

2 ЛАБОРАТОРНАЯ РАБОТА №2

«Исследование возможностей программирования на стороне клиента. Основы языка JavaScript»

2.1 Цель работы

Исследовать особенности написания программ для приложений на стороне клиента. Изучить основы языка JavaScript и объектной модели браузера. Приобрести практические навыки проверки HTML-форм с использованием JavaScript.

2.2 Вариант задания

На рисунке 2.1 указана проверка полей страницы теста для 5 пункта выполнения лабораторной работы, соответствующая варианту.

15	3	Количество слов в ответе не менее 35*
----	---	---------------------------------------

Рисунок 2.1 – Вариант проверки полей

2.3 Ход выполнения работы

2.3.1 В начале выполнения лабораторной работы была модифицирована страница «Фотоальбом», разработанная при выполнении предыдущей лабораторной работы. Был реализован вывод таблицы, содержащей фото, с использованием операторов циклов. Значения имен файлов фото и подписей к фото предварительно были размещены в массивах. Код функции представлен в листинге 2.1.

Листинг 2.1 – Функция вывода фотоальбома

```
function setPhotos(num, quant) {
```

```

const photos = {};
const titles = {};
const alts = {};
const ps = {};

for (let i = num; i < num + quant; i++) {
    titles[i] = ps[i] = "Мем №" + i;
    alts[i] = "meme" + i;
    photos[i] = "img/" + i + ".jpg";
}

for (let i = num; i < num + quant; i++) {
    const classPh = (quant === 3) ? "photo3" : "photo";
    document.write(`
        <div class="${classPh}">
            
            <p>${titles[i]}</p>
        </div>
    `);
}
}

```

На рисунке 2.2 можно увидеть готовый результат получившейся страницы.

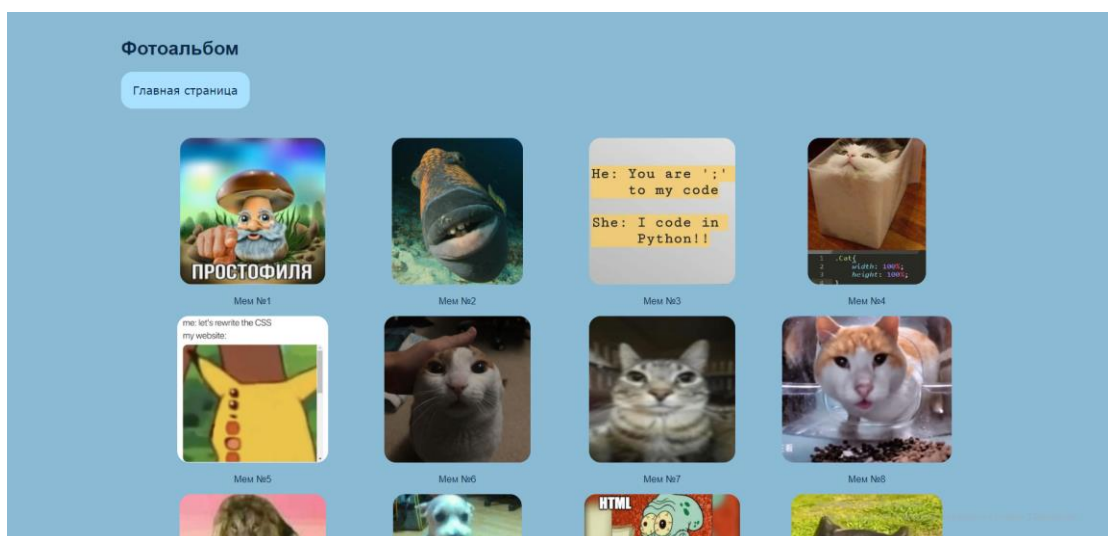


Рисунок 2.2 – Веб-страница «Фотоальбом»

Для главной страницы сайта была проведена валидация страницы в соответствии со стандартами, результат для mainpage.html показан на рисунке 2.3.

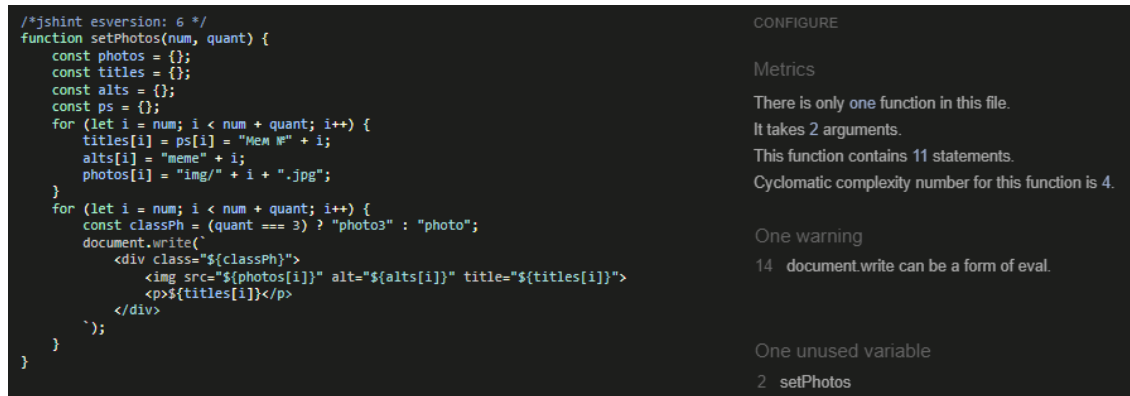


Рисунок 2.3 – Результаты валидации album.js

2.3.2 Далее была сделана модификация страницы «Мои интересы», а именно был реализован вывод списков с использованием JavaScript-функции с переменным числом аргументов. Код функции представлен в листинге 2.2.

Листинг 2.2 – Текст страницы сайта с автобиографией

```

function outputList(type) {
  document.write("<" + type + "l>");
  for (let i = 1; i < outputList.arguments.length; i+=2) {
    document.write(`<li><a href="${outputList.arguments[i]}">
      ${outputList.arguments[i+1]}</a></li>`);
  }
  document.write("</" + type + "l>")
}

```

Рисунок 2.4 демонстрирует внешний вид полученной страницы с использованием js-функции.

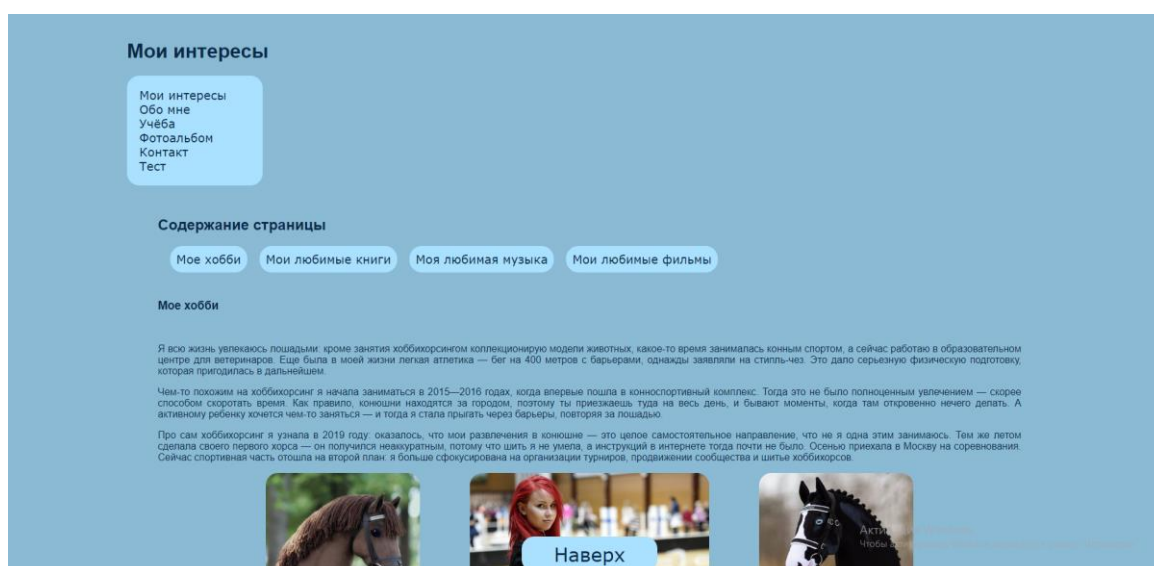


Рисунок 2.4 – Страница «Мои интересы»

Для данной страницы сайта была проведена валидация страницы в соответствии со стандартами, результат показан на рисунке 2.5.



Рисунок 2.5 – Результаты валидации myinterests.js

2.3.3 Была добавлена функция проверки заполненности форм для страницы «Контакт». В случае если какое-либо из полей формы осталось незаполненным при нажатии на кнопку отправить, выводится сообщение об ошибке и устанавливается фокус на незаполненный элемент. Для поля «Фамилия Имя Отчество» была добавлены функция проверки ввода 3 слов, разделенных пробелом. Поле «Телефон» проверяется другой функцией на ввод «+7» в начале, количество введенных цифр должно быть от 9 до 11, а также оно не должно содержать пробелы. Код функций представлен в листинге 2.3.

Листинг 2.3 – Функции проверки формы страницы «Контакт»

```
function checkInput() {  
    let form = document.forms["contactForm"];  
    if (form.fio.value === "") {  
        alert("Пожалуйста, введите ваше ФИО.");  
        form["fio"].focus();  
        return false;  
    }  
    if (!checkFio(document.forms["contactForm"])) return false;  
    if (!form["gender"].value) {  
        alert("Пожалуйста, выберите ваш пол.");  
        return false;  
    }  
    if (form["age"].value === "") {  
        alert("Пожалуйста, выберите ваш возраст.");  
        form["age"].focus();  
        return false;  
    }  
    if (form["email"].value === "") {  
        alert("Пожалуйста, введите ваш e-mail.");  
        form["email"].focus();  
        return false;  
    }  
    if (form["phone"].value === "") {  
        alert("Пожалуйста, введите ваш телефон.");  
        form["phone"].focus();  
        return false;  
    }  
    if (!checkPhone(document.forms["contactForm"])) return false;  
    if (form["message"].value === "") {  
        alert("Пожалуйста, введите сообщение.");  
        form["message"].focus();  
        return false;  
    }  
}
```

```

        return true;
    }

function checkFio(formName){
    let fio = formName["fio"].value;
    let cnt = 0;
    let startPos=0;
    while(fio.indexOf(" ",startPos)>=0){
        cnt++;
        startPos=fio.indexOf(" ",startPos)+1;
    }
    if (startPos<fio.length) {
        cnt++;
    }
    if(cnt !== 3){
        alert("Пожалуйста, введите корректные данные ФИО.");
        formName["fio"].focus();
        return false;
    }
    else return true;
}

function checkPhone(formName){
    let phone = formName["phone"];
    let substr = phone.value.substring(1,phone.value.length-1);
    if(phone.value[0]+phone.value[1] === "+7"){
        if(!isNaN(substr)){
            if((phone.value.length >= 10)&&(phone.value.length <= 12)){
                return true;
            }
        }
        else{
            alert("Пожалуйста, введите корректный номер телефона.");
            phone.focus();
            return false;
        }
    }
}

```

```

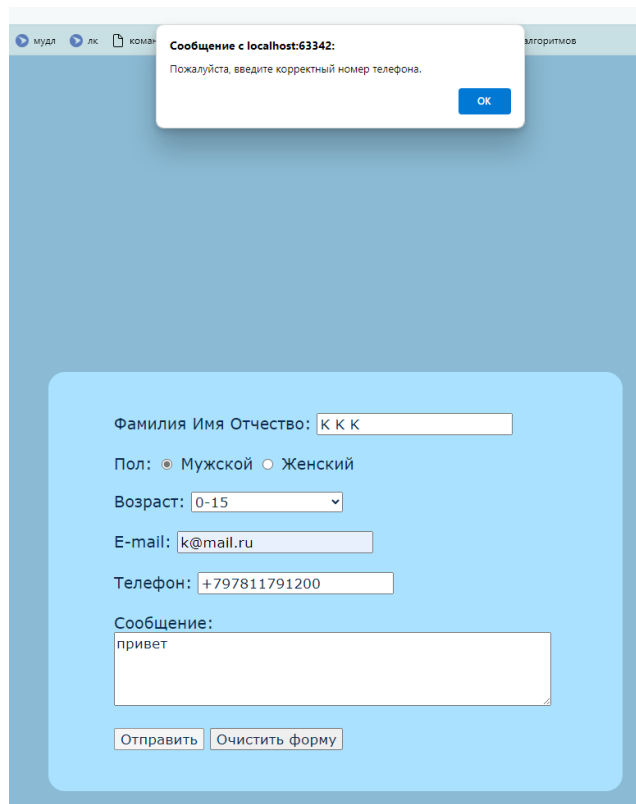
else{
    alert("Пожалуйста, используйте только цифры для ввода
номера.");
    phone.focus();
    return false;
}
}
else{
    alert("Номер должен начинаться с '+7'.");
    phone.focus();
    return false;
}
}
}

```

Рисунок 2.6 демонстрирует вывод ошибки при пропуске поля для ввода.
Рисунок 2.7 демонстрирует ошибку при вводе неверного формата номера.
Рисунок 2.8 показывает, какая ошибка вызывается при неверном вводе ФИО.

The screenshot shows a web browser window with a light blue background. A white alert dialog box is centered on the screen, displaying the text: "Сообщение с localhost:63342: Пожалуйста, введите ваш e-mail." with an "OK" button. Below the dialog, there is a light blue rounded rectangle containing a form. The form fields are: "Фамилия Имя Отчество:" with the value "Kriukova Ksenia Maksimov"; "Пол:" with radio buttons for "Мужской" (selected) and "Женский"; "Возраст:" with a dropdown menu showing "0-15"; "E-mail:" with an empty text input; "Телефон:" with the value "+79781179120"; and "Сообщение:" with a text area containing "привет". At the bottom of the form are two buttons: "Отправить" and "Очистить форму".

Рисунок 2.6 – Ошибка при пропуске поля для ввода



Сообщение с localhost:63342:
Пожалуйста, введите корректный номер телефона.

Фамилия Имя Отчество: К К К

Пол: ☒ Мужской ☐ Женский

Возраст: 0-15

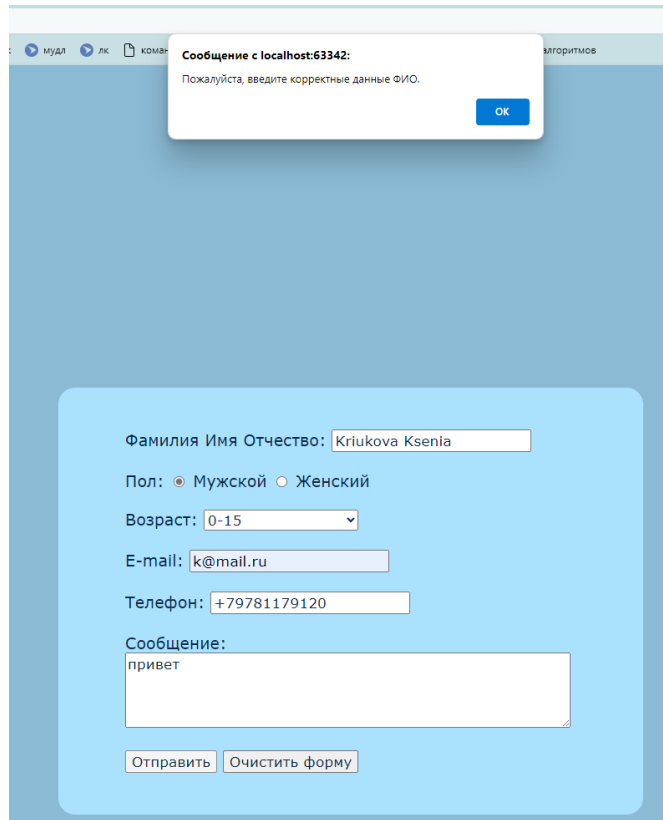
E-mail: k@mail.ru

Телефон: +797811791200

Сообщение:
привет

Отправить Очистить форму

Рисунок 2.7 – Ошибка при вводе неверного формата номера



Сообщение с localhost:63342:
Пожалуйста, введите корректные данные ФИО.

Фамилия Имя Отчество: Kriukova Ksenia

Пол: ☒ Мужской ☐ Женский

Возраст: 0-15

E-mail: k@mail.ru

Телефон: +79781179120

Сообщение:
привет

Отправить Очистить форму

Рисунок 2.8 – Ошибка при неверном вводе ФИО

Для данной страницы сайта была проведена валидация страницы в соответствии со стандартами, результат показан на рисунке 2.9.

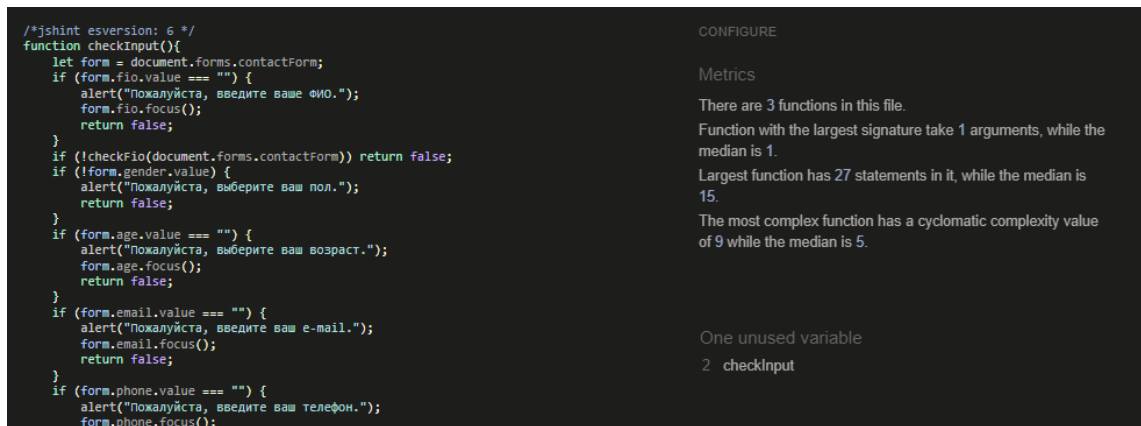


Рисунок 2.9 – Результаты валидации contact.js

2.3.4 Была добавлена функция проверки заполненности форм для страницы «Тест». В случае если какое-либо из полей формы осталось незаполненным при нажатии на кнопку отправить, выводится сообщение об ошибке и устанавливается фокус на незаполненный элемент. Для поля третьего тестового вопроса были добавлены функция проверки ввода не менее 35 слов, разделенных одним или более пробелами. Код функций представлен в листинге 2.4.

Листинг 2.4 – Функции проверки формы страницы «Тест»

```
function checkInputText() {
    let form = document.forms["testForm"];

    if (form["fio"].value === "") {
        alert("Пожалуйста, введите ваше ФИО.");
        form["fio"].focus();
        return false;
    }

    if (!checkFio(document.forms["testForm"])) return false;

    if (form["group"].value === "") {
        alert("Пожалуйста, выберите вашу группу.");
        form["group"].focus();
        return false;
    }
}
```

```

if (!form["question 1"].value) {
    alert("Пожалуйста, выберите ответ на 1 вопрос.");
    return false;
}
if (form["question 2"].value === "") {
    alert("Пожалуйста, выберите ответ на 2 вопрос.");
    form["question 2"].focus();
    return false;
}
if (form["question 3"].value === "") {
    alert("Пожалуйста, введите ответ на 3 вопрос.");
    form["question 3"].focus();
    return false;
}
if (!checkAnswer(document.forms["testForm"])) return false;
return true;
}

function checkAnswer(formName){
    let answer = formName["question 3"].value;
    let words = answer.split(/\s+/);
    if (words.length < 35) {
        alert("Пожалуйста, введите не менее 35 слов в ответе.");
        formName["fio"].focus();
        return false;
    }
    else return true;
}

```

Рисунок 2.10 демонстрирует вывод ошибки при пропуске поля для ввода. Рисунок 2.11 демонстрирует ошибку при вводе в поле менее 35 слов.

[illegible]

Рисунок 2.11 – Ошибка при вводе в поле менее 35 слов

Для данной страницы сайта была проведена валидация страницы в соответствии со стандартами, результат показан на рисунке 2.12.



Рисунок 2.12 – Результаты валидации test.js

Выводы

В ходе лабораторной работы были исследованы особенности написания программ для приложений на стороне клиента. Также были изучены основы языка JavaScript и объектной модели браузера. Были приобретены практические навыки проверки HTML-форм с использованием JavaScript.