

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

Практична робота №4  
з дисципліни «WEB-програмування JS»  
за темою: Робота з структурованими даними (об'єктами)

Виконав: 12.04.2020 \_\_\_\_\_ ст. гр. ПД-22 О.І. Мидинський  
(дата, підпис)

Перевірів: \_\_\_\_\_ викладач. В.О. Яскевич  
(дата, підпис)

Оцінка \_\_\_\_\_

Київ - 2020

## Мета роботи:

- закріпити знання про об'єкти користувача мови JavaScript;
- придбати практичні навички використання об'єктів, як структур даних.

### Завдання 1. Кількість об'єктів

---

#### Кількість об'єктів

Дописати у вікні нижче функцію, яка приймає на вхід масив об'єктів. Кожен об'єкт має два поля: *x* та *y*. Функція повинна

**Наприклад:**

Вхідні дані: [{x: 1, y: 1}, {x: 2, y: 3}, {x: 3, y: 3}, {x: 3, y: 4}, {x: 4, y: 5}]

Результат: 2

```
1 function getCount(objects){  
2     var count = 0;  
3  
4     for (var i = 0; i < objects.length; i++)  
5     {  
6         if (objects[i].x == objects[i].y) count++;  
7     }  
8  
9     return count;  
10 }  
11
```

chrome ▼ 14

OK

OK

OK

## Завдання 2. Shopping cart

```
// 10 + 10 + 50 = 70
```

```
1 function totalCost(goods, discounts) {  
2     var sum = 0;  
3  
4     for (var i = 0; i < goods.length; i++)  
5     {  
6         var flag = true;  
7         for (var j = 0; j < discounts.length; j++)  
8         {  
9             if (goods[i].name == discounts[j].name)  
10            {  
11                sum += (goods[i].value - goods[i].value * (parseInt(discounts[j].discount.slice(0, discounts[j].discount.indexOf('%')) / 100)) * goods[i].amount;  
12                flag = false;  
13            }  
14        }  
15        if (flag) sum += goods[i].value * goods[i].amount;  
16    }  
17  
18    return sum;  
19 }
```

chrome 14 | Перевірити

OK

OK

OK

OK

```
function totalCost(goods, discounts) {  
    var sum = 0;  
    for (var i = 0; i < goods.length; i++)  
    {  
        var flag = true;  
        for (var j = 0; j < discounts.length; j++)  
        {  
            if (goods[i].name == discounts[j].name)  
            {  
                sum += (goods[i].value - goods[i].value * (parseInt(discounts[j].discount.slice(0, discounts[j].discount.indexOf('%')) / 100)) * goods[i].amount;  
                flag = false;  
            }  
        }  
        if (flag) sum += goods[i].value * goods[i].amount;  
    }  
    return sum;  
}
```

### Завдання 3. Дужки (простий варіант)

```
1 function validSequenceSimple(braces){  
2     var open = 0, close = 0;  
3     for (var i = 0; i < braces.length; i++)  
4     {  
5         if (braces[i] == '(') open++;  
6         else if (braces[i] == ')') close++;  
7         if (close > open) return false;  
8     }  
9     if (open == close) return true;  
10    else return false;  
11 }  
12
```

chrome 14

OK

OK

OK

OK

OK

OK

OK

OK

OK

OK

OK

```
function validSequenceSimple(braces){  
    var open = 0, close = 0;  
    for (var i = 0; i < braces.length; i++)  
    {  
        if (braces[i] == '(') open++;  
        else if (braces[i] == ')') close++;  
        if (close > open) return false;  
    }  
    if (open == close) return true;  
    else return false;  
}
```

## Завдання 4. Дужки

### Дужки

Дописати у вікні нижче функцію, яка перевіряє чи правильна послідовність дужок на вході.

Види дужок: () {} [] <>

**Результат роботи:** true, якщо вхідний рядок містить правильну дужкову послідовність; або false, якщо послідовність є неправильною. Дужка закривається після відкриваючої.

```
14         if (obj[obj.length - 1] == '{') obj.pop();
15         else return false;
16     }
17     else if (braces[i] == ']')
18     {
19         if (obj[obj.length - 1] == '[') obj.pop();
20         else return false;
21     }
22     else if (braces[i] == '>')
23     {
24         if (obj[obj.length - 1] == '<') obj.pop();
25         else return false;
26     }
27 }
28
29 if (obj.length == 0) return true;
30 else return false;
31 }
32
```

chrome 14

OK

OK

OK

OK

OK

OK

OK

OK

OK

OK

OK

OK

```
function totalCost(braces) {
    var obj = [];

    for (var i = 0; i < braces.length; i++)
    {
        if (braces[i] == '(' || braces[i] == '{' || braces[i] == '<' || braces[i] == '[') obj.push(braces[i]);
        else if (braces[i] == ')')
        {
            if (obj[obj.length - 1] == '(') obj.pop();
            else return false;
        }
    }
}
```

```

    }
    else if (braces[i] == '}')
    {
        if (obj[obj.length - 1] == '{') obj.pop();
        else return false;
    }
    else if (braces[i] == ']')
    {
        if (obj[obj.length - 1] == '[') obj.pop();
        else return false;
    }
    else if (braces[i] == '>')
    {
        if (obj[obj.length - 1] == '<') obj.pop();
        else return false;
    }
}

if (obj.length == 0) return true;
else return false;
}

```

Відповіді на запитання:

1. Об'єкт - це набір властивостей, і кожне властивість складається з імені та значення, асоційованого з цим ім'ям.
2. Об'єкт може бути створений за допомогою фігурних дужок {...} з необов'язковим списком *властивостей*.
3. Асоціативний масив - це масив, у якого в якості ключів використовуються рядки.
4. Властивість - це пара «ключ: значення», де ключ - це рядок (також звана «ім'ям властивості»), а значенням може бути що завгодно.
7. Сам об'єкт зберігається десь в пам'яті. А в імені об'єкта лежить «посилання» на цю область пам'яті.
8. Якщо ми копіюємо за значенням, то створюємо новий об'єкт, ідентичний оригіналу.
9. Коли відбувається копіювання за посиланням, то створюється посилання на оригінал.
10. Коли змінна об'єкта копіюється - копіюється посилання, сам же об'єкт не дублюється. Якщо ми представляємо об'єкт як ящик, то змінна - це ключ до нього. Копіювання змінної дублює ключ, але не сам ящик.