

Москва, 2020

Цель: изучить методы работы с html-страницами в JavaScript и способы работы с сервером, выполнить задания, ориентированные на закрепление изученного материала.

Часть 1

Задание 1

С клавиатуры считывается число N. Далее считывается N строк. Необходимо создать массив и сохранять в него строки только с четной длиной. Получившийся массив необходимо преобразовать в строку JSON и сохранить в файл.

```
1  "use strict";
2
3  function work_n_str(){
4      const readlineSync = require('readline-sync');
5
6      const arr = [];
7      const fs = require("fs");
8      const nameString = "result.txt";
9
10     const n = readlineSync.question("Input amout of strings: ");
11     console.log()
12
13     if (n <= 0){
14         console.log("ОШИБКА: недопустимое значение n");
15         console.log("-----");
16         console.log();
17         return;
18     }
19
20     for (let i = 0; i < n; i++){
21         let str = readlineSync.question("Input string: ");
22
23         if (str.length % 2){
24             continue;
25         }
26
27         arr.push(str);
28     }
29
30     const jsonString = JSON.stringify(arr)
31     fs.writeFileSync(nameString, jsonString);
32
33     console.log()
34 }
```

Тесты:

| Тест | Результат |
|--|---------------------------------|
| Input amout of strings: -1 | ОШИБКА: недопустимое значение n |
| Input amout of strings: 1 Input string: 123 | 1 ["123"] |
| Input amout of strings: 2 Input string: 1 Input string: 1472583 | 1 ["1", "1472583"] |
| Input amout of strings: 3 Input string: 11 Input string: 22 Input string: 33 | 1 ["11", "22", "33"] |
| Input amout of strings: 2 Input string: Input string: 1 | 1 ["", "1"] |
| Input amout of strings: 3 Input string: 14 Input string: 2 Input string: 3698 | 1 ["14", "2", "3698"] |

Задание 2

Необходимо считать содержимое файла, в котором хранится массив строк в формате JSON. Нужно вывести только те строки на экран, в которых содержатся только гласные буквы.

```

1  "use strict";
2
3  ✓ function work_with_file(){
4      const fs = require("fs");
5
6      const nameString = "data.txt";
7      let found = false;
8
9      ✓ if (fs.existsSync(nameString)) {
10         console.log("Файл существует");
11         console.log();
12     }
13     ✓ else {
14         console.log("Файл не найден");
15         return;
16     }
17
18     let strings = fs.readFileSync(nameString, "utf8");
19     ✓ if (strings == ""){
20         console.log("Пустой файл");
21         console.log();
22         return;
23     }
24
25     strings = JSON.parse(strings);
26

```

```

27     for (let i = 0; i < strings.length; i++){
28         let print_str = true;
29
30         for (let j = 0; j < strings[i].length; j++){
31             if (vowels.indexOf(strings[i].charAt(j)) == -1){
32                 print_str = false;
33                 break;
34             }
35         }
36         if (print_str){
37             found = true;
38             console.log(strings[i])
39         }
40     }
41
42     if (found === false){
43         console.log("Не найдено");
44         console.log();
45     }
46
47     console.log();
48 }
49
50 let vowels = "УЕЭОАЫЯИЮЁуеэоаыяиюё";
51
52 function main(){
53     work_with_file();
54 }
55
56 main();

```

Тесты:

| Тест | Результат |
|---|--------------------------------|
| 1 ["абвг", "а", "ае", "клм", "56", "0"] | Файл существует а ае |
| 1 ["12346"] | Файл существует Не найдено |
| 1 ["ы", "ааааа"] | Файл существует ы ааааа |
| 1 | Файл существует Пустой файл |
| Ввод названия несуществующего файла | Файл не найден |

Задание 3

С клавиатуры считывается строка - название расширения файлов. Далее считывается строка - адрес папки. Необходимо перебрать все файлы в папке и вывести содержимое файлов, у которых расширение совпадает с введенным расширением.

```

1  "use strict";
2
3  function read_file(folder, fileName){
4      const fs = require("fs");
5
6      let strings = fs.readFileSync(folder + "/" + fileName, "utf8");
7
8      if (strings == ""){
9          console.log("Empty file");
10         console.log();
11         return;
12     }
13
14     console.log(strings);
15     console.log();
16 }

```

```

18 function choose_files(){
19     let found = false;
20
21     const readlineSync = require('readline-sync');
22
23     const ext = readlineSync.question("Input extention: ");
24     const folder = readlineSync.question("Input address: ");
25
26     console.log();
27
28     const fs = require("fs");
29
30     const arr = fs.readdirSync(folder);
31
32     for(let i = 0; i < arr.length; i++) {
33         const fileName = arr[i];
34         let temp_folder = fileName.split('.');
35         if (temp_folder.length != 2)
36             continue;
37
38         if (temp_folder[1] === ext){
39             console.log(fileName);
40             read_file(folder, fileName);
41             found = true;
42         }
43     }
44
45     if (!found){
46         console.log("NOT found");
47         console.log();
48     }
49 }
50
51 function main(){
52     choose_files();
53 }
54
55 main();

```

Тесты:

| Тест | Результат |
|--|--|
| <div>Input extention: txt Input address: C:\msys64\home\bryan\ComputerArchitecture\lab_02\task_13</div> <p>Содержание папки:</p> <ul style="list-style-type: none">node_modulesindexpackagepackage-locktest1test2test3 | <div>test1.txt Hi, это тест номер 1.</div> <div>test2.txt A это ТЕСТ 2</div> <div>test3.txt Empty file</div> |
| <div>Input extention: txt Input address: C:\msys64\home\bryan\ComputerArchitecture\lab_02\task_12</div> <p>Содержание папки:</p> <ul style="list-style-type: none">node_modulesindexpackagepackage-lock | NOT found |

Задание 4





Дана вложенная структура файлов и папок. Все файлы имеют расширение "txt". Необходимо рекурсивно перебрать вложенную структуру и вывести имена файлов, у которых содержимое не превышает по длине 10 символов.

```

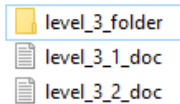
1  "use strict";
2
3  function show_all_subfolders(folder){
4      const fs = require("fs");
5
6      let arr = fs.readdirSync(folder, "utf8");
7
8      if (!arr.length)
9          return;
10
11     for(let i = 0; i < arr.length; i++) {
12         let fileName = arr[i];
13
14         if (fileName.split('.').length == 1)
15             show_all_subfolders(folder + "/" + fileName);
16         else{
17             let strings = fs.readFileSync(folder + "/" + fileName, "utf8");
18             if (strings.length <= 10){
19                 console.log(folder + "/" + fileName);
20                 console.log();
21             }
22         }
23     }
24 }
25
26 function main()
27 {
28     const folder = "." + "test";
29     show_all_subfolders(folder);
30 }
31
32 main();

```

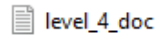
Тесты:

| Тест | Результат |
|---|--|
| <p>Папка test (файл level_1_doc удовлетворяет условию):</p> <div>  level_1_folder  level_1_doc </div> <p>Папка level_1_folder (файл level_2_doc удовлетворяет условию):</p> <div>  level_2_1_folder  level_2_2_folder  level_2_doc </div> <p>Папка level_2_1_folder (все тестовые</p> | <pre> ./test/level_1_doc.txt ./test/level_1_folder/level_2_1_folder/level_3_1_doc.txt ./test/level_1_folder/level_2_1_folder/level_3_2_doc.txt ./test/level_1_folder/level_2_doc.txt </pre> |

файл удовлетворяют условию):



Папка level_3_folder (файл не подходит):

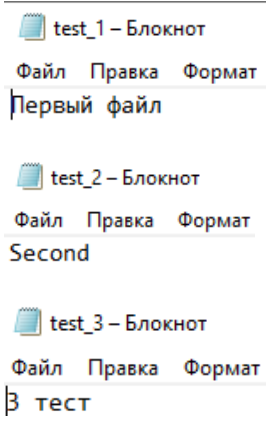
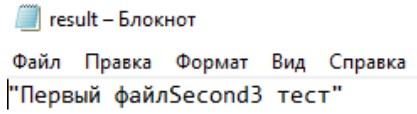


Папка level_2_2_folder (пустая)

Задание 5

С клавиатуры считывается число N. Далее считывается N строк - имена текстовых файлов. Необходимо склеить всё содержимое введенных файлов в одну большую строку и сохранить в новый файл.

```
1  "use strict";
2
3  function make_whole_file(){
4      let strings = "";
5
6      const readlineSync = require('readline-sync');
7
8      const arr = [];
9      const fs = require("fs");
10     const nameString = "result.txt";
11
12     const n = readlineSync.question("Input amount of files: ");
13     console.log()
14
15     if (n <= 0){
16         console.log("ОШИБКА: недопустимое значение n");
17         console.log();
18         return;
19     }
20
21     for (let i = 0; i < n; i++){
22         let fileName = readlineSync.question("Input name of file: ");
23         if (!fs.existsSync(fileName)) {
24             console.log("File was not found");
25             continue;
26         }
27         strings += fs.readFileSync(fileName, "utf8");
28     }
29
30     const jsonString = JSON.stringify(strings);
31     fs.writeFileSync(nameString, jsonString);
32 }
```

| Тест | Результат |
|--|--|
| <pre>Input amount of files: 3 Input name of file: test_1.txt Input name of file: test_2.txt Input name of file: test_3.txt</pre>  |  |

Задание 6

Написать код, который позволяет определить максимальный возможный уровень вложенности друг в друга полей в объекте, чтобы данный объект можно было преобразовать в строку формата JSON. Ответом является целое число.

```

1  "use strict";
2
3  const express = require("express");
4
5  class Obj
6  {
7      constructor (depth)
8      {
9          this.dpth = depth;
10         if (depth > 0)
11             this.next = new Obj(depth - 1);
12         else
13             this.next = null;
14     }
15 }
16
17 function find_depth()
18 {
19     let size = 1;
20
21     while (true)
22     {
23         try
24         {
25             let temp = new Obj(size);
26             let strjson = JSON.stringify(temp);
27             size += 1;
28         } catch (RangeError)
29         {
30             size -= 1;
31             break;
32         }
33     }
34
35     console.log(size);
36     console.log();
37 }

```

Результат:

964

Задание 7

Из файла считывается строка в формате JSON. В этой строке информация об объекте, в котором находится большое количество вложенных друг в друга полей. Объект представляет из себя дерево. Необходимо рекурсивно обработать дерево и найти максимальную вложенность в дереве. Необходимо вывести на экран ветку с максимальной вложенностью.

```

1  "use strict";
2
3  const fs = require("fs");
4  const file_name = "test.txt";
5
6  class Obj
7  {
8      constructor (depth)
9      {
10         this.dpth = depth;
11         this.leaf_1 = null;
12         this.leaf_2 = null;
13
14         if (Math.random() > 0.5)
15             this.leaf_1 = new Obj(depth + 1);
16         if (Math.random() > 0.5)
17             this.leaf_2 = new Obj(depth + 1);
18     }
19 }
20
21 function create_file()
22 {
23     let str = new Obj(0);
24     let jstr = JSON.stringify(str);
25     console.log(jstr);
26     fs.writeFileSync(file_name, jstr);
27 }

```

```

29 function processing_tree(tree)
30 {
31     let path = "";
32     let depth = 0;
33
34     for (let key in tree)
35     {
36         if (typeof(tree[key]) == "object" && tree[key] != null)
37         {
38             let res = processing_tree(tree[key]);
39             if (res[1] > depth)
40             {
41                 depth = res[1];
42                 path = key + "/" + res[0];
43             }
44         }
45     }
46
47     return [path, depth + 1];
48 }
49
50 function main()
51 {
52     //create_file();
53
54     let str = fs.readFileSync(file_name, "utf-8");
55     console.log("Стрнг: " + str);
56
57     let tree = JSON.parse(str);
58     let result = processing_tree(tree);
59
60     console.log();
61     console.log("Максимальная вложенность: " + result[0].slice(0, result[0].length - 1));
62     console.log("Максимальная длина: " + (result[1] - 1));
63     console.log();
64 }
65
66 main()

```

Тест:

[illegible]

Результат:

Максимальная вложенность: leaf_2/leaf_2/leaf_2/leaf_2/leaf_2/leaf_1/leaf_1/leaf_2/leaf_1/leaf_1/leaf_2/leaf_1/leaf_1/leaf_2/leaf_2/leaf_1/leaf_1
Максимальная длина: 17

Часть 2

Задание 1

Запустить сервер. Реализовать на сервере функцию для сравнения трёх чисел и выдачи наибольшего из них. Реализовать страницу с формой ввода для отправки запроса на сервер.

Файл index.js:

```

1  "use strict";
2
3  const fs = require("fs");
4
5  const express = require("express");
6
7  const app = express();
8  const port = 5015;
9  app.listen(port);
10 console.log("My server on port " + port);
11
12 app.get("/me/page", function(request, response) {
13   const nameString = request.query.p;
14   if (fs.existsSync(nameString)) {
15     const contentString = fs.readFileSync(nameString, "utf8");
16     response.end(contentString);
17   } else {
18     const contentString = fs.readFileSync("bad.html", "utf8");
19     response.end(contentString);
20   }
21 });
22
23 app.get("/find/max", function(request, response) {
24   const a = request.query.a;
25   const b = request.query.b;
26   const c = request.query.c;
27
28   const aInt = parseInt(a);
29   const bInt = parseInt(b);
30   const cInt = parseInt(c);
31
32   let res;
33
34   if (a > b)
35     res = a;
36   else
37     res = b;
38
39   if (c > res)
40     res = c;
41
42   const answerJSON = JSON.stringify({result: res});
43   response.end(answerJSON);
44 });

```

Файл a.html:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Максимум</title>
6  </head>
7  <body>
8    <h1>Максимум из A, B, C</h1>
9    <form method="GET" action="/find/max">
10     <p>Введите A</p>
11     <input name="a" spellcheck="false" autocomplete="off">
12     <p>Введите B</p>
13     <input name="b" spellcheck="false" autocomplete="off">
14     <p>Введите C</p>
15     <input name="c" spellcheck="false" autocomplete="off">
16     <br>
17     <br>
18     <input type="submit" value="Отправить запрос">
19   </form>
20 </body>
21 </html>

```

Файл bad.html совпадает с примером.

Тесты:

| Тест | Результат |
|---|--------------------------|
| <p>Максимум из А, В, С</p> <p>Введите А</p> <input type="text" value="-5"/> | <pre>{"result":56}</pre> |
| <p>Введите В</p> <input type="text" value="56"/> | |
| <p>Введите С</p> <input type="text" value="21"/> | |
| <input type="button" value="Отправить запрос"/> | |
| <p>Максимум из А, В, С</p> <p>Введите А</p> <input type="text" value="10"/> | <pre>{"result":10}</pre> |
| <p>Введите В</p> <input type="text" value="10"/> | |
| <p>Введите С</p> <input type="text" value="10"/> | |
| <input type="button" value="Отправить запрос"/> | |
| <p>Максимум из А, В, С</p> <p>Введите А</p> <input type="text" value="-55"/> | <pre>{"result":-6}</pre> |
| <p>Введите В</p> <input type="text" value="-6"/> | |
| <p>Введите С</p> <input type="text" value="-8"/> | |
| <input type="button" value="Отправить запрос"/> | |

Задание 2

Запустить сервер. На стороне сервера должен храниться файл, внутри которого находится JSON строка. В этой JSON строке хранится информация

о массиве объектов. Реализовать на сервере функцию, которая принимает индекс и выдает содержимое ячейки массива по данному индексу. Реализовать страницу с формой ввода для отправки запроса на сервер.

Файл index.js:

```
3 function show_element(ind){
4     const fs = require("fs");
5
6     const fileName = "data.txt";
7
8     let strings = fs.readFileSync(fileName, "utf8");
9     strings = JSON.parse(strings);
10
11     if (ind >= strings.length)
12         return "Wrong index: out of range";
13     else if (ind < 0)
14         return "Wrong index: <0";
15
16     return strings[ind];
17 }
```

```
19 function run_server(){
20     const fs = require("fs");
21
22     const express = require("express");
23
24     const app = express();
25     const port = 5015;
26     app.listen(port);
27     console.log("My server on port " + port);
28
29     app.get("/me/page", function(request, response) {
30         const nameString = request.query.p;
31         if (fs.existsSync(nameString)) {
32             const contentString = fs.readFileSync(nameString, "utf8");
33             response.end(contentString);
34         } else {
35             const contentString = fs.readFileSync("bad.html", "utf8");
36             response.end(contentString);
37         }
38     });
39
40     app.get("/find/ind", function(request, response) {
41         const ind = request.query.ind;
42         const answerJSON = JSON.stringify({result : show_element(ind)});
43         response.end(answerJSON);
44     });
45 }
```

Файл a.html:


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Значение по индексу</title>
6 </head>
7 <body>
8   <h1>Значение элемента массива по индексу</h1>
9   <form method="GET" action="/find/ind">
10     <p>Введите индекс</p>
11     <input name="ind" spellcheck="false" autocomplete="off">
12     <br>
13     <br>
14     <input type="submit" value="Отправить запрос">
15   </form>
16 </body>
17 </html>
```

Тесты:

| Тест | Результат |
|--|--|
| <p><code>[{"id": "1", "name": "Ann"}, {"id": "5", "name": "Patric"}, {"id": "6", "name": "Mila"}]</code></p> <p>Значение элемента массива по индексу</p> <p>Введите индекс</p> <p><input type="text" value="2"/></p> <p><input type="button" value="Отправить запрос"/></p> | <p><code>{"result":{"id":"6","name":"Mila"}}</code></p> |
| <p>Значение элемента массива по индексу</p> <p>Введите индекс</p> <p><input type="text" value="100"/></p> <p><input type="button" value="Отправить запрос"/></p> | <p><code>{"result":"Wrong index: out of range"}</code></p> |
| <p>Значение элемента массива по индексу</p> <p>Введите индекс</p> <p><input type="text" value="-56"/></p> <p><input type="button" value="Отправить запрос"/></p> | <p><code>{"result":"Wrong index: <0"}</code></p> |

Задание 3

Написать программу, которая на вход получает массив названий полей и адрес запроса (куда отправлять). Программа должна генерировать HTML разметку страницы, в которую встроена форма для отправки запроса.

```
1  "use strict";
2  //http://localhost:5015/me/page?p=a.html
3
4  const readFileSync = require('readline-sync');
5  const fs = require('fs');
6  const express = require('express');
7  const html_file = 'test.html';
8
9  function generate_html_file(fields, addr)
10 {
11     let page = '<!DOCTYPE html>\n\n'
12     <html>\n\n
13     <head>\n\n
14         <meta charset="UTF-8">\n\n
15         <title>Страница A</title>\n\n
16     </head>\n\n';
17
18     page += '<body>\n\n'
19     <h1>Сгенерированная страница</h1>\n\n
20     <form method="GET" action="" + addr + "">\n\n';
21
22     for (let i = 0; i < fields.length; i++)
23     {
24         page += '        <p>Введите ' + fields[i] + ': ' + '</p>\n\n'
25         <input name="" + fields[i] + "" spellcheck="false" autocomplete="off">\n\n';
26     }
27
28     page += '        <input type="submit" value="Отправить запрос">\n\n'
29     </form>\n\n
30 </body>\n\n
31 </html>';
32
33     return page;
34 }
```

```
36 function show_page_html()
37 {
38     const app = express();
39     const port = 5015;
40     app.listen(port);
41     console.log("My server on port " + port);
42
43     app.get("/me/page", function(request, response) {
44         let contentString;
45         if (fs.existsSync(html_file)) {
46             contentString = fs.readFileSync(html_file, "utf8");
47             response.end(contentString);
48         } else {
49             contentString = "ERROR: page isn's available now";
50         }
51         response.end(contentString);
52     });
53 }
54
55 function main()
56 {
57     const fields = readFileSync.question("Input fields separated ', ': ").split(", ");
58     const addr = readFileSync.question("Input address: ");
59     fs.writeFileSync(html_file, generate_html_file(fields, addr));
60
61     show_page_html();
62 }
63
64 main()
```

Тесты:

```
Input fields separated ', ': a, b, c, d, e, f
Input address: test/go
```

Результат:

Сгенерированная страница

Введите a:

Введите b:

Введите c:

Введите d:

Введите e:

Введите f:

Отправить запрос

Задание 4

Запустить сервер. Реализовать на сервере функцию, которая принимает на вход числа A, B и C. Функция должна выдавать массив целых чисел на отрезке от A до B, которые делятся на C нацело.

Файл index.js:

```
3 function show_numbers(a, b, c){
4   if (a > b)
5     return "Wrong borders";
6   if (c == 0)
7     return "Wrong c (prohibited c = 0)";
8
9   let arr = [];
10  for (let i = a; i <=b; i++){
11    if (i % c == 0)
12      arr.push(i);
13  }
14
15  return arr;
16 }
```

```

18 function run_server(){
19     const fs = require("fs");
20
21     const express = require("express");
22
23     const app = express();
24     const port = 5015;
25     app.listen(port);
26     console.log("My server on port " + port);
27
28     app.get("/me/page", function(request, response) {
29         const nameString = request.query.p;
30         if (fs.existsSync(nameString)) {
31             const contentString = fs.readFileSync(nameString, "utf8");
32             response.end(contentString);
33         } else {
34             const contentString = fs.readFileSync("bad.html", "utf8");
35             response.end(contentString);
36         }
37     });
38
39     app.get("/show/numbers", function(request, response) {
40         const a = request.query.a;
41         const b = request.query.b;
42         const c = request.query.c;
43         const aInt = parseInt(a);
44         const bInt = parseInt(b);
45         const cInt = parseInt(c);
46
47         const answerJSON = JSON.stringify({result: show_numbers(aInt, bInt, cInt)});
48         response.end(answerJSON);
49     });
50 }

```

Тесты:

| Тест | Результат |
|---|-----------------------|
| <p>Числа от А до В</p> <p>(делящиеся нацело на С)</p> <p>Введите А</p> <input type="text" value="-4"/> | {"result":[-3,0,3,6]} |
| Введите В | |
| <input type="text" value="7"/> | |
| Введите С | |
| <input type="text" value="3"/> | |
| <input type="button" value="Отправить запрос"/> | |

| | |
|---|---|
| <p>Числа от А до В</p> <p>(делящиеся нацело на С)</p> <p>Введите А</p> <input type="text" value="5"/> <p>Введите В</p> <input type="text" value="9"/> <p>Введите С</p> <input type="text" value="0"/> <input type="button" value="Отправить запрос"/> | <pre>{"result": "Wrong c (prohibited c = 0)"}</pre> |
| <p>Числа от А до В</p> <p>(делящиеся нацело на С)</p> <p>Введите А</p> <input type="text" value="1000"/> <p>Введите В</p> <input type="text" value="1"/> <p>Введите С</p> <input type="text" value="10 "/> <input type="button" value="Отправить запрос"/> | <pre>{"result": "Wrong borders"}</pre> |

Вывод: в процессе лабораторной работы были изучены методы работы с html-страницами, серверами. Были выполнены соответствующие задания на закрепление данного материала и составлен подробный отчет о работе.