

«Массивы, циклы, JSON»



JavaScript
Courses

vk.com/js.courses

js.courses.dp.ua/files

Массивы, когда переменных не хватает...

```
var a = [456, "lalala", 12.78, true];
```

Массивы – упорядоченный, сгруппированный набор элементов.

```
a = [ "a", "b", "c", "d", "e" ]
```

↑
0

↑
1

↑
2

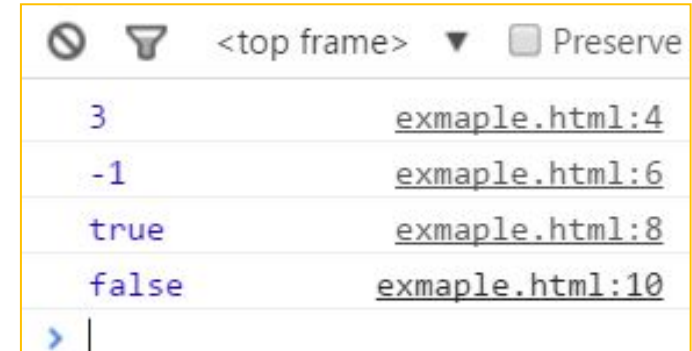
↑
3

↑
4



Массивы это коллекция пар «ключ => значение».

```
1 <script>
2   var arr = [1, 77, "Elena", 55.6, true];
3
4   console.log(arr.indexOf(55.6));
5
6   console.log(arr.indexOf("lalala"));
7
8   console.log(3 in arr);
9
10  console.log(7 in arr);
11 </script>
```



Value	Source
3	example.html:4
-1	example.html:6
true	example.html:8
false	example.html:10

```
0 => 1;
1 => 77;
2 => "Elena";
3 => 55.6;
4 => true;

arr.length == 5
```

В классических массива все ключи – числа, как правило нумерация начинается с 0, и идёт без пропусков.

Ассоциативные массивы



*Массивы это хранилища пар «ключ=>значение»,
но ключом может выступать не только целые
числа.*

Ассоциативные массивы

```
1 <script>
2
3   var mas = [];
4
5   mas["name"]      = "Ivan";
6   mas["age"]       = 19;
7   mas["height"]    = 1.81;
8   mas["smoke"]     = false;
9
10  console.log(mas);
11
12  console.log(mas.name, mas["name"]);
13
14  console.log(mas.length);
15
16  console.log(Object.keys(mas));
17
18  console.log(Object.keys(mas).length);
19
20  mas.push("55");
21
22  console.log(mas, mas.length);
23
24 </script>
```

```
exmaple.html:10
[name: "Ivan", age: 19, height: 1.81, smoke:
false]
Ivan Ivan                                exmaple.html:12
0                                         exmaple.html:14
["name", "age", "height", "smoke"]      exmaple.html:16
4                                         exmaple.html:18
exmaple.html:22
["55", name: "Ivan", age: 19, height: 1.81, smoke:
false]
1
```

Массивы это хранилища пар «ключ=>значение», но ключом может выступать не только целые числа.

Цикл for/in

```
1 <script>
2
3   var mas = [6, 8, 33, "text", true, 77.8];
4
5   for(var x in mas){
6     console.log(x, mas[x]);
7   }
8
9 </script>
```

0	6	exmaple.html:6
1	8	exmaple.html:6
2	33	exmaple.html:6
3	text	exmaple.html:6
4	true	exmaple.html:6
5	77.8	exmaple.html:6

Цикл for/in позволяет перебрать ключи массива не заворачиваясь с их количеством.

Цикл for/in и ассоциативные массивы

```
1 <script>
2
3     var mas = [];
4
5     mas["name"] = "Elena";
6     mas.age     = 18;
7     mas.smoke   = false;
8
9     for(var x in mas){
10         console.log(x, mas[x]);
11     }
12
13 </script>
```

name Elena	exmaple.html:10
age 18	exmaple.html:10
smoke false	exmaple.html:10

Цикл for/in позволяет перебрать ключи массива не заворачиваясь с их количеством.

Цикл for/in и свойства HTML элементов


```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <div>Text</div>
5 <script>
6   var div = document.querySelector("div");
7
8   for(var key in div){
9     console.log(key + ": " + div[key]);
10  }
11
12 </script>
13 </body>
14 </html>
```

tagName: DIV	ex1.html:9
id:	ex1.html:9
className:	ex1.html:9
classList:	ex1.html:9
attributes: [object NamedNodeMap]	ex1.html:9
innerHTML: Text	ex1.html:9
outerHTML: <div>Text</div>	ex1.html:9
shadowRoot: null	ex1.html:9
scrollTop: 0	ex1.html:9
scrollLeft: 0	ex1.html:9
scrollWidth: 403	ex1.html:9

Цикл for/in хороший инструмент для перебора свойств HTML элементов.

Объекты как ассоциативный массив

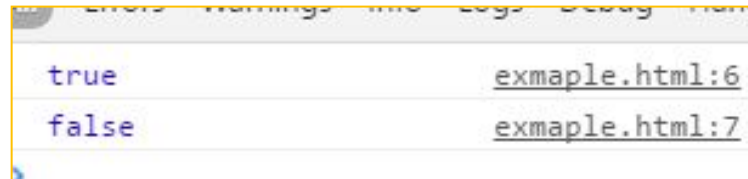
```
1 <script>
2
3   var mas = {name: "Ivan", age: 22};
4   console.log(mas);
5
6   for(var x in mas){
7       console.log(x, mas[x]);
8   }
9
10  mas["name"] = "John";
11  console.log(mas.name);
12
13  mas.push(55);
14
15 </script>
```

Object {name: "Ivan", age: 22}	exmaple.html:5
name Ivan	exmaple.html:8
age 22	exmaple.html:8
John	exmaple.html:13
 Uncaught TypeError: mas.push is not a function	exmaple.html:15

Объекты в JavaScript также можно использовать как ассоциативный массив, но в таком случае не будут доступны методы-помощники.

Как отличить массив от объекта

```
1 <script>
2
3   var arr1 = [1,2,3];
4   var arr2 = { title: "the news", year: 1999 };
5
6   console.log(Array.isArray(arr1));
7   console.log(Array.isArray(arr2));
8
9 </script>
```

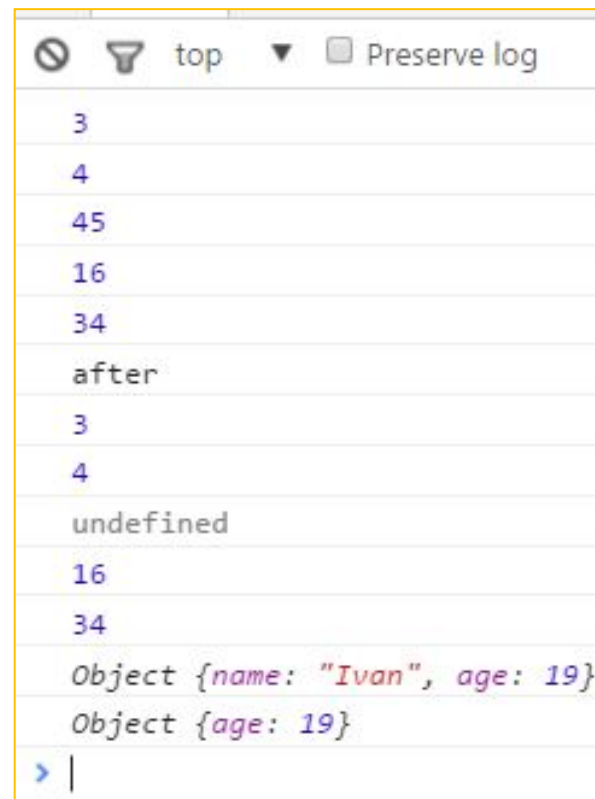


Message	Source
true	exmaple.html:6
false	exmaple.html:7

*Метод **Array.isArray()** возвращает **true** если полученный объект является классическим массивом **[]**, и **false** во всех остальных случаях.*

Удаление элементов массива

```
1 <script>
2
3   var arr = [3,4,45,16,34];
4
5   for(var i = 0; i < arr.length; i++){
6     console.log(arr[i]);
7   }
8
9   delete arr[2];
10  console.log("after");
11
12  for(var i = 0; i < arr.length; i++){
13    console.log(arr[i]);
14  }
15
16  //-----//
17
18  var mas = { name: "Ivan", age: 19 };
19
20  console.log(mas);
21
22  delete mas.name;
23
24  console.log(mas);
25
26 </script>
```



top	Preserve log
3	
4	
45	
16	
34	
after	
3	
4	
undefined	
16	
34	
Object {name: "Ivan", age: 19}	
Object {age: 19}	
>	

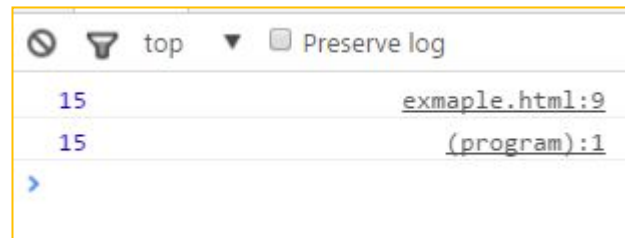
Удаление элементов массива по сути образует «дырки».

Задача

Разработать скрипт, проверяющий знания (умение) таблицы умножения двузначных чисел. Скрипт должен задать пользователю 12 задач на умножение двузначных чисел. По результатам проверки, пользователю выставляется оценка, а также выводиться два списка: верных ответов, и ошибочных ответов, указанием какой ответ был правильный.

eval

```
1 <script>
2
3   var a = 8;
4
5   var b = 7;
6
7   eval("var c = a + b;");
8
9   console.log(c);
10
11  eval("console.log(c);");
12
13 </script>
```



eval() выполняет JavaScript код, представленный строкой.

**Категорически не рекомендуется
использовать!!!**

JSON <http://www.json.org/json-ru.html>

```
1 <script>
2
3   var message = '{"name": "Vasya", "age": "19", "languages": ["English","Russian","French"] }';
4
5   var person = JSON.parse(message);
6
7   for(var x in person){
8       console.log(x + ": " + person[x]);
9   }
10
11   console.log(person);
12
13 </script>
```

name: Vasya	example.html:8
age: 19	example.html:8
languages: English,Russian,French	example.html:8
▼ Object {name: "Vasya", age: "19", languages: Array[3]}	example.html:11
age: "19"	
▶ languages: Array[3]	
name: "Vasya"	
▶ __proto__: Object	

Декодирование из JSON

JSON - текстовый формат текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. А по простому, это текстовый формат передачи массивов и объектов в JS.

JSON <http://www.json.org/json-ru.html>

```
1 <script>
2
3   var arr = [3, 67, "Computer", 'IFC', true, null, {title:"Harry Potter", autor: "J.K.Rowling"}, 768.22];
4
5   console.log(arr);
6
7   var text_format = JSON.stringify(arr);
8
9   console.log(text_format);
10
11 </script>
```

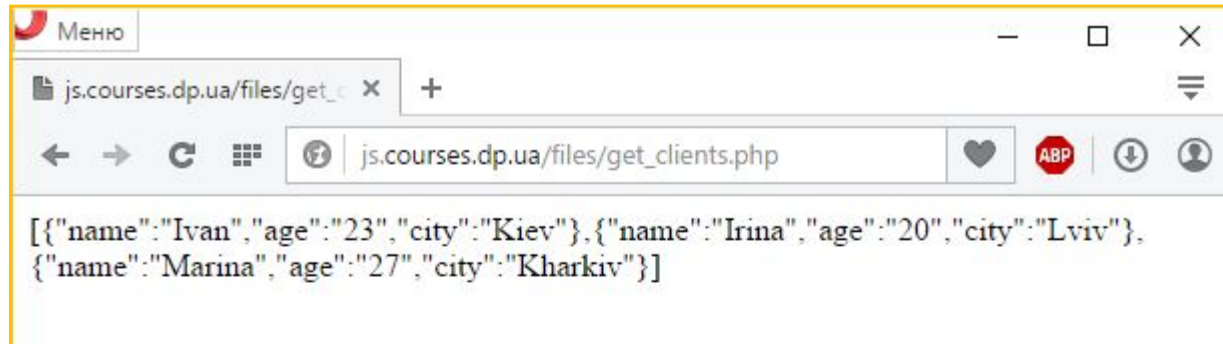
Кодирование в
JSON

```
▶ [3, 67, "Computer", "IFC", true, null, Object, 768.22]      example.html:5
[3,67,"Computer","IFC",true,null,{"title":"Harry Potter","autor":"J.K.Rowling"},768.22]  example.html:9
```

JSON - текстовый формат текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. А по простому, это текстовый формат передачи массивов и объектов в JS.

JSON в реальности

http://js.courses.dp.ua/files/get_clients.php



```
1 <?php
2     header('Access-Control-Allow-Origin: *');
3
4     $arr = array(
5         array("name" => "Ivan", "age" => "23", "city" => "Kiev"),
6         array("name" => "Irina", "age" => "20", "city" => "Lviv"),
7         array("name" => "Marina", "age" => "27", "city" => "Kharkiv")
8     );
9
10    echo json_encode($arr);
11
12 >?
```

Сценарий на PHP кодирует данные в формат JSON и возвращает строку с закодированными данными вместо страницы при запросе на адрес:

http://js.courses.dp.ua/files/get_clients.php . А наш JS код

декодирует данные и наполняет ими страницу.

JSON в реальности

```
1 <html>
2 <head></head>
3 <body>
4 <h1>Clients</h1>
5 <div id="clients"></div>
6 <script>
7     var XHR = new XMLHttpRequest();
8     XHR.onload = function() {
9
10        var data = XHR.responseText;
11        console.log("Recieve data: " + data);
12
13        var clients_arr = JSON.parse(XHR.responseText);
14        console.log(clients_arr);
15
16        var clients_div = document.getElementById("clients");
17
18        for(var id in clients_arr) {
19            var new_div = document.createElement("div");
20
21            for(var field in clients_arr[id]) {
22                new_div.innerHTML += "<b>" + field + "</b>: " +
23                    clients_arr[id][field] + "; ";
24            }
25            clients_div.appendChild(new_div);
26        }
27    };
28    XHR.open("get", "http://js.courses.dp.ua/files/get_clients.php",
29        true);
30    XHR.send();
31 </script>
</body></html>
```

Clients

name: Ivan; **age:** 23; **city:** Kiev;
name: Irina; **age:** 20; **city:** Lviv;
name: Marina; **age:** 27; **city:** Kharkiv;

JSON в реальности

The screenshot displays a web browser's developer tools interface. The top panel shows the HTML structure with a `<div id="clients">` containing three `<div>` elements, each representing a client's data. The middle panel shows the CSS styles for the selected `div` element, including `display: block;` and a box model diagram with dimensions of 1013 x 18. The bottom panel shows the console logs, including a message: `Recieve data: [{"name": "Ivan", "age": "23", "city": "Kiev"}, {"name": "Irina", "age": "20", "city": "Lviv"}, {"name": "Marina", "age": "27", "city": "Kharkiv"}]` and a subsequent log for `[Object, Object, Object]`.

Домашнее задание: задача от

«Apple»

```
2
3   var mas_length = Math.floor(Math.random()*20+20);
4   var arr = [];
5
6   for(var i = 0; i < mas_length; i++){
7       arr.push(Math.floor(Math.random()* 300 + 1));
8   }
9
10  console.log(arr);
11
12  /*
13
14     Вместо этого блока комментариев напишите свой код.
15
16  */
17
18 </script>
```

[269, 136, 69, 104, 254, 254, 237, 267, 110, 270, 159, 147, 245, 129, 154, 40, 207, 158, 290, 259, 137, 94, 126, 95, 79, 193, 23, 190]

Массив содержит котировки стоимости акций Apple (в течении одного дня торгов). Вы можете купить одну акцию и потом продать её. Ваша задача найти самый удачный момент для покупки и продажи акции (покупка разумеется должна быть до продажи), так чтобы ваша прибыль от этого действия была максимально возможной. В результате работы скрипта необходимо вывести рассчитанную прибыль, а также номера цен (в массиве) по которым бы производилась покупка и продажа.

JavaScript как язык программирования

его

концепции

**Переменные / Типы /
Операции
Ветвления (условные**

**операторы)
Циклы / Массивы (структуры**

**данных)
Функции**

Объект

ы