Методика выполнения лабораторной работы № 5.

Создайте два файла: index.html и index.js.

В файле index.js создайте асинхронную функцию parse()

```
async function parse (){}
```
Код будет в фигурных скобках.

Функция вызывается вне тела функции строкой `parse();`

В теле функции парсим информацию из API с помощью fetch:

```
let response = await fetch('https://randomuser.me/api/?results=10');
```
Обратите внимание, что мы указали число пользователей в запросе - 10.

Далее, конвертируем результат запроса в формат json в переменную parsed:

```
let parsed = await response.json();
```

Теперь в цикле будем создавать блоки html кода и вставлять в html файл.

```
for (let i = 0; i < 10; i++) {}
```
Код цикла также пишем в фигурных скобках

Создаём текстовую переменную card, где текст обозначиваем в так называемый backtick - машинописный обратный апостроф (`). Это нужно для многострочности.

```
let card = ``
```
Далее будем заполнять эту переменную html кодом блока карточек.

Здесь я использую Bootstrap Cards, вы можете сделать это по другому.

У меня получилась следующая переменная:

```
let card = `
<div class=card>
        <img src="${parsed.results[i].picture.large}" class="card-img-top" alt="...">
        <div class="card-body">
                <h5 class="card-title">${parsed.results[i].name.title +
parsed.results[i].name.first + parsed.results[i].name.last}</h5>
                <p class="card-text">${parsed.results[i].email}</p>
        </div>
</div>`;
```
В моём **примере** вы можете увидеть, что всю спарсенную информацию можно вставить в код (строку) с помощью форматирования ${javascript переменная}.

На место ссылки на фотографию я вставляю её ссылку из переменной parsed, которая содержит информацию в формате json. Как выглядит этот формат можете посмотреть по ссылке: https://randomuser.me/api/?results=10

Json это словарь, который имеет формат ключ: значение_1, значение_2, ...; при этом значения тоже могут быть ключами, благодаря чему осуществляется вложенность.

Чтобы достать значение по ключам необходимо перечислить из через точку:
`parsed.results[i].name.title`

При этом целочисленные типы указываются без точки, но в квадратных скобках. В нажем случае целочисленное значение - это номер пользователя. Мы перечисляем пользователей с помощью цикла от 0 до 10. **i** - это переменная-счётчик.

Вам необходимо добавить ещё несколько тегов `<p class="card-text">${}</p>` куда нужно достать прочую информацию по заданию:

2. Данные, которые должны быть выведены для каждого пользователя:
    a. Пол
    b. Фамилия и имя
    c. Адрес
    d. Email
    e. Дата рождения и возраст
    f. Телефон
    g. Аватар

После этого нужно в цикле вставить блоки с карточками в код html страницы:

`mainclass.insertAdjacentHTML("afterbegin", card);`
Имя cardclass должно соответствовать id главного блока html страницы, до этого мы ещё дойдём. Afterbegin здесь означает, что мы вставляем код `card` после зачала блока с id `mainclass`.

Итак, ваш javascript-файл должен выглядеть как-то так:

```
async function parse (){

    let response = await fetch('https://randomuser.me/api/?results=10');
    let parsed = await response.json();

    for (let i = 0; i < 10; i++) {


        let card = `
        <div class=card>
            <img src="${parsed.results[i].picture.large}" class="card-img-top" alt="...">
            <div class="card-body">
                <h5 class="card-title">${parsed.results[i].name.title + parsed.results[i].name.first + parsed.results[i].name.last}</h5>
                <p class="card-text">${parsed.results[i].email}</p>
            </div>
        </div>`;

        cardclass.insertAdjacentHTML("afterbegin", card);

    }
}

parse();

```

Теперь перейдём к заполнению нашего html файла.

Если вы хотите использовать Bootstrap, необходимо его подключить в head:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
```

```
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rt
R1zdB" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/
YESvpZ13" crossorigin="anonymous"></script>
```

Здесь же можете описать необходимые вам стили в теге style, или создать отдельный style.css и подключить его сюда.

Далее заполним наш body:

```
<body>
<div class="row row-cols-1 row-cols-md-5 g-4" id="mainclass"></div>
<script src="index.js"></script>
</body>
```

Обратите внимание, что id тега div соответствует названию в javascript файле (`mainclass`).

После div подключаем файл `index.js` как скрипт.

Ваш html файл должен выглядеть примерно так:

```
<!DOCTYPE html>
<html>
    <head>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7S
        <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9X

        <style>
            html, body {
                margin: 0;
                max-width: 90%;
                max-height: 90%;
            }

            body{
                margin-top: 5%;
                margin-left: 10%;

            }

            .card {
                height: auto;
                width: auto;
            }

        </style>

    </head>
    <body>

        <div class="row row-cols-1 row-cols-md-5 g-4" id="cardclass"></div>

        <script src="parse.js"></script>

    </body>
</html>
```

На этом всё. Будут вопросы - пишите vk.com/romo4ko .