

# Урок 5 JavaScript

## Заметки

- XHR-запросы

XHR нужен, чтобы JavaScript мог делать веб-запросы — GET, POST и другие. Дает возможность не перегружать страницу целиком, а получить данные сервера без перезагрузки страницы.

```
<script>
function xhrTest() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "test.txt", false);
    xhr.send();
    if (xhr.status != 200) {
        alert(xhr.status + ': ' + xhr.statusText);
    } else {
        alert(xhr.responseText);
    }
}
</script>
<button onclick="xhrTest()">Load data</button>
```

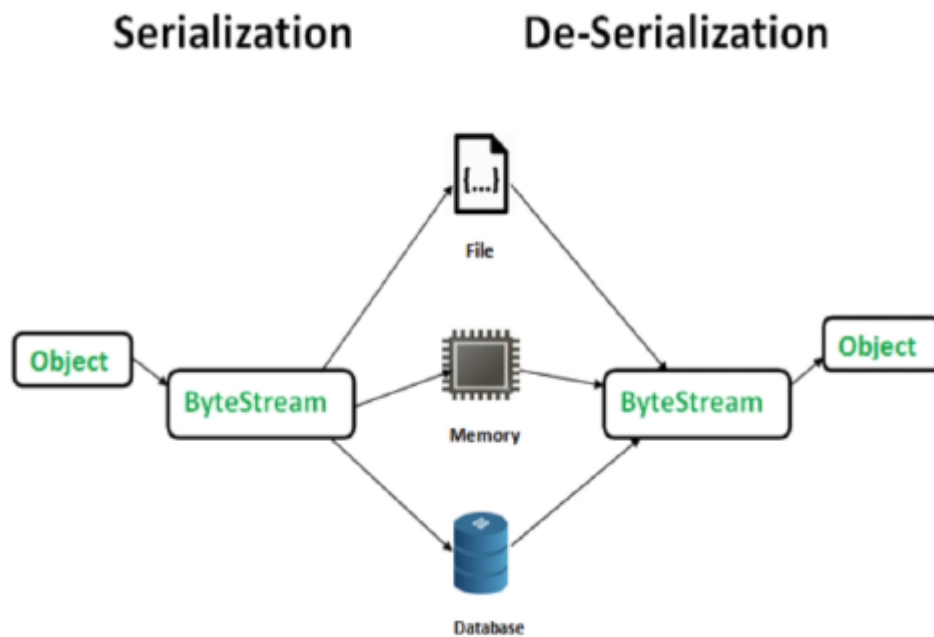
- Формат данных JSON

JSON придумали, чтобы можно было передавать сложный объект из браузера на сервер или наоборот.

```
<script>
function xhrTest() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "name.json", false);
    xhr.send();
    if (xhr.status != 200) {
        alert(xhr.status + ': ' + xhr.statusText);
    } else {
        var j = JSON.parse(xhr.responseText);
        alert(j["name"] + " " + j["lastname"]);
        var serialized = JSON.stringify(j);
        alert(serialized);
    }
}
```

```
</script>
```

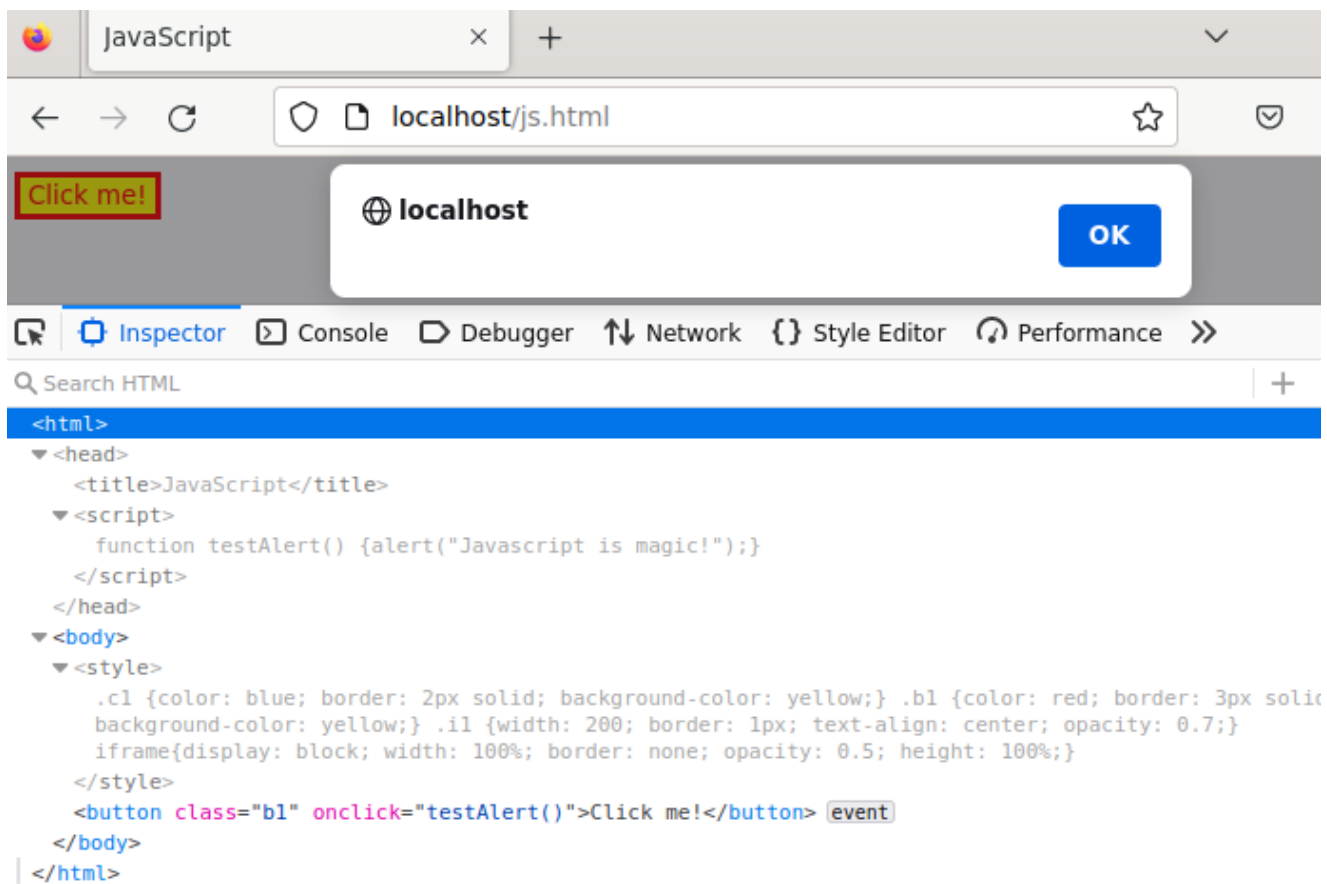
```
<button onclick="xhrTest()">Load data</button>
```



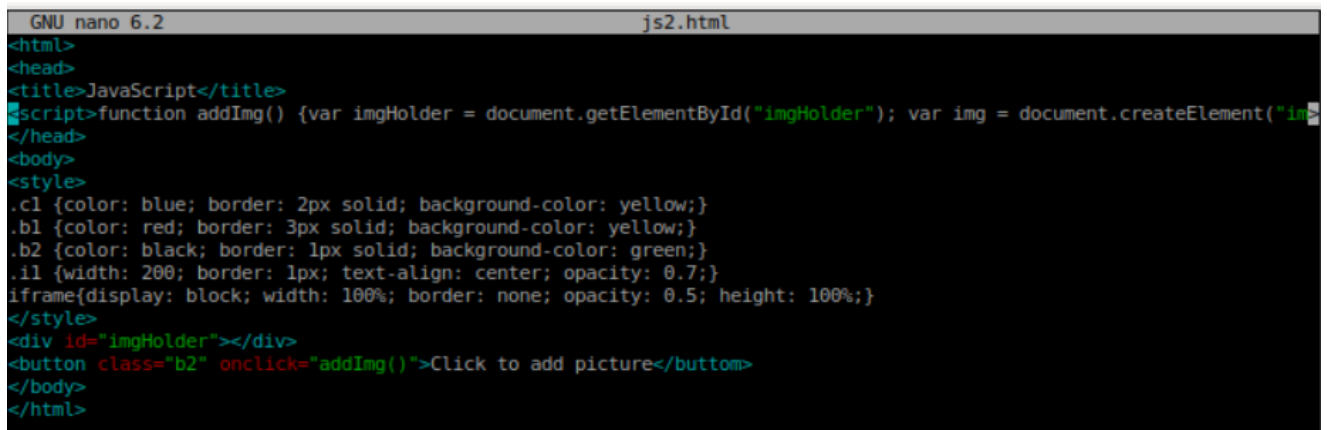
## Задание

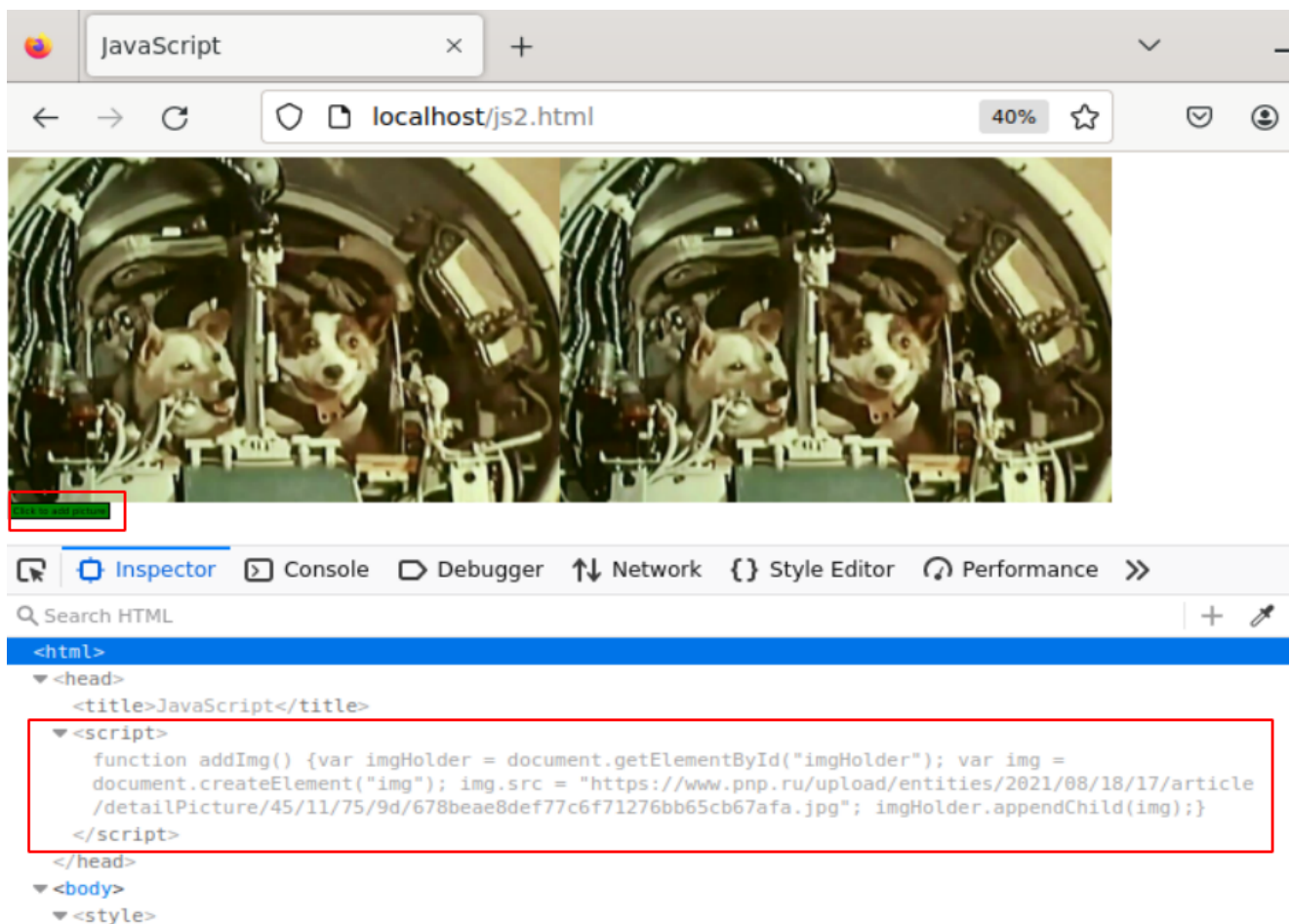
1. Создать страницу `javascript.html` с кнопкой, при нажатии на которую появлялся алерт с текстом `JavaScript is magic!`

```
GNU nano 6.2      js.html
<html>
<head>
<title>JavaScript</title>
<script>function testAlert() {alert("Javascript is magic!");}</script>
</head>
<body>
<style>
.c1 {color: blue; border: 2px solid; background-color: yellow;}
.b1 {color: red; border: 3px solid; background-color: yellow;}
.i1 {width: 200; border: 1px; text-align: center; opacity: 0.7;}
iframe{display: block; width: 100%; border: none; opacity: 0.5; height: 100%;}
</style>
<button class="b1" onclick="testAlert()">Click me!</button>
</body>
</html>
```



2. Написать функцию на языке JavaScript, которая будет создавать тег с картинкой и вставлять картинку в HTML.



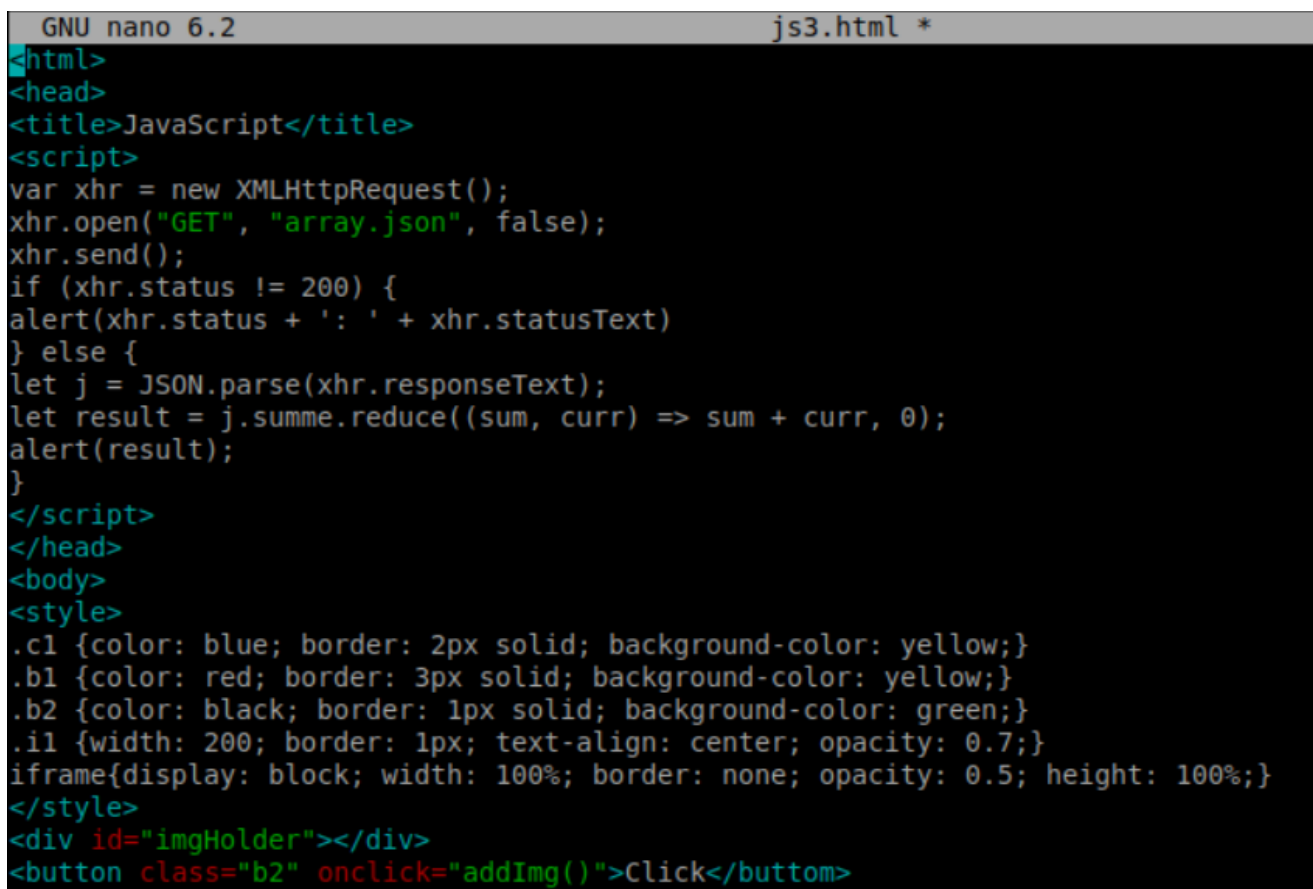
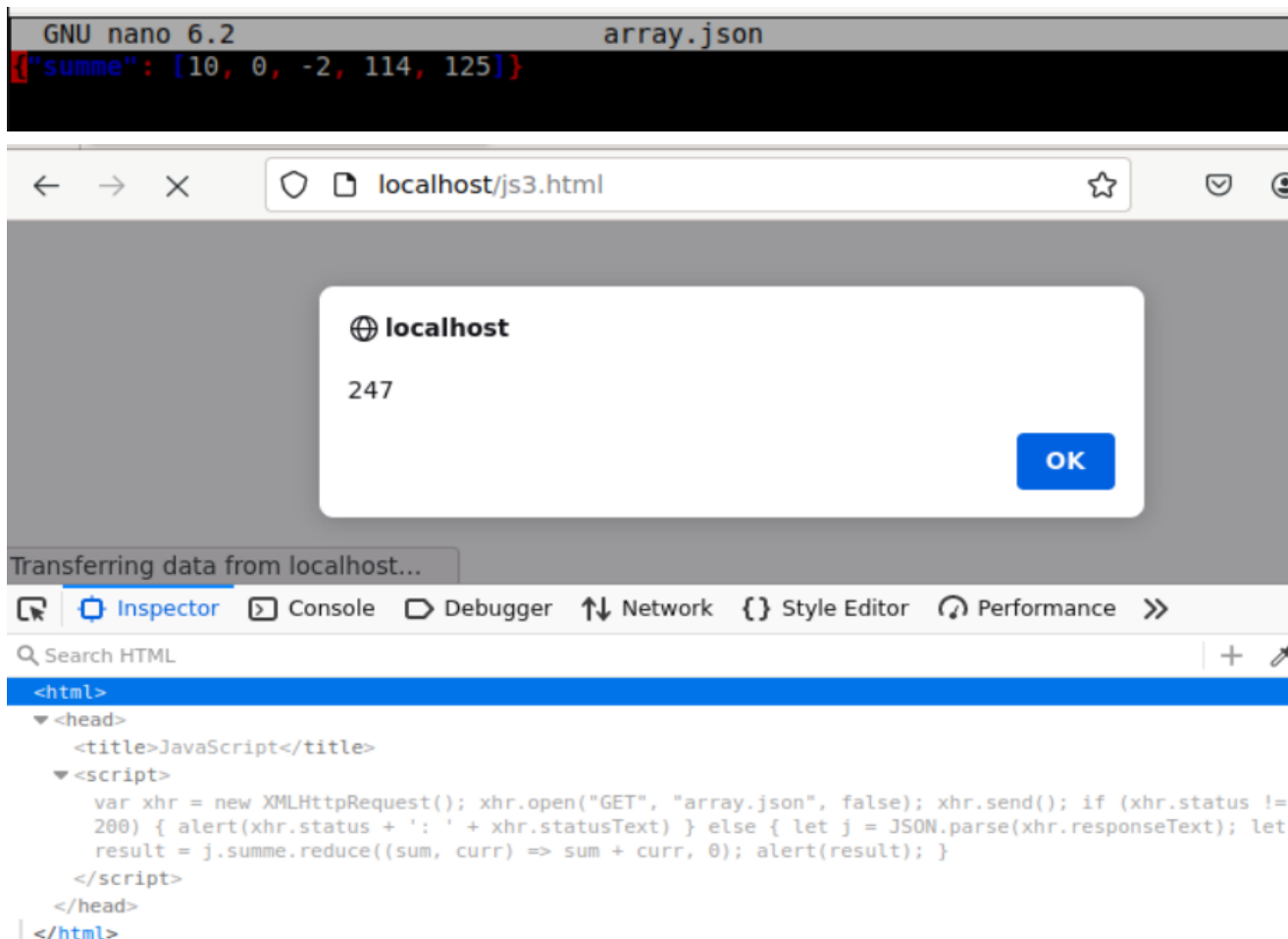


3. Создать файл `array.json` и разместить его в корневом каталоге своего сервера.

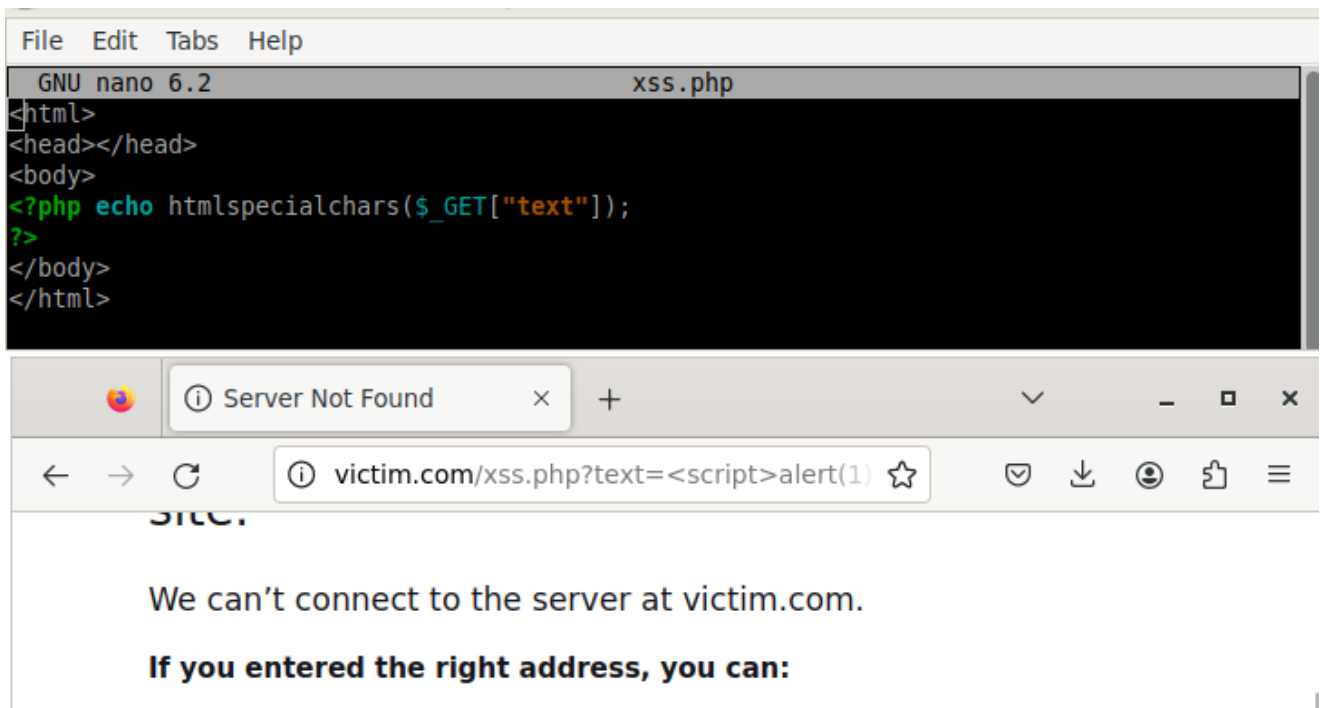
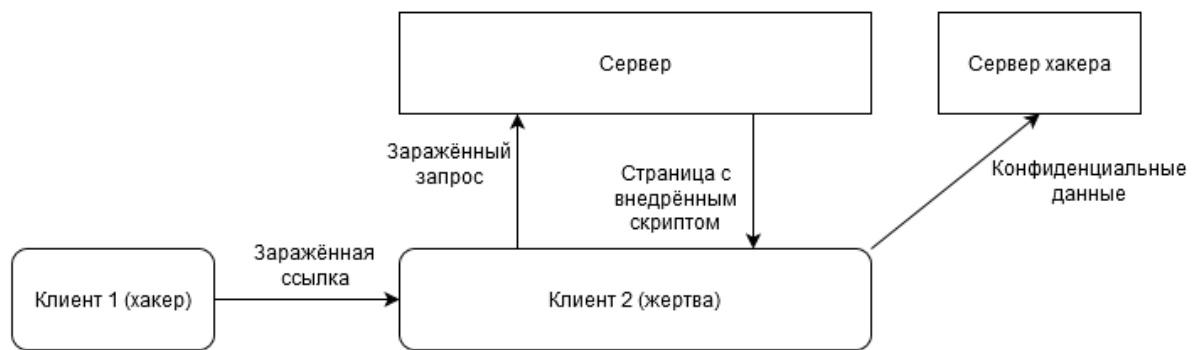
Содержимое файла `array.json`:

```
{ "summe": [10, 0, -2, 114, 125] }
```

Написать программу на языке JavaScript, которая будет делать XHR запрос на `http://localhost/array.json`, парсить полученный JSON и выводить сумму чисел массива `summe`.



4. (\*)Защититься от выполнения JavaScript кода на `victim.com/xss.php?text=`  
`<script>alert(1)</script>` (код страницы ниже) с помощью HTML экранирования.  
`<body> <?php echo $_GET["text"]; ?> </body>`



5. ( \* ) Написать функцию, которая будет сохранять ввод пользователя в переменную и отправлять значение переменной на <http://attacker.com>

