Лабораторная работа №13. Погружение в разработку на JavaScript

Список заданий

1. Погружение в разработку на JavaScript. Базовые понятия (https://github.com/5t0d/Lab_13_JS/blob/course-v2/les-1)

2. Основы JavaScript (https://github.com/5t0d/Lab_13_JS/blob/course-v2/les-2)

3. Разработка сайта (https://github.com/5t0d/Lab_13_JS/blob/course-v2/les-3)

4. Доработка проекта (https://github.com/5t0d/Lab_13_JS/blob/course-v2/les-4)

Требования

- Браузер (рекомендуем последние версии Chrome, Яндекс.Браузер, Firefox, Safari).

Начало работы

1. Зарегистрируйтесь на сайте Repl.IT (https://repl.it/).
2. Перейдите в раздел **my repls**.
3. Нажмите кнопку **Start coding now!**, если приступаете впервые, или **New Repl**, если у вас уже есть работы.
4. В списке языков выберите `Nodejs`.
5. Код пишите в левой части окна.
6. Посмотреть результат выполнения файла можно, нажав на кнопку **Run**. Результат появится в правой части

   окна.

Сдача работы:

1. Ожидается, что вы выложите ваш проект на публичный хостинг
2. Предоставьте ссылку на ваш сайт в отчёте
3. Добавьте результаты с описанием сделанной работы в отчёт
4. Сохраните в формате PDF и загрузите в курс (https://clck.ru/3EHRfD)

1. "Погружение в разработку на JavaScript. Базовые понятия"

Вы работаете над разработкой интернет-магазина с бонусной системой. Выполните задачи для реализации различных функций.

Задача 1

В интернет-магазине у зарегистрированного пользователя есть имя и бонусный баланс. Представим, что вы пользователь. Отобразите информацию об успешном входе и своем бонусном балансе.

Процесс реализации

1. Создайте две переменные `username` и `bonusBalance`.
2. Присвойте переменным значения `my name` и `1000` соответственно.
3. Выведите информацию о пользователе в консоль двумя строками. Значения X и Y возьмите из переменных:
    - В первой - «Пользователь X».
    - Во второй - «Баланс Y».

Задача 2

За каждую покупку мы добавляем на баланс фиксированную сумму в 50 бонусных баллов. Баллы сгорают со временем - каждый день сгорает 3 балла.

Посчитайте, какой баланс будет у пользователя через 7 дней, если обычно он раз в два дня делает покупку.

Процесс реализации

1. Используйте переменную `bonusBalance` со значением бонусного баланса из задачи №1.
2. Создайте переменные:
    - для хранения баллов, которые добавляются после каждой покупки;
    - для количества баллов, которые сгорают каждый день.
3. Посчитайте, какая сумма будет на балансе через 7 дней.
4. Выведите баланс пользователя на экран.

Инструкция по выполнению задания

1. Зарегистрируйтесь на сайте Repl.IT (http://repl.it/).
2. Перейдите в раздел **my repls**.
3. Нажмите кнопку **Start coding now!**, если приступаете впервые, или **New Repl**, если у вас уже есть работы.
4. В списке языков выберите `Nodejs`.
5. Код пишите в левой части окна.
6. Посмотреть результат выполнения файла можно, нажав на кнопку **Run**. Результат появится в правой части

   окна.

2. "Основы JavaScript"

Задача 1

Вы с другом обменивается сообщениями в чате. Первое сообщение отправляет Друг, второе — Вы и так далее.

Выведите беседу в хронологическом порядке.

Пример переписки:

> ***Вы:*** *Привет!*
>
> ***Друг:*** *Здорово, коль не шутишь!*

Шаги по решению задачи:

1. Инициализируйте массив сообщений.
2. Заполните его следующими данными:
    - "Пойдем гулять в парк?",
    - "Кажется, дождь собирается. Лучше пойдем в кино!",
    - "Давай, сегодня как раз вышел новый фильм.",
    - "Встречаемся через час у кинотеатра."
3. Напишите цикл для вывода сообщений.
4. В зависимости от номера сообщения нужно подставлять в начало сообщения либо “Друг”, либо “Вы”.

```
let arr = ["Пойдем гулять в парк?", "Кажется, дождь собирается. Лучше пойдем в кино!", "Давай, сегодня как раз вышел новый фильм.", "Встречаемся через час у кинотеатра."];
for(i = 0; i < arr.length; i++){
  if(i % 2 == 0){
    console.log("Друг: " + arr[i]);
   }
   else{
    console.log("Вы: " + arr[i]);
   }
}
```

```
~/13$ node 2.js
Друг: Пойдем гулять в парк?
Вы: Кажется, дождь собирается. Лучше пойдем в кино!
Друг: Давай, сегодня как раз вышел новый фильм.
Вы: Встречаемся через час у кинотеатра.
~/13$
```

Задача 2

Нужно добавить функцию поиска по тексту сообщений в нашем мессенджере.

Например, пользователь ищет слово “кино”, и ему отображаются все сообщения с таким текстом.

```
2.js                                                    Format
1  let arr = ["Пойдем гулять в парк?", "Кажется, дождь собирается. Лучше пойдем в кино!", "Давай, сегодня как раз вышел новый фильм.", "Встречаемся через час у кинотеатра."];
2  for(i = 0; i < arr.length; i++){
3    if(i % 2 == 0){
4      console.log("Друг: " + arr[i]);
5     }
6     else{
7       console.log("Вы: " + arr[i]);
8     }
9  }
10 for(i = 0; i < arr.length; i++){
11   if(arr[i].includes("кино")){
12     console.log(arr[i]);
13   }
14 }
```

```
[0] ~/13: bash
~/13$ node 2.js
Друг: Пойдем гулять в парк?
Вы: Кажется, дождь собирается. Лучше пойдем в кино!
Друг: Давай, сегодня как раз вышел новый фильм.
Вы: Встречаемся через час у кинотеатра.
Кажется, дождь собирается. Лучше пойдем в кино!
Встречаемся через час у кинотеатра.
~/13$
```

Шаги по решению задачи:

1. Инициализируйте переменную, в которой будет храниться искомый текст. Например, слово “кино”.
2. Для поиска воспользуйтесь методом includes (https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/includes) для строки.

Пример использования метода для поиска слова “зелёный” в строке “чёрный чай”:

```
"Чёрный чай".includes("зелёный"); // вернёт false
```

Метод возвращает `true`, если слово есть в строке, и `false`, если нет.

3. Напечатайте списком все сообщения, в которых есть искомая строка.

Инструкция по выполнению задания

1. Зарегистрируйтесь на сайте Repl.IT (https://repl.it/).
2. Перейдите в раздел **my repls**.
3. Нажмите кнопку **Start coding now!**, если приступаете впервые, или **New Repl**, если у вас уже есть работы.
4. В списке языков выберите `Nodejs`.
5. Код пишите в левой части окна.
6. Посмотреть результат выполнения файла можно, нажав на кнопку **Run**. Результат появится в правой части окна.

3. "Разработка сайта"

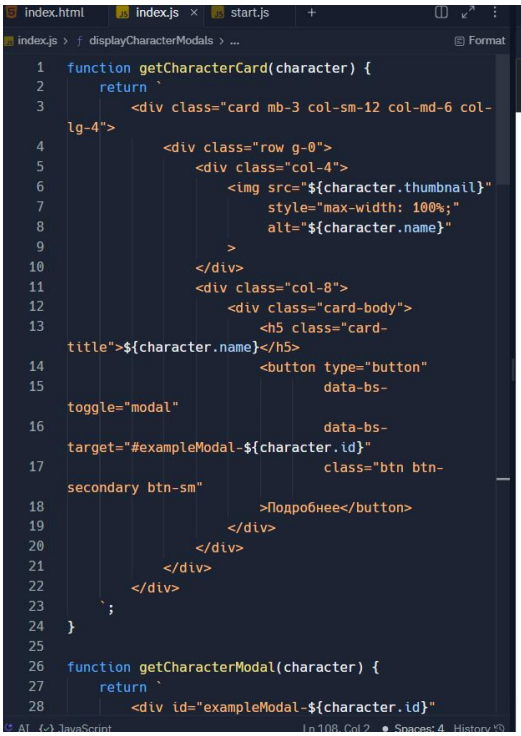
На занятии мы научились получать данные с API (https://jsfree-les-3-api.onrender.com/characters) и используя полученные данные оживили статичную верстку. В рамках домашней работы вам нужно повторить аналогичные операции.

1. Откройте стартовый шаблон (https://replit.com/@5t0dgm/Lab13JS-les-3-start-template) приложения и сделайте `Fork repl` чтобы создать копию приложения в своем аккаунте repl

> *Вся работа будет происходить в файле* `index.js`

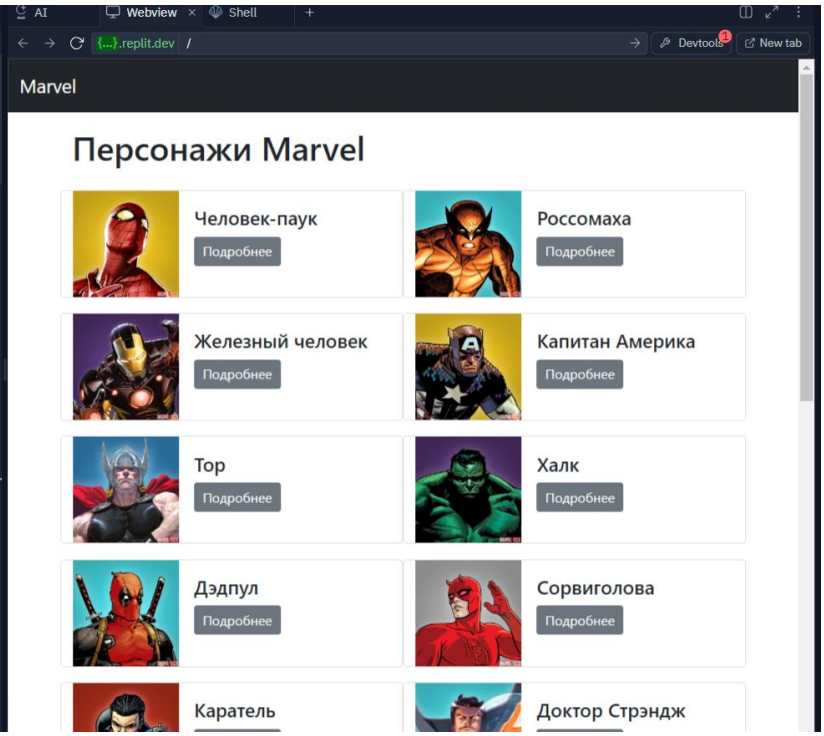
2. Внутри `function fetchCharacters() {}` реализуйте получение данных с API (https://jsfree-les-3-api.onrender.com/characters), используя Fetch (https://learn.javascript.ru/fetch)
3. Внутри `function getCharacterCards(characters) {}` реализуйте формирование массива карточек персонажей

получить одну карточку можно вызвав `getCharacterCard(character)` *4. Внутри function* `getCharacterModals (characters) {}` *реализуйте формирование массива модульных окон персонажей получить одно модальное окно можно вызвав* `getCharacterModal(character)`



```
function getCharacterCard(character) {
    return `
        <div class="card mb-3 col-sm-12 col-md-6 col-lg-4">
            <div class="row g-0">
                <div class="col-4">
                    <img src="${character.thumbnail}"
                         style="max-width: 100%;"
                         alt="${character.name}"
                    >
                </div>
                <div class="col-8">
                    <div class="card-body">
                        <h5 class="card-title">${character.name}</h5>
                        <button type="button"
                                data-bs-toggle="modal"
                                data-bs-target="#exampleModal-${character.id}"
                                class="btn btn-secondary btn-sm"
                        >Подробнее</button>
                    </div>
                </div>
            </div>
        </div>
    `;
}

function getCharacterModal(character) {
    return `
        <div id="exampleModal-${character.id}"
```



4. "Доработка проекта"

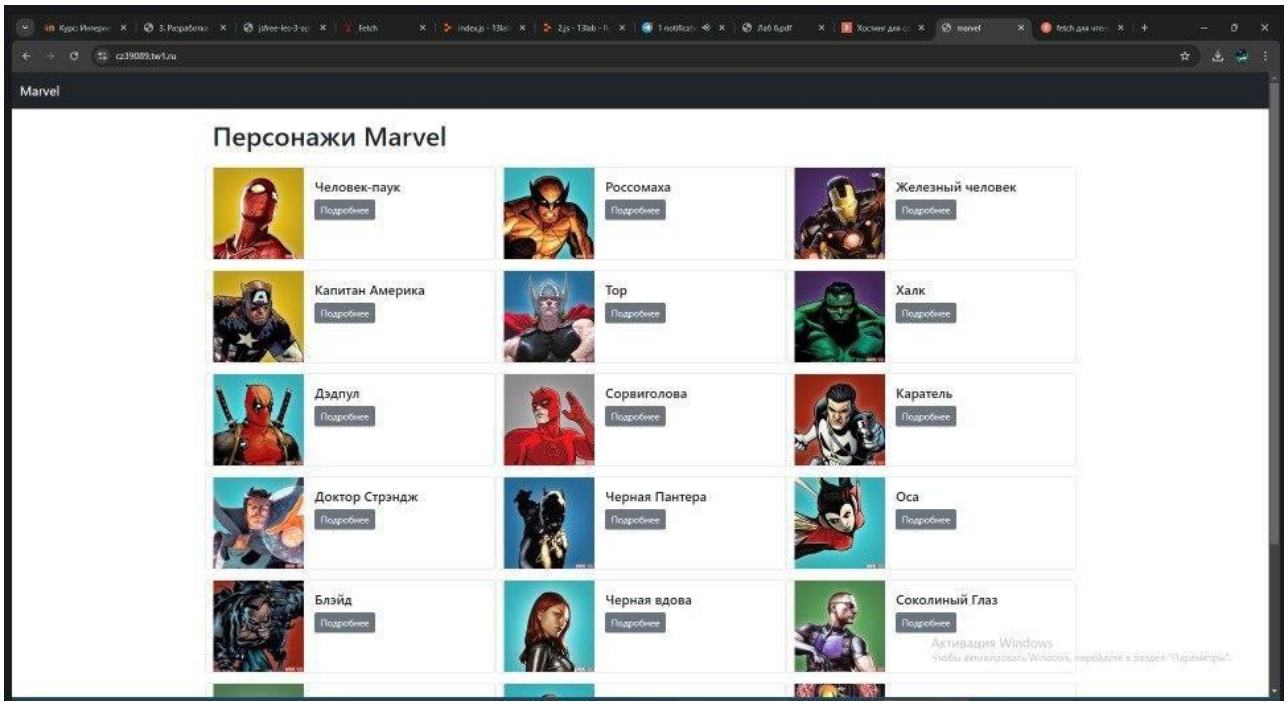
На занятии мы доработали свое приложение и опубликовали его на timeweb Hosting. В рамках работы вам нужно зарегистрировать аккаунт и опубликовать свое приложение на timeweb Hosting (https://timeweb.com/ru/).

###Размещение сайта Типичная схема размещения сайтов на серверах Timeweb выглядит следующим образом:

- `/` – корневая директория аккаунта;
- `/public_html` – директория для файлов главного сайта;
- `/new-site` – директория для дополнительного сайта;
- `/new-site/public_html` – директория для файлов дополнительного сайта (создается автоматически при создании новой директории в корне аккаунта).

Файлы сайта необходимо загружать в папку `public_html` необходимой директории, сохраняя структуру каталогов с файлами.

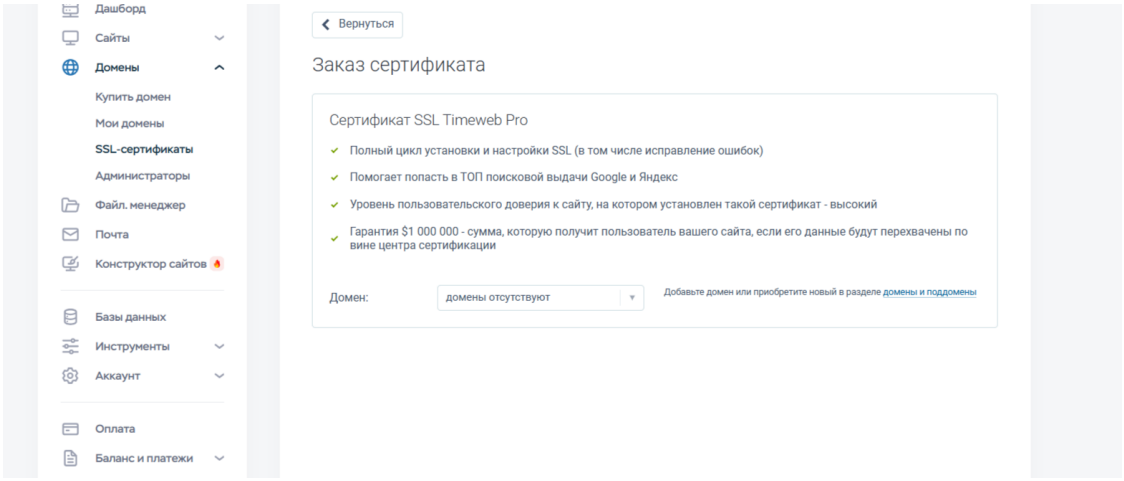
Для размещения сайта закажите хостинг (https://timeweb.com/ru/services/hosting/) **(Пробный период 10 дней бесплатно. ничего оплачивать не нужно!)** по нужному тарифу и зайдите в созданную панель управления аккаунтом, используя логин и пароль, высланные на ваш e-mail после регистрации на сайте.



Далее выполните следующие действия:

- Загрузите файлы сайта в папку `public_html` в директории сайта с помощью «Файлового менеджера» («Файл» — «Загрузить на сервер»).
- Проверьте работу сайта.
-

Для того, чтобы сайт работал по защищенному протоколу https (это защищает данные пользователей при передаче, повышает доверие пользователей к вашему сайту и его позиции в поисковых системах), можно получить SSL-сертификат, бесплатный Let's Encrypt формируется автоматически в течение 30 минут после регистрации учетной записи.



http://cz77368.tw1.ru/