

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Сибирский государственный университет телекоммуникаций и
информатики
СибГУТИ

ОТЧЕТ
по дисциплине «*WEB-технологии*»

по теме:

Погружение в разработку на JavaScript

Студент: *Пастухов А.А.*
Группы ИКС-432

Преподаватель: *Андреев А.В.*

Новосибирск, 2025

Создадим две переменные `username` и `bonusBalance`, и присвоим им значения. Используя `console.log()` выведем в консоль их значения. Создадим переменные для хранения баллов и сгорающих баллов. Напишем подсчет баллов используя созданные переменные.

```
index.js
1 console.log("Задание 1");
2 let username = "Андрей Валерьевич";
3 let bonusBalance = 1000;
4 console.log("Пользователь: " + username);
5 console.log("Баланс в долларах (1к1000): " + bonusBalance);
6
7 const bonusPerPurchase = 50;
8 const burnPerDay = 3;
9 const days = 7;
10 const purchases = 3;
11
12 bonusBalance += purchases * bonusPerPurchase;
13 bonusBalance -= days * burnPerDay;
14
15 console.log("Баланс через 7 дней:", bonusBalance);
```

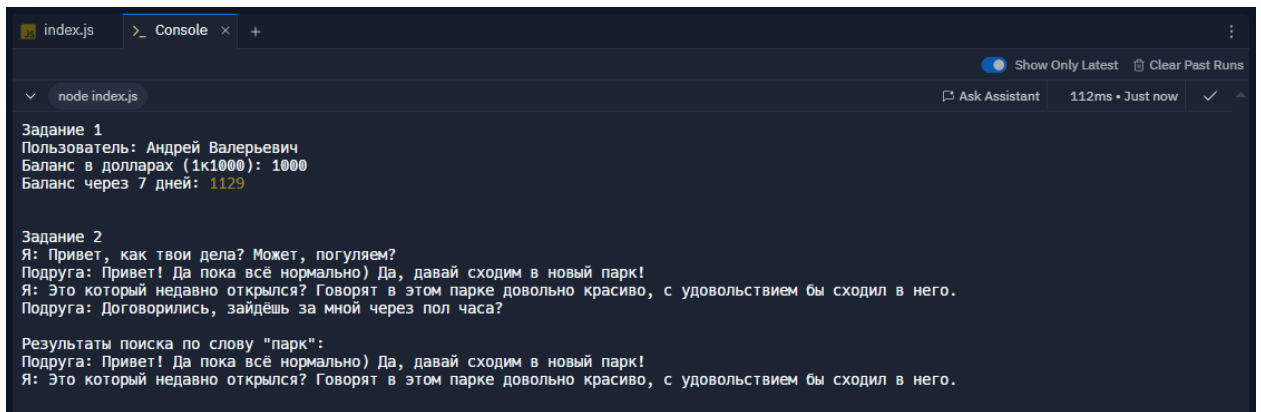
Рисунок 1 – Задание с деньгами

Инициализируем массив сообщений, заполним его данными. Напишем цикл для вывода сообщений и проверку на четность для выбора подписи. Добавим функцию поиска. Инициализируем переменную для хранения ключа и при помощи метода `includes` напишем проверку.

```
17 console.log("\n\nЗадание 2");
18 const messages = [
19   "Привет, как твои дела? Может, погуляем?",
20   "Привет! Да пока всё нормально) Да, давай сходим в новый парк!",
21   "Это который недавно открылся? Говорят в этом парке довольно красиво, с удовольствием бы сходил в него.",
22   "Договорились, зайдёшь за мной через пол часа?"
23 ];
24
25 for (let i = 0; i < messages.length; i++) {
26   const sender = i % 2 === 0 ? "Я" : "Подруга";
27   console.log(`${sender}: ${messages[i]}`);
28 }
29
30 const searchText = "парк";
31
32 console.log(`\n\nРезультаты поиска по слову "${searchText}":`);
33 for (let i = 0; i < messages.length; i++) {
34   if (messages[i].includes(searchText)) {
35     const sender = i % 2 === 0 ? "Я" : "Подруга";
36     console.log(`${sender}: ${messages[i]}`);
37   }
38 }
```

Рисунок 2 – Задание с массивом

Выведем результат программы на экран.



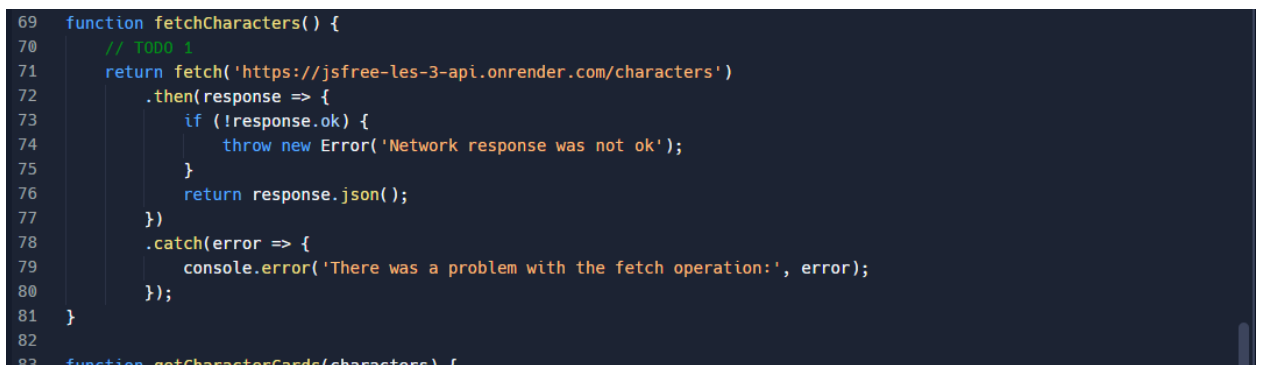
```
index.js  Console x +
node index.js
Задание 1
Пользователь: Андрей Валерьевич
Баланс в долларах (1к1000): 1000
Баланс через 7 дней: 1129

Задание 2
Я: Привет, как твои дела? Может, погуляем?
Подруга: Привет! Да пока всё нормально) Да, давай сходим в новый парк!
Я: Это который недавно открылся? Говорят в этом парке довольно красиво, с удовольствием бы сходил в него.
Подруга: Договорились, зайдёшь за мной через пол часа?

Результаты поиска по слову "парк":
Подруга: Привет! Да пока всё нормально) Да, давай сходим в новый парк!
Я: Это который недавно открылся? Говорят в этом парке довольно красиво, с удовольствием бы сходил в него.
```

Рисунок 3 – Общий вывод первой программы

Внутри функции `fetchCharacters` реализуем получение данных с API, используя `fetch('https://jsfree-les-3-api.onrender.com/characters')`. Добавим обработку ошибок.



```
69 function fetchCharacters() {
70   // TODO 1
71   return fetch('https://jsfree-les-3-api.onrender.com/characters')
72     .then(response => {
73       if (!response.ok) {
74         throw new Error('Network response was not ok');
75       }
76       return response.json();
77     })
78     .catch(error => {
79       console.error('There was a problem with the fetch operation:', error);
80     });
81 }
82
83 function getCharacterCards(characters) {
```

Рисунок 4 – Функция `fetchCharacters`

Внутри функции `getCharacterCards` реализуем формирование массива карточек используя `characters.map(character => getCharacterCard(character));`. Реализуем формирование модульных окон `characters.map(character => getCharacterModal(character));`;

```

26 function getCharacterModal(character) {
27   return `
28     <div id="exampleModal-${character.id}"
29       tabindex="-1"
30       aria-labelledby="exampleModalLabel-${character.id}"
31       class="modal fade"
32       style="display: none;"
33       aria-hidden="true"
34     >
35       <div class="modal-dialog">
36         <div class="modal-content">
37           <div class="modal-header">
38             <h5 class="modal-title">${character.name}</h5>
39             <button type="button"
40               data-bs-dismiss="modal"
41               aria-label="Close"
42               class="btn-close"
43             ></button>
44           </div>
45           <div class="modal-body">
46             
50             <div>
51               <p class="text-muted">${character.modified}</p>
52               <h5>Описание:</h5>
53               <p>${character.description}</p>
54             </div>
55           </div>
56           <div class="modal-footer">
57             <button type="button"

```

```

83 function getCharacterCards(characters) {
84   // TODO 2
85   // characters.length - длина массива characters
86   return characters.map(character => getCharacterCard(character));
87 }
88
89 function getCharacterModals(characters) {
90   // TODO 3
91   return characters.map(character => getCharacterModal(character));
92 }

```

Рисунок 5 – Функции обработки карточек

Выведем на отдельный экран результат нашей программы, а именно сайт с персонажами КВ Marvel.

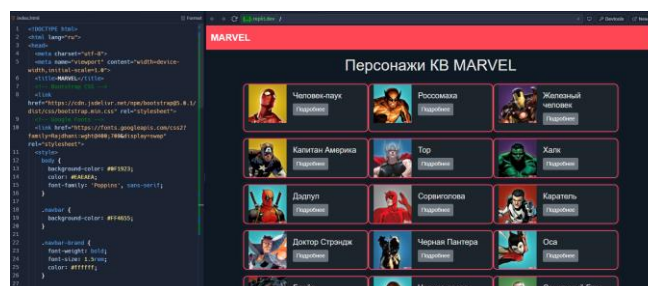


Рисунок 6 – Показ сайта на Replit

Создадим отдельно сайт на TimeWeb, выведем туда наше производство

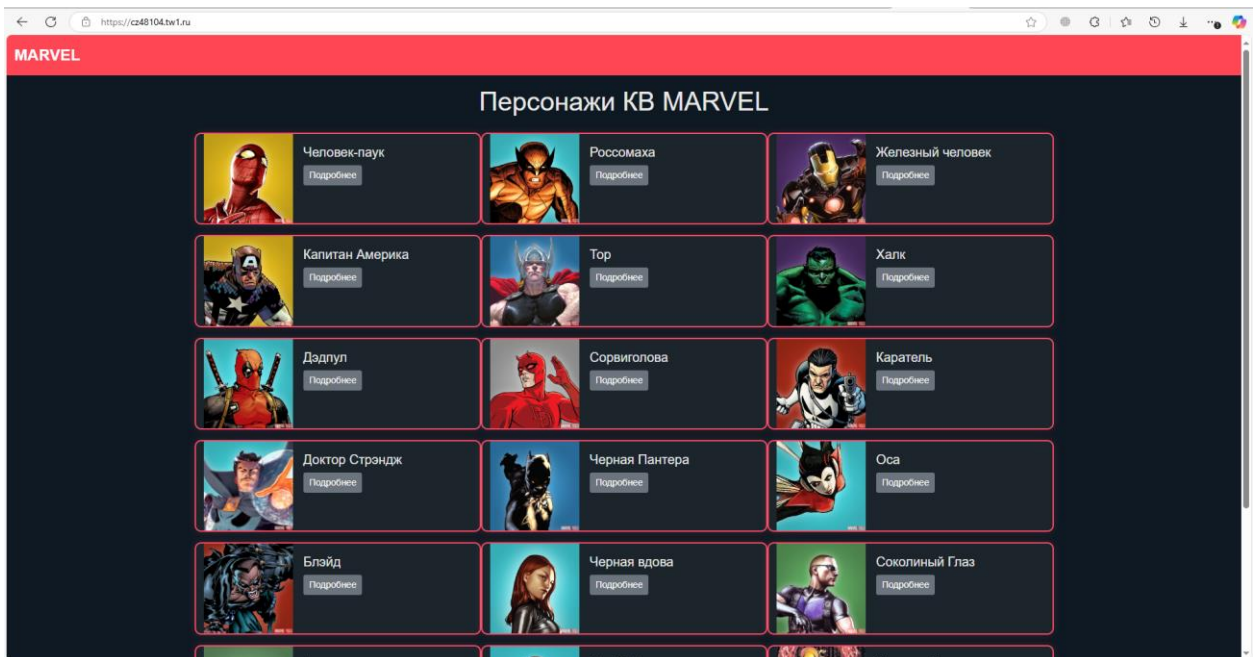


Рисунок 7 – Сайт на TimeWeb

<https://replit.com/@pastukhovaleksa/BaggyFlamboyantVolcano>

<https://replit.com/@pastukhovaleksa/BaggyFlamboyantVolcano>

<https://replit.com/@pastukhovaleksa/web13-chapter21>

<https://replit.com/@pastukhovaleksa/web13-chapter21>

<https://cz48104.tw1.ru/>

<https://cz48104.tw1.ru/>