

Об'єкти, масиви, формат JSON в JavaScript

Анна Шавурська
QAP INT

JavaScript - типізація

- Слабо чи сильно типізований?
- Зі статичним чи динамічним виділенням пам'яті?

- На які дві групи поділяються типи в JavaScript?

Типы в JavaScript

Simple (primitive) types

- Undefined,
- Null,
- Boolean,
- Number,
- String

Complex (reference) data type

- Object

Всього 6 типів

- Ви не маєте можливості створювати свої типи даних.
- Можливо, цього замало?

Об'єкти

•Що таке об'єкт?



Об'єкт в JavaScript

- **Об'єкт** –
- це сутність в пам'яті
- що володіє станом і поведінкою
- це не впорядкований список **name:value** пар:
 - name – string
 - value – будь-який тип, включаючи Object

Об'єкт в JavaScript

- Об'єкти – це екземпляри Object типу даних або його різновидностей.
- Об'єкти можуть містити дані і методи.
- Об'єкти можуть наслідуватись від інших об'єктів.

Native reference types

- Object,
- Array,
- Date,
- RegExp,
- ...

Object

OBJECT TYPE

- Тип, що найчастіше використовується.

ОБЈЕСТ TYPE

- Екземпляри цього типу не мають багато функціональності, але вони ідеально підходять для збереження і передання даних.

Створення нового об'єкту при допомозі object literal notation

```
var person = {  
  name : "Anton",  
  age : 30,  
  sayHello: function(){  
    alert("Hello world!");  
  }  
};
```

{ expression context }

{

key1: value1,

key2: value2

}

Доступ до полів об'єкта

1) dot notation

```
person.name = 'Anton';
```

2) bracket notation

```
person['name'] = 'Anton';
```

Створення нового об'єкту при допомозі `new Object()`

```
var person = new Object();  
person.name = "Anton";  
person.age = "30";
```


Constructor

- Це функція, ціллю якої є створити новий об'єкт.
- Це ім'я типу об'єкта, який створюємо.
- Object – конструктор.

Constructor - Приклад

```
function Person(name, age){  
    this.name = name;  
    this.age = age;  
}
```

Видалення полів об'єкту - delete

`delete object.property`

`delete object[property]`

Secret Linkage

- При створенні об'єкт отримує секретне посилання на інший об'єкт (батьківський об'єкт).
- Якщо властивість не знайдена в самому об'єкті, пошук автоматично буде здійснюватись в батьківському об'єкті.
- За замовчуванням об'єкти отримують посилання на об'єкт типу Object.

for ... in loop

```
for (key in object) {  
    object[key] = value;  
}
```

Цикл по всім **злічуваним** властивостям об'єкта, включаючи унаслідовані від батьківських об'єктів.

Object.keys(obj)

- Повертає масив всіх злічуваних властивостей об'єкта у вигляді строк. **Не включає** властивості батьківських об'єктів.

Властивості Object

- toLocaleString()
- toString()
- valueOf()
- constructor
- hasOwnProperty(propertyName)
- isPrototypeOf(object)
- propertyIsEnumerable(propertyName)

Качина типізація - *Duck typing*



- *«If it looks like a duck, swims like a duck and quacks like a duck, then it probably is a duck».*

Завдання

- Створити функцію, яка приймає об'єкт та пару: ключ і значення, та додає нове поле в переданий об'єкт (за відповідним ключем), якому буде присвоєне передане значення. Якщо в об'єкті вже було передане поле, то його не потрібно перезаписувати новим значенням.
- Двома різними методами перевернути і вивести всі поля об'єкта.

Завдання

- Написати функцію, що імплементує функціональність `Object.assign()`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/assign

МАСИВИ



Масив

- Що це таке?

Масив в С

- це структура даних, представлена в вигляді комірок **ОДНОГО** типу, об'єднаних під одним іменем.

Масив в JavaScript

- це структура даних, представлена в вигляді комірок **БУДЬ-ЯКОГО** типу, об'єднаних під одним іменем.

Приклади створення масиву

- `var colors = ["red", "blue", "green"];`
- `var primitives = [5, "Anton", false];`
- `var anything = [135,
 {name: "Anton", age: 30},
 function(){ alert("Hello world")}]`

Синтаксис створення масиву

1) Array конструктор:

```
var arrayName = new Array();
```

```
var arrayName = new Array(numberOfItems);
```

```
var arrayName = new Array(item1, item2, item3);
```

2) array literal notation

```
var arrayName = [];
```

```
var arrayName = [item1, item2, item3];
```

Доступ до комірок масиву

```
arrayName[index] = item;
```

Приклад:

```
colors[2] = "black";
```

Розмір масиву

```
arrayName.length; // get number of items  
arrayName.length = numberOfItems // set
```

Приклад:

```
var colors = ["red", "blue", "green"];  
colors.length = 2;  
alert(colors[2]); //undefined
```

Максимальна кількість комірок

4 294 967 295

Arrays

- Array унаслідований від Object.
- Індеси конвертуються в рядки і використовуються для пошуку значень.
- Можна створювати розріджені масиви.
- Не потрібно вказувати довжину чи тип при створенні.

Перевірка чи змінна є масивом

```
if (value instanceof Array){  
    //do something on the array  
}
```

```
if (Array.isArray(value)){  
    //do something on the array  
}
```

```
if (value.constructor === Array){  
    //do something on the array  
}
```

```
typeof value // “object”
```

Методи конвертації

- `toLocaleString()`;
- `toString()`;
- `valueOf()`;

Методи стеку та черги

```
var a = [2, 3, 5], item;  
a.push(1);           // [2, 3, 5, 1]  
item = a.pop();     // [2, 3, 5]  
item = a.shift();   // [3, 5]  
a.unshift(6, 10);  // [6, 10, 3, 5]
```


Методи перестановки

```
values.sort(compare);  
values.reverse()
```

```
function compare(value1, value2) {  
  if (value1 < value2) {  
    return -1;  
  } else if (value1 > value2) {  
    return 1;  
  } else {  
    return 0;  
  }  
}
```

Методи маніпуляції

```
var colors = ["red", "green", "blue"];  
var colors2 = colors.concat("yellow", ["black",  
"brown"]);
```

```
alert(colors); //red,green,blue
```

```
alert(colors2);
```

```
//red,green,blue,yellow,black,brown
```

Методи маніпуляції

```
var colors = ["red", "green", "blue", "yellow",  
"purple"];
```

```
var colors2 = colors.slice(1);
```

```
var colors3 = colors.slice(1, 4);
```

```
var colors4 = colors.slice(-2, -1);
```

```
alert(colors2); //green,blue,yellow,purple
```

```
alert(colors3); //green,blue,yellow
```

```
alert(colors4); //yellow
```

Методи маніпуляції (!= delete)

```
var colors = ["red", "green", "blue"];  
var removed = colors.splice(0,1); //remove the first item  
alert(colors); //green,blue  
alert(removed); //red - one item array
```

```
removed = colors.splice(1, 0, "yellow", "orange"); //insert two  
items at position 1  
alert(colors); //green,yellow,orange,blue  
alert(removed); //empty array
```

```
removed = colors.splice(1, 1, "red", "purple"); //insert two  
values, remove one  
alert(colors); //green,red,purple,orange,blue  
alert(removed); //yellow - one item array
```

Визначення позиції елемента

```
var numbers = [1,2,3,4,5,4,3,2,1];
```

```
alert(numbers.indexOf(4)); //3
```

```
alert(numbers.lastIndexOf(4)); //5
```

Методи перебору елементів

- every()
- filter()
- forEach()
- map()
- some()

Reduction methods

```
var values = [1,2,3,4,5];  
var sum = values  
    .reduce(function(prev, cur, index, array){  
        return prev + cur;  
    });  
alert(sum); //15
```

```
var values = [1,2,3,4,5];  
var sum = values  
    .reduceRight(function(prev, cur, index, array){  
        return prev + cur;  
    });  
alert(sum); //15
```

Завдання

1. Створити третій масив з унікальних елементів, що зустрічаються хоча б в одному з двох інших масивів.
2. Вивести парні числа з додатніх елементів масиву.
3. Вивести будь-яке повідомлення, якщо всі елементи масиву – масиви.
4. Вивести будь-яке повідомлення, якщо хоча б один елемент масиву дорівнює нулю.
5. Вивести індекс елемента масиву, значення якого = 5.
6. Обрахувати добуток всіх елементів масиву.
7. Відсортувати елементи масиву:
 - а) в алфавітному порядку,
 - б) чисельний масив, не беручи до уваги знак.



{JSON}

JSON - JavaScript Object Notation

- Легковісний, текстовий, незалежний від мови формат обміну даними.
Douglas Crockford, 2006.

<http://json.org/>

JSON - формат даних

- JSON – це не мова програмування.
- JSON – це не частина JavaScript.
- Парсери JSON існують також в багатьох інших мовах програмування.



JSON is all about representing

Приклад

```
{
  "books": [
    {
      "title": "Professional JavaScript",
      "authors": [
        "Nicholas C. Zakas"
      ],
      "edition": 3,
      "year": 2011
    },
    {
      "title": "Professional JavaScript",
      "authors": [
        "Nicholas C. Zakas"
      ],
      "edition": 2,
      "year": 2009
    }
  ]
}
```

Типи даних, що можуть бути представлені в форматі JSON

1. Simple Values: Strings, Numbers, Booleans, null
2. Objects
3. Arrays

Для опису цих типів даних використовується літерал, а не конструктор.

JSON не підтримує

- Undefined
- Змінні
- Функції

Різниця з синтаксисом JavaScript

- Number
 - Boolean
 - Null
 - Strings – лише “ ”, а не ‘ ’
- не відрізняється
-

Різниця з синтаксисом JavaScript

- Objects:

Назви властивостей в лапках:

```
{  
  "name": "Nicholas",  
  "age": 29  
}
```

Парсинг JSON в порівнянні з XML

JSON

- Парситься в JavaScript об'єкт.
- `books[2].title`

XML

- Парситься в DOM.
- `doc.getElementsByTagName("book")[2].getAttribute("title")`

JSON - натівний глобальний об'єкт

- Об'єкт для роботи з даними у форматі JSON.
- Методи:
 - `JSON.stringify()`,
 - `JSON.parse()`.

JSON. stringify(data, filter, separator)

1. Без параметрів

```
• var book = {  
    title: "Professional JavaScript",  
    authors: [ "Nicholas C. Zakas" ],  
    edition: 3,  
    year: 2011  
};
```

```
var jsonText = JSON.stringify(book);
```

```
"{"title": "Professional  
JavaScript", "authors": ["Nicholas C.  
Zakas"], "edition": 3, "year": 2011}"
```

JSON. stringify(data, filter, separator)

- 2. 1 Filter: array | function

```
var jsonText = JSON.stringify(book,  
  ["title", "edition"]);  
  
{  
  "title": "Professional JavaScript",  
  "edition": 3  
};
```

JSON. stringify(data, filter, separator)

2.2 Filter: array | function

```
var jsonText = JSON.stringify(book, function(key, value) {
    switch (key) {
        case "authors":
            return value.join(",");
        case "year":
            return 5000;
        case "edition":
            return undefined;
        default:
            return value;
    }
});
{"title":"Professional JavaScript","authors":"Nicholas C.
Zakas","year":5000})
```

JSON. stringify(data, filter, separator)

3. Separator: number | string

```
var jsonText = JSON.stringify(book, null, 4);  
{  
  "title": "Professional JavaScript",  
  "authors": [  
    "Nicholas C. Zakas"  
  ],  
  "edition": 3,  
  "year": 2011  
}
```

JSON. stringify(data, filter, separator)

3. Separator: number | string

```
•var jsonText = JSON.stringify(book,  
  null, "  -  ");  
{  
  --"title": "Professional JavaScript",  
  --"authors": [  
    ---"Nicholas C. Zakas"  
  --],  
  --"edition": 3,  
  --"year": 2011  
}
```


toJSON()

- Метод для налаштування серіалізації об'єкта.

```
var book = {  
    "title": "Professional JavaScript",  
    "authors": [  
        "Nicholas C. Zakas"  
    ],  
    edition: 3,  
    year: 2011,  
    toJSON: function () {  
        return this.title;  
    }  
};  
var jsonText = JSON.stringify(book);
```

JSON.parse(jsonText, reviver)

```
var bookCopy = JSON.parse(jsonText);
```

JSON.parse(jsonText, reviver)

- Reviver: function

```
var book = {  
  "title": "Professional JavaScript",  
  "authors": [  
    "Nicholas C. Zakas"  
  ],  
  edition: 3,  
  year: 2011,  
  releaseDate: new Date(2011, 11, 1)  
};  
var jsonText = JSON.stringify(book);
```

JSON.parse(jsonText, reviver)

- Reviver: function

```
var bookCopy = JSON.parse(jsonText,  
    function (key, value) {  
        if (key == "releaseDate") {  
            return new Date(value);  
        } else {  
            return value;  
        }  
    }  
);  
bookCopy.releaseDate.getFullYear();
```

Завдання

- Створити об'єкт галереї, що буде містити перелік картинок. Кожна картинка описана об'єктом: ім'я, шлях до картинки та дата додавання. Серіалізувати об'єкт галереї в формат JSON такими способами (а результат вивести в консоль):
 - Зберегти всю інформацію.
 - Так, щоб зберегти лише імена картинок.
 - Таким чином, що якщо картинка не має імені, то не зберігати її взагалі.
- Розпартити сереалізовані об'єкти та вивести їх в консоль таким чином, щоб дата була об'єктом Date, а не строкою.
- <http://jsfiddle.net/AnnaShavurska/7Lnxsjzg/5/>

ПИТАННЯ?



Нове Закарпаття

Література

1. Professional JavaScript™ for Web Developers -
Nicholas C. Zakas
2. Douglas Crockford: The JavaScript Programming
Language
<https://www.youtube.com/watch?v=v2ifWcnQs6M&list=PL62E185BB8577B63D>