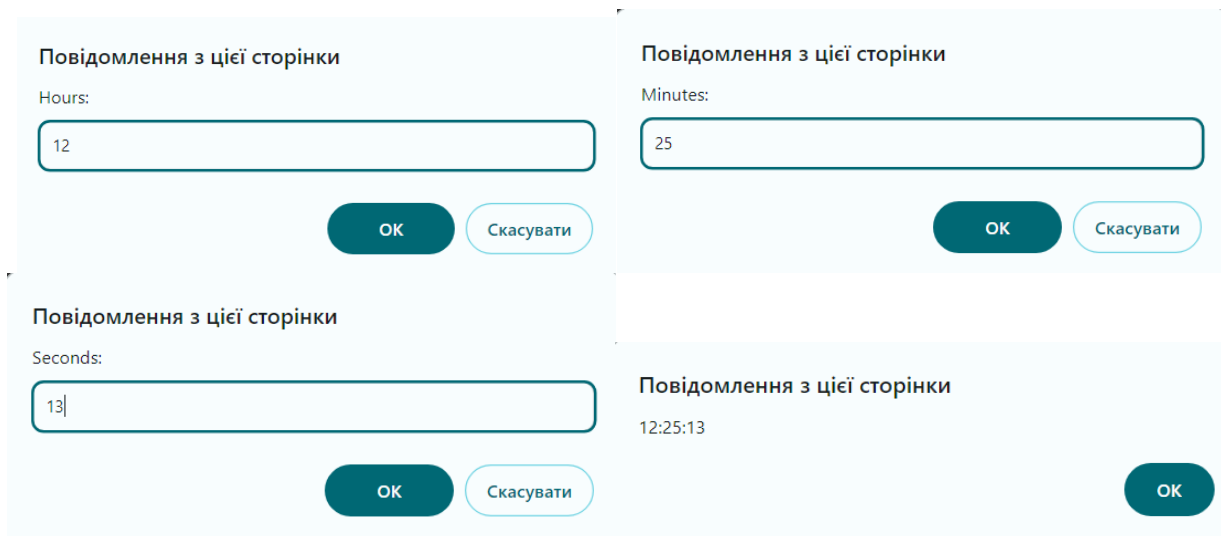


Рис. 2.1 Результат виконання функції виведення часу на екран

```

addseconds : function(seconds) {
    let totalSeconds = this.hours * 3600 + this.minutes * 60 + this.seconds +
seconds;
    this.hours = Math.floor(totalSeconds / 3600) % 24;
    this.minutes = Math.floor((totalSeconds % 3600) / 60);
    this.seconds = totalSeconds % 60;
},

adds = parseInt(prompt("Add seconds:"));
time.addseconds(adds);
time.showtime();

```

Повідомлення з цієї сторінки
Add seconds:

ОК

Скасувати

Повідомлення з цієї сторінки
12:26:13

ОК

Рис. 2.2 Результат виконання функції додавання секунд

```

addMinutes : function(minutes) {
    var date = new Date();
    date.setHours(this.hours);
    date.setMinutes(this.minutes + minutes);
    date.setSeconds(this.seconds);

    this.hours = date.getHours();
    this.minutes = date.getMinutes();
    this.seconds = date.getSeconds();
},

adds = parseInt(prompt("Add minutes:"));
time.addMinutes(adds);
time.showtime();

```

Повідомлення з цієї сторінки
Add minutes:

ОК

Скасувати

Повідомлення з цієї сторінки
13:10:13

ОК

Рис. 2.3 Результат виконання функції додавання хвилин

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.125.11.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

addHours : function(hours) {
    var date = new Date();
    date.setHours(this.hours + hours);
    date.setMinutes(this.minutes);
    date.setSeconds(this.seconds);

    this.hours = date.getHours();
    this.minutes = date.getMinutes();
    this.seconds = date.getSeconds();
}

time.hours = parseInt(prompt("Hours:"));
time.minutes = parseInt(prompt("Minutes:"));
time.seconds = parseInt(prompt("Seconds:"));

```

Рис. 2.4 Результат виконання функції додавання годин

Створіть масив «список покупок». Кожен елемент масиву є об'єктом, який містить назву продукту, необхідну кількість і чи був він куплений. Напишіть декілька функцій для роботи із таким масивом.

1. Виведення всього списку на екран таким чином, щоб спочатку йшли некуплені продукти, а потім куплені.
2. Додавання покупки до списку. Враховуйте, що при додаванні покупки з вже існуючим продуктом у списку, необхідно збільшувати кількість в існуючій покупці, а не додавати нову.
3. Купівля продукту. Функція приймає назву продукту і помічає його як придбаний.

```

let list = [];

function addItem(name, quantity) {
    let findIndex = list.findIndex(item => item.name === name);
    if (findIndex !== -1) {
        list[findIndex].quantity += quantity;
    } else {

```



```

let newItem = {
    name: name,
    quantity: quantity,
    bought: false
};
list.push(newItem);
}
alert(`${name} in number of ${quantity} is added`);
}

let f = true;
do {
    let name = prompt("name?");
    let q = parseInt(prompt("quantity?"));
    addItem(name, q);
    f = confirm("add another item?");
} while (f)

```

The figure illustrates the sequence of dialog boxes generated by the JavaScript code for adding items:

- Dialog 1:** Title "Повідомлення з цієї сторінки". Prompt "name?". Input: "pear". Buttons: "ОК", "Скасувати".
- Dialog 2:** Title "Повідомлення з цієї сторінки". Prompt "quantity?". Input: "4". Buttons: "ОК", "Скасувати".
- Dialog 3:** Title "Повідомлення з цієї сторінки". Message: "pear in number of 4 is added". Button: "ОК".
- Dialog 4:** Title "Повідомлення з цієї сторінки". Prompt "add another item?". Buttons: "ОК", "Скасувати".
- Dialog 5:** Title "Повідомлення з цієї сторінки". Prompt "name?". Input: "bread". Buttons: "ОК", "Скасувати".
- Dialog 6:** Title "Повідомлення з цієї сторінки". Prompt "quantity?". Input: "2". Buttons: "ОК", "Скасувати".
- Dialog 7:** Title "Повідомлення з цієї сторінки". Message: "bread in number of 2 is added". Button: "ОК".
- Dialog 8:** Title "Повідомлення з цієї сторінки". Prompt "name?". Input: "milk". Buttons: "ОК", "Скасувати".
- Dialog 9:** Title "Повідомлення з цієї сторінки". Prompt "quantity?". Input: "1". Buttons: "ОК", "Скасувати".
- Dialog 10:** Title "Повідомлення з цієї сторінки". Message: "milk in number of 1 is added". Button: "ОК".

Рис. 3.1 Результат роботи функції додавання продуктів

```
function display() {
    let output = "list:\n\n";
    let sortedlist = list.sort((a, b) => (a.bought === b.bought) ? 0 : a.bought ? 1 :
-1);
    sortedlist.forEach((item, index) => {
        output += `${index + 1}. ${item.name} - ${item.quantity} - ${item.bought ?
'bought' : 'not bought'}\n`;
    });
    alert(output);
}

display();
```

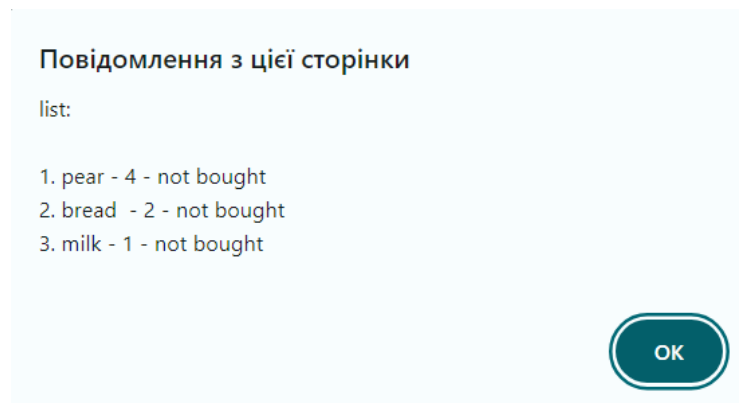


Рис. 3.2 Результат роботи функції виведення списку

```
function bought(name){
    let item = list.find(item => item.name === name);
    if (item) {
        item.bought = true;
        alert(`${item.name} is marked as bought`);
    } else {
        alert(`no such product in list`);
    }
}

f = false;
do {
    let name = prompt("name of bought item");
    bought(name);
    display();
    f = confirm("end shopping?");
} while (!f)
```

Повідомлення з цієї сторінки
name of bought item

bread

OK

Скасувати

Повідомлення з цієї сторінки
bread is marked as bought

OK

Рис. 3.3 Позначення продукту як купленого

Повідомлення з цієї сторінки
list:

1. pear - 4 - not bought  
2. milk - 1 - not bought  
3. bread - 2 - bought

OK

Рис. 3.4 Відсортований оновлений список

Повідомлення з цієї сторінки
end shopping?

OK

Скасувати

Рис. 3.5 Завершення покупки

Створіть масив, який описує чек у магазині. Кожен елемент масиву складається із назви товару, кількості та ціни за одиницю товару.

Напишіть наступні функції:

1. Виведення чека на екран.
2. Підрахунок загальної суми покупки.
3. Отримання найдорожчої покупки у чеку.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.125.11.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

#### 4. Підрахунок середньої вартості одного товару у чеку.

```
let check = [
  {name: "pear", quantity: 4, price: 23},
  {name: "bread", quantity: 2, price: 34},
  {name: "milk", quantity: 1, price: 50}
];

function total(){
  let total = 0;
  check.forEach(item => {total += item.price * item.quantity});
  return total
}

function exp(){
  let a = check.reduce((first, second) => {
    if (first.price > second.price){
      return first.name
    }
    else{
      return second.name
    }
  })
  return a;
}

function average(){
  let quantity = 0
  check.forEach((item) => {
    quantity += 1
  })
  return total() / quantity
}

function display() {
  let list = 'Your Check:\n';
  check.forEach(item => {
    list += `${item.name} - ${item.quantity} - ${item.price} money per item\n`;
  });
  list += "\nAverage: " + average() + " money";
  list += "\nMost expensive: " + exp() + "\n";
  list += "\nTotal: " + total() + "\n";
  alert(list);
}

display();
```

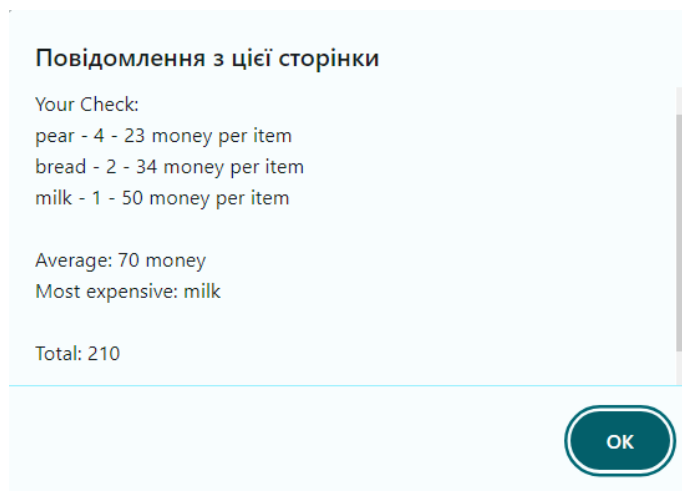


Рис. 3.5 Робота функцій для виведення чеку

**Висновок:** На цій лабораторній роботі ми вивчи навчилися створювати функції в мові JavaScript. Ознайомилися з методами для обробки масивів. Закріпили навички роботи з рядками.